

32nd International Symposium on Graph Drawing and Network Visualization

GD 2024, September 18–20, 2024, Vienna, Austria

Edited by

Stefan Felsner

Karsten Klein



Editors

Stefan Felsner 

TU Berlin, Germany
felsner@math.tu-berlin.de

Karsten Klein 

Universität Konstanz, Germany
karsten.klein@uni-konstanz.de

ACM Classification 2012

Mathematics of computing → Graph theory; Mathematics of computing → Graph algorithms; Human-centered computing → Graph drawings

ISBN 978-3-95977-343-0

Published online and open access by

Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at <https://www.dagstuhl.de/dagpub/978-3-95977-343-0>.

Publication date

October, 2024

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <https://portal.dnb.de>.

License

This work is licensed under a Creative Commons Attribution 4.0 International license (CC-BY 4.0): <https://creativecommons.org/licenses/by/4.0/legalcode>.



In brief, this license authorizes each and everybody to share (to copy, distribute and transmit) the work under the following conditions, without impairing or restricting the authors' moral rights:

- Attribution: The work must be attributed to its authors.

The copyright is retained by the corresponding authors.

Digital Object Identifier: 10.4230/LIPIcs.GD.2024.0

ISBN 978-3-95977-343-0

ISSN 1868-8969

<https://www.dagstuhl.de/lipics>

LIPICs – Leibniz International Proceedings in Informatics

LIPICs is a series of high-quality conference proceedings across all fields in informatics. LIPICs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

Editorial Board

- Luca Aceto (Reykjavik University, IS and Gran Sasso Science Institute, IT)
- Christel Baier (TU Dresden, DE)
- Roberto Di Cosmo (Inria and Université Paris Cité, FR)
- Faith Ellen (University of Toronto, CA)
- Javier Esparza (TU München, DE)
- Daniel Král' (Masaryk University, Brno, CZ)
- Meena Mahajan (*Chair*, Institute of Mathematical Sciences, Chennai, IN)
- Anca Muscholl (University of Bordeaux, FR)
- Chih-Hao Luke Ong (Nanyang Technological University, SG)
- Phillip Rogaway (University of California, Davis, US)
- Eva Rotenberg (Technical University of Denmark, Lyngby, DK)
- Raimund Seidel (Universität des Saarlandes, Saarbrücken, DE and Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Wadern, DE)
- Pierre Senellart (ENS, Université PSL, Paris, FR)

ISSN 1868-8969

<https://www.dagstuhl.de/lipics>

■ Contents

Preface	
<i>Stefan Felsner and Karsten Klein</i>	0:xi
Organization	
.....	0:xiii

Invited Talks

How Can Biclique Covers Help in Matching Problems	
<i>Otfried Cheong</i>	1:1–1:1
How Can Algorithms Help in Protecting Our Privacy	
<i>Monika Henzinger</i>	2:1–2:1

Regular Papers

The Euclidean MST-Ratio for Bi-Colored Lattices	
<i>Sebastiano Cultrera di Montesano, Ondřej Draganov, Herbert Edelsbrunner, and Morteza Saghafian</i>	3:1–3:23
Enumeration of Intersection Graphs of x -Monotone Curves	
<i>Jacob Fox, János Pach, and Andrew Suk</i>	4:1–4:12
Holes in Convex and Simple Drawings	
<i>Helena Bergold, Joachim Orthaber, Manfred Scheucher, and Felix Schröder</i>	5:1–5:9
1-Planar Unit Distance Graphs	
<i>Panna Gehér and Géza Tóth</i>	6:1–6:9
The Density Formula: One Lemma to Bound Them All	
<i>Michael Kaufmann, Boris Klemz, Kristin Knorr, Meghana M. Reddy, Felix Schröder, and Torsten Ueckerdt</i>	7:1–7:17
Note on Min- k -Planar Drawings of Graphs	
<i>Petr Hliněný and Lili Ködmön</i>	8:1–8:10
Partitioning Complete Geometric Graphs on Dense Point Sets into Plane Subgraphs	
<i>Adrian Dumitrescu and János Pach</i>	9:1–9:10
Constrained Outer-String Representations	
<i>Therese Biedl, Sabine Cornelsen, Jan Kratochvíl, and Ignaz Rutter</i>	10:1–10:18
Monotone Arc Diagrams with Few Biarcs	
<i>Steven Chaplick, Henry Förster, Michael Hoffmann, and Michael Kaufmann</i>	11:1–11:16
The Parameterized Complexity Of Extending Stack Layouts	
<i>Thomas Depian, Simon D. Fink, Robert Ganian, and Martin Nöllenburg</i>	12:1–12:17



The Price of Upwardness <i>Patrizio Angelini, Therese Biedl, Markus Chimani, Sabine Cornelsen, Giordano Da Lozzo, Seok-Hee Hong, Giuseppe Liotta, Maurizio Patrignani, Sergey Pupyrev, Ignaz Rutter, and Alexander Wolff</i>	13:1–13:20
Bounding the Treewidth of Outer k -Planar Graphs via Triangulations <i>Oksana Firman, Grzegorz Gutowski, Myroslav Kryven, Yuto Okada, and Alexander Wolff</i>	14:1–14:17
Bundling-Aware Graph Drawing <i>Daniel Archambault, Giuseppe Liotta, Martin Nöllenburg, Tommaso Piselli, Alessandra Tappini, and Markus Wallinger</i>	15:1–15:19
GRAPHTRIALS: Visual Proofs of Graph Properties <i>Henry Förster, Felix Klesen, Tim Dwyer, Peter Eades, Seok-Hee Hong, Stephen G. Kobourov, Giuseppe Liotta, Kazuo Misue, Fabrizio Montecchiani, Alexander Pastukhov, and Falk Schreiber</i>	16:1–16:18
Connectivity-Faithful Graph Drawing <i>Amyra Meidiana, Seok-Hee Hong, and Yongcheng Jing</i>	17:1–17:17
On the Uncrossed Number of Graphs <i>Martin Balko, Petr Hliněný, Tomáš Masařík, Joachim Orthaber, Birgit Vogtenhuber, and Mirko H. Wagner</i>	18:1–18:13
Weakly Leveled Planarity with Bounded Span <i>Michael A. Bekos, Giordano Da Lozzo, Fabrizio Frati, Siddharth Gupta, Philipp Kindermann, Giuseppe Liotta, Ignaz Rutter, and Ioannis G. Tollis</i>	19:1–19:19
Quantum Algorithms for One-Sided Crossing Minimization <i>Susanna Caroppo, Giordano Da Lozzo, and Giuseppe Di Battista</i>	20:1–20:9
The Perception of Stress in Graph Drawings <i>Gavin J. Mooney, Helen C. Purchase, Michael Wybrow, Stephen G. Kobourov, and Jacob Miller</i>	21:1–21:17
Boundary Labeling in a Circular Orbit <i>Annika Bonerath, Martin Nöllenburg, Soeren Terziadis, Markus Wallinger, and Jules Wulms</i>	22:1–22:17
Intersection Graphs with and Without Product Structure <i>Laura Merker, Lena Scherzer, Samuel Schneider, and Torsten Ueckerdt</i>	23:1–23:19
Upward Pointset Embeddings of Planar st -Graphs <i>Carlos Alegría, Susanna Caroppo, Giordano Da Lozzo, Marco D’Elia, Giuseppe Di Battista, Fabrizio Frati, Fabrizio Grosso, and Maurizio Patrignani</i> ..	24:1–24:18
Parameterized Algorithms for Beyond-Planar Crossing Numbers <i>Miriam Münch and Ignaz Rutter</i>	25:1–25:16
Storylines with a Protagonist <i>Tim Hegemann and Alexander Wolff</i>	26:1–26:22
On k -Planar Graphs Without Short Cycles <i>Michael A. Bekos, Prosenjit Bose, Aaron Büngener, Vida Dujmović, Michael Hoffmann, Michael Kaufmann, Pat Morin, Saeed Odak, and Alexandra Weinberger</i>	27:1–27:17

On the Edge Density of Bipartite 3-Planar and Bipartite Gap-Planar Graphs <i>Aaron Büngener and Maximilian Pfister</i>	28:1–28:21
Improving the Crossing Lemma by Characterizing Dense 2-Planar and 3-Planar Graphs <i>Aaron Büngener and Michael Kaufmann</i>	29:1–29:22
Flips in Colorful Triangulations <i>Rohan Acharya, Torsten Mütze, and Francesco Verciani</i>	30:1–30:20
Revisiting ILP Models for Exact Crossing Minimization in Storyline Drawings <i>Alexander Dobler, Michael Jünger, Paul J. Jünger, Julian Meffert, Petra Mutzel, and Martin Nöllenburg</i>	31:1–31:19
On the Complexity of Recognizing k^+ -Real Face Graphs <i>Michael A. Bekos, Giuseppe Di Battista, Emilio Di Giacomo, Walter Didimo, Michael Kaufmann, and Fabrizio Montecchiani</i>	32:1–32:22
Crossing Numbers of Beyond Planar Graphs Re-Visited: A Framework Approach <i>Markus Chimani, Torben Donzelmann, Nick Kloster, Melissa Koch, Jan-Jakob Völlering, and Mirko H. Wagner</i>	33:1–33:17
Separable Drawings: Extendability and Crossing-Free Hamiltonian Cycles <i>Oswin Aichholzer, Joachim Orthaber, and Birgit Vogtenhuber</i>	34:1–34:17
On k -Plane Insertion into Plane Drawings <i>Julia Katheder, Philipp Kindermann, Fabian Klute, Irene Parada, and Ignaz Rutter</i>	35:1–35:11
Noncrossing Longest Paths and Cycles <i>Greg Aloupis, Ahmad Biniaz, Prosenjit Bose, Jean-Lou De Carufel, David Eppstein, Anil Maheshwari, Saeed Odak, Michiel Smid, Csaba D. Tóth, and Pavel Valtr</i>	36:1–36:17
Rectilinear Crossing Number of Graphs Excluding a Single-Crossing Graph as a Minor <i>Vida Dujmović and Camille La Rose</i>	37:1–37:18
Harborth’s Conjecture for 4-Regular Planar Graphs <i>Daniel J. Chang and Timothy Sun</i>	38:1–38:9
Drawing Planar Graphs and 1-Planar Graphs Using Cubic Bézier Curves with Bounded Curvature <i>David Eppstein, Michael T. Goodrich, and Abraham M. Illickan</i>	39:1–39:17
Morphing Planar Graph Drawings via Orthogonal Box Drawings <i>Therese Biedl, Anna Lubiw, and Jack Spalding-Jamieson</i>	40:1–40:16

Graph Drawing Contest Report

Graph Drawing Contest Report <i>Sara Di Bartolomeo, Fabian Klute, Debajyoti Mondal, and Jules Wulms</i>	41:1–41:13
--	------------

Poster Abstracts

From Planar via Outerplanar to Outerpath – Engineering NP-Hardness Constructions <i>Joshua Geis and Johannes Zink</i>	42:1–42:3
Minimizing Switches in Cased Graph Drawings <i>Robert Ganian, Martin Nöllenburg, and Sebastian Röder</i>	43:1–43:3
Graph-Drawing Supported Identification of Influential Students at Schools <i>Markus Chimani, Lea Kröger, Juliane Liedtke, Jonah Mevert, Maor Shani, and Maarten van Zalk</i>	44:1–44:3
GdMetriX – A NetworkX Extension For Graph Drawing Metrics <i>Martin Nöllenburg, Sebastian Röder, and Markus Wallinger</i>	45:1–45:3
AdMaTilE: Visualizing Event-Based Adjacency Matrices in a Multiple-Coordinated-Views System <i>Nikolaus-Mathias Herl and Velitchko Filipov</i>	46:1–46:3
Strict Upward Planar Grid Drawings of Binary Trees with Minimal Area <i>Maarten Löffler</i>	47:1–47:3
Determining Sugiyama Topology with Model Order <i>Sören Domrös and Reinhard von Hanxleden</i>	48:1–48:3
Introducing Fairness in Graph Visualization <i>Seok-Hee Hong, Giuseppe Liotta, Fabrizio Montecchiani, Martin Nöllenburg, and Tommaso Piselli</i>	49:1–49:3
Level Planarity Is More Difficult Than We Thought <i>Simon D. Fink, Matthias Pfretzschner, Ignaz Rutter, and Peter Stumpf</i>	50:1–50:3
Evolutionary Algorithms for One-Sided Bipartite Crossing Minimisation <i>Jakob Baumann, Ignaz Rutter, and Dirk Sudholt</i>	51:1–51:3
Polygonally Anchored Graph Drawing <i>Alvin Chiu, Ahmed Eldawy, and Michael T. Goodrich</i>	52:1–52:3
String Graph with Cop Number 4 <i>Stephane Durocher, Myroslav Kryven, and Maarten Löffler</i>	53:1–53:3
Approximating the Crossing Number of Dense Graphs <i>Oriol Solé Pi</i>	54:1–54:3
 Software Abstracts	
yFiles – From Data to Meaningful Visualizations <i>Evmorfia Argyriou and Benjamin Niedermann</i>	55:1–55:3
The Eclipse Layout Kernel <i>Maximilian Kasperowski, Sören Domrös, and Reinhard von Hanxleden</i>	56:1–56:2
Immersive Analytics of Graphs in Virtual Reality with GAV-VR <i>Stefan P. Feyer, Wilhelm Kerle-Malcharek, Ying Zhang, Falk Schreiber, and Karsten Klein</i>	57:1–57:3

Graph Harvester <i>Julius Deynet, Tim Hegemann, Sebastian Kempf, and Alexander Wolff</i>	58:1–58:3
CentralityViz: Comprehending Node-Centrality in Large Networks <i>Garima Jindal and Kamalakar Karlapalem</i>	59:1–59:3
NodeXL – A few Clicks to Network Insights <i>Harald Meier and Arber Ceni</i>	60:1–60:2
Knowledge Graph Builder – Constructing a Graph from Arbitrary Text Using an LLM <i>Andreas Benno Kollegger, Alexander Erdl, and Michael Hunger</i>	61:1–61:2

■ Preface

This volume contains the papers presented at GD 2024, the 32nd International Symposium on Graph Drawing and Network Visualization, held during September 18–20, 2024 in Vienna, Austria. Graph Drawing is concerned with the geometric representation of graphs and constitutes the algorithmic core of Network Visualization. Graph Drawing and Network Visualization are motivated by applications where it is crucial to visually analyze and interact with relational datasets. Examples of such application areas include data science, social sciences, web computing, information systems, biology, geography, business intelligence, information security and software engineering.

The symposium on Graph Drawing and Network Visualization has been the main annual event in this area for more than 30 years. Its focus is on combinatorial and algorithmic aspects of graph drawing as well as the design of network visualization systems and interfaces. Information about the conference series and past symposia is maintained at <http://www.graphdrawing.org>.

A total of 125 participants from 20 different countries attended the conference. Regular papers could be submitted to one of two distinct tracks: Track 1 for papers on fundamental theoretical graph drawing advances, such as on combinatorial and algorithmic aspects, and Track 2 for papers on practical aspects of graph drawing, such as experimental, applied, and network visualization aspects. Short papers were given a separate category, which welcomed both theoretical and applied contributions. An additional track was devoted to poster submissions, and short abstracts were invited for describing software participating in the software exhibition.

All the tracks were handled by a single Program Committee. As in previous editions of GD, the papers in the different tracks did not compete directly with each other, but all program committee members were invited to review papers from either track in a “light-weight double-blind” process.

In response to the call for papers, the Program Committee received a total of 108 submissions, consisting of 94 papers (77 in Track 1, 17 in Track 2, and 21 in the short paper category) and 14 posters. Close to 300 reviews were provided, almost half having been contributed by external sub-reviewers. After extensive electronic discussions by the Program Committee via EasyChair, interspersed with virtual meetings of the Program Chairs producing incremental accept/reject proposals, 31 long papers, 7 short papers, and 13 posters were selected for inclusion in the scientific program of GD 2024. This resulted in an overall paper acceptance rate (not considering posters) of 40% (41% in Track 1, 35% in Track 2, and 33% in the short paper category). As is common in GD, some hard choices had to be made in particular during the final acceptance/rejection round, where several papers that clearly had merit still did not make the cut. However, the number of submitted high-quality papers speaks for the community.

For the first time in GD history, the proceedings are published by LIPICs, creating an additional challenge for authors and editors as the established schedule and workflow for the creation of the proceedings had to be changed. The generous support by the Dagstuhl LIPICs team helped to ease efforts in this transition.

There were two invited talks at GD 2024. Otfried Cheong from SCALGO (Denmark) gave answers to the question “How Can Biclique Covers Help in Matching Problems” and Monika Henzinger from the Institute of Science and Technology Austria discussed the question “How Can Algorithms Help in Protecting our Privacy”. Abstracts of both invited talks are included in these proceedings.



The conference gave out best paper awards in Track 1 and Track 2, as well as a best presentation award and a best poster award. The award for the best paper in Track 1 was given to “The Density Formula: One Lemma to Bound Them All” by Michael Kaufmann, Boris Klemz, Kristin Knorr, Meghana M. Reddy, Felix Schröder and Torsten Ueckerdt (Paper 7 in this volume), and the award for the best paper in Track 2 was given to “GraphTrials: Visual Proofs of Graph Properties” by Henry Förster, Felix Klesen, Tim Dwyer, Peter Eades, Seok-Hee Hong, Stephen Kobourov, Giuseppe Liotta, Kazuo Misue, Fabrizio Montecchiani, Alexander Pastukhov and Falk Schreiber (Paper 16). Based on a majority vote of the conference participants, the best presentation award was given to Soeren Terziadis for the presentation of “Boundary Labeling in a Circular Orbit” (Paper 22) and the best poster award was given to Simon D. Fink, Matthias Pfretzschner, Ignaz Rutter, and Peter Stumpf for the poster titled “Level Planarity Is More Difficult Than We Thought” (Paper 50).

A PhD School was held on the two days prior to the conference. Four half-day sessions led by Philipp Kindermann, Tamara Mchedlidze, Thekla Hamm, and Manfred Scheucher covered both theoretical and practical topics in graph drawing and network visualization.

As is traditional, the 31st Annual Graph Drawing Contest was held during the conference. The contest was divided into two parts, creative topics and the live challenge. The creative topics task featured a single graph that represents the history of medal wins by countries at the Olympic games. The live challenge focused on minimizing the number of crossings on point set embeddings. There were two categories: manual and automatic. We thank the Contest Committee, chaired by Fabian Klute, for preparing interesting and challenging contest problems. A report on the contest is included in these proceedings.

Many people and organizations contributed to the success of GD 2024. We would like to thank all members of the Program Committee and the external reviewers for carefully reviewing and discussing the submitted papers and posters; this was crucial for putting together a strong and interesting program. We would also like to thank all authors who chose GD 2024 as the publication venue for their research and the participants that showcased their software.

We are grateful for the support of our “Gold” sponsors Tom Sawyer Software and yWorks, our “Bronze” sponsor Neo4J, and our “Contributors” vrvis and TU Wien. Their generous support helped to ensure the continued success of this conference. Furthermore, we would like to thank the LIPIcs team for their support during the creation of these proceedings.

Our special thanks go to all the members of the organizing committee based at TU Wien and St. Pölten University of Applied Sciences, Austria, in particular to the co-chairs Martin Nöllenburg and Robert Ganian.

The 33rd International Symposium on Graph Drawing and Network Visualization (GD 2025) will take place during September 24–26, 2025, in Norrköping, Sweden, and will be hosted by Linköping University. Andreas Kerren and Kostiantyn Kucher will co-chair the Organizing Committee, and Vida Dujmović and Fabrizio Montecchiani will co-chair the Program Committee.

■ Organization

Program Committee

- Daniel Archambault, Newcastle University
- David Auber, University of Bordeaux
- Benjamin Bach, University of Edinburgh
- Martin Balko, Charles University in Prague
- Carla Binucci, University of Perugia
- Giuseppe Di Battista, Third University of Rome
- Vida Dujmović, University of Ottawa
- Tim Dwyer, Monash University
- David Eppstein, University of California, Irvine
- Stefan Felsner (co-chair), Technische Universität Berlin
- Michael Hoffmann, ETH Zürich
- Seok-Hee Hong, University of Sydney
- Yifan Hu, Amazon
- Tony Huang, University of Technology Sydney
- Jonathan Klawitter, University of Auckland
- Karsten Klein (co-chair), Konstanz University
- Linda Kleist, Technische Universität Braunschweig
- Boris Klemz, University of Würzburg
- Maarten Löffler, Utrecht University
- Maurizio Patrignani, Third University of Rome
- Arnaud Sallaberry, University of Montpellier
- Manfred Scheucher, Technical University Berlin
- Alessandra Tappini, University of Perugia
- Géza Tóth, Rényi Institute, Budapest
- Torsten Ueckerdt, KIT Karlsruhe
- Birgit Vogtenhuber, Graz University of Technology
- Hsiang-Yun Wu, St. Pölten University of Applied Sciences

Organizing Committee

- Doris Brazda, TU Wien, Austria
- Thomas Depian, TU Wien, Austria
- Alexander Dobler, TU Wien, Austria
- Simon Dominik Fink, TU Wien, Austria
- Robert Ganian, TU Wien (co-chair), Austria
- Martin Nöllenburg, TU Wien (co-chair), Austria
- Hsiang-Yun Wu, St. Pölten University of Applied Sciences, Austria

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).
Editors: Stefan Felsner and Karsten Klein



Leibniz International Proceedings in Informatics
LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany


Graph Drawing Contest Committee

- Sara Di Bartolomeo, University of Konstanz, Germany
- Fabian Klute (chair), UPC Barcelona, Spain
- Wouter Meulemans, TU Eindhoven, Netherlands
- Debajyoti Mondal, University of Saskatchewan, Canada
- Jules Wulms, TU Eindhoven, Netherlands

Additional Reviewers

Ackerman, Eyal	Hamm, Thekla	Rahat, Alma
Alves Radtke, Yan	Hegemann, Tim	Rieck, Christian
Angelini, Patrizio	Itoh, Takayuki	Saumell, Maria
Antić, Todor	Jung, Attila	Schlipf, Lena
Behrisch, Michael	Kaufmann, Michael	Schnider, Patrick
Bekos, Michael	Kindermann, Philipp	Schröder, Felix
Bergold, Helena	Klesen, Felix	Silveira, Rodrigo
Biedl, Therese	Klost, Katharina	Soukup, Jan
Biniáz, Ahmad	Klute, Fabian	Stumpf, Peter
Brötzner, Anna	Kramer, Peter	Swanepoel, Konrad
Brückner, Guido	Kryven, Myroslav	Symvonis, Antonios
Caroppo, Susanna	Lin, Chun-Cheng	Takahashi, Shigeo
Chang, Hsien-Chih	Liotta, Giuseppe	Tancer, Martin
Chaplick, Steven	McGee, Fintan	Tejel, Javier
Chimani, Markus	Meidiana, Amyra	Telea, Alexandru
Cortes Kühnast, Fernando	Melançon, Guy	Terziadis, Soeren
Cuenca, Erick	Merker, Laura	Tollis, Ioannis
D'Elia, Marco	Miller, Jacob	Toth, Csaba
Da Lozzo, Giordano	Montecchiani, Fabrizio	Valla, Tomáš
De Meyer, Lucas	Mooney, Gavin J.	Weinberger, Alexandra
Di Giacomo, Emilio	Morin, Pat	Wesolek, Alexandra
Didimo, Walter	Mulzer, Wolfgang	Wittmann, Maximilian Tito
Filipov, Velitchko	Nederlof, Jesper	Wolff, Alexander
Firman, Oksana	Nöllenburg, Martin	Wulf, Lasse
Frati, Fabrizio	Ortali, Giacomo	Wulms, Jules
Fulek, Radoslav	Orthaber, Joachim	Wybrow, Michael
Förster, Henry	Pahr, Daniel	Yen, Hsu-Chun
Giovannangeli, Loann	Pelsmajer, Michael	Zaidi, Faraz
Glišić, Jelena	Perz, Daniel	Zeng, Ji
Goetze, Miriam	Piselli, Tommaso	Zink, Johannes
Gonçalves, Daniel	Pournin, Lionel	van Renssen, André
Grilli, Luca	Pupyrev, Sergey	
Grosso, Fabrizio	Purchase, Helen	

How Can Biclique Covers Help in Matching Problems

Otfried Cheong   

SCALGO, Aarhus, Denmark

Abstract

In several settings one encounters assignment or matching problems between objects of two different types, and needs to run a computation on a bipartite graph. While this graph can potentially be dense, it can sometimes be represented compactly using a biclique cover. This is in particular often the case when the objects are geometric – we will look at examples, and see how recent progress on maximum flow can be combined with such biclique covers to obtain faster algorithms.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases Matching problems

Digital Object Identifier 10.4230/LIPIcs.GD.2024.1

Category Invited Talk



© Otfried Cheong;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).



Editors: Stefan Felsner and Karsten Klein; Article No. 1; pp. 1:1–1:1

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

How Can Algorithms Help in Protecting Our Privacy

Monika Henzinger   

Institute of Science and Technology, Klosterneuburg, Austria

Abstract

Decisions are increasingly automated using rules that were learnt from personal data. Thus, it is important to guarantee that the privacy of the data is protected during the learning process. To formalize the notion of an algorithm that protects the privacy of its data, differential privacy was introduced by Dwork, McSherry, Nissim, and Smith in 2006. It is a rigorous mathematical definition to analyze the privacy properties of an algorithm – or the lack thereof. In this talk I will give an introduction to differential privacy with an emphasis on differential private algorithms that can handle dynamically changing input data.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases Matching problems

Digital Object Identifier 10.4230/LIPIcs.GD.2024.2

Category Invited Talk

Funding This work has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (Grant agreement No. 101019564) and the Austrian Science Fund (FWF) grant DOI 10.55776/Z422, grant DOI 10.55776/I5982, and grant DOI 10.55776/P33775 with additional funding from the netidee SCIENCE Stiftung, 2020–2024.



© Monika Henzinger;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).



Editors: Stefan Felsner and Karsten Klein; Article No. 2; pp. 2:1–2:1

Leibniz International Proceedings in Informatics





LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



The Euclidean MST-Ratio for Bi-Colored Lattices

Sebastiano Cultrera di Montesano  

Institute of Science and Technology Austria (ISTA), Klosterneuburg, Austria

Ondřej Draganov  

Institute of Science and Technology Austria (ISTA), Klosterneuburg, Austria

Herbert Edelsbrunner  

Institute of Science and Technology Austria (ISTA), Klosterneuburg, Austria

Morteza Saghafian  

Institute of Science and Technology Austria (ISTA), Klosterneuburg, Austria

Abstract

Given a finite set, $A \subseteq \mathbb{R}^2$, and a subset, $B \subseteq A$, the *MST-ratio* is the combined length of the minimum spanning trees of B and $A \setminus B$ divided by the length of the minimum spanning tree of A . The question of the supremum, over all sets A , of the maximum, over all subsets B , is related to the Steiner ratio, and we prove this sup-max is between 2.154 and 2.427. Restricting ourselves to 2-dimensional lattices, we prove that the sup-max is 2, while the inf-max is 1.25. By some margin the most difficult of these results is the upper bound for the inf-max, which we prove by showing that the hexagonal lattice cannot have MST-ratio larger than 1.25.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases Minimum spanning Trees, Steiner Ratio, Lattices, Partitions

Digital Object Identifier 10.4230/LIPIcs.GD.2024.3

Related Version *arXiv Version*: <https://arxiv.org/abs/2403.10204>

Funding This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme, grant no. 788183, from the Wittgenstein Prize, Austrian Science Fund (FWF), grant no. Z 342-N31, and from the DFG Collaborative Research Center TRR 109, “Discretization in Geometry and Dynamics”, Austrian Science Fund (FWF), grant no. I 02979-N35.

1 Introduction

The recent development of measuring the interaction between two or more sets of points with methods from topological data analysis motivates the discrete geometric question about minimum spanning trees studied in this paper; see [1, 7] for background in this general area. We refer to the measured interaction as *mingling*, in which higher values corresponding to more mingling. The ambiguity of the term is deliberate and leaves the concrete meaning to the geometric and algebraic constructions described in [5]. As explained in the appendix of the current paper, one of these measurements can be expressed in elementary terms:

► **Definition.** *Given a finite set, $A \subseteq \mathbb{R}^2$, we write $\text{MST}(A)$ for the (Euclidean) minimum spanning tree of the complete graph on A , with edge weights equal to the distances between the points. For $B \subseteq A$, the MST-ratio of A and B is the combined length of the minimum spanning trees of B and $A \setminus B$, divided by the length of the minimum spanning tree of A :*

$$\mu(A, B) = \frac{|\text{MST}(B)| + |\text{MST}(A \setminus B)|}{|\text{MST}(A)|}. \quad (1)$$



© Sebastiano Cultrera di Montesano, Ondřej Draganov, Herbert Edelsbrunner, and Morteza Saghafian; licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 3; pp. 3:1–3:23

Leibniz International Proceedings in Informatics



LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

To make use of this measure for statistical or other purposes, we ought to know how small and how large the ratio can get (the extremal question), and how it behaves for random data. A first result in the latter direction can be found in [6], who prove that for points chosen uniformly at random in the unit square, the expected MST-ratio for a random partition into two subsets is at least $\sqrt{2} - \varepsilon$, for any $\varepsilon > 0$. In the non-random setting, we study the maximum MST-ratio, over all partitions of A into two sets, and consider both the infimum and supremum of the maximum, over all sets in a class of point sets. If these sets are infinite, like for example 2-dimensional lattices, then we talk about the supremum rather than the maximum MST-ratio.

Given any set, A , the minimum MST-ratio is achieved by removing the longest edge from $\text{MST}(A)$ and letting B and $A \setminus B$ be the vertices of the resulting two trees, so it is less than 1. Indeed, any other partition of A would produce two minimum spanning trees that together are at least as long as $\text{MST}(B)$ and $\text{MST}(A \setminus B)$. More interestingly, the maximum MST-ratio is related to the *Steiner ratio* of the Euclidean plane [8, 10], and we exploit this connection to prove that the supremum is between 2.154 and 2.427 (Theorem 2.1 in Section 2). The infimum of the maximum is again less interesting: allowing ourselves to pick points arbitrarily close to each other, and one far away, this infimum can be seen to be arbitrarily close to 1.

This motivates us to study the MST-ratio for a restricted class of sets, and our choice are the (Euclidean) lattices, which are well studied objects with many applications in mathematics and beyond; see e.g. [12]. Since we optimize over subsets of an infinite set, we talk about the supremum rather than the maximum, and taking a sequence of progressively larger but finite portions of such a lattice, we have well defined minimum spanning trees and can study the asymptotic behavior of the MST-ratio. Our main result is that the supremum MST-ratio of the hexagonal lattice is 1.25 (Theorem 4.2 in Section 4). Observe that this is an upper bound on the infimum, over all lattices, of the supremum MST-ratio. We complement this with a matching lower bound (Claim 3.5 in Section 3), and with matching lower and upper bounds for the supremum, over all lattices, of the supremum MST-ratio, which we establish is 2 (Claims 3.2 and 3.4 in Section 3).

2 The Maximum MST-ratio for Finite Sets

The main question we ask is to what extent two minimum spanning trees can be longer than a single minimum spanning tree of the same points; see the definition of the MST-ratio of a set $A \subseteq \mathbb{R}^2$ and a subset $B \subseteq A$ in the introduction. We are interested in the maximum MST-ratio, over all subsets $B \subseteq A$, and in the supremum and infimum of this maximum, over all finite sets $A \subseteq \mathbb{R}^2$.

The supremum is related to the well-studied Steiner tree problem. Given a finite set, $X \subseteq \mathbb{R}^2$, the *Steiner tree* of X is the minimum spanning tree of $X \cup B$, in which $B = B(X)$ is chosen to minimize the length of this tree. The *Steiner ratio* of the Euclidean plane is the infimum of the length ratio, $|\text{MST}(X \cup B)|/|\text{MST}(X)|$, over all finite sets X and B in the plane. There are sets $X \subseteq \mathbb{R}^2$ for which the ratio is only $\sqrt{3}/2 = 0.866\dots$; take for example the vertices of an equilateral triangle as X and the barycenter of this triangle as the sole point in B . It is conjectured that $\sqrt{3}/2$ is the Steiner ratio of the Euclidean plane [8], but the current best lower bound proved in [2] is only $0.824\dots$. We use this bound to prove upper and lower bounds for the supremum maximum MST-ratio:

► **Theorem 2.1.** *The supremum, over all finite $A \subseteq \mathbb{R}^2$, of the maximum, over all subsets $B \subseteq A$, of the MST-ratio satisfies $2.154 \leq \sup_A \max_B \mu(A, B) \leq 2.427$.*

Proof. We first prove the upper bound. Since B is a subset of A , the MST of A cannot be shorter than the Steiner tree of B . Similarly, the MST of A cannot be shorter than the Steiner tree of $A \setminus B$. Hence, $|\text{MST}(A)| \geq 0.824\dots \cdot |\text{MST}(B)|$ and $|\text{MST}(A)| \geq 0.824\dots \cdot |\text{MST}(A \setminus B)|$. It follows that the ratio satisfies

$$\mu(A, B) \leq \frac{2 \cdot [|\text{MST}(B)| + |\text{MST}(A \setminus B)|]}{0.824\dots \cdot [|\text{MST}(B)| + |\text{MST}(A \setminus B)|]} = 2.426\dots \quad (2)$$

This inequality holds for every $B \subseteq A$. We second prove the lower bound for the sup-max by constructing a set A of seven points that implies the inequality. Let $B \subseteq A$ be the three vertices of an equilateral triangle with unit length edges, and let $A \setminus B$ be the vertices of another equilateral triangle with unit length edges, but this time together with the barycenter. Hence, $|\text{MST}(B)| = 2$ and $|\text{MST}(A \setminus B)| = \sqrt{3}$. Assuming the distance between corresponding vertices of the two equilateral triangles is less than $\varepsilon > 0$, we have $|\text{MST}(A)| < \sqrt{3} + 3\varepsilon$. This implies

$$\mu(A, B) > \frac{2 + \sqrt{3}}{\sqrt{3} + 3\varepsilon} > 2.154\dots - 4\varepsilon. \quad (3)$$

Since we can make $\varepsilon > 0$ arbitrarily small, this implies the claimed lower bound. \blacktriangleleft

The example used to establish the lower bound can be extended to larger numbers of points, e.g. the following disjoint union of three lattices: B is the hexagonal lattice (to be defined shortly), and $A \setminus B$ is a slightly shifted copy of the hexagonal lattice, together with the barycenters of the triangles in every fourth row, which is a rectangular lattice with distances 1 and $\sqrt{3}$ between consecutive rows and columns.

The question about the infimum of the maximum MST-ratio turns out to be less interesting, with 1 as answer. To see the lower bound, set $B = A$, in which case $|\text{MST}(B)| = |\text{MST}(A)|$ and $|\text{MST}(A \setminus B)| = 0$. The ratio is therefore 1. We get the upper bound by constructing a set A of $n \geq 2$ points. It contains the origin, $n - 2$ points each at distance at most $\varepsilon > 0$ from the origin, and another point, which we call b , at unit distance from the origin. Assume $b \in B$, and consider the case in which B contains at least one other point of A . Then

$$1 \leq |\text{MST}(A)| \leq 1 + 2(n - 2)\varepsilon, \quad (4)$$

$$1 - \varepsilon \leq |\text{MST}(B)| \leq 1 + 2(n - 2)\varepsilon, \quad (5)$$

$$0 \leq |\text{MST}(A \setminus B)| \leq 2(n - 3)\varepsilon. \quad (6)$$

For any given $\delta > 0$, we can choose $\varepsilon > 0$ sufficiently small such that the ratio is smaller than $1 + \delta$. In the other case, in which $B = \{b\}$, we have $|\text{MST}(B)| = 0$ and $|\text{MST}(A \setminus B)| \leq 2(n - 2)\varepsilon$, so we can make the ratio arbitrarily small and certainly smaller than 1.

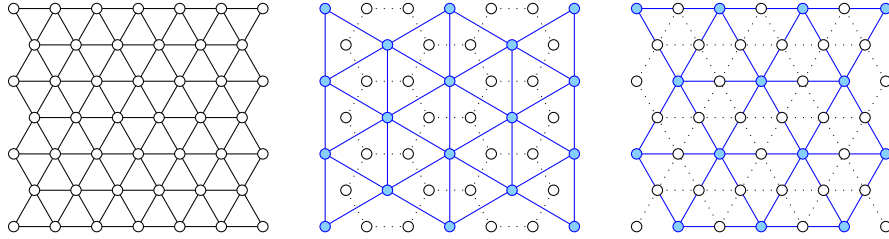
3 Two-dimensional Lattices

Motivated by the triviality of the infimum maximum MST-ratio for general finite sets, we aim for a restriction that disallows extremely unbalanced distributions. There are many choices, and we opt for a maximally restricted setting in which the MST-ratio is still an interesting question. Specifically, we focus on 2-dimensional lattices.

► Definition. *The (Euclidean) lattice spanned by two linearly independent vectors, $\mathbf{u}, \mathbf{v} \in \mathbb{R}^2$, consists of all integer combinations of these vectors: $\Lambda(\mathbf{u}, \mathbf{v}) = \{i\mathbf{u} + j\mathbf{v} \mid i, j \in \mathbb{Z}\}$.*

3:4 The Euclidean MST-Ratio of Bi-Colored Lattices

By definition, lattices are infinite. To cope with the difficulty of constructing the minimum spanning tree of infinitely many points, we take progressively larger but finite portions of a lattice and monitor the sequence of MST-ratios. Specifically, we fix a partition of the infinite lattice, take rhombi centered at the origin and spanned by the vectors of the shortest basis of the lattice, for each rhombus get the MST-ratio for the points inside the rhombus, and consider the sequence of MST-ratios as the size of the rhombus increases. If this sequence converges, we call the limit the *MST-ratio* of the chosen partition of the lattice.



■ **Figure 1** *Left*: a portion of the hexagonal lattice and all its shortest edges. *Middle*: a partition into one and two thirds of the points, with MST-ratio converging to $(2 + \sqrt{3})/3 = 1.245 \dots$. *Right*: a partition into one and three quarters of the points, with MST-ratio converging to 1.25.

A particularly interesting lattice is the *triangular* or *hexagonal lattice*, which is spanned by $\mathbf{u} = (1, 0)$ and $\mathbf{v} = \frac{1}{2}(1, \sqrt{3})$; see the left panel in Figure 1. The minimum distance between its points is 1, so all edges of the MST have length 1. The two partitions illustrated in the middle and right panels of Figure 1 have MST-ratios $1.245 \dots$ and 1.25 , respectively. In one way or another, we use this lattice to prove all four bounds claimed in the following theorem.

► **Theorem 3.1.** *The supremum and infimum, over all 2-dimensional lattices, Λ , of the supremum, over all subsets, $B \subseteq \Lambda$, of the MST-ratio are $C_0 = \sup_{\Lambda} \sup_B \mu(\Lambda, B) = 2$ and $c_0 = \inf_{\Lambda} \sup_B \mu(\Lambda, B) = 1.25$.*

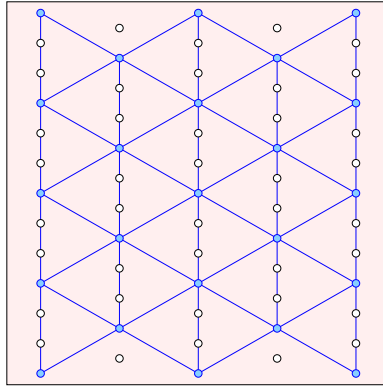
Each of the subsequent subsections restates and proves one of the four bounds, except for the last subsection, which only sketches the proof strategy, with the proof presented in Section 4.

3.1 Lower Bound for Sup-Sup

This subsection exhibits a lattice, and a partition of this lattice into two sets, such that the MST-ratio of progressively larger finite portions of the lattice approaches the supremum of the supremum MST-ratio claimed in Theorem 3.1 from below.

▷ **Claim 3.2.** $C_0 \geq 2$.

Proof. Let Λ be the hexagonal lattice horizontally stretched by a factor 9, and let $B \subseteq \Lambda$ be the one third of the points drawn blue in Figure 2. The (vertical) distance between neighboring points in a column of Λ is $\sqrt{3}$, and the (horizontal) distance between two neighboring columns is $\frac{9}{2}$. For each $r \geq 0$, let $\Lambda_r \subseteq \Lambda$ and $B_r \subseteq B$ be the points in $[-r, r]^2$. Hence, Λ_r consists of $p_r = 2\lfloor 2r/9 \rfloor + 1$ vertical columns, which alternate between $q_r = 2\lfloor r/\sqrt{3} \rfloor + 1$ and $q_r - 1$ or $q_r + 1$ points. Observe that p_r and q_r are both odd, and that $n_r = q_r p_r \pm (p_r - 1)/2$ is the cardinality of Λ_r . The number of points of B_r in the columns alternates between $b_r = 2\lfloor r/(3\sqrt{3}) \rfloor + 1$ and $b_r - 1$ or $b_r + 1$, so $m_r = b_r p_r \pm (p_r - 1)$ is the cardinality of B_r . It is easy to see that $n_r - 2p_r \leq 3m_r \leq n_r + 2p_r$.



■ **Figure 2** The portion of the horizontally stretched hexagonal lattice, Λ , and the subset of *blue* points, B , inside a square centered at the origin. The edges show the union of all possible minimum spanning trees of the *blue* points.

By choice of the stretch factor, B is a hexagonal lattice with distance $3\sqrt{3}$ between closest points. Hence, $|\text{MST}(B_r)| = 3\sqrt{3}(m_r - 1)$. Compare this with a minimum spanning tree of Λ_r , which first connects the points in each column and second connects neighboring columns with one edge for each pair. Hence,

$$|\text{MST}(\Lambda_r)| = \sqrt{3}(n_r - p_r) + \sqrt{21}(p_r - 1), \quad (7)$$

because every point, except the last in each column, has a neighbor at distance $\sqrt{3}$ below, and any two neighboring columns have points at distance $\sqrt{21}$ from each other. Similarly, any minimum spanning tree of $\Lambda_r \setminus B_r$ first connects the points in each column and second connects neighboring columns with one edge for each pair. Its length is therefore the same as that of $\text{MST}(\Lambda_r)$. Using $3m_r = n_r + o(n_r)$, this implies

$$\frac{|\text{MST}(B_r)| + |\text{MST}(\Lambda_r \setminus B_r)|}{|\text{MST}(\Lambda_r)|} = \frac{3\sqrt{3}(m_r - 1) + \sqrt{3}(n_r - p_r) + \sqrt{21}(p_r - 1)}{\sqrt{3}(n_r - p_r) + \sqrt{21}(p_r - 1)} \quad (8)$$

$$= \frac{2\sqrt{3}n_r + o(n_r)}{\sqrt{3}n_r + o(n_r)} \xrightarrow{r \rightarrow \infty} 2. \quad (9)$$

For any $\varepsilon > 0$, we can choose r sufficiently large such that the MST-ratio exceeds $2 - \varepsilon$, which implies the claimed lower bound. \triangleleft

3.2 Upper Bound for Sup-Sup

This subsection proves the upper bound that matched the lower bound established in the preceding subsection. Given any lattice and any partition of this lattice into two sets, we show that for any $\varepsilon > 0$, the MST-ratio cannot exceed $2 + \varepsilon$. We begin with a bound for the length of the minimum spanning tree of any finite set in a square.

► **Lemma 3.3.** *The length of the minimum spanning tree of any n or fewer points in $[0, n]^2$ is at most $2n\sqrt{n}$.*

Proof. Assuming the number of points is $k \leq n$, the minimum spanning tree has $k - 1$ edges, and we write $\ell_1, \ell_2, \dots, \ell_{k-1}$ for their lengths. The sum of the squares of these lengths is at most $4n^2$, as proved in [8]. By the Cauchy–Schwarz inequality, the sum of the ℓ_i is maximized when all terms are the same, namely $\ell_i^2 = 4n^2/(k - 1)$ for all i . This implies

$$\sum_{i=1}^{k-1} \ell_i \leq (k - 1)\sqrt{4n^2/(k - 1)} = 2n\sqrt{k - 1}, \quad (10)$$

from which the claimed bound follows. \blacktriangleleft

3:6 The Euclidean MST-Ratio of Bi-Colored Lattices

Lemma 3.3 will provide a crucial step in the proof of the upper bound for the supremum maximum MST-ratio, which we present next.

▷ Claim 3.4. $C_0 \leq 2$.

Proof. We show that the MST-ratio of any lattice $\Lambda \subseteq \mathbb{R}^2$ and any subset $B \subseteq \Lambda$ is at most the claimed upper bound. Let \mathbf{u} be the shortest non-zero vector in Λ , and \mathbf{v} the shortest non-zero vector that is not a multiple of \mathbf{u} , breaking ties arbitrarily if necessary. Suppose their lengths satisfy $1 = \|\mathbf{u}\| \leq \|\mathbf{v}\| = \nu$. To simplify language, we call the points on a line parallel to \mathbf{u} a *row* of Λ . For every positive integer, n , let $\Lambda_n \subseteq \Lambda$ contain all points $\alpha\mathbf{u} + \beta\mathbf{v}$, with $0 \leq \alpha, \beta \leq n$. The minimum spanning tree of Λ_n first connects the points in each row and then the neighboring rows, so

$$|\text{MST}(\Lambda_n)| = (n+1)n + n\nu. \quad (11)$$

Set $B_n = B \cap \Lambda_n$. We construct a spanning tree, $T(B_n)$, by first connecting the points within the rows. This allows for the possibility that some rows do not contain any points of B_n . In each of the other rows, we choose an arbitrary but fixed point of B_n , write $B'_n \subseteq B_n$ for the chosen points, construct $\text{MST}(B'_n)$, and add its edges to $T(B_n)$. Since $T(B_n)$ spans B_n but is not necessarily the shortest such tree, so $|\text{MST}(B_n)| \leq |T(B_n)|$. To bound the latter, recall that there are $n+1$ rows, each of length at most n . Furthermore, B'_n consists of at most $n+1$ points that fit inside a square of side length $n(\nu+1)$, in which ν is independent of n . Lemma 3.3 implies $|\text{MST}(B'_n)| \leq 2(\nu+1)\sqrt{\nu+1} \cdot n\sqrt{n}$. Hence,

$$|\text{MST}(B_n)| \leq (n+1)n + 2(\nu+1)\sqrt{\nu+1} \cdot n\sqrt{n}. \quad (12)$$

By symmetry, we have the same upper bound for the length of $\text{MST}(\Lambda_n \setminus B_n)$. Comparing this with the minimum spanning tree of Λ_n , we get

$$\frac{|\text{MST}(B_n)| + |\text{MST}(\Lambda_n \setminus B_n)|}{|\text{MST}(\Lambda_n)|} \leq \frac{2n^2 + 2n + 4(\nu+1)^{3/2} \cdot n\sqrt{n}}{n^2 + n + \nu n} \xrightarrow{n \rightarrow \infty} 2. \quad (13)$$

For every $\varepsilon > 0$, we can choose n large enough so that the MST-ratio is less than $2 + \varepsilon$. This works for every lattice and partition, which implies the claimed upper bound. ◁

3.3 Lower Bound for Inf-Sup

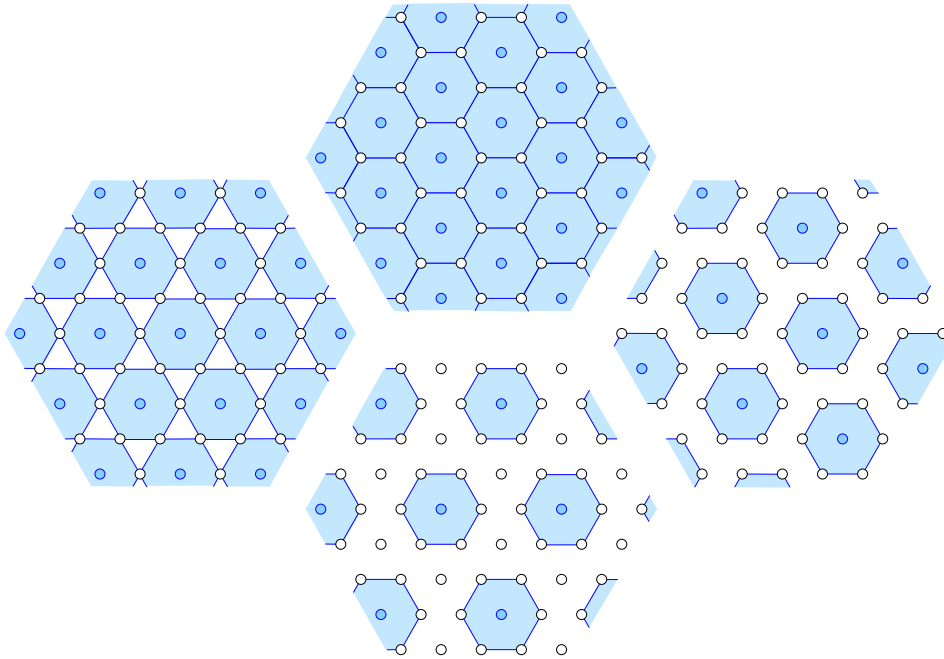
This subsection establishes the lower bound for the infimum, over all lattices, of the supremum MST-ratio. We do this by establishing a partition into one and three quarters that can be defined for any lattice and has MST-ratio at least as large as claimed in Theorem 3.1.

▷ Claim 3.5. $c_0 \geq 1.25$.

Proof. Let \mathbf{u} and \mathbf{v} be two vectors spanning Λ , and let B be the sublattice spanned by $2\mathbf{u}$ and $2\mathbf{v}$. Assuming the minimum distance between two points in Λ is 1, most edges of $\text{MST}(\Lambda)$ have length 1, while most edges of $\text{MST}(B)$ have length 2. Write $\Lambda_n \subseteq \Lambda$ for the points $i\mathbf{u} + j\mathbf{v}$, with $-2n \leq i, j \leq 2n+1$, and $B_n \subseteq \Lambda_n$ for the points with even i and j . Since B_n contains only a quarter of the points, this implies $\lim_{n \rightarrow \infty} |\text{MST}(B_n)|/|\text{MST}(\Lambda_n)| = \frac{1}{2}$. The complement of B_n contains three quarters of the points, and the edges in its minimum spanning tree have length at least 1, which implies $\lim_{n \rightarrow \infty} |\text{MST}(\Lambda_n \setminus B_n)|/|\text{MST}(\Lambda_n)| \geq \frac{3}{4}$. Hence, the MST-ratio of $B \subseteq \Lambda$ is at least $\frac{1}{2} + \frac{3}{4} = 1.25$. ◁

3.4 Upper Bound for Inf-Sup

The upper bound for the infimum of the supremum MST-ratio will be proved in Section 4. This proof is carefully constructed from a network of inequalities that require attention to detail. This subsection makes an argument why it is not unreasonable to believe that significant short-cuts may be difficult to find.



■ **Figure 3** Four partitions of the hexagonal lattice into two sets, in which we draw each (*blue*) point of the smaller set with its hexagonal neighborhood. The proportions of *blue* versus *white* points are 1 : 2 in the *upper middle*, 1 : 3 on the *left*, 1 : 6 on the *right*, and 1 : 8 in the *lower middle*. The corresponding MST-ratios are approximately 1.245, 1.25, 1.236, and 1.222, in this sequence.

The lattice that is most resistant to large MST-ratios is the hexagonal lattice, Λ , of which four different subsets, $B \subseteq \Lambda$, are illustrated as packings of hexagonal neighborhoods in Figure 3. Starting at the upper middle, then left, then right, and finally the lower middle, the density of the packing decreases monotonically as the minimum distance between points of B increases from $\sqrt{3}$ to 2, to $\sqrt{7}$, and finally to 3. Correspondingly, B contains one third, one quarter, one seventh, and one ninth of the points. Perhaps surprisingly, the MST-ratio does not vary monotonically and attains the largest value for the subset B that contains one quarter of the points. The purpose of Section 4 is to prove that no other subset of Λ achieves a larger MST-ratio; that is: 1.25 is the supremum MST-ratio of the hexagonal lattice.

▷ **Claim 3.6.** $c_0 \leq 1.25$.

Because the value matches the lower bound stated in Claim 3.5, this implies that 1.25 is indeed the infimum, over all 2-dimensional lattices, of the supremum MST-ratio. Prior to studying the hexagonal lattice, the authors of this paper proved that the supremum MST-ratio of the integer lattice is $\sqrt{2}$ – which happens to match the ratio found for random sets [6] – and the optimizing subset are the points whose coordinates add up to even integers. The proof is similar to the one for the hexagonal lattice presented in Section 4, and almost as long. If instead we consider the points whose coordinates add up to odd integers, we get

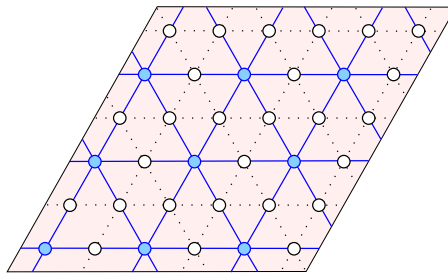
the same MST-ratio, so the integer lattice has at least two globally optimal partitions that are far from each other if the difference is measured in terms of the color changes needed to turn one into the other. Similarly, the hexagonal lattice has at least four globally optimal partitions, and moving from one to the other (by flipping colors) means walking a path along which the MST-ratio is sometimes barely below 1.25. To support the hypothesis of a rugged but shallow landscape, we conducted computational experiments for finite subsets of the integer lattice, which identified many local maxima that prevent local improvement strategies from reaching any global maximum. We feel that these findings justify the exhaustive case analysis in Section 4, and the many delicate inequalities in that section give evidence for how close the paths get to the supremum MST-ratio.

4 Hexagonal Lattice on Torus

In this section, we prove Claim 3.6 for the hexagonal lattice on the torus. We begin by constructing this lattice from a portion of the hexagonal lattice in the plane and proving that the minimum spanning trees in the two topologies are not very different in length. In the remaining subsections, we give a precise statement of the theorem that implies Claim 3.6 and prove the theorem with a packing argument in six steps.

4.1 Plane versus Torus

We consider the hexagonal lattice on the torus rather than in \mathbb{R}^2 in order to eliminate boundary effects, which appear when we study a finite portion of the hexagonal lattice. Let \mathbf{u} and \mathbf{v} be two unit vectors with a 60° degree angle between them, and write $\Lambda \subseteq \mathbb{R}^2$ for the hexagonal lattice they span. For every positive $n \in \mathbb{Z}$, let $\Lambda_n \subseteq \Lambda$ contain the n^2 points $a = \alpha\mathbf{u} + \beta\mathbf{v}$ with $0 \leq \alpha, \beta \leq n - 1$. We write Λ'_n for the same n^2 points but with the topology of the torus, which we get by identifying a with $a + i\mathbf{u} + j\mathbf{v}$ for all $i, j \in \mathbb{Z}$, and defining the distance as the minimum Euclidean distance between any two representatives. Equivalently, consider the rhombus of points $\varphi\mathbf{u} + \psi\mathbf{v}$ for real coefficients $-\frac{1}{2} \leq \varphi, \psi \leq n - \frac{1}{2}$,



■ **Figure 4** The hexagonal lattice of 36 points on the torus, obtained by gluing opposite sides of the rhombus. The sublattice with twice the distance between neighboring points is shown in blue.

and glue this rhombus along opposite sides as illustrated for $n = 6$ in Figure 4. Call the boundary of this rhombus the *seam*. Its length is $4n$ in the plane but only $2n$ on the torus since the sides are glued in pairs. Note also that every point of Λ has distance at least $\sqrt{3}/4$ from the nearest point in the seam.

► **Lemma 4.1.** *Let $\Lambda \subseteq \mathbb{R}^2$ be the hexagonal lattice, $\Lambda_n \subseteq \Lambda$ the subset of n^2 points, and Λ'_n the same n^2 points but on the torus, as described above. For any subset $B_n \subseteq \Lambda_n$ and the corresponding subset $B'_n \subseteq \Lambda'_n$ on the torus, the lengths of the minimum spanning trees satisfy $|\text{MST}(B'_n)| \leq |\text{MST}(B_n)| \leq |\text{MST}(B'_n)| + 32\sqrt{2} \cdot n\sqrt{n}$.*

Proof. Fix two minimum spanning trees, T of B_n in \mathbb{R}^2 and T' of B'_n on the torus. Since the distances on the torus are smaller than or equal to those in \mathbb{R}^2 , we have $|T'| \leq |T|$, which is the first claimed inequality. Let E' be the edges of T' that have the same length in both topologies, and let E'' be the other edges of T' , which are shorter on the torus than in \mathbb{R}^2 . To draw an edge of E'' in the plane so its length matches the length on the torus, we need to connect representatives of the endpoints that lie in different rhombi. Assuming one endpoint is in Λ_n , this edge crosses the seam. In contrast, every edge in E' can be drawn between two points of Λ_n , so without crossing the seam. We will prove shortly that the distance between two crossings measured along the seam is at least $\frac{1}{2}$. Since the length of the seam is $2n$, this implies that E'' contains at most $4n$ edges. Let $V'' \subseteq \Lambda_n$ be the set of at most $8n$ endpoints of the edges in E'' , and let T'' be a minimum spanning tree of V'' , with distances measured in \mathbb{R}^2 . Since Λ_n easily fits inside a square with sides of length $8n$, Lemma 3.3 implies $|T''| \leq 32\sqrt{2} \cdot n\sqrt{n}$. The edges in E' together with the edges of T'' form a connected graph with vertices Λ_n . Hence,

$$|T| \leq |T'| + |T''| \leq |T'| + 32\sqrt{2} \cdot n\sqrt{n}, \quad (14)$$

which is the second claimed inequality. It remains to show that the distance between two crossings along the seam is at least $\frac{1}{2}$. Let ab and xy be two edges in E'' , and recall that the greedy construction of the minimum spanning tree prohibits x and y to lie inside the smallest circle that passes through a and b , and vice versa. If the edges share an endpoint, then the angle between them is at least 60° . Since the common endpoint is at distance at least $\sqrt{3}/4$ from the seam, this implies the claimed lower bound on the distance between the two crossings. So assume a, b, x, y are distinct, and let $c \in ab$ and $z \in xy$ be the points that minimize the distance between the edges, and observe that $\|c - z\|$ is a lower bound for the distance between the crossings. At least one of c and z must be an endpoint, so suppose $z = x$. But since x lies outside the smallest circle of a and b , and outside the unit circles centered at a and b , the distance of x to any point of ab is at least 1. \blacktriangleleft

The inequalities in Lemma 3.3 generalize to all 2-dimensional lattices. Letting \mathbf{u} and \mathbf{v} be two shortest vectors that span a lattice, and assuming $1 = \|\mathbf{u}\| \leq \|\mathbf{v}\| = \nu$, we get $2(4 + 4\nu)^{3/2} \cdot n\sqrt{n}$ as an upper bound for the difference in length, in which we compare a rhombus of $n \times n$ points in \mathbb{R}^2 and on the torus, as before.

4.2 Statement of Theorem

We fix n to an even integer and write $\Delta = \Lambda'_n$ for the hexagonal lattice on the torus. Since n is even, $\Delta_1 = \{2x \mid x \in \Delta\}$ is a hexagonal sublattice of Δ , and we set $\Delta_3 = \Delta \setminus \Delta_1$; see Figure 4. The lengths of the three minimum spanning trees are easy to determine because they use only the shortest available edges, which have length 1 for Δ and Δ_3 , and length 2 for Δ_1 . The MST-ratio is therefore

$$\mu(\Delta, \Delta_1) = \frac{|\text{MST}(\Delta_1)| + |\text{MST}(\Delta_3)|}{|\text{MST}(\Delta)|} = \frac{2(n^2/4 - 1) + (3n^2/4 - 1)}{n^2 - 1} \xrightarrow{n \rightarrow \infty} 1.25. \quad (15)$$

Call an edge *short* if its length is 1. All other edges have length larger than the desired average, which is $\frac{5}{4} = 1.25$, so we call them *long*. While $\text{MST}(\Delta_3)$ has only short edges, and $\text{MST}(\Delta_1)$ uses only the shortest edges connecting its points, we claim that their combined length is as large as it can be.

► Theorem 4.2. *Let Δ be a hexagonal lattice on the torus. Then the maximum MST-ratio of Δ converges to $\frac{5}{4} = 1.25$ from below.*

The proof consists of six steps, which are presented in the same number of subsections: 4.3 introduces the hexagonal distance, compares its MST with the Euclidean MST, and uses the former to formulate the proof strategy; 4.4 introduces the main tool, which are hexagonal-neighborhoods of the lattice points; 4.5 constructs a hierarchy of such neighborhoods aimed at counting the short edges; 4.6 introduces so-called satellites, which provide additional short edges needed in the proof; 4.7 forms loop-free subgraphs of short edges and bounds their sizes; and 4.8 does the final accounting while paying special attention to the cases in which all long edges have length between $\sqrt{3}$ and 3. Throughout this proof, we use the fact that the minimum spanning tree can be computed by greedily adding the shortest available edge that does not form a cycle to the tree [9, 11].

4.3 Hexagonal Distance and Proof Strategy

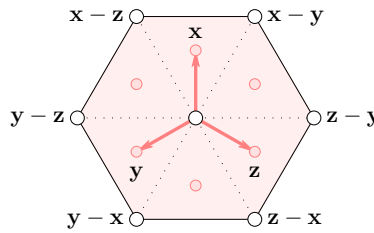
It is convenient to write the points in Δ with three integer coordinates. To explain this, let

$$\mathbf{x} = \frac{1}{\sqrt{3}}(0, 1), \quad \mathbf{y} = \frac{1}{\sqrt{3}}\left(-\frac{\sqrt{3}}{2}, -\frac{1}{2}\right), \quad \mathbf{z} = \frac{1}{\sqrt{3}}\left(\frac{\sqrt{3}}{2}, -\frac{1}{2}\right) \tag{16}$$

be three vectors, each of length $\sqrt{3}/3$, that mutually enclose an angle of 120° . These are the projections of the unit coordinate vectors of \mathbb{R}^3 onto the plane normal to the diagonal direction, scaled such that the three points are mutually one unit of distance apart. The plane consists of all points $u = a\mathbf{x} + b\mathbf{y} + c\mathbf{z}$ for which $a + b + c = 0$, and such a point belongs to the hexagonal lattice iff $a, b, c \in \mathbb{Z}$; see Figure 5. Given a second point, $v = \alpha\mathbf{x} + \beta\mathbf{y} + \gamma\mathbf{z}$, we write $i = a - \alpha$, $j = b - \beta$, $k = c - \gamma$ to compute the squared Euclidean distance between u and v . Since $\mathbf{x}^2 = \mathbf{y}^2 = \mathbf{z}^2 = \frac{1}{3}$ and $\mathbf{xy} = \mathbf{yz} = \mathbf{zx} = -\frac{1}{6}$, we get

$$\|u - v\|^2 = \|i\mathbf{x} + j\mathbf{y} + k\mathbf{z}\|^2 = \frac{1}{3}(i^2 + j^2 + k^2) - \frac{1}{3}(ij + ik + jk) = i^2 + ij + j^2, \tag{17}$$

in which we get the final expression using $k = -(i + j)$. For points of the hexagonal lattice, i and j are integers, and so is the squared Euclidean distance between them. It follows that the minimum distance between two points in Δ is 1.



■ **Figure 5** The unit disk under the hexagonal distance in the plane. The edges that connect the origin to the corners at $\pm(\mathbf{x} - \mathbf{y})$, $\pm(\mathbf{y} - \mathbf{z})$, $\pm(\mathbf{z} - \mathbf{x})$ decompose the hexagon into six equilateral triangles, whose barycenters are $\pm\mathbf{x}$, $\pm\mathbf{y}$, $\pm\mathbf{z}$.

We adapt the notion of distance to construct neighborhoods in the hexagonal lattice. By definition, the *hexagonal distance* between points $u = a\mathbf{x} + b\mathbf{y} + c\mathbf{z}$ and $v = \alpha\mathbf{x} + \beta\mathbf{y} + \gamma\mathbf{z}$ is

$$\|u - v\|_{hex} = \max\{|a - \alpha|, |b - \beta|, |c - \gamma|\} = \max\{|i|, |j|, |i + j|\}. \tag{18}$$

The *unit disk* under this distance consists of all points with hexagonal distance at most 1 from the origin: $\mathbb{H} = \{u \in \mathbb{R}^2 \mid \|u - 0\|_{hex} \leq 1\}$. It is the regular hexagon with unit length sides that is the convex hull of the points $\pm(\mathbf{x} - \mathbf{y})$, $\pm(\mathbf{y} - \mathbf{z})$, $\pm(\mathbf{z} - \mathbf{x})$; see Figure 5. For

$B \subseteq \Delta$, we write $\text{MST}_{\text{hex}}(B)$ for the spanning tree that minimizes the hexagonal length. We construct it by adding the edges in sequence of non-decreasing hexagonal length, breaking ties with Euclidean length, and breaking the remaining ties arbitrarily. Since $\text{MST}_{\text{hex}}(B)$ is a spanning tree but not necessarily the one that minimizes Euclidean length, we have

$$|\text{MST}(B)| \leq |\text{MST}_{\text{hex}}(B)|, \quad (19)$$

in which we measure the Euclidean length on both sides. To prove Theorem 4.2, we show that for every $B \subseteq \Delta$, the average (Euclidean) length of the long edges in $\text{MST}_{\text{hex}}(B)$ and the short edges in $\text{MST}_{\text{hex}}(\Delta \setminus B)$ is at most $\frac{5}{4}$. Interchanging B and $\Delta \setminus B$, we get the same relation by symmetry. Using (19), this implies

$$|\text{MST}(B)| + |\text{MST}(\Delta \setminus B)| \leq |\text{MST}_{\text{hex}}(B)| + |\text{MST}_{\text{hex}}(\Delta \setminus B)| \leq \frac{5}{4}(n^2 - 2). \quad (20)$$

Compare this with (15), which establishes $|\text{MST}(\Delta_1)| + |\text{MST}(\Delta_3)| = \frac{5}{4}n^2 - 3$ for the partition $\Delta = \Delta_1 \sqcup \Delta_3$. The right-hand side differs from the upper bound in (20) by only a small additive constant. We thus conclude that the maximum MST-ratio of Δ converges to $\frac{5}{4}$ from below, as claimed by Theorem 4.2.

4.4 Hierarchy of Habitats

Let T_ℓ be the subset of edges in $\text{MST}_{\text{hex}}(B)$ whose hexagonal lengths are at most ℓ , together with the endpoints of these edges. For example, T_0 has zero edges, T_1 consist of all short edges, and $T_\ell = \text{MST}_{\text{hex}}(B)$ for sufficiently large ℓ . All edges connecting points in different components of T_ℓ have hexagonal length $\ell + 1$ or larger. We thus write $k\mathbb{H}$ for the scaled copy of the unit disk and call

$$D_k(B) = \bigcup_{u \in B} (k\mathbb{H} + u) \quad (21)$$

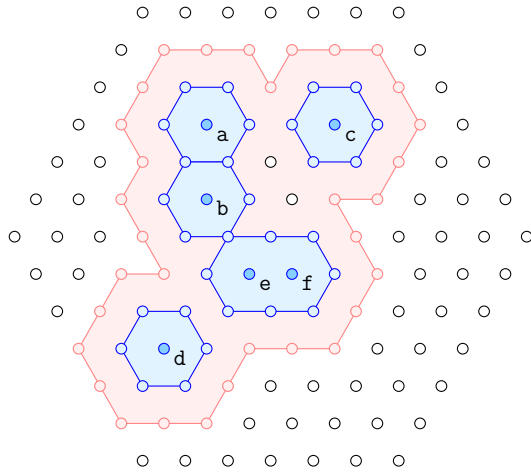
the k -th thickening of B , in which $k\mathbb{H} + u$ is the translate of $k\mathbb{H}$ whose center is u . As illustrated in Figure 6, the k -th thickenings of points u and v *overlap*, *touch*, are *disjoint* if the hexagonal distance between u and v is less than, equal to, larger than $2k$, respectively.

The boundary of $k\mathbb{H}$ passes through $6k$ points of the hexagonal lattice, which we call the *vertices* of $k\mathbb{H}$. Furthermore, we call the $6k$ (short) edges that connect these points in cyclic order the *edges* of $k\mathbb{H}$. Let $B_k \subseteq B$ be the vertex set of a component of T_{2k-1} , and observe that for all $u, v \in B_k$ there is a sequence of points $u = x_1, x_2, \dots, x_m = v$ in B_k such that $k\mathbb{H} + x_i$ and $k\mathbb{H} + x_{i+1}$ overlap for all $1 \leq i \leq m - 1$. We define the *frontier* of the component, denoted $\partial D_k(B_k)$, as the lattice points and the connecting (short) edges in the boundary of $D_k(B_k)$. Furthermore, $\partial D_k(B)$ is the union of frontiers of the components of T_{2k-1} . These notions are illustrated in Figure 6, which shows $\partial D_1(B)$ and $\partial D_2(B)$ for six marked points. Note that the edge shared by $\mathbb{H} + \mathbf{a}$ and $\mathbb{H} + \mathbf{b}$ is part of $\partial D_1(B)$.

4.5 Subdivided Foreground and Background

Consider the 1-st thickening of B , which for the time being we call the *foreground*. Letting $B_1 \subseteq B_2$ be the vertex sets of two nested components of T_1 and T_2 , we call $D_1(B_1)$ a *room* and $D_1(B_2)$ a *block* of the foreground. We say two rooms are *adjacent* if they share at least one edge. In Figure 6, there are five rooms, two of which are adjacent, and three blocks, one of which contains three rooms.

To make a finer distinction, observe that for any edge, its Euclidean length is smaller than or equal to the hexagonal length. The two notions agree on edges with slope 0, $\sqrt{3}$, and $-\sqrt{3}$. Consider T_2 and T_3 after removing all edges whose Euclidean length equals 2



■ **Figure 6** The *blue* 1-st thickening and the *pink* 2-nd thickening of $B = \{a, b, c, d, e, f\}$ in the hexagonal lattice. $\mathbb{H} + a$ and $\mathbb{H} + b$ share an edge and therefore form two rooms in a common house, while $\mathbb{H} + e$ and $\mathbb{H} + f$ overlap and thus form a one-room house in $D_1(B)$. These two houses form a block, and together with $\mathbb{H} + d$, they form a compound of two blocks. $\mathbb{H} + c$ is a room, a house, a block, and a compound by itself. The two compounds lie in the interior of a room in $D_2(B)$.

and 3, respectively, and let B'_2 and B'_3 be the vertex sets of the components that satisfy $B_1 \subseteq B'_2 \subseteq B_2 \subseteq B'_3$. Observe that any two rooms in $D_1(B'_2)$ have a sequence of pairwise adjacent rooms connecting them. We therefore call $D_1(B'_2)$ a *house*. For comparison, any two rooms in $D_1(B_2)$ have a sequence of rooms connecting them such that any two consecutive rooms share at least a vertex but not necessarily a full edge. Similarly, for any two blocks in $D_1(B'_3)$, there is a sequence of blocks connecting them such that the channel separating any two consecutive blocks at its narrowest place is only $\sqrt{3}/2$ wide. We therefore call $D_1(B'_3)$ a *compound*; see Figure 6 for examples. For comparison, the channel that separates two compounds is at its narrowest place at least one unit of distance wide. A few observations:

- (i) all vertices of $\partial D_1(B)$ are points in $\Delta \setminus B$;
- (ii) all edges of $\partial D_1(B)$ are short;
- (iii) the frontier of a room consists of at least six (short) edges.

We call the complement of the foreground the *background*, and the components of the background its *backyards*. We say a backyard is *adjacent* to a house if the two share a non-empty portion of their boundary. There are configurations in which the number of backyards is twice the number of houses; see Figure 3 on the left, where each backyard is adjacent to three houses, and each house is adjacent to six backyards. In general, we distinguish between backyards adjacent to at most two and at least three houses, denoting their numbers α_1 and β_1 , respectively. We prove an upper bound for β_1 in terms of the number of houses and blocks.

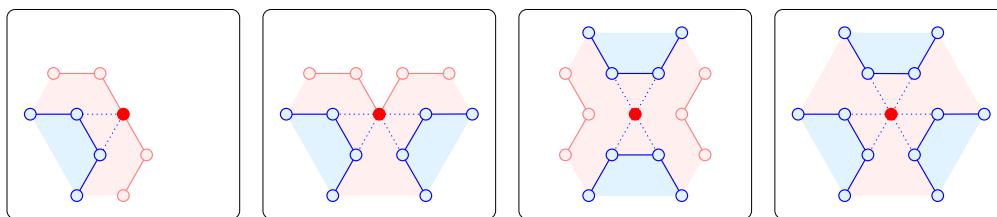
► **Lemma 4.3.** *Given h_1 houses arranged in b_1 blocks, the number of backyards adjacent to three or more houses satisfies $\beta_1 \leq 2h_1 - 2b_1 + 2$.*

Proof. We construct a graph $G = G(B)$ on the torus by placing a node inside each house, and whenever two houses meet at a boundary vertex, we connect the corresponding nodes with a curved arc that passes through the shared vertex. This can be done such that no two of the arcs cross and each face of G contains one backyard. A face bounded by a single arc (*loop*) or two arcs (*multi-arcs*) contains a backyard adjacent to at most two houses and

thus does not count toward β_1 . We remove this face by deleting the loop or one of the two multi-arcs. The resulting graph has h_1 nodes, b_1 components, and β_1 faces. Write a_1 for the number of arcs. If the graph is connected and all faces are bounded by three arcs, we have $h_1 - a_1 + \beta_1 = 0$ because the Euler characteristic of the torus is 0. Whenever we remove an arc from this graph, we either merge two faces or split a component, but it is also possible that the removal of the arc has neither of those two side-effects. Hence, we have $h_1 - a_1 + \beta_1 \geq b_1 - 1$ in the general case. Since $2a_1 \geq 3\beta_1$, this implies $\beta_1 \leq 2h_1 - 2b_1 + 2$, as claimed. ◀

4.6 Satellites

By definition, compounds cannot be packed as tightly as blocks; see Figure 3 with lattice points between the compounds in the lower middle but no such points between the blocks on the right. Recall that each component of $D_1(B)$ is contained in a room of $D_2(B)$. For each such room, we single out the largest compound it contains – breaking ties arbitrarily – and call this the *big compound* of the room. All others are *small compounds* of the room. We refer to certain lattice points close to one or more compounds as satellites. The targeted lattice points are at distance $\sqrt{3}/2$ outside $D_1(B)$ and either on the boundary or in the interior of $D_2(B)$.



■ **Figure 7** From *left to right*: a single, a double, another double, and a triple satellite in *red*. In the *left* two cases, the satellite belongs to the frontier of a room of the 2-nd thickening of B , while in the *right* two cases, the satellite lies in the interior of such a room.

The difference between small and large compounds influences which lattice points we call satellites. For each small compound we find three satellites as follows: sandwich the compound between three lines with slopes $0, \pm\sqrt{3}$, choose a (short) edge as the basis of an equilateral triangle outside the compound on each line, and pick the vertex of this triangle opposite to the basis as a *satellite*. Observe that the Euclidean distance between any two satellites of the same compound is at least 3. In contrast, we pick six lattice points as the satellites of the big compound by sandwiching it between six lines, two each of slope $0, \pm\sqrt{3}$, choosing one basis on each line, and picking the vertex of the equilateral triangle opposite to the basis as a satellite. The Euclidean distance between any two such satellites is at least $\sqrt{3}$.

As illustrated in Figure 7, a lattice point can be a satellite of one, two, or three compounds in the same room. Accordingly, we call the point a *single*, *double*, or *triple satellite* of the room, respectively. A single satellite is necessarily a vertex on the frontier of the room, a triple satellite is necessarily in the interior of the room, and a double satellite can be one or the other. For a room, R , we write $s(R)$ and $d(R)$ for the number of single and double satellites on its frontier, and $e(R)$ and $t(R)$ for the number of double and triple satellites in its interior. Summing over all rooms in $D_2(B)$, we set $s_1 = \sum s(R)$, $d_1 = \sum d(R)$, $e_1 = \sum e(R)$, $t_1 = \sum t(R)$, and refer to s_1, d_1, e_1, t_1 as the *satellite sums* of $D_2(B)$. Furthermore, let c_1 be the number of compounds of $D_1(B)$ and r_2 the number of rooms of $D_2(B)$. Since

$s(R) + 2d(R) + 2e(R) + 3t(R)$ is three times the number of small compounds in R plus six for the big compound, the satellite sums satisfy a linear relation, which we state together with a property of short edges connecting satellites in the interior:

- (iv) if $c_1 > 1$, then the satellite sums of $D_2(B)$ satisfy $s_1 + 2d_1 + 2e_1 + 3t_1 = 3c_1 + 3r_2$;
- (v) any unit length edge connecting blocks of $D_1(B)$ inside a room of $D_2(B)$ with each other or to satellites in the interior of this room is contained in the interior of this room.

By construction, there are $s(R) + d(R)$ satellites that are vertices of R . We prove a stronger lower bound on the number of vertices, which also strengthens Claim (iii).

► **Lemma 4.4.** *Assume $r_2 \geq 2$ and let R be a room of $D_2(B)$. Then the frontier of R has at least $6 + \frac{2}{3}s(R) + \frac{4}{3}d(R)$ vertices.*

Proof. Let p , s , d be the number of non-satellite lattice points, single satellites, double satellites, and write $\text{per}(R)$ for the *perimeter*, which is the length of or the number of (short) edges in the frontier of R . To begin note that a satellite in the frontier of R is in the boundary of at most one backyard. This is because the external angle is 180° at a single satellite and 60° at a double satellite. The internal angle at any vertex of another room is at least 120° , so there is not enough space for two backyards around a satellite; see the left two panels in Figure 7. This implies that we may assume that the frontier of R is a simple polygon, or a collection of such. Indeed, if the polygon touches itself at a vertex, this must be a non-satellite, which we can duplicate, and if the polygon touches itself along a sequence of edges, we can remove these edges and their shared vertices. This operation neither changes the number of single and double satellites, nor does it increase the perimeter. A room that contains only one compound can have perimeter as small as 12, but a room with at least two compounds has significantly larger perimeter, certainly larger than 15. For $\text{per}(R) \leq 15$, we thus get only one compound and, by construction, only 6 single and no double satellites. This implies the claimed inequality. We therefore assume (22), aim at proving (23), and note that (24) follows as the convex combination of (22) and (23) with coefficients $\frac{1}{3}$ and $\frac{2}{3}$:

$$\text{per}(R) \geq 16; \tag{22}$$

$$\text{per}(R) \geq 1 + s + 2d; \tag{23}$$

$$\text{per}(R) \geq \frac{1}{3}16 + \frac{2}{3}(1 + d + 2d) = 6 + \frac{2}{3}s + \frac{4}{3}d. \tag{24}$$

It remains to prove (23). Call the endpoints of an edge in the frontier of R *neighbors*. Two neighbors cannot both be double satellites, else they would belong to a common compound, which contradicts that the distance between them is at least $\sqrt{3}$. Furthermore, if a double satellite neighbors a single satellite, then this is only possible if they are vertices of an equilateral triangle bounding a backyard, as in Figure 8 on the left. For lack of space around this triangle, its third vertex is a non-satellite. The contribution of these three vertices to the right-hand side of (23) is $2 + 1 + 0 = 3$. Hence, we can remove the three edges from the left-hand side and the three vertices from the right-hand side of (23) without affecting the validity of the inequality. As illustrated in Figure 8 on the left, two such triangles may touch at a non-satellite vertex, but this does not matter and we can remove the edges and vertices of both triangles from (23).

We can therefore assume that both neighbors of a double satellite are non-satellites. Hence, between any two double satellites there is at least one non-satellite, which implies $p \geq d$. But $p = d$ only if $p = d = 0$ or there is strict alternation between double satellites and non-satellites. It is not possible that all vertices in the frontier are single satellites,

because this contradicts that the distance between any two of them is at least $\sqrt{3}$. Strict alternation is possible, but only for the polygon of 12 edges shown in Figure 8 on the right. By assumption, $D_2(B)$ has at least two rooms, so not all backyards of R can be bounded by such 12-gons. But this implies $p \geq d + 1$, so $\text{per}(R) = p + s + d \geq 1 + s + 2d$, as claimed. ◀

To generalize the above concepts to $k \geq 1$, we let $B_{2k-1} \subseteq B_{2k}$ be the vertex sets of two nested components of T_{2k-1} and T_{2k} , and call $D_k(B_{2k-1})$ a *room* and $D_k(B_{2k})$ a *block* of $D_k(B)$. The rooms that share edges join to form *houses*, and the blocks separated by channels that are only $\sqrt{3}/2$ wide join to form *compounds*. Write r_k, h_k, b_k, c_k for the number of rooms, houses, blocks, compounds of $D_k(B)$, α_k, β_k for the number of backyards adjacent to at most 2, at least 3 houses, and s_k, d_k, e_k, t_k for the satellite sums of $D_{k+1}(B)$. We can now extend Claims (i) to (v) and Lemmas 4.3 and 4.4 merely by substituting $D_k(B)$ for $D_1(B)$, β_k for β_1 , c_k for c_1 , etc. In particular, the extension of Claim (iv) to

$$s_k + 2d_k + 2e_k + 3t_k = 3c_k + 3r_{k+1} \tag{25}$$

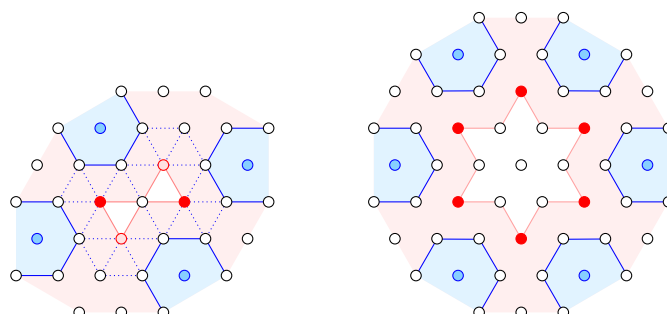
assuming $c_k > 1$ will be needed shortly. We note that (25) and the extension of Lemma 4.4 can be strengthened, but it is not necessary for the purpose of proving Theorem 4.2.

4.7 Loop-free Subgraphs

Let V_k be the vertices of $D_k(B)$ together with all double and triple satellites that lie in the interior of rooms in $D_{k+1}(B)$, and note that $V_j \cap V_k = \emptyset$ whenever $j \neq k$. Let V'_k be V_k together with the remaining satellites of $D_k(B)$, and note that $V_j \cap V'_k = \emptyset$ if $j < k$, but V'_k and V_{k+1} may share points. To account for this difference, let ℓ be the smallest integer such that $r_{\ell+1} = 1$, and define

$$V = \begin{cases} V_1 & \text{if } \ell = 0; \\ V_1 \sqcup \dots \sqcup V_{\ell-1} \sqcup V_\ell & \text{if } \ell \geq 1 \text{ and } c_\ell = 1; \\ V_1 \sqcup \dots \sqcup V_{\ell-1} \sqcup V'_\ell & \text{if } \ell \geq 1 \text{ and } c_\ell > 1. \end{cases} \tag{26}$$

By construction, all points in V belong to $\Delta \setminus B$, and all unit length edges connecting these points are candidates for $\text{MST}_{\text{hex}}(\Delta \setminus B)$. We therefore let U be a maximal loop-free graph whose vertices are the points in V and whose edges all have unit length. Since U has no loops, there is an $\text{MST}_{\text{hex}}(\Delta \setminus B)$ that contains U as a subgraph. We are therefore motivated to study the number of edges in U . Using a slight abuse of notation, we denote this number $\#U$. For every k , let U_k and U'_k be the subgraphs of U induced by V_k and V'_k , respectively. We first count the edges in U_1 and U'_1 in Lemma 4.5.



■ **Figure 8** *Left*: two touching triangular backyards. Their shared vertex is a non-satellite, the two *red* vertices are double satellites, and the two *pink* vertices are single satellites. *Right*: unique polygon with strictly alternating double satellites and non-satellites. On *both sides*, all (partially drawn) *blue* compounds are different and belong to the same (partially drawn) *pink* room.

► **Lemma 4.5.** *Let $r_1 \geq h_1 \geq b_1 \geq c_1$ be the number of rooms, houses, blocks, and compounds of $D_1(B)$, and s_1, d_1, e_1, t_1 the satellite sums of $D_2(B)$. Then*

$$\#U_1 \geq 2r_1 + h_1 + 3b_1 + (e_1 + t_1) - r_2 - 4; \tag{27}$$

$$\#U'_1 \geq 2r_1 + h_1 + 3b_1 + (s_1 + d_1 + e_1 + t_1) - 5, \tag{28}$$

in which we assume $c_1 > r_2 = 1$ for the second inequality.

Proof. We argue in three steps: first counting edges in $\partial D_1(B)$, second counting edges connecting blocks, and third counting edges connecting the satellites. In each case, we count only unit length edges, and we make sure that the edges we count do not form loops.

For the first step, it is convenient to count *half-edges*, which are the two sides of an edge. These two sides either face two rooms, or one faces a room and the other faces the background. For a house, H , we make its $r(H)$ rooms accessible from the outside by removing $r(H) - 1$ edges shared by adjacent rooms plus 1 edge shared with the background. By (iii), each room was originally faced by at least 6 half-edges, so we still have at least $4r(H) + 1$ of them left. Doing this for each house, we make all r_1 rooms accessible from the background, and we have at least $4r_1 + h_1$ half-edges left facing these rooms.

Observe that the convex hull of a house contains at least six of the (short) edges that bound the house. One may have been removed, so we still have at least 5 half-edges facing the background. Keeping in mind that the cycles that bound backyards still need to be opened, we now have at least $4r_1 + h_1 + 5h_1$ half-edges and therefore at least $2r_1 + 3h_1$ edges. If a backyard is adjacent to at most two houses, then it has two consecutive (short) edges that enclose an angle less than π and that are both shared with the same house. Hence, the complementary angle on the side of the house is larger than π , which implies that these two edges cannot belong to the convex hull of the house. We remove one of them and use the half-edge facing the backyard of the other to compensate for the removed half-edge facing the room. Since both edges have not yet been accounted for, we still have at least $2r_1 + 3h_1$ edges. If a backyard is adjacent to three or more houses, we also remove one edge, but this time count one less. Recalling that β_1 is the number of such backyards, we still have at least $2r_1 + 3h_1 - \beta_1 \geq 2r_1 + h_1 + 2b_1 - 2$ edges, in which we get the right-hand side from Lemma 4.3.

For the second step, we connect the $b(R)$ blocks inside a common room of $D_2(B)$ with $b(R) - 1$ short edges. A total of b_1 blocks are hierarchically organized in r_2 rooms, so we add $b_1 - r_2$ short edges to those counted in the first step. Similarly, we add $e_1 + t_1$ short edges that connect the double and triple satellites in the interiors of the rooms to the vertices in the frontier of $D_1(B)$. Finally, we remove two edges to open the meridian and longitudinal cycles of the graph, if they exist. The final count is therefore at least $2r_1 + h_1 + 3b_1 + (e_1 + t_1) - r_2 - 4$, which is the claimed lower bound for $\#U_1$.

For the third step, we assume $c_1 > r_2 = 1$. Since there is only one room, there are no shared satellites between different rooms, and we can connect them to the frontier of $D_1(B)$ with $s_1 + d_1$ short edges without creating any loop. This implies that the number of edges in U'_1 is at least $2r_1 + h_1 + 3b_1 + (s_1 + d_1 + e_1 + t_1) - 5$, as claimed. ◀

The bounds in Lemma 4.5 generalize to $k > 1$, but there are differences. Most important is the existence of a loop-free graph for thickness $k - 1$. In particular, we have satellites that affect the structure and size of U_k and U'_k .

► **Lemma 4.6.** *Let $r_k \geq h_k \geq b_k \geq c_k$ be the number of rooms, houses, blocks, compounds of $D_k(B)$, and s_k, d_k, e_k, t_k the satellite sums of $D_{k+1}(B)$. Then for $k \geq 2$, we have*

$$\#U_k \geq (3r_k + \frac{1}{3}s_{k-1} + \frac{2}{3}d_{k-1}) + 4h_k + 3b_k + (e_k + t_k) - r_{k+1} - 4; \quad (29)$$

$$\#U'_k \geq (3r_k + \frac{1}{3}s_{k-1} + \frac{2}{3}d_{k-1}) + 4h_k + 3b_k + (s_k + d_k + e_k + t_k) - 5, \quad (30)$$

in which we assume $c_k > r_{k+1} = 1$ for the second inequality.

Proof. We argue again in three steps: first counting edges in $\partial D_k(B)$, second counting edges connecting blocks, and third counting edges connecting to the satellites. Each of these three steps is moderately more involved than the corresponding step in the proof of Lemma 4.5, and we emphasize the differences.

The first step starts the construction with Lemma 4.4, which implies that the rooms in $D_k(B)$ are faced by a total of at least $6r_k + \frac{2}{3}s_{k-1} + \frac{4}{3}d_{k-1}$ half-edges. After making all rooms accessible to the background, we still have at least $(4r_k + \frac{2}{3}s_{k-1} + \frac{4}{3}d_{k-1}) + h_k$ half-edges. Adding the at least 11 half-edges per house facing the background, we have at least $(4r_k + \frac{2}{3}s_{k-1} + \frac{4}{3}d_{k-1}) + 12h_k$ half-edges and thus at least $(2r_k + \frac{1}{3}s_{k-1} + \frac{2}{3}d_{k-1}) + 6h_k$ edges. Let α_k and β_k be the number of backyards adjacent to at most two and at least three houses, respectively. By extension of Lemma 4.3, we have $\beta_k \leq 2h_k - 2b_k + 2$. We remove an edge per backyard, which for the first type does not affect the current edge count, while the backyards of the second type reduce the count to $(2r_k + \frac{1}{3}s_{k-1} + \frac{2}{3}d_{k-1}) + 4h_k + 2b_k - 2$.

For the second step, we connect the blocks of $D_k(B)$ inside a common room of $D_{k+1}(B)$ with $b_k - r_{k+1}$ edges. Furthermore, we add r_k edges to connect the blocks of $D_{k-1}(B)$ inside a common room of $D_k(B)$ – which inductively are already connected to each other – to the frontier of this room, and we add at least $e_k + t_k$ edges connecting to the triple satellites of compounds inside the rooms of $D_{k+1}(B)$. After removing two additional edges to break the meridian and longitudinal loops, if they exist, we arrive at a lower bound of at least $(3r_k + \frac{1}{3}s_{k-1} + \frac{2}{3}d_{k-1}) + 4h_k + 3b_k + (e_k + t_k) - r_{k+1} - 4$ edges in U_k .

For the third step, we assume $c_k > r_{k+1} = 1$, in which case we can add at least $s_k + d_k$ edges connecting to the single and double satellites. This implies $\#U'_k \geq (3r_k + \frac{1}{3}s_{k-1} + \frac{2}{3}d_{k-1}) + 4h_k + 3b_k + (s_k + d_k + e_k + t_k) - 5$. ◀

4.8 Book-keeping

The goal is to show that the average (Euclidean) length of the long edges in $\text{MST}_{\text{hex}}(B)$ and the short edges in $\text{MST}_{\text{hex}}(\Delta \setminus B)$ is at most $\frac{5}{4}$. We thus assign a *credit* of $\alpha = \frac{1}{4}$ to every short edge and set the *cost* of a long edge to be its Euclidean length minus $\frac{5}{4}$. For convenience, we set the value of α to 1 Euro and convert the costs into Euros; see Table 1.

■ **Table 1** The Euclidean lengths of the edges with hexagonal lengths 2 to 5, and their costs in Euros, each truncated beyond the first two digits after the decimal point.

hex	2	2	3	3	4	4	4	5	5	5
L_2	$\sqrt{3}$	$\sqrt{4}$	$\sqrt{7}$	$\sqrt{9}$	$\sqrt{12}$	$\sqrt{13}$	$\sqrt{16}$	$\sqrt{19}$	$\sqrt{21}$	$\sqrt{25}$
cost	1.92	3.00	5.58	7.00	8.85	9.42	11.00	12.43	13.33	15.00

For the accounting, we need the costs of the last two edges for each hexagonal length. Letting w_k, x_k and y_k, z_k be the costs of the two longest edges with hexagonal length $2k$ and $2k + 1$, respectively, we have

3:18 The Euclidean MST-Ratio of Bi-Colored Lattices

$$w_k = \frac{1}{\alpha} \left[\sqrt{4k^2 - 2k + 1} - \frac{5}{4} \right], \quad x_k = \frac{1}{\alpha} \left[2k - \frac{5}{4} \right], \quad (31)$$

$$y_k = \frac{1}{\alpha} \left[\sqrt{4k^2 + 2k + 1} - \frac{5}{4} \right], \quad z_k = \frac{1}{\alpha} \left[(2k + 1) - \frac{5}{4} \right]; \quad (32)$$

see Table 1, which shows the values of $w_1, x_1, y_1, z_1, w_2, x_2, y_2, z_2$ in boldface. Listing the edges in sequence, we need bounds for the cost differences between consecutive edges:

$$2 \leq w_k - z_{k-1} \leq 2.928\dots; \quad 1.071\dots \leq x_k - w_k \leq 2; \quad (33)$$

$$2 \leq y_k - x_k \leq 2.583\dots; \quad 1.414\dots \leq z_k - y_k \leq 2, \quad (34)$$

which are not difficult to prove using elementary computations. We use accounting with credits and costs to prove that the average (Euclidean) edge length of the two minimum spanning trees is less than $\frac{5}{4}$. Note that the hexagonal lattice on the torus is obtained by gluing a regular hexagonal portion of the Euclidean hexagonal lattice along opposite sides. If we choose $12n^2$ points, then this hexagon has $2n + 1$ vertices and therefore $2n$ edges per side. Taking only every other point – so $3n^2$ of the $12n^2$ – we still get an integer number of edges per side. It follows that the $3n^2$ points are the minority color in a 1 : 3 coloring of the $12n^2$ points.

► **Lemma 4.7.** *Let Δ be the hexagonal lattice with $12n^2$ points and unit minimum distance on the torus, and $B \subseteq \Delta$. Then $|\text{MST}(B)| + |\text{MST}(\Delta \setminus B)| \leq 15n^2 - \frac{5}{2}$.*

Proof. By (19), it suffices to prove the inequality for $\text{MST}_{\text{hex}}(B)$ and $\text{MST}_{\text{hex}}(\Delta \setminus B)$. For $k \geq 1$, we compare the edges of hexagonal length $2k$ and $2k + 1$ in $\text{MST}_{\text{hex}}(B)$ with the (short) edges in U_k or possibly in U'_k . Since $T_{2k+1} \setminus T_{2k-1}$ is the set of these long edges, we can do this in one step by comparing $T_{2\ell+1}$ with U , for sufficiently large ℓ and U as defined right after the definition of V in (26). Recall that r_k is the number of components of T_{2k-1} or, equivalently, the number of rooms of $D_k(B)$. These rooms are organized hierarchically into h_k houses, b_k blocks, and c_k compounds. Hence, $r_1 \geq h_1 \geq b_1 \geq c_1 \geq r_2$, etc. This implies that there are

- $r_1 - h_1$ edges of hexagonal length 2 and Euclidean length less than 2 that connect the rooms pairwise inside the h_1 houses;
- $h_1 - b_1$ edges of hexagonal and Euclidean length 2 that connect the houses pairwise inside the b_1 blocks;
- $b_1 - c_1$ edges of hexagonal length 3 and Euclidean length less than 3 that connect the blocks pairwise inside the c_1 compounds;
- $c_1 - r_2$ edges of hexagonal and Euclidean length 3 that connect the compounds pairwise inside the r_2 rooms of $D_2(B)$, etc.

The costs for these edges are w_1, x_1, y_1, z_1 , respectively. Setting $z_0 = 0$, and generalizing to $k \geq 1$, we observe that the total cost satisfies

$$\text{cost} \leq \sum_{k \geq 1} [w_k(r_k - h_k) + x_k(h_k - b_k) + y_k(b_k - c_k) + z_k(c_k - r_{k+1})] \quad (35)$$

$$= \sum_{k \geq 1} [(w_k - z_{k-1})r_k + (x_k - w_k)h_k + (y_k - x_k)b_k + (z_k - y_k)c_k] \quad (36)$$

$$\leq [2r_1 + h_1 + 3b_1 + c_1 - 7] + \sum_{k \geq 2} [3r_k + h_k + 3b_k + c_k - 8]. \quad (37)$$

To see how (37) derives from (36), we first make the sums finite by letting ℓ be the smallest integer such that $r_{\ell+1} = 1$. Then the last non-zero term in (35) is $z_\ell(c_\ell - r_{\ell+1})$ and, correspondingly, the last term in (36) is $z_\ell r_{\ell+1} = z_\ell$, which by (32) is equal to $8\ell - 1$. But this is

the same as the sum of constants in (37). Furthermore, we note that if $r_k = h_k = b_k = c_k = 1$, for every k , then (36) vanishes because (35) vanishes, and (37) vanishes because for any k the corresponding sum of four terms minus the constant vanishes. Hence, the difference between (37) and (36) vanishes. To prove the inequality, we reintroduce the variables, which satisfy $r_1 \geq h_1 \geq \dots \geq c_\ell$, and look at their coefficients. The first is $2 - w_1 + z_0$, which is positive because $w_1 < 2$ and $z_0 = 0$. Indeed, using the inequalities in (33) and (34), we observe that the coefficients alternate between positive and negative. For example, $3 - w_k + z_{k-1}$ is positive because $w_k - z_{k-1} < 3$, and $1 - x_k + w_k$ is negative because $x_k - w_k > 1$. This implies that the difference is non-negative, so (37) follows.

The difficult cases are the edges of hexagonal lengths 2 and 3. We therefore consider the special cases in which all edges in $\text{MST}_{\text{hex}}(B)$ have Euclidean length at most $\sqrt{3}, \sqrt{4}, \sqrt{7}, \sqrt{9}$, so $h_1 = 1, b_1 = 1, c_1 = 1, r_2 = 1$, respectively; see Figure 3. From (37), we get

$$\text{cost} \leq \begin{cases} 2r_1 - 2 & \text{if } r_1 > h_1 = 1; \\ 2r_1 + h_1 - 3 & \text{if } h_1 > b_1 = 1; \\ 2r_1 + h_1 + 3b_1 - 6 & \text{if } b_1 > c_1 = 1; \\ 2r_1 + h_1 + 3b_1 + c_1 - 7 & \text{if } c_1 > r_2 = 1. \end{cases} \quad (38)$$

The cost needs to be paid from the credit contributed by the (short) edges in U , which in these four cases is either U_1 or U'_1 . Recall that after the conversion, each short edge contributes one Euro of credit, so Lemma 4.5 provides lower bounds:

$$\text{credit} \geq \begin{cases} 2r_1 - 1 & \text{if } r_1 > h_1 = 1; \\ 2r_1 + h_1 - 2 & \text{if } h_1 > b_1 = 1; \\ 2r_1 + h_1 + 3b_1 - 5 & \text{if } b_1 > c_1 = 1; \\ 2r_1 + h_1 + 3b_1 + (s_1 + d_1 + e_1 + t_1) - 5 & \text{if } c_1 > r_2 = 1. \end{cases} \quad (39)$$

Comparing (39) with (38), we get $\text{cost} \leq \text{credit}$ trivially in the first three cases. Using Claim (iv), we get use $s_1 + d_1 + e_1 + t_1 \geq c_1 \geq (s_1 + 2d_1 + 2e_1 + 3t_1) - r_2 = c_1$, which supports the same in the fourth case. To compare the cost with the credit in the remaining cases, we use Lemmas 4.5 and 4.6 to compute a lower bound for the latter, assuming that $\ell > 1$ is the smallest integer for which $r_{\ell+1} = 1$:

$$\begin{aligned} \text{credit} &\geq \#U_1 + \sum_{k=2}^{\ell-1} \#U_k + \#U'_\ell & (40) \\ &\geq [2r_1 + h_1 + 3b_1 + (\frac{1}{3}s_1 + \frac{2}{3}d_1 + e_1 + t_1) - r_2 - 4] \\ &\quad + \sum_{k=2}^{\ell-1} [3r_k + 4h_k + 3b_k + (\frac{1}{3}s_k + \frac{2}{3}d_k + e_k + t_k) - r_{k+1} - 4] \\ &\quad + [3r_\ell + 4h_\ell + 3b_\ell + (s_\ell + d_\ell + e_\ell + t_\ell) - 5], & (41) \end{aligned}$$

in which we group the terms with index $k - 1$ that appear in the bounds for $\#U_k$ and $\#U'_k$ with the terms that have the same index. Using the extension of Claim (iv) to $k \geq 1$ stated in (25), we get $\frac{1}{3}s_k + \frac{2}{3}d_k + e_k + t_k \geq \frac{1}{3}(s_k + 2d_k + 2e_k + 3t_k) = c_k + r_{k+1}$, so the lower bound in (41) exceeds the upper bound in (37). Hence, $\text{cost} \leq \text{credit}$. In other words, the average Euclidean length of the edges in $\text{MST}_{\text{hex}}(B)$ and $\text{MST}_{\text{hex}}(\Delta \setminus B)$ is at most $\frac{5}{4}$. It follows that their total Euclidean length is at most $\frac{5}{4}(n^2 - 2)$, which by (19) implies the same for $\text{MST}(B)$ and $\text{MST}(\Delta \setminus B)$. \blacktriangleleft

By Lemma 4.7, the average Euclidean length of the edges in $\text{MST}(B)$ and $\text{MST}(\Delta \setminus B)$ is less than $\frac{5}{4}$. Together with (15), this implies Theorem 4.2.

5 Discussion

This paper proves bounds on the supremum and infimum of the maximum MST-ratio for finite sets, as well as of the supremum MST-ratio for lattices in the plane. There are many directions of generalization, and their connection to the topological analysis of colored point sets started in [5] provides a potential path to relevance outside of mathematics.

- What about sets in the plane that are less restrictive than lattices but still disallow arbitrarily dense clusters of points, such as periodic sets or Delone sets? A first result in this direction is the lower bound of $1 + 1/(11(2c + 1)^2)$ for the maximum MST-ratio of a set of n points with spread at most $c\sqrt{n}$ proved in [6].
- What about partitions of $A \subseteq \mathbb{R}^2$ into three or more sets? For example, is it true that the supremum MST-ratio of the hexagonal lattice partitioned into three subsets is $\sqrt{3}$, as realized by the unique partition into three congruent hexagonal grids? Is $\sqrt{3}$ the infimum, over all lattices in \mathbb{R}^2 , of the supremum, over all partitions into three subsets?
- What about three and higher dimensions? Consider for example the FCC lattice in \mathbb{R}^3 (all integer points whose sums of coordinates are even), and partition it into 2FCC and the rest. The MST-ratio of this example is $\frac{9}{8} = 1.125$. Is it true that this is the supremum MST-ratio of the FCC lattice? Is 1.125 the infimum, over all lattices in \mathbb{R}^3 , of the supremum, over all partitions into two subsets?

Beyond these extensions in discrete geometry, it would be interesting to study the MST-ratio stochastically, to determine the computational complexity of the maximum MST-ratio, and to frame notions of mingling as measured by homology classes of dimension 1 and higher in elementary geometric terms.

References

- 1 Gunnar E. Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46:255–308, 2009. doi:10.1090/S0273-0979-09-01249-X.
- 2 F. Chung and R. Graham. A new bound for Euclidean Steiner minimal trees. *Annals of the New York Academy of Sciences*, 440:328–346, 1985. doi:10.1111/j.1749-6632.1985.tb14564.x.
- 3 David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete Comput. Geom.*, pages 103–120, 2007. doi:10.1007/s00454-006-1276-5.
- 4 David Cohen-Steiner, Herbert Edelsbrunner, John Harer, and Dmitriy Morozov. Persistent homology for kernels, images, and cokernels. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1011–1020, January 2009. doi:10.1137/1.9781611973068.110.
- 5 Sebastiano Cultrera di Montesano, Ondřej Draganov, Herbert Edelsbrunner, and Morteza Saghafian. Chromatic alpha complexes, 2024. doi:10.48550/arXiv.2212.03128.
- 6 Adrian Dumitrescu, János Pach, and Géza Tóth. Two trees are better than one, 2023. doi:10.48550/arXiv.2312.09916.
- 7 Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. Springer, January 2010. doi:10.1007/978-3-540-33259-6_7.
- 8 E. N. Gilbert and H. O. Pollak. Steiner minimal trees. *SIAM J. Appl. Math.*, 16(1):1–29, January 1968. doi:10.1137/0116001.
- 9 V. Jarník. O jistém problému minimálním: (z dopisu panu O. Borůvkovi). *Mor. přírodovědecká společnost*, 1930. URL: <http://dml.cz/dmlcz/500726>.
- 10 V. Jarník and M. Kössler. O minimálních grafech, obsahujících n daných bodů. *Časopis pro pěstování matematiky a fyziky*, 063(8):223–235, 1934. doi:10.21136/CPMF.1934.122548.

- 11 Joseph B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956. doi:10.2307/2033241.
- 12 B. Zhilinskii. *Introduction to Louis Michel's Lattice Geometry Through Group Action*. Current natural sciences. EDP Sciences, 2015. URL: <http://library.oapen.org/handle/20.500.12657/25012>.

A Connection to Chromatic Persistence

As mentioned in the introduction, the study of the MST-ratio is motivated by a recent topological data analysis method for measuring the “mingling” of points in a colored configuration; see Figure 9, which shows six persistence diagrams measuring various aspects of the mingling in a bi-colored configuration. This appendix addresses the meaning of some of these diagrams and explains the connection to the MST-ratio, while referring to [5] for a detailed account of the method. In particular, we short-cut the description by ignoring the discrete structures that are necessary for the algorithm. We first sketch the general background from [7] and [4], and then explain the specific setting that motivates the MST-ratio.

Let $A \subseteq \mathbb{R}^2$ be a finite set of points, $\chi: A \rightarrow \{0, 1\}$ a bi-coloring, and write $B = \chi^{-1}(0)$ and $C = A \setminus B = \chi^{-1}(1)$. Let $\mathbf{a}: \mathbb{R}^2 \rightarrow \mathbb{R}$ be the function that maps every $x \in \mathbb{R}^2$ to the minimum Euclidean distance between x and the points in A , and let $\mathbf{b}: \mathbb{R}^2 \rightarrow \mathbb{R}$ and $\mathbf{c}: \mathbb{R}^2 \rightarrow \mathbb{R}$ be the similarly defined functions for B and C . Furthermore, write $A_r = \mathbf{a}^{-1}[0, r]$, $B_r = \mathbf{b}^{-1}[0, r]$, and $C_r = \mathbf{c}^{-1}[0, r]$ for the sublevel sets at distance threshold $r \geq 0$. Each is a union of disks with radius r centered at the points of A , B , and C , respectively. The inclusions $B_r \subseteq A_r$ and $C_r \subseteq A_r$ induce homomorphisms in p -th homology, $b_r: H_p(B_r) \rightarrow H_p(A_r)$ and $c_r: H_p(C_r) \rightarrow H_p(A_r)$, for each dimension $p \in \mathbb{Z}$ and every threshold $r \geq 0$. Assuming field coefficients in the construction of the homology groups, the latter are vector spaces and the homomorphisms are linear maps.

We also have $A_r \subseteq A_s$ whenever $r \leq s$, so there are also linear maps from $H_p(A_r)$ to $H_p(A_s)$. By now it is tradition in the field to consider the *filtration* of the A_r , for r from 0 to ∞ , and the corresponding sequence of homology groups together with the linear maps between them. Reading this sequence from left to right, we see homology classes being born and dying. There is a unique way to pair the births with the deaths that regards the identity of the classes, and the *persistence diagram* summarizes this information by drawing a point $(r, s) \in \mathbb{R}^2$ for every homology class that is born at A_r and dies entering A_s ; see e.g. [7, Chapter VII]. Every death is paired with a birth, but it is possible that a birth remains unpaired – when the homology class is of the domain – in which case the corresponding point is at infinity. We write $\text{Dgm}_p(\mathbf{a})$ for the persistence diagram defined by the sublevel sets of \mathbf{a} , noting that it is a multi-set of points vertically above the diagonal.

Besides $\text{Dgm}_p(\mathbf{a})$, we consider $\text{Dgm}_p(\mathbf{b})$ and $\text{Dgm}_p(\mathbf{c})$, which are the persistence diagrams of the sublevel sets of \mathbf{b} and \mathbf{c} , respectively, and work with the disjoint union, $B_r \sqcup C_r$. Conveniently, the p -th persistence diagram of $\mathbf{b} \sqcup \mathbf{c}: \mathbb{R}^2 \sqcup \mathbb{R}^2 \rightarrow \mathbb{R}$ is the disjoint union of $\text{Dgm}_p(\mathbf{b})$ and $\text{Dgm}_p(\mathbf{c})$, for all p . Write $b_r \oplus c_r: H_p(B_r) \oplus H_p(C_r) \rightarrow H_p(A_r)$ for the corresponding map in homology. As proved in [4], the sequence of images of the $b_r \oplus c_r$ admit linear maps between them and thus define another persistence diagram, denoted $\text{Dgm}_p(\text{im } \mathbf{b} \sqcup \mathbf{c} \rightarrow \mathbf{a})$. Similarly, the kernels of the $b_r \oplus c_r$ define a persistence diagram, denoted $\text{Dgm}_p(\text{ker } \mathbf{b} \sqcup \mathbf{c} \rightarrow \mathbf{a})$. To simplify the notation, we write $\kappa_r = b_r \oplus c_r$ and use mnemonic notation to indicate whether a persistence diagram belongs to the domain, image, or kernel of the map:

3:22 The Euclidean MST-Ratio of Bi-Colored Lattices

$$\text{Dom}_p(\kappa) = \text{Dgm}_p(\mathbf{b} \sqcup \mathbf{c}), \quad (42)$$

$$\text{Im}_p(\kappa) = \text{Dgm}_p(\text{im } \mathbf{b} \sqcup \mathbf{c} \rightarrow \mathbf{a}), \quad (43)$$

$$\text{Ker}_p(\kappa) = \text{Dgm}_p(\text{ker } \mathbf{b} \sqcup \mathbf{c} \rightarrow \mathbf{a}). \quad (44)$$

The 1-norm of a persistence diagram, D , is the sum of the absolute differences between birth- and death-coordinates over all points in D , denoted $\|D\|_1$. To cope with points at infinity, we use a cut-off – e.g. the maximum finite homological critical value, denoted ω_0 – so that the contribution of a point at infinity to the 1-norm is finite.

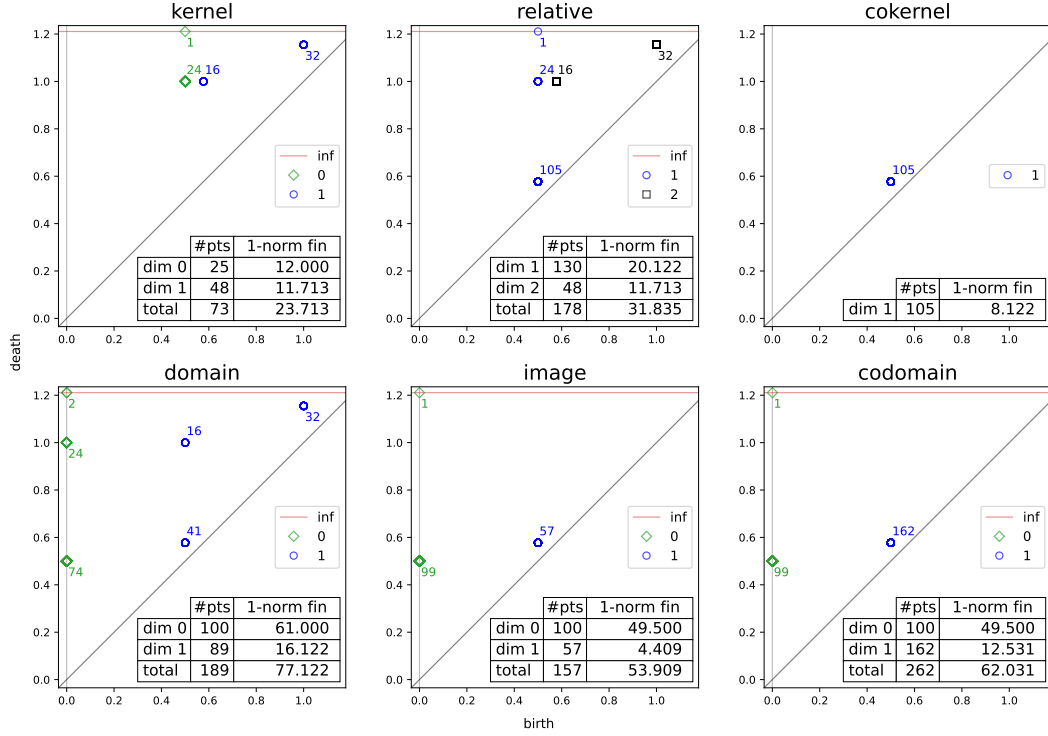


Figure 9 The six-pack for the 10×10 portion of the hexagonal lattice with coloring as in Figure 4. Important for the current discussion are the *diamond-shaped* points in the domain, image, and kernel diagrams. To get the MST-ratio, the 1-norms of the diagrams are computed while ignoring the points at infinity, giving 61.0 and 49.5 for the domain and the image diagrams, respectively. Compare the ratio of 1.232... with the upper bound of 1.25 proved in Theorem 4.2.

The kernel, domain, and image form a short exact sequence that splits, which implies $\|\text{Im}_p(\kappa)\|_1 + \|\text{Ker}_p(\kappa)\|_1 = \|\text{Dom}_p(\kappa)\|_1$; see [5, Theorem 5.3]. For dimension $p = 0$, all three 1-norms can be rewritten in terms of minimum spanning trees. Indeed, $\|\text{Dgm}_0(\mathbf{b})\|_1 = \frac{1}{2}|\text{MST}(B)| + \omega_0$ because every edge in the minimum spanning tree of B marks the death of a connected component in the sublevel set, and ω_0 is contributed by the one component that never dies. Similarly, $\|\text{Dgm}_0(\mathbf{c})\|_1 = \frac{1}{2}|\text{MST}(C)| + \omega_0$, which implies (45):

$$\|\text{Dom}_0(\kappa)\|_1 = \|\text{Dgm}_0(\mathbf{b})\|_1 + \|\text{Dgm}_0(\mathbf{c})\|_1 = \frac{1}{2}|\text{MST}(B)| + \frac{1}{2}|\text{MST}(C)| + 2\omega_0; \quad (45)$$

$$\|\text{Im}_0(\kappa)\|_1 = \frac{1}{2}|\text{MST}(A)| + \omega_0. \quad (46)$$

Since persistence diagrams are stable, as originally proved in [3], these relations imply that minimum spanning trees are similarly stable. (46) deserves a proof. There are two ways a connected component of B_r can die in the image: by merging with a component of C_r ,

or with another component of B_r . In the first case, the death corresponds to an edge of $\text{MST}(A)$ that connects a point in B with a point in C , and in the second case, it corresponds to an edge of $\text{MST}(A)$ that connects two points in B . There is also the symmetric case in which the edge connects two points in C . This establishes a bijection between the deaths in $\text{Im}_0(\kappa)$ and the edges of $\text{MST}(A)$. There is one component that never dies, which accounts for the extra cut-off term and implies (46).

The 1-norm of the kernel diagram is the difference between the 1-norms of the domain diagram and the image diagram: $\|\text{Ker}_0(\kappa)\|_1 = \|\text{Dom}_0(\kappa)\|_1 - \|\text{Im}_0(\kappa)\|_1$. It thus makes sense to call $\|\text{Im}_0(\kappa)\|_1/\|\text{Dom}_0(\kappa)\|_1$ and $\|\text{Ker}_0(\kappa)\|_1/\|\text{Dom}_0(\kappa)\|_1$ the *image share* and *kernel share*, respectively. Observe that both are real numbers between 0 and 1 and that they add up to 1. The intuition is that the kernel share is a measure of the amount of “0-dimensional mingling” of B and C . In other words, the smaller the image share, the more the two colors mingle. We therefore get

$$\mu(A, B) = \frac{|\text{MST}(B)| + |\text{MST}(C)|}{|\text{MST}(A)|} = \frac{\|\text{Dom}_0(\kappa)\|_1 - 2\omega_0}{\|\text{Im}_0(\kappa)\|_1 - \omega_0}, \quad (47)$$

for the MST-ratio, which besides the cut-off terms is the reciprocal of the image share. Hence, the larger the MST-ratio the more the two colors mingle. In this interpretation, Theorem 3.1 says that among all lattices in \mathbb{R}^2 , the hexagonal lattice is most restrictive to mingling as it does not permit MST-ratios larger than the inf-max, which for 2-dimensional lattices is 1.25.

Enumeration of Intersection Graphs of x -Monotone Curves

Jacob Fox ✉

Department of Mathematics, Stanford University, CA, USA

János Pach ✉

HUN-REN Alfréd Rényi Institute of Mathematics, Budapest, Hungary

Andrew Suk ✉

Department of Mathematics, University of California San Diego, La Jolla, CA, USA

Abstract

A curve in the plane is x -monotone if every vertical line intersects it at most once. A family of curves are called *pseudo-segments* if every pair of them have at most one point in common. We construct $2^{\Omega(n^{4/3})}$ families, each consisting of n labelled x -monotone pseudo-segments such that their intersection graphs are different. On the other hand, we show that the number of such intersection graphs is at most $2^{O(n^{3/2-\varepsilon})}$, where $\varepsilon > 0$ is a suitable constant. Our proof uses an upper bound on the number of set systems of size m on a ground set of size n , with VC-dimension at most d . Much better upper bounds are obtained if we only count *bipartite* intersection graphs, or, in general, intersection graphs with bounded chromatic number.

2012 ACM Subject Classification Mathematics of computing → Combinatorics

Keywords and phrases Enumeration, intersection graphs, pseudo-segments, x -monotone

Digital Object Identifier 10.4230/LIPIcs.GD.2024.4

Funding *Jacob Fox*: supported by NSF Award DMS-2154129.

János Pach: Rényi Institute, Budapest. Supported by NKFIH grant K-176529, Austrian Science Fund Z 342-N31 and ERC Advanced Grant “GeoScape.”

Andrew Suk: Supported by NSF CAREER award DMS-1800746, and NSF awards DMS-1952786 and DMS-2246847.

Acknowledgements We would like to thank Zixiang Xu for pointing out the reference [3] to us.

1 Introduction

The intersection graph of a collection \mathcal{C} of sets has vertex set \mathcal{C} and two sets in \mathcal{C} are adjacent if and only if they have nonempty intersection. A curve is a subset of the plane which is homeomorphic to the interval $[0, 1]$. A *string graph* is the intersection graph of a collection of curves. It is straightforward to show the intersection graph of any collection of arcwise connected sets in the plane is a string graph. A collection of curves in the plane is called a collection of *pseudo-segments* if every pair of them have at most one point in common. Finally, we say that a curve in the plane is x -monotone if every vertical line intersects it in at most one point.

For a family \mathcal{F} of simple geometric objects (namely those that can be defined by semi-algebraic relations of bounded description complexity), such as segments or disks in the plane, Warren’s theorem [24] can be used to show that the number of labelled graphs on n vertices which can be obtained as the intersection graph of a collection of n objects from \mathcal{F} is $2^{O(n \log n)}$ (see [16, 15]). Moreover, for many simple geometric objects, a result of Sauermann [19] shows that these bounds are essentially tight. Unfortunately, for general curves, Warren’s theorem cannot be applied. In this paper, we estimate the number of graphs which can be obtained as the intersection graph of curves in the plane under various constraints.



© Jacob Fox, János Pach, and Andrew Suk;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 4; pp. 4:1–4:12

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In [17], Pach and Tóth showed that the number of intersection graphs of n labelled pseudo-segments is at most $2^{o(n^2)}$. This bound was later improved by Kynčl [13] to $2^{O(n^{3/2} \log n)}$. It was noted in both papers that the best known lower bound on the number of intersection graphs of n labelled pseudo-segments is $2^{\Omega(n \log n)}$, the number of different labellings of the vertex set. Our first result significantly improves this bound.

► **Theorem 1.** *There are at least $2^{\Omega(n^{4/3})}$ labelled n -vertex intersection graphs of x -monotone pseudo-segments.*

In the other direction, we prove the following.

► **Theorem 2.** *There is an absolute constant $\varepsilon \in (0, 1)$ such that the following holds. There are at most $2^{O(n^{3/2-\varepsilon})}$ labelled n -vertex intersection graphs of x -monotone pseudo-segments in the plane.*

The ε in the theorem above can be taken to be roughly $1/D$, where D is the VC-dimension of intersection graphs of pseudo-segments in the plane. A result of Pach and Tóth [17] states that D is at most a tower of 2's of height 8.

In the case of small clique number, we obtain the following.

► **Theorem 3.** *There are at most $2^{O(kn \log^2 n)}$ labelled n -vertex intersection graphs of x -monotone pseudo-segments with clique number at most k . Moreover, for $k < n^{1/3}$, this bound is tight up to a polylogarithmic factor in the exponent.*

In [17], Pach and Tóth showed that the number of string graphs on n labelled vertices is $2^{\frac{3}{4} \binom{n}{2} + o(n^2)}$. Moreover, their result holds for x -monotone curves. Our next result shows that there are far fewer *bipartite* intersection graphs of x -monotone curves in the plane.

► **Theorem 4.** *There are at most $2^{O(n \log^2 n)}$ labelled n -vertex bipartite intersection graphs of x -monotone curves in the plane.*

Let us remark that the x -monotone condition in the theorem above cannot be removed. An interesting construction due to Keszegh and Pálvölgyi [12] implies that the number of n -vertex *bipartite* string graphs is at least $2^{\Omega(n^{4/3})}$.

For the non-bipartite case, suppose G is an n -vertex intersection of graph of x -monotone curves, such that G has chromatic number $q \geq 3$. Then we can partition $V(G)$ into q parts such that each part is an independent set. By further partitioning each part, arbitrarily, such that the size of each remaining part is at most n/q , we end up with at most $2q$ parts. By applying Theorem 4 to each pair of parts, we obtain the following corollary.

► **Corollary 5.** *There are at most $2^{O(qn \log^2 n)}$ labelled n -vertex intersection graphs of x -monotone curves with chromatic number at most q .*

Two drawings of a graph are *isomorphic* if the intersection graphs of their edges (with edges labelled by their endpoints) are the same. A *topological graph* is a graph drawn in the plane with possibly intersecting edges, and it is called *simple* if every pair of edges intersect at most once. A topological graph is *k -quasiplanar* if it has no k pairwise crossing edges with distinct endpoints.

The above results can be used to get upper bounds on the number of non-isomorphic drawings of a graph with certain properties. The next result is an immediate corollary of Theorem 3, combined with the theorem of Valtr [23] stating that the number of edges of a k -quasiplanar simple topological graph on n vertices with x -monotone edges is $O_k(n \log n)$.

► **Corollary 6.** *Given any n -vertex graph G , the number of non-isomorphic drawings of G as a k -quasiplanar simple topological graph with x -monotone edges is $2^{O_k(n \log^3 n)}$.*

In Theorems 3 and 4 and Corollaries 5 and 6, we conjecture that one of the logarithmic factors in the exponent can be removed. (In the case of Corollary 6, perhaps a factor $\log^2 n$ in the exponent can be removed). We discuss what is known from below at the end of the paper.

Our paper is organized as follows. In the next section, we prove Theorem 1. In Section 3, we establish a bound on the number of set systems of size m on a ground set of size n with VC-dimension d . After completing this work, we learned that Alon, Moran, and Yehudayoff [3] also found a proof of this result. Together with the well-known cutting lemma, we prove Theorem 2 in Section 4. In Section 5, we prove Theorem 4. We conclude the paper with some remarks.

2 Proof of Theorem 1

The proof of Theorem 1 is based on a well-known construction from incidence geometry. We prove the following more general result.

► **Theorem 7.** *For $k \leq n^{1/3}$, there are at least $2^{\Omega(kn)}$ n -vertex labelled intersection graphs of x -monotone pseudo-segments with clique number at most k .*

Proof. Let k and n be integers such that $k \leq n^{1/3}$. Take

$$P = \{(a, b) \in \mathbb{N}^2 : a < n^{1/3}, b < n^{2/3}\}$$

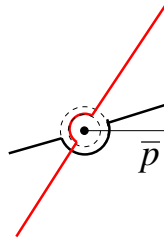
and

$$\mathcal{L} = \{a'x + b' = y : a', b' \in \mathbb{N}, a' < k, b' < n^{2/3}/2\}.$$

Then we have $|P| \leq n$ and $|\mathcal{L}| \leq kn^{2/3}/2 \leq n$, and each line in \mathcal{L} is incident to at least $n^{1/3}/4$ points from P . For each point $p = (a, b)$ in P , we replace p with a very short horizontal segment \bar{p} with endpoints (a, b) and $(a + \epsilon, b)$. Let \mathcal{H} be the resulting set of horizontal segments.

For each line $\ell \in \mathcal{L}$, we modify ℓ in a small neighborhood of each point in P that is incident to ℓ as follows. Let $\ell : y = a'x + b'$ and $p \in \ell$. Inside the circle C centered at p with radius $\frac{\epsilon}{2a'}$, we modify ℓ so that it is a half-circle along C that lies either above or below p . After performing this operation at each point p on ℓ , and performing a small perturbation, we obtain an x -monotone curve $\tilde{\ell}$. Moreover, any two resulting x -monotone curves will cross at most once. See Figure 1. Let \mathcal{L}_x be the resulting set of x -monotone curves, and note that $\mathcal{H} \cup \mathcal{L}_x$ is a set of x -monotone pseudo-segments.

We now count the number of intersection graphs between \mathcal{H} and \mathcal{L}_x . Since each line $\ell \in \mathcal{L}$ was incident to at least $n^{1/3}/4$ points in P , the number of different neighborhoods that can be generated for $\tilde{\ell}$ is $2^{\Omega(n^{1/3})}$. Moreover, two x -monotone curves $\tilde{\ell}, \tilde{\ell}' \in \mathcal{L}_x$ cross if and only if their original line configuration $\ell, \ell' \in \mathcal{L}$ have distinct slope. Thus, the intersection graph of $\mathcal{H} \cup \mathcal{L}_x$ has clique number at most k , and number of such intersection graphs we can create between \mathcal{H} and \mathcal{L}_x is at least $2^{\Omega(kn)}$. This completes the proof of Theorem 7. ◀



■ **Figure 1** Modifying lines through p .

3 Tools from VC-dimension theory

In this section, we recall and prove results related to the notion of VC-dimension. The *VC-dimension of a set system* \mathcal{F} on a ground set V is the *largest* integer d for which there exists a d -element set $S \subset V$ such that for every subset $B \subset S$, one can find a member $A \in \mathcal{F}$ with $A \cap S = B$. Note that for a multiset system (which allows for the sets to necessarily be distinct), the VC-dimension is the same as for the set system where we include each set that appears once.

Given a graph $G = (V, E)$, we define the *VC-dimension of G* to be the VC-dimension of the set system formed by the neighborhoods of the vertices, where the neighborhood of $v \in V$ is $N(v) = \{u \in V : uv \in E\}$. In [2], Alon et al. proved that the number of bipartite graphs with parts of size n and VC-dimension at most d is at most

$$2^{O(n^{2-1/d}(\log n)^{d+2})}.$$

They further asked if the logarithmic factors can be removed. We make progress on this question, obtaining a better bound for a more general problem. By following their proof but using the Haussler packing lemma [11] (stated below) instead of Lemma 26 in [2], one can obtain a stronger and more general bound. In addition to this, we use a different counting strategy that further removes an additional logarithmic factor.

For the sake of completeness, we include the short proof below. First, we will need some definitions. Given two sets $A, B \in \mathcal{F}$, the *distance* between A and B is $d(A, B) := |A \triangle B|$, where $A \triangle B = (A \cup B) \setminus A \cap B$ is the symmetric difference of A and B . We say that the set system \mathcal{F} is δ -separated if the distance between any two members in \mathcal{F} is at least δ . The following *packing lemma* was proved by Haussler in [11].

► **Lemma 8** ([11]). *Let $\delta > 0$ and \mathcal{F} be a set system on an n -element ground set V such that \mathcal{F} has VC-dimension d . If \mathcal{F} is δ -separated, then $|\mathcal{F}| \leq c_1(n/\delta)^d$ where $c_1 = c_1(d)$.*

Let $h_d(m, n)$ denote the number of multiset systems consisting of m subsets of $[n]$ that have VC-dimension at most d . Let $h'_d(m, n)$ denote the number of set systems of m subsets of $[n]$ that have VC-dimension at most d . Clearly, $h'_d(m, n) \leq h_d(m, n)$. For simplicity, we let $\binom{n}{\leq d} := \sum_{i=0}^d \binom{n}{i}$. The Sauer-Shelah lemma [18, 21] says that any set system with ground set $[n]$ and VC-dimension d has size at most $\binom{n}{\leq d}$. It follows that $h'_d(m, n) = 0$ if $m > \binom{n}{\leq d}$. Further, we can relate the two as follows. If we pick a multiset system consisting of m sets that has VC-dimension at most d , then by throwing out repeated sets, we get a set system on the same ground set consisting of $m' \leq m$ sets. We then have to fill out these m' sets to m sets with repeats, including each set at least once. We thus have

$$h_d(m, n) = \sum_{m' \leq m} h'_d(m', n) \binom{m-1}{m'-1}. \tag{1}$$

In what follows, d is fixed and the implicit constant in the big-O depends on d .

► **Theorem 9.** *Let $d \geq 2$ be fixed and $n, m \geq 2$. Then the number $h_d(m, n)$ of multiset systems of m subsets of $[n]$ with VC-dimension at most d satisfies*

$$h_d(m, n) = 2^{O(m^{1-1/d} n \log m)}.$$

Furthermore, if $m > n^d$, then

$$h_d(m, n) = 2^{O(n^d \log m)}.$$

Proof. Consider a linear ordering of the subsets of $[n]$. Let \mathcal{F} be a multiset system of m subsets of $[n]$. Let S_1 be the first set in \mathcal{F} by the linear ordering. We will order the sets in \mathcal{F} as S_1, S_2, \dots, S_m as follows. After picking S_1, \dots, S_{i-1} , let

$$\delta_i = \max_{S \in \mathcal{F} \setminus \{S_1, \dots, S_{i-1}\}} \min_{1 \leq j \leq i-1} d(S, S_j),$$

and S_i be a set S that obtains the maximum, and j_i be a j that obtains the minimum $d(S_i, S_j)$. By our choice of the sets, we have the minimum of $d(S_a, S_b)$ over all $1 \leq a < b \leq i$ is $d(S_{j_i}, S_i)$. By the Haussler packing lemma, we thus have $i = O((n/\delta_i)^d)$, or equivalently, $\delta_i = O(i^{-1/d} n)$.

We now upper bound the number of choices of \mathcal{F} . There are at most 2^n choices of S_1 . Each j_i is a positive integer at most $i - 1$, so there are at most $(m - 1)! \leq m^m$ choices of j_2, \dots, j_m . Having picked out this sequence of j_i 's, and having picked S_1, \dots, S_{i-1} , we know S_i must have symmetric difference at most $t_i := O(i^{-1/d} n)$ from S_{j_i} . Thus given this information, the number of choices for S_i is at most $\binom{n}{\leq t_i}$. Thus we get the number of choices of \mathcal{F} is at most

$$\begin{aligned} 2^n m^m \prod_{i=2}^m \binom{n}{\leq t_i} &\leq 2^n m^m \prod_{i=2}^m (O(i^{1/d}))^{O(i^{-1/d} n)} \\ &= 2^n m^m 2^{n \sum_{i=2}^m O(i^{-1/d} \log i)} \\ &= 2^n m^m 2^{O(m^{1-1/d} n \log m)}. \end{aligned}$$

Note that the 2^n factor is at most the last factor. Hence we get that the count is at most $m^m 2^{O(m^{1-1/d} n \log m)}$. If $m \leq n^d$, then the last factor is largest and this gives the desired bound.

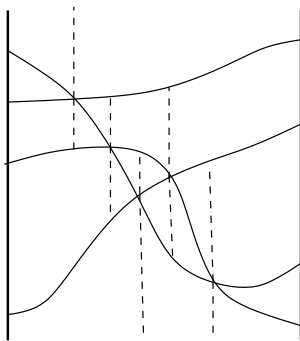
So we may assume we are in the case $m > n^d > \binom{n}{\leq d}$. In this case, by equation (1), the fact that $h'_d(m', n) = 0$ for $m' > n^d$ and $h'_d(m', n) \leq h_d(n^d, n)$, we get

$$h_d(m, n) \leq h_d(n^d, n) \sum_{m' \leq n^d} \binom{m-1}{m'-1} = 2^{O(n^d \log m)}.$$

Notice that in this case, the first bound still holds, as $m^{1-1/d} n \log m \geq n^d \log m$. ◀

4 Intersection graphs of x -monotone pseudo-segments

In this section, we prove Theorem 2. We will need the following lemmas. Recall that a *pseudoline* is a two-way infinite x -monotone curve in the plane. An *arrangement of pseudolines* is a finite collection of pseudolines such that any two members have at most one point in common, at which they cross, and each intersection point has a unique x -coordinate. Given an arrangement \mathcal{A} of n pseudolines, we obtain a sequence of permutations of $1, \dots, n$



■ **Figure 2** Vertical decomposition of \mathcal{A} .

by sweeping a directed vertical line across \mathcal{A} . This sequence of permutations is often referred to as an *allowable sequence of permutations*, which starts with the identity permutation $(1, \dots, n)$, such that i) the move from one permutation to the next consists of swapping two adjacent elements, and ii) each pair of elements switch exactly once. We say that two pseudoline arrangements \mathcal{A}_1 and \mathcal{A}_2 are *x -isomorphic* if they give rise to the same sequence of permutations, that is, a sweep with a vertical line meets the crossings in the same order.

► **Lemma 10** ([22]). *The number of arrangements of m pseudolines, up to x -isomorphism, is at most $2^{\Theta(m^2 \log m)}$.*

We will also need the following result, known as the *zone lemma* for pseudolines.

► **Lemma 11** ([8]). *Let \mathcal{A} be an arrangement of m pseudolines. Then for any $\alpha \in \mathcal{A}$, the sum of the numbers of sides in all the cells in the arrangement of \mathcal{A} that are supported by α is at most $O(m)$.*

The next lemma we will need is the following result due to Pach and Tóth.

► **Lemma 12** ([17]). *Let G be the intersection graph of a collection of pseudo-segments in the plane. Then the VC-dimension of G is at most an absolute constant d .*

We say that a collection \mathcal{A} of x -monotone pseudo-segments in the plane is *double grounded* if there are vertical lines ℓ_1 and ℓ_2 (called grounds) such that each curve in \mathcal{A} has its left endpoint on ℓ_1 and its right endpoint on ℓ_2 . We start by bounding the number of intersection graphs between a family \mathcal{A} of double grounded x -monotone curves and a family \mathcal{B} of x -monotone curves such that $\mathcal{A} \cup \mathcal{B}$ is a collection of pseudo-segments.

Let \mathcal{A} be a collection of double grounded x -monotone pseudo-segments in the plane. The *vertical decomposition* of the arrangement of \mathcal{A} is obtained by drawing a vertical segment from each crossing point and endpoint in the arrangement, in both directions, and extend it until it meets the arrangement of \mathcal{A} , else to $\pm\infty$. Since \mathcal{A} is double grounded, the grounds will appear in the vertical decomposition. The vertical decomposition of \mathcal{A} partitions the plane into cells called *generalized trapezoids*, where each generalized trapezoid is bounded by at most two curves from \mathcal{A} from above or below, and at most two vertical segments on the sides. See Figure 2. By applying standard random sampling arguments (e.g., see [6] or Lemma 4.6.1 in [14]), we obtain the following result known as the *weak cutting lemma*.

► **Lemma 13** ([6, 14]). *Let \mathcal{A} be a collection of m double grounded x -monotone pseudo-segments in the plane. Then for any parameter r , where $1 \leq r \leq m$, there is a set of at most $s = 6r \log m$ curves in \mathcal{A} whose vertical decomposition partitions the plane $\mathbb{R}^2 = \Delta_1 \cup \dots \cup \Delta_t$ into t generalized trapezoids, such that $t = O(s^2)$, and the interior of each Δ_i crosses at most m/r members in \mathcal{A} .*

Let $f(m, n)$ denote the number of labelled intersection graphs between a collection \mathcal{A} of m double grounded x -monotone curves whose grounds are the vertical lines at $x = 0$ and $x = 1$, and a collection \mathcal{B} of n x -monotone curves whose endpoints lie inside the strip $S = [0, 1] \times \mathbb{R}$ such that $\mathcal{A} \cup \mathcal{B}$ is a collection of pseudo-segments. By Lemma 12, there is an absolute constant d such that the VC-dimension of any intersection graph of pseudo-segments in the plane is at most d . We now prove the following.

► **Lemma 14.** *For $m, n \geq 1$, we have*

$$f(m, n) \leq 2^{O(n^{d/(2d-1)} m^{(2d-2)/(2d-1)} \log^2 m)} + 2^{O(n^{3/2-1/d} \log n)} + 2^{O(m \log^3 m)}.$$

Proof. We can assume that m, n are sufficiently large. Let $\mathcal{A} \cup \mathcal{B}$ be a collection of pseudo-segments where \mathcal{A} and \mathcal{B} are as above.

Suppose $n > m^2$. Then by the first part of Theorem 9, the number of intersection graphs between \mathcal{A} and \mathcal{B} is at most

$$h_d(n, m) \leq 2^{O(mn^{1-1/d} \log n)} \leq 2^{O(n^{3/2-1/d} \log n)}. \quad (2)$$

If $n < m^{1/d} \log^2 m$, then by the second part of Theorem 9, the number of intersection graphs between \mathcal{A} and \mathcal{B} is at most

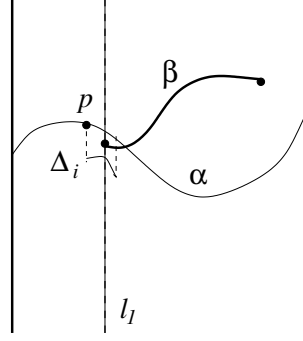
$$h_d(m, n) \leq 2^{O(m^{1-1/d} n \log m)} \leq 2^{O(m \log^3 m)}. \quad (3)$$

Let us assume that $m^{1/d} < n < m^2$. Set $r = \frac{n^{d/(2d-1)}}{(m \log^{2d} m)^{1/(2d-1)}}$ and $s = 6r \log m$. Since m and n are sufficiently large, we have $1 \leq r < s \leq m$. For a set of m double grounded x -monotone curves whose grounds are on the vertical lines $x = 0$ and $x = 1$, there are $(m!)^2$ ways to order the left and right endpoints. Let us fix such an ordering.

Let $\mathcal{A}' \subset \mathcal{A}$ be a set of at most $s = 6r \log m$ curves from \mathcal{A} whose arrangement gives rise to a vertical decomposition satisfying Lemma 13 with parameter r . Note that there are at most m^s choices for \mathcal{A}' , and by Lemma 10, there are at most $2^{O(s^2 \log s)}$ ways to fix the arrangement, up to x -isomorphism, for \mathcal{A}' . Once the arrangement of \mathcal{A}' is fixed, the vertical decomposition of \mathcal{A}' is determined.

Let $\mathbb{R}^2 = \Delta_1 \cup \dots \cup \Delta_t$ be the vertical decomposition corresponding to \mathcal{A}' , where $t = O(s^2)$. Let $\mathcal{A}_i \subset \mathcal{A}$ be the curves in \mathcal{A} that cross the cell Δ_i . For each curve $\alpha \in \mathcal{A}'$, by Lemma 11, at most $O(s)$ vertical segments from the vertical decomposition have an endpoint on α . Moreover, at most m curves from \mathcal{A} cross α . Among these $O(s+m)$ points along α , let us fix the order in which they appear along α , from left to right. Since there are at most s^2 vertical segments, there are at most $(s^2 + m)^{O(s+m)} = m^{O(m)}$ ways to fix this ordering, and therefore, there are at most $m^{O(sm)}$ ways to fix such an ordering for each curve $\alpha \in \mathcal{A}'$.

Let $\beta \in \mathcal{B}$. Then there are $O(s^4)$ choices for the cells Δ_i for which the endpoints of β lie in. Suppose that the left endpoint of β lies in cell Δ_i and the right endpoint lies in Δ_j , and consider the vertical lines ℓ_1 and ℓ_2 that goes through the left and right endpoint of β respectively. Then for each $\alpha' \in \mathcal{A} \setminus (\mathcal{A}_i \cup \mathcal{A}_j)$, we have already determined if α' crosses β . Indeed, let us consider the left endpoint of β and the cell Δ_i . By the vertical decomposition, Δ_i is bounded either above or below by some curve $\alpha \in \mathcal{A}'$. Without loss of generality, let us assume that Δ_i is bounded from above by α . Let p be the point on α that intersects the left side of Δ_i . Then for any $\alpha' \in \mathcal{A} \setminus (\mathcal{A}_i \cup \mathcal{A}_j)$, we have already determined if the left endpoint of α' is above or below the left endpoint of α along the ground $x = 0$. Moreover, we have already determined if α' crosses α to the left of point p . Since α' does not cross Δ_i , we have determined if α' crosses ℓ_1 above or below β . See Figure 3. By the same argument, we have determined if α' crosses ℓ_2 above or below the right endpoint of β . Therefore, by the pseudo-segment condition, we have determined if α' crosses β .



■ **Figure 3** Cell Δ_i bounded above by α and contains the left endpoint of β .

It remains to determine how many ways β can cross the curves in \mathcal{A}_i and \mathcal{A}_j . By Lemma 13, $|\mathcal{A}_i| \leq m/r$. Let \mathcal{B}_i denote the curves in \mathcal{B} that has at least one endpoint in the cell Δ_i . Set $n_i = |\mathcal{B}_i|$. By Theorem 9, there are at most

$$h_d(|\mathcal{A}_i|, |\mathcal{B}_i|) \leq 2^{O((m/r)^{1-1/d} n_i \log m)}$$

ways the curves in \mathcal{A}_i cross the curves in \mathcal{B}_i . Putting everything together, the number of ways the curves in \mathcal{A} cross the curves in \mathcal{B} is at most

$$(m!)^2 m^s 2^{O(s^2 \log s)} m^{O(sm)} (s^4)^n \prod_{i=1}^t 2^{O((m/r)^{1-1/d} n_i \log m)}.$$

Since $t = O(s^2)$, $r = \frac{n^{d/(2d-1)}}{(m \log^{2d} m)^{1/(2d-1)}}$, and $s = 6r \log m \leq m$, this quantity is at most

$$2^{O((m/r)^{1-1/d} n \log m + s^2 (m/r) \log m)} \leq 2^{O(n^{d/(2d-1)} m^{(2d-2)/(2d-1)} \log^2 m)}. \quad (4)$$

Combining (2), (3), and (4), we have

$$f(m, n) \leq 2^{O(n^{d/(2d-1)} m^{(2d-2)/(2d-1)} \log^2 m)} + 2^{O(n^{3/2-1/d} \log n)} + 2^{O(m \log^3 m)}. \quad \blacktriangleleft$$

Hence, we have $f(n, n) \leq 2^{O(n^{3/2-1/(4d-2)} \log^2 n)}$, where d is the absolute constant from Lemma 12.

Proof of Theorem 2. Let d be the absolute constant from Lemma 12. Let $g(n; p)$ be the number of labeled intersection graphs of at most n x -monotone pseudo-segments in the vertical strip $[0, 1] \times \mathbb{R}$, such that there are at most p endpoints with x -coordinate in $(0, 1)$. Note that some pseudo-segments may contribute two endpoints to p . Then we have the following recurrence.

▷ **Claim 15.** We have

$$g(n; p) \leq 2^{O(n^{3/2-1/(4d-2)} \log^2 n)} g^2(\lceil p/2 \rceil; \lceil p/2 \rceil).$$

Proof. For n x -monotone curves in the strip $S = [0, 1] \times \mathbb{R}$, with p endpoints in the interior of S , we can assume that these p endpoints have distinct x -coordinates. We partition the interval $[0, 1]$ into two parts I_1, I_2 , so that the interior of each strip $S_i = I_i \times \mathbb{R}$ has at most $\lceil p/2 \rceil$ endpoints. Next, we upper bound the number of labeled intersection graphs of the curves restricted to the strip S_i . Note that there are $n!$ ways to label the curves.

Among the curves restricted to the strip S_i , let \mathcal{A}_i denote the set of curves that go entirely through S_i , and let \mathcal{B}_i be the curve what at least one endpoint in the interior of S_i . There are at most $n!$ ways to determine the intersection graph among the curves in \mathcal{A}_i . By Lemma 14, there are at most

$$f(|\mathcal{A}_i|, |\mathcal{B}_i|) \leq f(n, n) \leq 2^{O(n^{3/2-1/(4d-2)} \log^2 n)}$$

ways to determine the intersection graph between \mathcal{A}_i and \mathcal{B}_i . Finally, there are at most $g(\lceil p/2 \rceil; \lceil p/2 \rceil)$ ways to determine the intersection graph among the curves in \mathcal{B}_i . Putting everything above together gives the desired recurrence. \triangleleft

Since $p \leq 2n$, the recurrence above gives

$$g(n; 2n) \leq 2 \sum_{i=1}^{\log n} 2^i O((n/2^i)^{3/2-1/(4d-2)} \log^2(n/2^i)) \quad g(1; 1) \leq 2^{O(n^{3/2-1/(4d-2)} \log^2 n)}. \quad \blacktriangleleft$$

Intersection graphs with small clique number

In this subsection, we prove Theorem 3.

Proof of Theorem 3. Next, we prove Theorem 3. Let $g_k(n; p)$ be the number of labeled intersection graphs of at most n x -monotone pseudosegments with clique number at most k in the vertical strip $[0, 1] \times \mathbb{R}$, such that there are at most p endpoints with x -coordinate in $(0, 1)$. Similar to above, we will show

$$g_k(n; p) \leq n^{6n+2kp} g_k^2(\lceil p/2 \rceil; \lceil p/2 \rceil).$$

Indeed, for n x -monotone pseudosegments in the strip $S = [0, 1] \times \mathbb{R}$, with p endpoints in the interior of S , we can assume that these p endpoints have distinct x -coordinate. We partition the interval $[0, 1]$ into two parts I_1, I_2 , so that the interior of each strip $S_i = I_i \times \mathbb{R}$ has at most $\lceil p/2 \rceil$ endpoints. We now bound the number of labeled intersection graphs of the curves restricted to S_1 .

There are at most $n!$ ways to label the curves in S_1 . There are at most 2^n ways to choose the set \mathcal{A} of pseudo-segments that goes entirely through S_1 . Let $G_{\mathcal{A}}$ denote its intersection graph of \mathcal{A} , restricted to S_1 . Then $G_{\mathcal{A}}$ depends entirely on the permutation of the endpoints of \mathcal{A} . Hence, there are at most $n!$ ways to determine $G_{\mathcal{A}}$. Since $G_{\mathcal{A}}$ has clique number at most k , by Dilworth's theorem [7], $G_{\mathcal{A}}$ has chromatic number at most k . Thus, there are at most k^n ways to properly color the vertices of $G_{\mathcal{A}}$. After fixing such a coloring, let $\mathcal{A}_1, \dots, \mathcal{A}_k$ denote the color classes. Since the curves in \mathcal{A}_i are pairwise disjoint and goes through S_1 , for each curve γ with an endpoint in the interior of S_1 , there are at most n^2 ways γ can intersect the curves in \mathcal{A}_i . Therefore, there are at most $(n^2)^k$ ways γ can intersect the curves in \mathcal{A} . Since $k \leq n$, there are at most

$$n! 2^n n! k^n (n^2)^{kp/2} g_k(\lceil p/2 \rceil; \lceil p/2 \rceil) \leq n^{4n+kp} g_k(\lceil p/2 \rceil; \lceil p/2 \rceil)$$

labeled intersection graphs among the curves restricted to S_1 . A similar argument holds for the curves restricted to S_2 . Hence,

$$g_k(n; p) \leq n^{8n+2kp} g_k^2(\lceil p/2 \rceil; \lceil p/2 \rceil).$$

Iterating the inequality above t times gives

$$g_k(n; p) \leq n^{8n+2kp} \left(\frac{p}{2}\right)^{8p+2kp} \left(\frac{p}{2^2}\right)^{8p+2kp} \dots \left(\frac{p}{2^{t-1}}\right)^{8p+2kp} g^{2^t}(\lceil p/2^t \rceil; \lceil p/2^t \rceil).$$

4:10 Enumeration of Intersection Graphs of x -Monotone Curves

Hence for $t = \lceil \log_2 n \rceil$, we have

$$g_k(n; p) \leq n^{8n+2kp} p^{(8p+2kp)t}.$$

By setting $p = 2n$, we have

$$g_k(n; 2n) \leq 2^{O(kn \log^2 n)},$$

and Theorem 3 follows. ◀

5 Bipartite intersection graphs of x -monotone curves

In this section, we prove Theorem 4. The proof is very similar to the proof of Theorem 3 above. Let $w(n; p)$ be the number of labeled bipartite intersection graphs of at most n x -monotone curves in the vertical strip $[0, 1] \times \mathbb{R}$, such that there are at most p endpoints with x -coordinate in $(0, 1)$. We establish the following recurrence.

► **Lemma 16.** *We have*

$$w(n; p) \leq n^{6n} w^2(\lceil p/2 \rceil; \lceil p/2 \rceil).$$

Proof. For n x -monotone curves in the strip $S = [0, 1] \times \mathbb{R}$, with p endpoints in the interior of S , we can assume that these p endpoints have distinct x -coordinate. We partition the interval $[0, 1]$ into two parts I_1, I_2 , so that the interior of each strip $S_i = I_i \times \mathbb{R}$ has at most $\lceil p/2 \rceil$ endpoints. Next, we upper bound the number of labeled intersection graphs of the curves restricted to the strip S_i . Note that there are $n!$ ways to label the curves.

For each curve γ , as the graph is bipartite, let us count the number of ways γ intersects the set of pairwise disjoint curves that go entirely through S_i . By ordering these pairwise disjoint curves vertically, this intersection set is an interval with respect to this vertical ordering. Hence, γ has at most n^2 ways to intersect the family of curves that goes entirely through S_i . This gives a total of at most $n!(n^2)^n < n^{3n}$ ways of determining the intersection graph in S_i , apart from the induced subgraph on the curves with at least one endpoint in the interior of S_i . Since there are $p/2$ such endpoints, there are at most $p/2$ such curves. Thus we have at most $w(\lceil p/2 \rceil; \lceil p/2 \rceil)$ possible such intersection graphs of the curves with one endpoint in S_i . Thus we have at most $n^{3n} w(\lceil p/2 \rceil; \lceil p/2 \rceil)$ possible intersection graphs restricted to S_i . Since the intersection graph of all n curves is the union of the intersection graphs on S_1 and S_2 , we get in total at most $(n^{3n} w(\lceil p/2 \rceil; \lceil p/2 \rceil))^2$ such choices. ◀

Proof of Theorem 4. It suffices to bound $w(n; 2n)$ as the original n curves have $2n$ endpoints. Iterating the recurrence in Lemma 16 t times gives

$$w(n; p) \leq n^{6n} \left(\frac{p}{2}\right)^{6p} \left(\frac{p}{2^2}\right)^{6p} \cdots \left(\frac{p}{2^{t-1}}\right)^{6p} w^{2^t}(\lceil p/2^t \rceil; \lceil p/2^t \rceil).$$

Thus for $t = \lceil \log_2 n \rceil$, we get

$$w(n; p) \leq n^{6n} p^{6pt}.$$

Hence,

$$w(n; 2n) \leq 2^{O(n \log^2 n)}. \quad \blacktriangleleft$$

Let us remark that in [9], the first two authors showed that there is an absolute constant $c > 0$ such that every n -vertex string graph with clique number k has chromatic number at most $(C \frac{\log n}{\log k})^{c \log k}$. Together with Corollary 5, we obtain the following.

► **Corollary 17.** *For every $\epsilon > 0$, there is $\delta > 0$ such that the number of intersection graphs of n x -monotone curves with clique number at most n^δ is at most $2^{n^{1+\epsilon}}$.*

6 Concluding remarks

An important motivation for enumerating intersection graphs of curves of various kinds came from a question in graph drawing [17]: How many ways can one draw a graph? The number of different (non-isomorphic) drawings of K_n , a complete graph of n vertices, can be upper-bounded by the number intersection graphs of $\binom{n}{2}$ curves. By [17], this is at most $2^{(3/2^5 + o(1))n^4}$.

The number of non-isomorphic *straight-line drawings* of K_n cannot exceed the the number of different intersection graphs of $\binom{n}{2}$ segments in the plane, which is $2^{(4+o(1))n^2 \log n}$; see [19, 16]. However, the true order of magnitude of the number of straight-line drawings of K_n is much smaller. As was pointed out in [17], this quantity is equal to the number of *order types* of n points in general position in the plane. The latter quantity is $2^{(4+o(1))n \log n}$, according to seminal results of Goodman–Pollack [10] and Alon [1], based on Warren’s theorem in real algebraic geometry [24].

Recall that Theorem 9 in Section 3 shows that for $d \geq 2$ fixed and $m, n \geq 2$, the number $h'_d(m, n)$ of set systems of m subsets of $[n]$ that have VC-dimension at most d is at most $2^{O(nm^{1-1/d} \log m)}$. It would be interesting to remove the logarithmic factor in the exponent, which would answer the question of Alon et al. [2] mentioned in the beginning of Section 3. A natural approach, which has worked for similar enumerative problems, is to recast the problem as counting independent sets in an auxiliary hypergraph and use the hypergraph container method. Consider the 2^{d+1} -uniform hypergraph H with vertex set $2^{[n]}$ (so the vertices are just the subsets of $[n]$) and a 2^{d+1} -tuple of vertices forms an edge if they shatter a subset of the ground set of size $d + 1$. The function $h'_d(m, n)$ then just counts the number of independent sets of size m in H . The hypergraph container method (introduced in [4, 20], see also [5]) is a powerful tool that is useful for counting independent sets in similar settings. It would be interesting if one could adapt these techniques to give better bounds on $h'_d(m, n)$.

The last five results in the introduction give upper bounds on the number of intersection graphs or the number of non-isomorphic drawings of graphs under various constraints. It would be interesting to close the gap between these upper bounds and lower bounds.

The following simple construction shows that there are $2^{\Omega(n \log n)}$ unlabelled bipartite graphs on n vertices that are intersection graphs of segments. One can fix the first $k = n / \log n$ segments to be vertical and cross the x -axis, and then have the freedom to choose the remaining $n - k$ segments to be horizontal, deciding which interval of vertical segments (ordered by x -axis intersection point) to intersect. By having, for $i \in [k]$, a horizontal segment that intersects precisely the first i of the vertical segments, we can fix the underlying ordering of the vertical segments, up to reversing the order, and use the remaining $n - 2k$ horizontal segments to pick any interval of the vertical segments to intersect. One gets $2^{(2-o(1))n \log_2 n}$ labelled bipartite intersection graphs (and hence at least $2^{(1-o(1))n \log_2 n}$ unlabelled bipartite intersection graphs). This shows that Theorem 4 is tight up to a single logarithmic factor in the exponent.

Viewing the same construction as a drawing of a matching (with the endpoints of segments as vertices of the matching), gives $2^{\Omega(n \log n)}$ non-isomorphic straight-line drawings of a matching on n vertices whose edge-intersection graph is bipartite, providing a lower bound for Corollary 6.

References

- 1 N. Alon. The number of polytopes, configurations and real matroids. *Mathematika*, 33:62–71, 1986.
- 2 N. Alon, J. Balogh, B. Bollobás, and R. Morris. The structure of almost all graphs in a hereditary property. *J. Combin. Theory Ser. B*, 101:85–110, 2011. doi:10.1016/J.JCTB.2010.10.001.
- 3 N. Alon, S. Moran, and A. Yehudayoff. Sign rank versus vc dimension. *Mat. Sbornik*, 208:1724–1757, 2017.
- 4 J. Balogh, R. Morris, and W. Samotij. Independent sets in hypergraphs. *J. Amer. Math. Soc.*, 28:669–709, 2015.
- 5 J. Balogh, R. Morris, and W. Samotij. The method of hypergraph containers. In *Proceedings of the International Congress of Mathematicians-Rio de Janeiro 2018. Vol. IV. Invited lectures*, pages 3059–3092. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2018.
- 6 K.L. Clarkson and P. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989. doi:10.1007/BF02187740.
- 7 R. Dilworth. A decomposition theorem for partially ordered sets. *Ann. of Math.*, 51:161–166, 1950.
- 8 D. Eppstein, P. Plassmann, and F. Yao. Horizon theorems for lines and polygons. In *Discrete and Computational Geometry: Papers from the DIMACS Special Year, J.E. Goodman, R. Pollack, and W. Steiger, editors*, pages 45–66. AMS, Providence, 1991.
- 9 Jacob Fox, János Pach, and Andrew Suk. Quasiplanar graphs, string graphs, and the Erdős-Gallai problem. In Patrizio Angelini and Reinhard von Hanxleden, editors, *Graph Drawing and Network Visualization*, pages 219–231, Cham, 2023. Springer International Publishing.
- 10 J. E. Goodman and R. Pollack. Upper bounds for configurations and polytopes in \mathbf{R}^d . *Discrete Comput. Geom.*, 1:219–227, 1986. doi:10.1007/BF02187696.
- 11 D. Haussler. Sphere packing numbers for subsets of the Boolean n -cube with bounded Vapnik Chervonenkis dimension. *J. Combin. Theory Ser. A*, 69:217–232, 1995.
- 12 B. Keszegh and D. Pálvölgyi. The number of tangencies between two families of curves. preprint, arXiv:2111.08787, 2024+.
- 13 J. Kynčl. Improved enumeration of simple topological graphs. *Discrete Comput. Geom.*, 10:727–770, 2013.
- 14 J. Matoušek. *Lectures on Discrete Geometry*. Springer-Verlag, New York, 2002.
- 15 C. McDiarmid and T. Müller. The number of disk graphs. *European J. Combin.*, 35:413–431, 2014. doi:10.1016/J.EJC.2013.06.037.
- 16 J. Pach and J. Solymosi. Crossing patterns of segments. *J. Combin. Ser. A.*, 96:316–325, 2001. doi:10.1006/JCTA.2001.3184.
- 17 J. Pach and G. Tóth. How many ways can one draw a graph? *Combinatorica*, 26:559–576, 2006. doi:10.1007/S00493-006-0032-Z.
- 18 N. Sauer. On the density of families of sets. *J. Combinat. Theory Ser. A*, 13:145–147, 1972. doi:10.1016/0097-3165(72)90019-2.
- 19 L. Sauerermann. On the speed of algebraically defined graph classes. *Adv. Math.*, 380:Article 107593, 2021.
- 20 D. Saxton and A. Thomason. Hypergraph containers. *Invent. Math.*, 201:925–992, 2015.
- 21 S. Shelah. A combinatorial problem, stability and order for models and theories in infinitary languages. *Pacific J. Math.*, 41:247–261, 1972.
- 22 R. P. Stanley. On the number of reduced decompositions of elements of Coxeter groups. *European J. Combin.*, 5:359–372, 1984. doi:10.1016/S0195-6698(84)80039-6.
- 23 P. Valtr. Graph drawing with no k pairwise crossing edges. In *Graph drawing (Rome, 1997), Lecture Notes in Comput. Sci.*, pages 205–218. Springer-Verlag, Berlin, 1997.
- 24 H. E. Warren. Lower bounds for approximation by linear manifolds. *Trans. Amer. Math. Soc.*, 133:167–178, 1968.

Holes in Convex and Simple Drawings

Helena Bergold  

Institute of Computer Science, Freie Universität Berlin, Germany
TUM School of Computation, Information and Technology,
Technische Universität München, Germany

Joachim Orthaber  

Institute of Software Technology, Graz University of Technology, Austria

Manfred Scheucher  

Institut für Mathematik, Technische Universität Berlin, Germany

Felix Schröder  

Department of Applied Mathematics, Faculty of Mathematics and Physics,
Charles University, Prague, Czech Republic
Institute of Mathematics, Technische Universität Berlin, Germany

Abstract

Gons and holes in point sets have been extensively studied in the literature. For simple drawings of the complete graph a generalization of the Erdős–Szekeres theorem is known and empty triangles have been investigated. We introduce a notion of k -holes for simple drawings and study their existence with respect to the convexity hierarchy. We present a family of simple drawings without 4-holes and prove a generalization of Gerken’s empty hexagon theorem for convex drawings. A crucial intermediate step will be the structural investigation of pseudolinear subdrawings in convex drawings.

2012 ACM Subject Classification Mathematics of computing → Discrete mathematics; Theory of computation → Computational geometry

Keywords and phrases simple topological graph, convexity hierarchy, k -gon, k -hole, empty k -cycle, Erdős–Szekeres theorem, Empty Hexagon theorem, planar point set, pseudolinear drawing

Digital Object Identifier 10.4230/LIPIcs.GD.2024.5

Related Version *Full Version*: <https://arxiv.org/abs/2409.01723> [11]

Funding *Helena Bergold*: Supported by DFG-Research Training Group “Facets of Complexity” (DFG-GRK 2434).

Joachim Orthaber: Supported by the Austrian Science Fund (FWF) grant W1230.

Manfred Scheucher: Supported by DFG Grant SCHE 2214/1-1.

Felix Schröder: Supported by the GAČR Grant no. 23-04949X.

1 Introduction

A classic theorem from combinatorial geometry is the Erdős–Szekeres theorem [15]. It states that for every $k \in \mathbb{N}$ every sufficiently large point set in general position (i.e., no three points on a line) contains a subset of k points that are the vertices of a convex polygon, a so called k -gon. In this article we will focus on a prominent variant of the Erdős–Szekeres theorem suggested by Erdős himself [14], which asks for the existence of *empty* k -gons, also known as k -holes. A k -hole H in a point set P is a k -gon with the property that there are no points of P in the interior of the convex hull of H . It is known that every sufficiently large point set contains a 6-hole [18, 23] and that there are arbitrarily large point sets without 7-holes [21].

Point sets in general position are in correspondence with *geometric drawings* of the complete graph where vertices are mapped to points and edges are drawn as straight-line segments between the vertices. In this article we generalize the notion of holes to simple



© Helena Bergold, Joachim Orthaber, Manfred Scheucher, and Felix Schröder;
licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 5; pp. 5:1–5:9

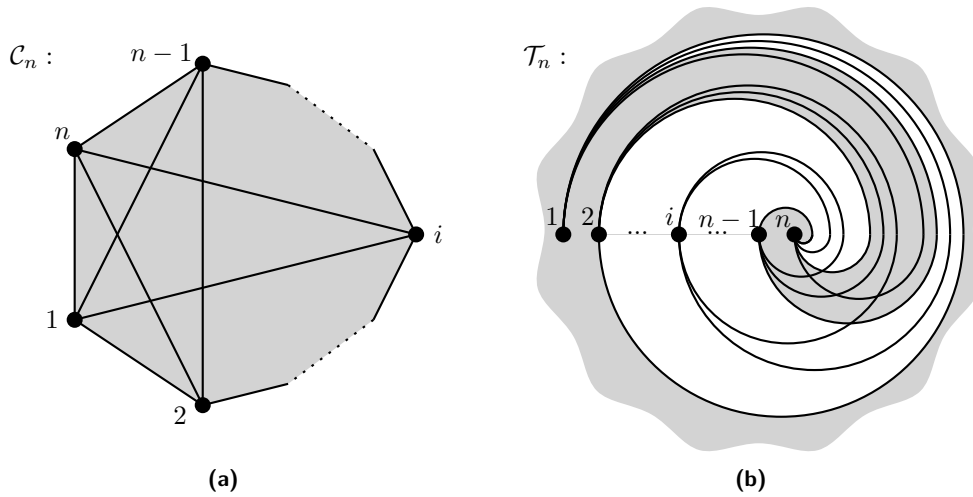
Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

drawings of the complete graph K_n . In a *simple drawing*, vertices are mapped to distinct points in the plane (or on the sphere) and edges are mapped to simple curves connecting the two corresponding vertices such that two edges have at most one point in common, which is either a common vertex or a proper crossing. In the course of this article, we will see that many important properties do not depend on the full drawing but only on the underlying combinatorics, more specifically on the isomorphism class of a drawing. We call two simple drawings of the same graph *isomorphic*¹ if there is a bijection between their vertex sets such that the corresponding pairs of edges cross. Note that this isomorphism is independent of the choice of the outer cell and thus only encodes the simple drawing on the sphere.

To study k -holes, we first extend the notion of k -gons to simple drawings of K_n . A k -gon \mathcal{C}_k is a subdrawing isomorphic to the geometric drawing on k points in convex position; see Figure 1(a). In terms of crossings, a k -gon \mathcal{C}_k is a (sub)drawing with vertices v_1, \dots, v_k such that $\{v_i, v_\ell\}$ crosses $\{v_j, v_m\}$ exactly if $i < j < \ell < m$. In contrast to the geometric setting where every sufficiently large geometric drawing contains a k -gon, simple drawings of complete graphs do not necessarily contain k -gons [19]. For example, the twisted drawing \mathcal{T}_n depicted in Figure 1(b) does not contain any 5-gon. In terms of crossings, \mathcal{T}_n can be characterized as a drawing of K_n with vertices v_1, \dots, v_n such that $\{v_i, v_m\}$ crosses $\{v_j, v_\ell\}$ exactly if $i < j < \ell < m$. A theorem by Pach, Solymosi and Tóth [24] states that, for every k , every sufficiently large simple drawing of K_n contains \mathcal{C}_k or \mathcal{T}_k . The currently best known estimate is due to Suk and Zeng [28] who showed that every simple drawing of K_n with $n > 2^{9 \cdot \log_2(a) \log_2(b) a^2 b^2}$ contains \mathcal{C}_a or \mathcal{T}_b . Convex drawings, which we define in the next paragraph, are a class of drawings nested between geometric drawings and simple drawings. In particular, convex drawings do not contain \mathcal{T}_5 as a subdrawing. Hence every convex drawing of K_n contains a k -gon \mathcal{C}_k with $k = (\log n)^{1/2 - o(1)}$.



■ **Figure 1** A drawing of (a) an n -gon \mathcal{C}_n and (b) a twisted \mathcal{T}_n for $n \geq 4$.

In the last decades, holes were intensively studied for the setting of point sets. Our focus will be on determining the existence of holes in various layers of the convexity hierarchy introduced by Arroyo et al. [5], which give a more fine-grained layering between geometric drawings and simple drawings. The basis to define convexity are *triangles*, which are

¹ This isomorphism is often referred to as “weak isomorphism” since there also exist stronger notions.

subdrawings induced by three vertices. Since in a simple drawing incident edges do not cross, a triangle separates the plane (resp. the sphere) into two connected components. The closure of each of the components is called a *side* of the triangle. A side S is *convex* if, for every pair of vertices in S , the connecting edge is fully contained in S . A simple drawing \mathcal{D} of K_n is

- *convex* if every triangle in \mathcal{D} has a convex side;
- *h-convex* (hereditarily convex) if there is a choice of a convex side S_T for every triangle T such that, for every triangle T' contained in S_T , it holds $S_{T'} \subseteq S_T$;
- *f-convex* (face convex) if there is a marking face F in the plane such that for all triangles the side not containing F is convex.

The class of f-convex drawings is related to pseudolinear drawings. A *pseudolinear drawing* is a simple drawing in the plane such that the edges can be extended to an arrangement of pseudolines. A *pseudoline* is a simple curve partitioning the plane into two unbounded components and in an *arrangement* each pair of pseudolines has exactly one point in common, which is a proper crossing. As shown by Arroyo et al. [4], a simple drawing of K_n is pseudolinear if and only if it is f-convex and the marking face F is the unbounded face. For more information about the convexity hierarchy we refer the reader to [4, 5, 6, 10].

Before we define k -holes, consider the case of 3-holes, also known as empty triangles. A triangle is *empty* if one of its two sides does not contain any vertex in its interior. Harborth [19] proved that every simple drawing of K_n contains at least two empty triangles and conjectured that the minimum among all simple drawings of K_n is $2n - 4$. While $2n - 4$ is obtained by \mathcal{T}_n and all generalized twisted drawings [17], the best known lower bound is n [3].

In the geometric setting, the number of empty triangles behaves differently: every point set has $\Omega(n^2)$ empty triangles, and this bound is asymptotically optimal [8]. Note that the notion of empty triangles in point sets slightly differs from the one in simple drawings since the complement of the convex hull of a point set can be an empty triangle. The class of convex drawings behaves similarly to the geometric setting: the minimum number of empty triangles is asymptotically quadratic [4, Theorem 5].

In the drawing \mathcal{C}_k with $k \geq 4$, all triangles have exactly one empty side, which is the unique convex side. The *convex side* of \mathcal{C}_k is the union of convex sides of its triangles; see the grey shaded regions in Figure 1. Given a k -gon \mathcal{C}_k in a simple drawing of K_n , we call vertices in the interior of the convex side of \mathcal{C}_k *interior vertices*. A k -hole in a simple drawing of K_n is a k -gon that has no interior vertices. For example, the vertices $1, 2, n - 1, n$ form a 4-hole in \mathcal{T}_n ; marked grey in Figure 1(b). In convex drawings, as in the geometric setting, edges from an interior vertex to a vertex of \mathcal{C}_k and edges between two interior vertices are contained in the convex side of \mathcal{C}_k [5, Lemma 3.5]. For more details see the full version [11].

In this paper, using the notion of k -holes in simple drawings defined above, we resolve the questions of existence of 4-, 5- and 6-holes in simple and convex drawings of K_n . In particular, we show the existence of 6-holes in sufficiently large convex drawings (Theorem 2.1), generalizing Gerken's empty hexagon theorem [18]. The key ingredient of the proof is that any subdrawing induced by a minimal k -gon together with its interior vertices is f-convex (Lemma 2.2). This allows to transfer various existential results from the geometric, pseudolinear, and f-convex settings to convex drawings. Besides the existence of 6-holes, we also show the existence of monochromatic generalized 4-holes in two-colored convex drawings (Corollary 3.1), generalizing a result by Aichholzer et al. [2]. For this we discuss two variants of generalized holes (Section 3) in the setting of simple drawings of K_n and show the existence of plane cycles of length 4 such that one side does not contain other vertices (Theorem 3.3).

2 Holes in convex drawings

In this section, we show that convex drawings behave similarly to geometric point sets when it comes to the existence of holes. We show that every sufficiently large convex drawing contains a 6-hole and hence a 5-hole and a 4-hole. This is tight, as the construction by Horton [21] gives arbitrarily large point sets, that is geometric drawings, without 7-holes.

► **Theorem 2.1** (Empty Hexagon theorem for convex drawings). *For every sufficiently large n , every convex drawing of K_n contains a 6-hole.*

For the proof we use the existence of k -gons in sufficiently large convex drawings [24, 28]. Our key lemma is that the subdrawing induced by a minimal k -gon together with its interior vertices is f -convex, a fact that had been known only for h -convex drawings [5, Lemma 4.7]. A k -gon is *minimal* if its convex side does not contain the convex side of another k -gon.

► **Lemma 2.2.** *Let \mathcal{C}_k be a minimal k -gon in a convex drawing \mathcal{D} of K_n with $n \geq k \geq 5$. Then the subdrawing \mathcal{D}' induced by the vertices in the convex side of \mathcal{C}_k is f -convex.*

Proof. Let v_1, \dots, v_k be the vertices of the minimal k -gon \mathcal{C}_k in \mathcal{D} and F be a face contained in the non-convex side of \mathcal{C}_k . We show that for every triangle spanned by three vertices of the convex side of \mathcal{C}_k , the side not containing F is convex and hence \mathcal{D}' is f -convex. Suppose towards a contradiction that there exists a triangle spanned by vertices t_1, t_2, t_3 from the convex side of \mathcal{C}_k , such that the side not containing F is not convex. The non-convex side S_N is the side contained in the convex side of \mathcal{C}_k . Since \mathcal{D} is convex, the other side containing F and all vertices v_1, \dots, v_k is convex and is denoted by S_C . If we additionally assume that S_N is not contained in (the closure of) a single cell of the subdrawing induced by \mathcal{C}_k , then some edge $\{v_i, v_j\}$ has a crossing with one of the edges $\{t_\ell, t_m\}$. This shows that S_C is not convex; a contradiction. Hence, S_N lies in (the closure of) a cell of \mathcal{C}_k .

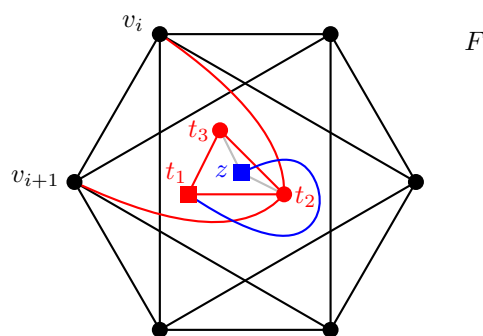
Since \mathcal{C}_k is minimal, there are no interior vertices in the convex side of a triangle $\{v_i, v_{i+1}, v_{i+2}\}$. For details see the full version [11].

Since all cells in the convex side of \mathcal{C}_k incident to the vertex v_{i+1} are inside this triangle, the vertex v_{i+1} is not part of the triangle spanned by t_1, t_2, t_3 . This holds for every $i = 1, \dots, k$ and hence the vertices t_1, t_2, t_3 are interior vertices of \mathcal{C}_k and S_N lies in a cell of the convex side of \mathcal{C}_k that is not covered by the convex side of any triangle $\{v_i, v_{i+1}, v_{i+2}\}$. Since S_N is not convex, there exists a vertex z in the interior of S_N such that the subdrawing induced by $\{t_1, t_2, t_3, z\}$ has a crossing [5, Corollary 2.5]. We assume without loss of generality that the edge $\{t_1, z\}$ crosses $\{t_2, t_3\}$. Moreover, exactly one of the following two conditions holds: Either the triangle $\{t_1, t_3, z\}$ separates t_2 and F or the triangle $\{t_1, t_2, z\}$ separates t_3 and F . We assume that the former holds as otherwise we exchange the roles of t_2 and t_3 . Figure 2 gives an illustration.

Now we consider all edges from t_2 to the vertices v_1, \dots, v_k of \mathcal{C}_k . Since S_C is convex and contains v_1, \dots, v_k , the edges $\{t_2, v_i\}$ are contained in S_C . This shows that none of the edges $\{t_2, v_i\}$ crosses any of the triangle edges and, in particular, they do not cross $\{t_1, t_3\}$.

The edges $\{t_2, v_1\}, \dots, \{t_2, v_k\}$ partition the convex side of \mathcal{C}_k into triangles t_2, v_i, v_{i+1} . Hence there is an index i such that the three vertices t_1, t_3, z lie in the convex side of the triangle $\{t_2, v_i, v_{i+1}\}$. However, the edge $\{t_1, z\}$ is not fully contained in this side; a contradiction to convexity. This completes the proof of Lemma 2.2. ◀

Recently, Heule and Scheucher [20] used SAT to show that every set of 30 points has a 6-hole. Since their result is about the more general case of pseudoconfigurations of points, it holds for pseudolinear drawings. To prove Theorem 2.1, we combine this fact with Lemma 2.2.



■ **Figure 2** Illustration of the proof of Lemma 2.2.

Proof of Theorem 2.1. Let \mathcal{D} be a convex drawing of K_n with $n > 2^{225 \log_2(5)} \cdot 30^2 \log_2(30)$. Since convex drawings do not contain the twisted drawing \mathcal{T}_5 , it follows from [28] that \mathcal{D} contains a 30-gon. To find a 6-hole in \mathcal{D} , we choose a minimal 30-gon G . By Lemma 2.2, the subdrawing \mathcal{D}' induced by G and its interior vertices is f-convex. Since the existence of holes is invariant under the choice of the outer cell, we can assume without loss of generality that \mathcal{D}' is pseudolinear as we may otherwise choose the face F as the unbounded face. According to [7], \mathcal{D}' corresponds to a pseudoconfiguration of points, and hence there exists a 6-hole H in \mathcal{D}' [20]. Hence the convex side of H does not contain any vertex of \mathcal{D}' . Moreover, every vertex of \mathcal{D} in the convex side of H would be an interior vertex of G and therefore belong to \mathcal{D}' . This shows that H is a 6-hole in \mathcal{D} . ◀

The existence of 6-holes further implies the existence of 4- and 5-holes. However, it remains a challenging task to determine the smallest integer $n(k)$ such that every convex drawing of K_n with $n \geq n(k)$ contains a k -hole for $k = 4, 5, 6$.

For 6-holes, one can slightly improve the estimate from Theorem 2.1 by utilizing the fact that every 9-gon in a point set yields a 6-hole [18]. As shown in [27] this result transfers to pseudolinear drawings. It follows from Lemma 2.2 that every convex drawing of K_n with $n > 2^{225 \log_2(5)} \cdot 9^2 \log_2(9)$ contains a 6-hole.

A similar improvement is possible for 5-holes: as the textbook proof for the existence of 5-holes in every 6-gon of a point set (see e.g. Section 3.2 in [22]) applies to pseudolinear drawings, every convex drawing with more than $2^{225 \log_2(5)} \cdot 6^2 \log_2(6)$ vertices contains a 5-hole.

For 4-holes, we can combine the proof of Bárány and Füredi [8, Theorem 3.3] for the quadratic number of empty 4-holes in point sets and the proof of Arroyo et al. [4, Theorem 5] for the quadratic number of empty triangles in convex drawings to obtain:

► **Lemma 2.3.** *Every crossed edge in a convex drawing of K_n is a chord of a 4-hole, that is, it is one of the crossing edges of a 4-hole.*

Since the number of uncrossed edges in drawings of K_n is at most $2n - 2$ [26], Lemma 2.3 implies that there are $\Omega(n^2)$ empty 4-holes in every convex drawing of K_n . A detailed proof is provided in the full version [11]. Since every drawing of K_5 contains a crossing, Lemma 2.3 also implies that every convex drawing of K_n with $n \geq 5$ contains a 4-hole. In contrast to the convex setting, 4-holes can be avoided in simple drawings as we show in the next section.

3 Generalized Holes

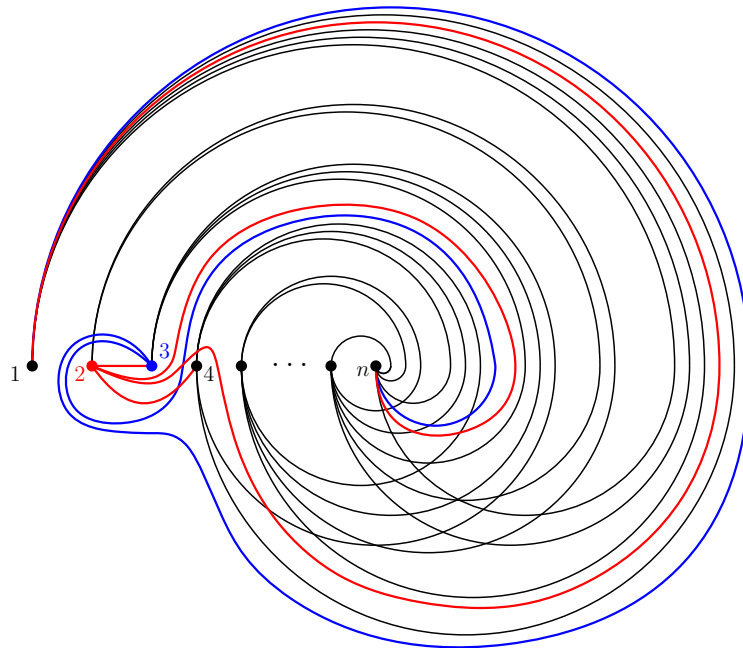
Devillers et al. [13] showed that sufficiently large two-colored point sets in general position contain a monochromatic 3-hole and constructed arbitrarily large two-colored sets without monochromatic 5-holes. The existence of monochromatic 4-holes, however, remains a

longstanding open problem [12, Problem 8.2.7]. A weaker version was shown by Aichholzer et al. [2]. They showed that every two-colored point set $P = A \dot{\cup} B$ contains a monochromatic generalized 4-hole. A *generalized k -hole* is a simple polygon (not necessarily convex) which is spanned by k points of P and does not contain any point of P in its interior. Apparently, their proof transfers to the pseudolinear setting, which allows us to generalize this result to convex drawings in the same way as the Empty Hexagon theorem 2.1 using Lemma 2.2.

► **Corollary 3.1.** *Every sufficiently large convex drawing on vertices $V = A \dot{\cup} B$ has an empty 4-triangulation induced only by vertices from A or only by vertices from B .*

To define generalized k -holes in simple drawings we consider plane cycles. A plane cycle divides the plane into two components whose closures we call *sides*. An *empty k -cycle* in a simple drawing is a plane cycle of length k such that one of its sides is empty. For $k = 3$ this definition coincides with empty triangles. Since polygons in point sets can be triangulated, we say that an empty k -cycle is an *empty k -triangulation* if its empty side is the disjoint union of empty triangles. As the following construction (Figure 3) shows, there are simple drawings of K_n without empty 4-triangulations. For the construction, we start with the twisted drawing \mathcal{T}_n and reroute some edges such that the drawing is still crossing maximal. The resulting drawing \mathcal{T}'_n does not contain 4-holes. A precise description and proof of Proposition 3.2 is given in the full version [11].

► **Proposition 3.2.** *For $n \geq 6$ the simple drawing \mathcal{T}'_n contains no empty 4-triangulation.*



■ **Figure 3** The drawing \mathcal{T}'_n without empty 4-triangulations for $n \geq 6$.

If instead of empty 4-triangulations we only ask for empty 4-cycles, then we can actually guarantee their existence in all simple drawings of K_n .

► **Theorem 3.3.** *Let \mathcal{D} be a simple drawing of K_n with $n \geq 4$ and let v be a vertex of \mathcal{D} . Then \mathcal{D} contains an empty 4-cycle passing through v .*

This resolves one case of a recent conjecture by Bergold et al. [9]. They showed that every convex drawing contains an empty k -cycle for all $3 \leq k \leq n$ and conjectured that this holds for simple drawings.

► **Conjecture 3.4** ([9]). *Every simple drawing of K_n contains an empty k -cycle for each $3 \leq k \leq n$.*

While the case $k = 3$ follows by Harborth’s result [19], the $k = n$ case coincides with Rafla’s conjecture concerning the existence of plane Hamiltonian cycles in all simple drawings of K_n [25]. For the proof of the case $k = 4$ of Conjecture 3.4 (Theorem 3.3), we use results on plane subdrawings by García, Pilz, and Tejel [16].

Proof of Theorem 3.3. For a fixed vertex v , we consider the spanning star S_v centered at v . By [16, Corollary 3.4], there is a plane subdrawing \mathcal{D}' of \mathcal{D} that consists of the star S_v and some spanning tree T on the other $n - 1$ vertices. Note that \mathcal{D}' has exactly $2n - 3$ edges and $n - 1$ faces. Every face F of \mathcal{D}' contains v on its boundary because the tree T is cycle-free and since \mathcal{D}' is 2-connected [16, Theorem 3.1], F is bounded by exactly two edges of S_v .

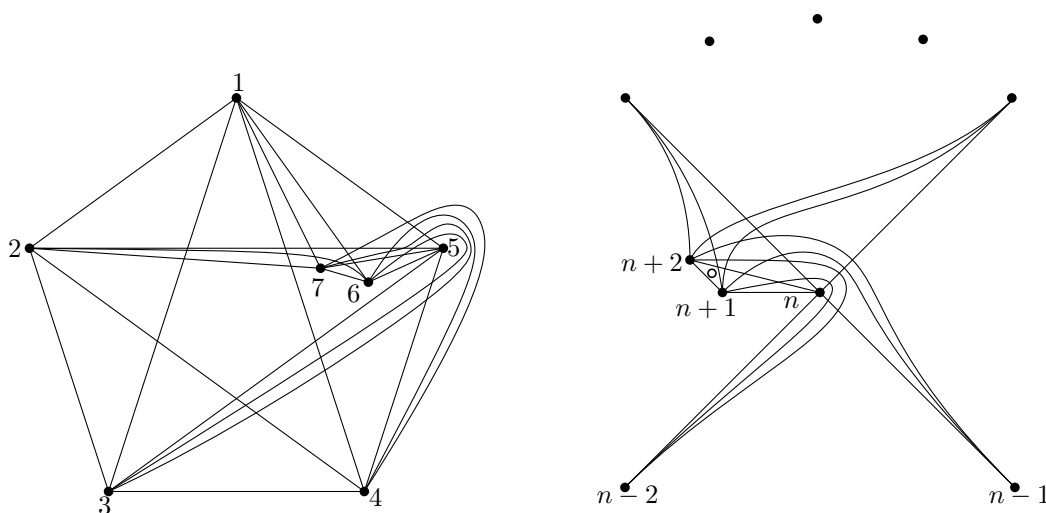
If there is a face of \mathcal{D}' with exactly 4 boundary edges or if there are two adjacent triangular faces, we obtain an empty 4-cycle passing through v and the statement follows. Otherwise we count the number of edges $|E|$ in \mathcal{D}' : At most half of the $n - 1$ faces are triangles so that none of them are adjacent. All other faces have at least 5 boundary edges. Since every edge is incident to exactly two faces, we have $|E| \geq \frac{1}{2} (5(n - 1) - 2 \lfloor \frac{n-1}{2} \rfloor) \geq 2n - 2$. This is a contradiction to the fact that \mathcal{D}' contains exactly $2n - 3$ edges. ◀

The above theorem implies a linear lower bound on the number of empty 4-cycles. This is similar to the minimum number of empty triangles which is asymptotically linear as well [3].

► **Corollary 3.5.** *Every simple drawing of K_n with $n \geq 4$ contains at least $\frac{n}{4}$ empty 4-cycles.*

While the twisted drawing \mathcal{T}_n is conjectured to minimize the number of empty triangles, it contains $\Theta(n^3)$ empty 4-cycles. This is certainly not minimal as there exist drawings with $\Theta(n^2)$ empty 4-cycles; see Figure 4 and the full version [11].

This seems to be in contrast to the geometric setting, where the number of empty k -cycles with $k \geq 4$ is conjectured to be super-quadratic [1].



■ **Figure 4** Constructing the drawing \mathcal{D}_n of K_n , n odd, with few empty 4-cycles from K_5 .

References


- 1 Oswin Aichholzer, Ruy Fabila-Monroy, Hernán González-Aguilar, Thomas Hackl, Marco A. Heredia, Clemens Huemer, Jorge Urrutia, Pavel Valtr, and Birgit Vogtenhuber. On k -gons and k -holes in point sets. *Computational Geometry*, 48(7):528–537, 2015. doi:10.1016/j.comgeo.2014.12.007.
- 2 Oswin Aichholzer, Thomas Hackl, Clemens Huemer, Ferran Hurtado, and Birgit Vogtenhuber. Large bichromatic point sets admit empty monochromatic 4-gons. *SIAM Journal on Discrete Mathematics*, 23(4), 2010. doi:10.1137/090767947.
- 3 Oswin Aichholzer, Thomas Hackl, Alexander Pilz, Pedro Ramos, Vera Sacristán, and Birgit Vogtenhuber. Empty triangles in good drawings of the complete graph. *Graphs and Combinatorics*, 31:335–345, 2015. doi:10.1007/s00373-015-1550-5.
- 4 Alan Arroyo, Dan McQuillan, R. Bruce Richter, and Gelasio Salazar. Levi’s Lemma, pseudo-linear drawings of K_n , and empty triangles. *Journal of Graph Theory*, 87(4):443–459, 2018. doi:10.1002/jgt.22167.
- 5 Alan Arroyo, Dan McQuillan, R. Bruce Richter, and Gelasio Salazar. Convex drawings of the complete graph: topology meets geometry. *Ars Mathematica Contemporanea*, 22(3), 2022. doi:10.26493/1855-3974.2134.ac9.
- 6 Alan Arroyo, R. Bruce Richter, and Matthew Sunohara. Extending drawings of complete graphs into arrangements of pseudocircles. *SIAM Journal on Discrete Mathematics*, 35(2):1050–1076, 2021. doi:10.1137/20M1313234.
- 7 Martin Balko, Radoslav Fulek, and Jan Kynčl. Crossing Numbers and Combinatorial Characterization of Monotone Drawings of K_n . *Discrete & Computational Geometry*, 53(1):107–143, 2015. doi:10.1007/s00454-014-9644-z.
- 8 Imre Bárány and Zoltán Füredi. Empty simplices in Euclidean space. *Canadian Mathematical Bulletin*, 30(4):436–445, 1987. doi:10.4153/cmb-1987-064-1.
- 9 Helena Bergold, Stefan Felsner, Meghana M. Reddy, Joachim Orthaber, and Manfred Scheucher. Plane Hamiltonian cycles in convex drawings. In *Proceedings of the 40th International Symposium on Computational Geometry (SoCG 2024)*, pages 18:1–18:16, 2024. doi:10.4230/LIPIcs.SocG.2024.18.
- 10 Helena Bergold, Stefan Felsner, Manfred Scheucher, Felix Schröder, and Raphael Steiner. Topological Drawings meet Classical Theorems from Convex Geometry. *Discrete & Computational Geometry*, 2022. doi:10.1007/s00454-022-00408-6.
- 11 Helena Bergold, Joachim Orthaber, Manfred Scheucher, and Felix Schröder. Holes in convex and simple drawings. *arXiv*, 2024. URL: <https://arxiv.org/abs/2409.01723>.
- 12 Peter Brass, William O. J. Moser, and János Pach. *Research Problems in Discrete Geometry*. Springer, 2005. doi:10.1007/0-387-29929-7.
- 13 Olivier Devillers, Ferran Hurtado, Gyula Károlyi, and Carlos Seara. Chromatic variants of the Erdős–Szekeres theorem on points in convex position. *Computational Geometry*, 26(3):193–208, 2003. doi:10/bb9h93.
- 14 Paul Erdős. Some problems on elementary geometry. *Australian Mathematical Society*, 2:2–3, 1978. URL: https://users.renyi.hu/~p_erdos/1978-44.pdf.
- 15 Paul Erdős and George Szekeres. A combinatorial problem in geometry. *Compositio Mathematica*, 2:463–470, 1935. URL: http://www.renyi.hu/~p_erdos/1935-01.pdf.
- 16 Alfredo García, Alexander Pilz, and Javier Tejel. On plane subgraphs of complete topological drawings. *Ars Mathematica Contemporanea*, 20(1):69–87, 2021. doi:10.26493/1855-3974.2226.e93.
- 17 Alfredo García, Javier Tejel, Birgit Vogtenhuber, and Alexandra Weinberger. Empty triangles in generalized twisted drawings of K_n . In *Graph Drawing and Network Visualization*, volume 13764 of *LNCS*, pages 40–48. Springer, 2022. doi:10.1007/978-3-031-22203-0_4.
- 18 Tobias Gerken. Empty Convex Hexagons in Planar Point Sets. *Discrete & Computational Geometry*, 39(1):239–272, 2008. doi:10.1007/s00454-007-9018-x.

- 19 Heiko Harborth. Empty triangles in drawings of the complete graph. *Discrete Mathematics*, 191(1–3):109–111, 1998. doi:10.1016/S0012-365X(98)00098-3.
- 20 Marijn J. H. Heule and Manfred Scheucher. Happy ending: An empty hexagon in every set of 30 points. In Bernd Finkbeiner and Laura Kovács, editors, *Proc. 30th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'24)*, volume 14570 of *LNCS*, pages 61–80. Springer, 2024. doi:10.1007/978-3-031-57246-3_5.
- 21 Joseph D. Horton. Sets with no empty convex 7-gons. *Canadian Mathematical Bulletin*, 26:482–484, 1983. doi:10.4153/CMB-1983-077-8.
- 22 Jiří Matoušek. *Lectures on Discrete Geometry*. Springer, 2002. doi:10.1007/978-1-4613-0039-7.
- 23 Carlos M. Nicolás. The Empty Hexagon Theorem. *Discrete & Computational Geometry*, 38(2):389–397, 2007. doi:10.1007/s00454-007-1343-6.
- 24 János Pach, József Solymosi, and Géza Tóth. Unavoidable configurations in complete topological graphs. *Discrete & Computational Geometry*, 30(2):311–320, 2003. doi:10.1007/s00454-003-0012-9.
- 25 Nabil H. Rafla. *The good drawings D_n of the complete graph K_n* . PhD thesis, McGill University, Montreal, 1988. URL: <https://escholarship.mcgill.ca/concern/theses/x346d4920>.
- 26 Gerhard Ringel. Extremal problems in the theory of graphs. In *Theory of Graphs and its Applications (Proc. Sympos. Smolenice, 1963)*, volume 8590. Publ. House Czechoslovak Acad. Sci Prague, 1964.
- 27 Manfred Scheucher. A SAT Attack on Erdős–Szekeres Numbers in \mathbb{R}^d and the Empty Hexagon Theorem. *Computing in Geometry and Topology*, 2(1):2:1–2:13, 2023. doi:10.57717/cgt.v2i1.12.
- 28 Andrew Suk and Ji Zeng. Unavoidable patterns in complete simple topological graphs. In *Graph Drawing and Network Visualization*, volume 13764 of *LNCS*, pages 3–15. Springer, 2022. doi:10.1007/978-3-031-22203-0_1.

1-Planar Unit Distance Graphs

Panna Gehér 

Eötvös Loránd University, Budapest, Hungary

Géza Tóth 

Alfréd Rényi Institute of Mathematics, Budapest, Hungary

Abstract

A matchstick graph is a plane graph with edges drawn as unit distance line segments. This class of graphs was introduced by Harborth who conjectured that a matchstick graph on n vertices can have at most $\lfloor 3n - \sqrt{12n - 3} \rfloor$ edges. Recently his conjecture was settled by Lavollée and Swanepoel. In this paper we consider 1-planar unit distance graphs. We say that a graph is a 1-planar unit distance graph if it can be drawn in the plane such that all edges are drawn as unit distance line segments while each of them are involved in at most one crossing. We show that such graphs on n vertices can have at most $3n - \sqrt[4]{n}/10$ edges.

2012 ACM Subject Classification Mathematics of computing → Discrete mathematics

Keywords and phrases unit distance graph, 1-planar, matchstick graph

Digital Object Identifier 10.4230/LIPIcs.GD.2024.6

Related Version *Previous Version:* <https://arxiv.org/abs/2310.00940>

Funding *Panna Gehér:* Supported by Lendület Programme of the Hungarian Academy of Sciences, LP2021-1/2021, ERC Advanced Grant “GeoScape” No. 882971 and by the National Research, Development and Innovation Office, NKFIH, K-131529.

Géza Tóth: Supported by ERC Advanced Grant “GeoScape” No. 882971 and by the National Research, Development and Innovation Office, NKFIH, K-131529.

1 Introduction

A graph is called a matchstick graph if it can be drawn in the plane with no crossings such that all edges are drawn as unit segments. This graph class was introduced by Harborth in 1981 [7, 9]. He conjectured that the maximum number of edges of a matchstick graph with n vertices is $\lfloor 3n - \sqrt{12n - 3} \rfloor$. He managed to prove it in a special case where the unit distance is also the smallest distance among the points [8]. Recently his conjecture was settled by Lavollée and Swanepoel [12].

Other interesting classes of graphs are the k -planar graphs. For any $k \geq 0$, a graph G is called k -planar if it can be drawn in the plane such that each edge is involved in at most k crossings. Let $e_k(n)$ denote the maximum number of edges of a k -planar graph on n vertices. Since 0-planar graphs are the well known planar graphs, $e_0(n) = 3n - 6$ for $n \geq 3$. We have $e_1(n) = 4n - 8$ for $n \geq 4$ [19], $e_2(n) \leq 5n - 10$, which is tight for infinitely many n [19], $e_3(n) \leq 5.5n - 11$, which is tight up to an additive constant [17] and $e_4(n) \leq 6n - 12$, which is also tight up to an additive constant [1]. For general k we have $e_k(n) \leq c\sqrt{kn}$ for some constant c , which is tight apart from the value of c [19, 1].

A k -planar unit distance graph is a graph that can be drawn in the plane such that each edge is a unit segment and involved in at most k crossings. Let $u_k(n)$ be the maximum number of edges of a k -planar unit distance graph. Since 0-planar unit distance graphs are exactly the matchstick graphs, by the result of Lavollée and Swanepoel we have $u_0(n) = \lfloor 3n - \sqrt{12n - 3} \rfloor$. We do not have any better lower bound for $u_1(n)$ than the value of $u_0(n)$. That is, allowing to use one crossing on each edge does not seem to help, still a proper piece of the triangular grid is the best known construction. Somewhat surprisingly, we prove an almost matching upper bound.



© Panna Gehér and Géza Tóth;

licensed under Creative Commons License CC-BY 4.0

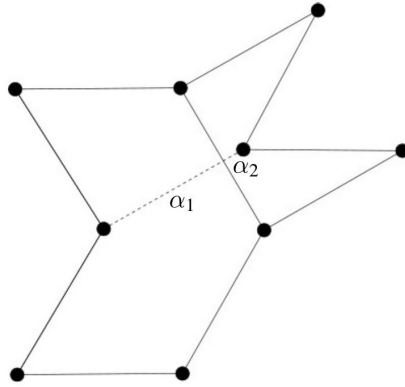
32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 6; pp. 6:1–6:9

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** An edge in E_1 can be partitioned into two halfedges, α_1 and α_2 .

► **Theorem 1.** *For the maximum number of edges of a 1-planar unit distance graph, $u_1(n)$, we have $\lfloor 3n - \sqrt{12n - 3} \rfloor \leq u_1(n) \leq 3n - \sqrt[4]{n}/10$.*

For general k , the best known lower bound is due to Günter Rote (personal communication, 2023).

► **Theorem 2 (Rote).** *For the maximum number of edges of a k -planar unit distance graph, $u_k(n)$, we have $u_k(n) \geq 2^{\Omega(\log k / \log \log k)} n$.*

We have the following upper bound.

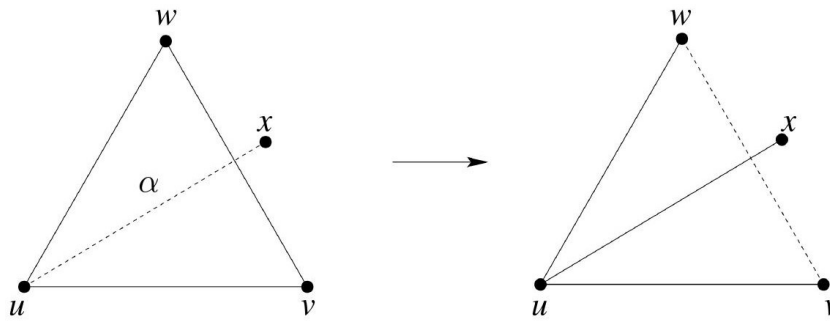
► **Theorem 3.** *For any $n, k \geq 0$ we have $u_k(n) \leq c\sqrt[4]{kn}$ for some constant $c > 0$.*

2 1-planar unit distance graphs

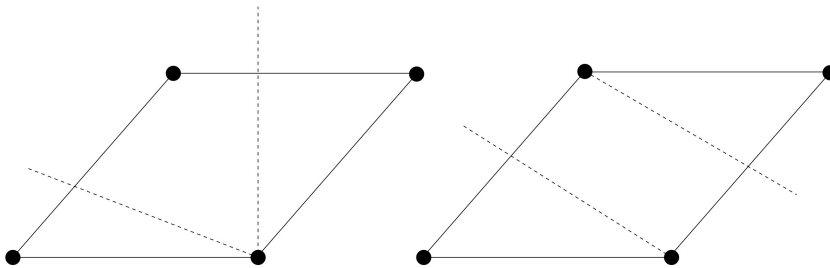
Proof of Theorem 1. The lower bound follows directly from Harborth’s construction for matchstick graphs [8]. We prove the upper bound. Let G be a 1-planar unit distance graph with n vertices and consider a 1-plane unit distance drawing of G . Let E be the set of edges, $|E| = e$. Let G_0 be a plane subgraph of G with maximum number of edges, and among those one with the minimum number of triangular faces. Let $E_0 \subset E$ denote the set of edges of G_0 and $E_1 = E \setminus E_0$ denote the set of remaining edges, $|E_0| = e_0$, $|E_1| = e_1$. Let f be the number of faces of G_0 , including the unbounded face and let $\Phi_1, \Phi_2 \dots \Phi_f$ be the faces of G_0 . For any face Φ_i , $|\Phi_i|$ is the number of bounding edges of it, counted with multiplicity. That is, if an edge bounds Φ_i from both sides, then it is counted twice. Due to the maximality of G_0 and 1-planarity of G , every edge $\alpha \in E_1$ crosses an edge in E_0 and connects two vertices that belong to neighbouring faces of G_0 . Therefore, we can partition every edge $\alpha \in E_1$ into two *halfedges*, α_1 and α_2 at the unique crossing point on α . See Figure 1. Each halfedge is contained in a face Φ , one of its endpoints is a vertex of Φ and the other endpoint is an interior point of a bounding edge.

▷ **Claim 4.** A triangular face of G_0 does not contain any halfedge.

Proof. Let $\Phi = uvw$ be a triangular face of G_0 that contains a halfedge α_1 , which is part of the edge $\alpha = ux$. Then α crosses the edge vw . Replace the edge vw by α in G_0 . See Figure 2. Since vw is the only edge of G that crosses α , we obtain another plane subgraph of G . It has the same number of edges.



■ **Figure 2** The number of triangles in G_0 can be reduced by edge flips.



■ **Figure 3** A quadrilateral can have at most two halfedges.

We claim that it has fewer triangular faces. The triangular face Φ disappeared. Suppose that we have created a new triangular face. Then α should be a side of it. But then either uv or uw is also a side, suppose without loss of generality that it is uv . But then uvx is also a unit equilateral triangle. If the two equilateral triangles uvw and uvx are on the same side of uv then $x = w$, if they are on opposite sides then vw and ux can not cross. \triangleleft

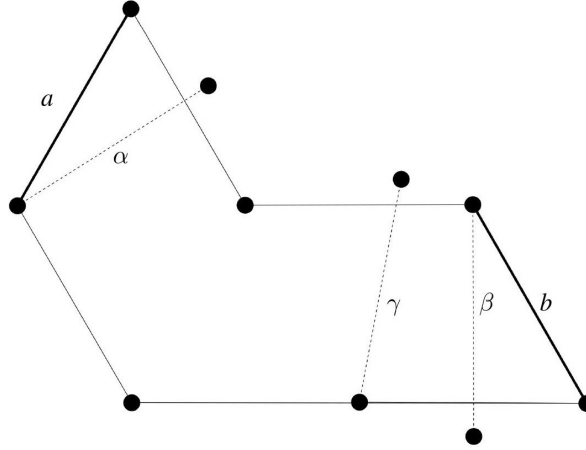
Assign weight $1/2$ to each halfedge. For any face Φ_i , let $s(\Phi_i)$ be the sum of the weights of its halfedges. Clearly, we have $\sum_{i=1}^f s(\Phi_i) = |E_1|$. For any face Φ of G_0 , let $t(\Phi)$ denote the number of edges in a triangulation of Φ . A straightforward consequence of Euler's formula is the following statement. If the boundary of Φ_i has m connected components, then $t(\Phi_i) = |\Phi_i| + 3m - 6$.

\triangleright **Claim 5.** For any face Φ of G_0 we have (a) $s(\Phi) \leq t(\Phi)$, and if $|\Phi| \geq 5$ then (b) $s(\Phi) \leq t(\Phi) - |\Phi|/10$.

Proof. Suppose first that the boundary of Φ is not connected, that is, $m \geq 2$. Each of the $|\Phi_i|$ edges on the boundary of Φ_i is crossed by at most one halfedge, therefore, $s(\Phi) \leq |\Phi|/2$. On the other hand, $t(\Phi) \geq |\Phi|$. Therefore, $t(\Phi) \geq |\Phi| \geq |\Phi|/2 + |\Phi|/10 \geq s(\Phi) + |\Phi|/10$ and we are done in this case.

Suppose now that the boundary of Φ_i is connected, that is, $m = 1$. If $|\Phi| = 3$, then Φ is a triangle. Then $t(\Phi) = 0$ and by Claim 4, $s(\Phi) = 0$. If $|\Phi| = 4$, then Φ is a quadrilateral (actually, a rhombus). Then $t(\Phi) = 1$. Figure 3 shows all possible cases when Φ has two halfedges. On the other hand, it is shown in [19] by an easy case analysis that no more halfedges can be added. Therefore, $s(\Phi) \leq 1 = t(\Phi)$. This finishes part (a).

Suppose that $|\Phi_i| \geq 5$. We can assume that Φ_i has at least two halfedges, otherwise we are done. A halfedge α in Φ divides Φ into two parts. Let $a(\alpha)$ and $b(\alpha)$ be the number of vertices of Φ in the two parts. If a vertex appears on the boundary more than once, then



■ **Figure 4** Halfedges α and β are minimal, edges a and b are uncrossed.

it is counted with multiplicity. Since the halfedges in Φ do not cross each other, all other halfedges are entirely in one of these two parts. If one part does not contain any halfedge, then α is called a *minimal halfedge*. Let α be a halfedge for which $M = \min\{a(\alpha), b(\alpha)\}$ is minimal. Then there are M vertices of Φ_i on one side of α . Clearly, this part cannot contain any halfedge, so α is minimal. Now for any other halfedge $\beta \neq \alpha$, let $c(\beta)$ be the number of vertices of Φ_i on the side of β not containing α . Take a halfedge β for which $c(\beta)$ is minimal. Then β is also a minimal halfedge. So, we can conclude that there are at least two minimal halfedges in Φ , say, α and β .

Then α and β together partition Φ into three parts, two parts contain no other halfedges but both contain an edge of Φ . So, at most $|\Phi| - 2$ edges of Φ are crossed by a halfedge, therefore, there are at most $|\Phi| - 2$ halfedges in Φ , consequently $s(\Phi) \leq (|\Phi| - 2)/2$. See Figure 4.

On the other hand, $t(\Phi) = |\Phi| - 3$. Since $|\Phi| \geq 5$, we have $t(\Phi) = |\Phi| - 3 \geq (|\Phi| - 2)/2 + |\Phi|/10 \geq s(\Phi) + |\Phi|/10$. This concludes the proof of the Claim. \triangleleft

Return to the proof of Theorem 1. For $i \geq 3$, let f_i denote the number of faces Φ of G_0 with $|\Phi| = i$. By definition, $\sum_{i=3}^{\infty} f_i = f$ and $\sum_{i=3}^{\infty} i f_i = 2e_0$. Let $F_{\geq 5} = \sum_{i=5}^{\infty} i f_i$. By the maximality of G_0 , every edge in E_1 crosses an edge in E_0 , and by 1-planarity, every edge in E_0 is crossed by at most one edge in E_1 . Consequently, $|E_0| = e_0 \geq |E_1| = e_1$.

If $e_0 \leq n$, then $e = e_0 + e_1 \leq 2e_0 \leq 2n < 3n - c\sqrt[3]{n}$, so we are done. Therefore, for the rest of the proof we can assume that $e_0 \geq n$. It follows that $3f_3 + 4f_4 + F_{\geq 5} = 2e_0 \geq 2n$.

▷ **Claim 6.** Suppose that $F_{\geq 5} \geq p$. Then $e = e_0 + e_1 \leq 3n - p/10$.

Proof. By the previous observations,

$$\begin{aligned}
 e = e_0 + e_1 &= e_0 + \sum_{\substack{\alpha \text{ is a} \\ \text{halfedge}}} 1/2 = e_0 + \sum_{i=1}^f s(\Phi_i) = e_0 + \sum_{|\Phi|=3} s(\Phi) + \sum_{|\Phi|=4} s(\Phi) + \sum_{|\Phi|\geq 5} s(\Phi) \\
 &\leq e_0 + \sum_{|\Phi|=3} t(\Phi) + \sum_{|\Phi|=4} t(\Phi) + \sum_{|\Phi|\geq 5} (t(\Phi) - |\Phi|/10) \\
 &\leq e_0 + \sum_{|\Phi|} t(\Phi) - \sum_{|\Phi|\geq 5} |\Phi|/10 \leq 3n - 6 - F_{\geq 5}/10 \leq 3n - p/10. \quad \triangleleft
 \end{aligned}$$

▷ Claim 7. Suppose that $f_3 \geq p$. Then $e = e_0 + e_1 \leq 3n - \sqrt{p}/5$.

Proof. We can assume that Ψ , the unbounded face of G_0 has at least 5 edges. If not, the statement holds trivially. Since we have p equilateral triangles in G_0 , the union of all bounded faces, R , has area at least $\sqrt{3}p/4$. The Isoperimetric inequality states that if a polygon has perimeter l and area A , then $l^2 \geq 4\pi A$ [5]. It implies that R has perimeter at least $\sqrt[4]{3}\sqrt{\pi p} > 2\sqrt{p}$. That is $|\Psi| \geq 2\sqrt{p}$. Therefore,

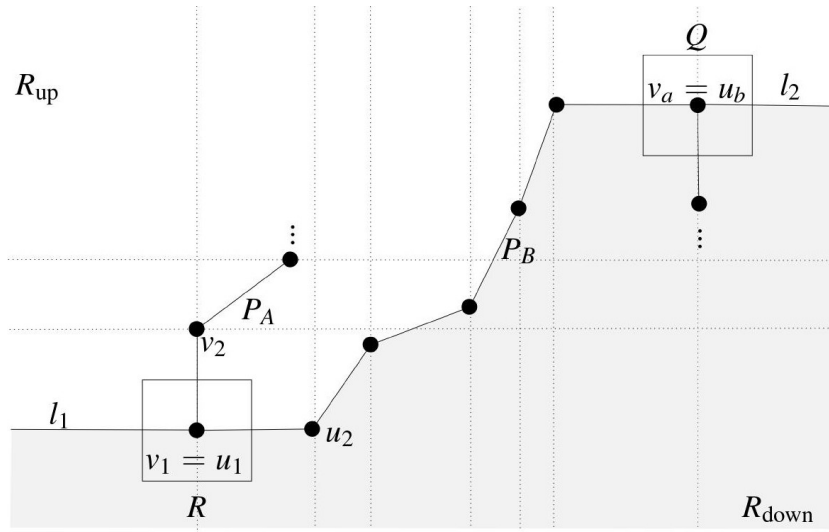
$$\begin{aligned} e = e_0 + e_1 &= e_0 + \sum_{\substack{\alpha \text{ is a} \\ \text{halfedge}}} 1/2 = e_0 + \sum_{i=1}^f s(\Phi_i) = e_0 + \sum_{\Phi \neq \Psi} s(\Phi) + s(\Psi) \\ &\leq e_0 + \sum_{\Phi \neq \Psi} t(\Phi) + t(\Psi) - |\Psi|/10 = 3n - 6 - |\Psi|/10 \leq 3n - 6 - \sqrt{p}/5. \quad \triangleleft \end{aligned}$$

We can assume that $n \geq 5$, otherwise Theorem 1 holds trivially. If $F_{\geq 5} \geq n/2$, then by Claim 6, $e \leq 3n - n/20 \leq 3n - \sqrt[4]{n}/10$ and we are done. If $f_3 \geq n/9$, then by Claim 7, $e \leq 3n - \sqrt{n}/15 \leq 3n - \sqrt[4]{n}/10$ and we are done again. So, we can assume that $F_{\geq 5} \leq n/2$, $f_3 \leq n/9$. Since $3f_3 + 4f_4 + F_{\geq 5} = 2e_0 \geq 2n$, it follows that $f_4 \geq n/4$.

Suppose without loss of generality that none of the edges of G are vertical. Otherwise apply a rotation. Define an auxiliary graph H as follows. The vertices represent the quadrilateral faces of G_0 . Since all edges are of unit length, all these faces are rhombuses. Two vertices are connected by an edge if the corresponding rhombuses have a common edge. The edges of H correspond to the edges of G_0 with a rhombus face on both sides. For every edge of H define its weight as the slope of the corresponding edge of G_0 . A path in H , such that all of its edges have the same weight w , is called a w -chain, or briefly a *chain*. A chain corresponds to a sequence of rhombuses such that the consecutive pairs share a side and all these sides are parallel. A chain, with at least two vertices (rhombuses) is called *maximal* if it cannot be extended. With one-vertex chains we have to be careful. Suppose that v is a vertex of H , R is the corresponding rhombus, and let w_1, w_2 be the slopes of its sides. The one-vertex chain v is *maximal* if it cannot be extended to a larger w_1 -chain or a larger w_2 -chain. Each vertex of H is in exactly two maximal chains.

▷ Claim 8. The intersection of two chains is empty or forms a chain.

Proof. If the intersection is just one vertex then the statement clearly holds. Suppose that A and B are chains with at least two common vertices, and their intersection is not a chain. Let $A = v_1, v_2, \dots, v_a$. We can assume without loss of generality that $v_1, v_a \in B$ but no other vertex of A is in B . Otherwise we can delete some vertices of A to obtain this situation. Delete all vertices of B which are not between v_1 and v_a . Now $B = u_1, u_2, \dots, u_b$ where $v_1 = u_1, v_a = u_b$ and these are the only common points of A and B . Let R be the rhombus that represents $v_1 = u_1$ in G_0 . Its sides have slopes w_1 and w_2 such that A is a w_1 -chain, B is a w_2 -chain. Apply an affine transformation so that R is a unit square, w_1 is the horizontal, w_2 is the vertical direction. Suppose that Q is the rhombus that represents $v_a = u_b$. Then its sides also have slopes w_1 and w_2 , so Q is also an axis-parallel unit square. Represent each vertex $v_1, v_2, \dots, v_a, u_1, u_2, \dots, u_b$ by the center of the corresponding rhombus. For simplicity we call these points also $v_1, v_2, \dots, v_a, u_1, u_2, \dots, u_b$, respectively. Assume without loss of generality that the point $v_a = u_b$ has larger x and y coordinates, than $v_1 = u_1$. Connect the consecutive points in both chains by straight line segments. Since A is a w_1 -chain and w_1 is the horizontal direction, the polygonal chain $P_A = v_1, v_2, \dots, v_a$ is y -monotone, and similarly,



■ **Figure 5** The intersection of two chains is empty or forms a chain.

the polygonal chain $P_B = u_1, v_2, \dots, v_b$ is x -monotone. Let l_1 be the horizontal halfline from $v_1 = u_1$, pointing to the left and let l_2 be the horizontal halfline from $v_a = u_b$, pointing to the right. The bi-infinite curve $l_1 \cup P_B \cup l_2$ is simple, because P_B is x -monotone. It divides the plane into two regions, R_{down} , which is below it and its complement, R_{up} , see Figure 5.

Observe that the initial part of P_A , near $v_1 = u_1$ is in R_{up} , while the final part, near $v_a = u_b$ is in R_{down} . On the other hand, P_A does not intersect the boundary of R_{down} and R_{up} . Indeed, it does not intersect l_1 and l_2 since it is y -monotone, and does not intersect P_B by assumption. This is clearly a contradiction which proves the Claim. \triangleleft

▷ **Claim 9.** There are at least $\sqrt{n}/\sqrt{2}$ disjoint maximal chains.

Proof. For any vertex of H (that is, for any rhombus face in G_0) there are exactly two maximal chains containing it. Therefore, the total length of all the maximal chains is $2f_4 \geq n/2$. If there are less than $\sqrt{n}/\sqrt{2}$ disjoint maximal chains, then one of them has length at least $\sqrt{n}/\sqrt{2}$. Through each of its vertices, there is another maximal chain and by Claim 8 all of these chains are different. \triangleleft

By Claim 9, we have at least $\sqrt{n}/\sqrt{2}$ disjoint maximal chains. Each of them has two ending rhombuses with sides that bound a face of size different than 4. All of these bounding edges are different, therefore, $3f_3 + F_{\geq 5} \geq \sqrt{2}\sqrt{n}$, which implies that either $3f_3 \geq \sqrt{n}/\sqrt{2}$, or $F_{\geq 5} \geq \sqrt{n}/\sqrt{2}$.

In the first case, by Claim 7 we have $e \leq 3n - \sqrt[4]{n}/10$. In the second case, by Claim 6 we have $e \leq 3n - \sqrt{n}/10$. This concludes the proof of Theorem 1. \blacktriangleleft

3 k -planar unit distance graphs

Proof of Theorem 2. Suppose that $n, k > 100$. The following is a well known result in number theory (see [14], [16]). For any m , there is an $r < m$ such that r can be written as $a^2 + b^2$ in $2^{\Omega(\log m / \log \log m)}$ different ways where a and b are integers. For any fixed m let r be the product of the first l primes congruent to 1 mod 4, such that l is maximal with the property that $r < m$. This r satisfies the requirements.

Erdős [6] used it to construct a set of n points that determine $n2^{\Omega(\log n / \log \log n)}$ unit distances. Clearly, r is square-free, therefore, whenever $r = a^2 + b^2$, $(a, b) = 1$.

Apply the above result for $m = \sqrt{k}/5$. We obtain $r < \sqrt{k}/5$ that can be written as the sum of two integer squares, $r = a^2 + b^2$ in $2^{\Omega(\log m / \log \log m)} = 2^{\Omega(\log k / \log \log k)}$ different ways. Take a $\sqrt{n} \times \sqrt{n}$ unit square grid and connect two points by a straight line segment if they are at distance \sqrt{r} . Then each vertex has degree $2^{\Omega(\log k / \log \log k)}$, so our graph has $n2^{\Omega(\log k / \log \log k)}$ edges. Observe that no edge contains a vertex in its interior.

Let uv be an edge. Consider all vertices adjacent to an edge that crosses uv . All these vertices are at distance at most \sqrt{r} from uv . This region has area $(2 + \pi)r$, so the number of vertices in this region is less than $6r$. Each of these vertices have degree at most $4r$, so uv is crossed by at most $24r^2 < k$ edges. Scale the picture by a factor of $1/\sqrt{r}$ and we obtain a k -planar unit distance graph of n vertices and $2^{\Omega(\log k / \log \log k)}$ edges. ◀

For the proof of Theorem 3 we need some introduction. Let $\text{CR}(G)$ denote the *crossing number* of graph G , that is, the minimum number of edge crossing over all drawings of G in the plane. According to the Crossing Lemma [3, 13], for every graph G with n vertices and $e \geq 4n$ edges, $\text{CR}(G) \geq \frac{1}{64} \frac{e^3}{n^2}$. It is asymptotically tight in general for simple graphs [19]. However, there are better bounds for graphs satisfying some monotone property [15], or for monotone drawing styles [10].

A drawing style \mathcal{D} is a subset of all drawings of a graph G . so some drawings belong to \mathcal{D} , others do not. It is monotone if removing edges retains the drawing style. A vertex split is the following operation. (a) Replace a vertex v of G by two vertices, v_1 and v_2 , both very close to v . Connect each edge of G incident to v either to v_1 or v_2 by locally modifying them such that no additional crossing is created. Or as an extreme or limiting case, (b) place both v_1 and v_2 to the same point where v was, connect each edge incident to v either to v_1 or v_2 without modifying them, such that the edges incident to v in G that are connected to v_1 (resp. v_2) after the split form an interval in the clockwise order from v . A drawing style \mathcal{D} is split-compatible if performing vertex splits retains the drawing style.

The bisection width $b(G)$ of a graph G is the smallest number of edges whose removal splits G into two graphs, G_1 and G_2 , such that $|V(G_1)|, |V(G_2)| \geq |V(G)|/5$. For a drawing style \mathcal{D} the \mathcal{D} -bisection width $b_{\mathcal{D}}(G)$ of a graph G in drawing style \mathcal{D} is the smallest number of edges whose removal splits G into two graphs, G_1 and G_2 , both in drawing style \mathcal{D} such that $|V(G_1)|, |V(G_2)| \geq |V(G)|/5$. Let $\Delta(G)$ denote the maximum degree in G . The following result is a generalization of the Crossing Lemma.

► **Theorem 10** (Kaufmann-Pach-Tóth-Ueckerdt [10]). *Suppose that \mathcal{D} is a monotone and split-compatible drawing style, and there are constants $k_1, k_2, k_3 > 0$ and $b > 1$ such that each of the following holds for every n -vertex e -edge graph G in drawing style \mathcal{D} :*

1. *If $\text{CR}_{\mathcal{D}}(G) = 0$, then $e \leq k_1 \cdot n$.*
2. *The \mathcal{D} -bisection width satisfies $b_{\mathcal{D}}(G) \leq k_2 \sqrt{\text{CR}_{\mathcal{D}}(G) + \Delta(G)} \cdot e + n$.*
3. *$e \leq k_3 \cdot n^b$.*

Then there exists a constant $\alpha > 0$ such that for any n -vertex e -edge graph G in drawing style \mathcal{D} we have $\text{CR}_{\mathcal{D}}(G) \geq \alpha \frac{e^{1/(b-1)+2}}{n^{1/(b-1)+1}}$ provided $e > (k_1 + 1)n$.

In [10] only vertex split of type (a) was allowed, but the proof works also for type (b).

► **Theorem 11** (Spencer-Szemerédi-Trotter [20]). *Let G be a unit distance graph on n vertices. The number of edges in G is at most $cn^{4/3}$ where $c > 0$ is a constant.*

Proof of Theorem 3. Consider now the following drawing style \mathcal{D} for a graph G .

1. Vertices are represented by not necessarily distinct points.
2. Edges are represented by unit segments between the corresponding points.
3. The intersection of two edges is empty or a point, that is, they cannot overlap.
4. If a point p represents more than one vertex, say, v_1, \dots, v_m , then the sets of edges incident to v_1, \dots, v_m , respectively, form an interval in the clockwise order from point p .

Clearly, \mathcal{D} satisfies the following properties.

1. The drawing style \mathcal{D} is monotone and split-compatible.
2. If $\text{CR}(G) = 0$, then $e \leq 3n - 6$. In fact, by [12], $e \leq \lfloor 3n - \sqrt{12n - 3} \rfloor$.
3. For any graph G , we have $b(G) \leq 10\sqrt{\text{CR}(G) + \Delta(G)} \cdot e + n$ by the result of Pach, Shahrokhi and Szegedy [18]. But if G is drawn in drawing style \mathcal{D} , then all of its subgraphs are also drawn in drawing style \mathcal{D} . Therefore, $b_{\mathcal{D}}(G) \leq 10\sqrt{\text{CR}(G) + \Delta(G)} \cdot e + n$.
4. By [2], any n -vertex graph in drawing style \mathcal{D} has less than $1.94n^{4/3}$ edges.

Summarizing, we can apply Theorem 10 with $k_1 = 3$, $k_2 = 10$, $k_3 = 1.94$, $b = 4/3$ and obtain the following. For any graph G in drawing style \mathcal{D} with n vertices and $e > 4n$ edges we have $\text{CR}_{\mathcal{D}}(G) \geq \alpha \frac{e^{1/(b-1)+2}}{n^{1/(b-1)+1}} = \alpha \frac{e^5}{n^4}$ for some $\alpha > 0$.

Consider now a k -plane drawing of a unit distance graph G with n vertices and e edges. If $e \leq 4n$, we are done, suppose that $e \geq 4n$. Since each edge contains at most k crossings, the total number of crossings $c(G)$ satisfies $c(G) \leq ek/2$. On the other hand, we have $c(G) \geq \alpha \frac{e^5}{n^4}$. Therefore, $ek/2 \geq \alpha \frac{e^5}{n^4}$ so $e \leq \beta \sqrt[4]{kn}$ for some $\beta > 0$. ◀

4 Open questions

In this paper we proved that a 1-planar unit distance graph on n vertices can have at most $u_1(n) \leq 3n - \sqrt[3]{n}/10$ edges. However, the best known lower bound construction for $u_1(n)$ is the same as for $u_0(n)$.

► **Problem 12.** *Is it true that $u_0(n) = u_1(n)$?*

For $k = 2$ there is a slightly better construction by Dániel Simon (personal communication, 2023) of roughly $3n - \sqrt{\frac{192}{23}n}$ edges and for $k = 3$ there is an easy construction (a piece of a unit triangular grid and its shifted copy by a unit vector) with $3.5n - c\sqrt{n}$ edges.

For a larger k our lower and upper bounds for $u_k(n)$ are very far from each other.

► **Problem 13.** *Determine the maximum number of edges of a k -planar unit distance graph.*

There are r -regular matchstick graphs for $r \leq 4$ [9, 21] and there are no r -regular matchstick graphs for $r \geq 5$ [4, 11]. It follows from Theorem 1 that there are no r -regular 1-planar unit distance graphs for $r \geq 6$.

► **Problem 14.** *Are there 5-regular 1-planar unit distance graphs?*


References

- 1 Eyal Ackerman. On topological graphs with at most four crossings per edge. *Computational Geometry*, 85:101574, 2019. doi:10.1016/J.COMGEO.2019.101574.
- 2 Péter Ágoston and Dömötör Pálvölgyi. An improved constant factor for the unit distance problem. *Studia Scientiarum Mathematicarum Hungarica*, 59(1):40–57, 2022.
- 3 Miklós Ajtai, Vašek Chvátal, Monroe M Newborn, and Endre Szemerédi. Crossing-free subgraphs. In *North-Holland Mathematics Studies*, volume 60, pages 9–12. Elsevier, 1982.

- 4 Aart Blokhuis. Regular finite planar maps with equal edges. *arXiv preprint*, 2014. [arXiv:1401.1799](#).
- 5 Manfredo P Do Carmo. *Differentialgeometrie von Kurven und Flächen*, volume 55. Springer-Verlag, 2013.
- 6 Paul Erdős. On sets of distances of n points. *The American Mathematical Monthly*, 53(5):248–250, 1946.
- 7 H Harborth. Point sets with equal numbers of unit-distant neighbors. *Discrete Geometry*, pages 12–18, 1981.
- 8 Heiko Harborth. Solution to problem 664a. *Elem. Math*, 29:14–15, 1974.
- 9 Heiko Harborth. Match sticks in the plane. In *The lighter side of mathematics. Proceedings of the Eugne Strens memorial conference on recreational mathematics and its history*, pages 281–288, 1994.
- 10 Michael Kaufmann, János Pach, Géza Tóth, and Torsten Ueckerdt. The number of crossings in multigraphs with no empty lens. *Journal of Graph Algorithms and Applications*, 25(1):383–396, 2021. [doi:10.7155/JGAA.00563](#).
- 11 Sascha Kurz and Rom Pinchasi. Regular matchstick graphs. *The American Mathematical Monthly*, 118(3):264–267, 2011. [doi:10.4169/AMER.MATH.MONTHLY.118.03.264](#).
- 12 Jérémy Lavollée and Konrad Swanepoel. A tight bound for the number of edges of matchstick graphs. *Discrete & Computational Geometry*, pages 1–15, 2023.
- 13 Frank Thomson Leighton. New lower bound techniques for vlsi. *Mathematical systems theory*, 17:47–70, 1984. [doi:10.1007/BF01744433](#).
- 14 Jiří Matoušek. *Lectures on discrete geometry*, volume 212. Springer Science & Business Media, 2013.
- 15 J Pach, J Spencer, and G Tóth. New bounds on crossing numbers. *Discrete & Computational Geometry*, 24:623–644, 2000. [doi:10.1007/S004540010011](#).
- 16 János Pach and Pankaj K Agarwal. *Combinatorial geometry*. John Wiley & Sons, 2011.
- 17 János Pach, Radoš Radoičić, Gábor Tardos, and Géza Tóth. Improving the crossing lemma by finding more crossings in sparse graphs. *Discrete & Computational Geometry*, 36(4):527–552, 2006. [doi:10.1007/S00454-006-1264-9](#).
- 18 János Pach, Farhad Shahrokhi, and Mario Szegedy. Applications of the crossing number. In *Proceedings of the tenth annual symposium on Computational geometry*, pages 198–202, 1994. [doi:10.1145/177424.177629](#).
- 19 János Pach and Géza Tóth. Graphs drawn with few crossings per edge. *Combinatorica*, 17(3):427–439, 1997. [doi:10.1007/BF01215922](#).
- 20 Joel Spencer, Endre Szemerédi, and William T Trotter. Unit distances in the euclidean plane. In *Graph theory and combinatorics*, pages 294–304. Academic Press, 1984.
- 21 Mike Winkler, Peter Dinkelacker, and Stefan Vogel. On the existence of 4-regular matchstick graphs. *arXiv preprint*, 2017. [arXiv:1705.00293](#).

The Density Formula: One Lemma to Bound Them All

Michael Kaufmann ✉ 
University of Tübingen, Germany

Boris Klemz ✉ 
Universität Würzburg, Germany

Kristin Knorr ✉ 
Freie Universität Berlin, Germany

Meghana M. Reddy ✉ 
ETH Zürich, Switzerland

Felix Schröder ✉ 
Technische Universität Berlin, Germany
Charles University, Prague, Czech Republic

Torsten Ueckerdt ✉ 
Karlsruhe Institute of Technology, Germany

Abstract

We introduce the Density Formula for (topological) drawings of graphs in the plane or on the sphere, which relates the number of edges, vertices, crossings, and sizes of cells in the drawing. We demonstrate its capability by providing several applications: we prove tight upper bounds on the edge density of various beyond-planar graph classes, including so-called k -planar graphs with $k = 1, 2$, fan-crossing / fan-planar graphs, k -bend RAC-graphs with $k = 0, 1, 2$, quasiplanar graphs, and k^+ -real face graphs. In some cases (1-bend and 2-bend RAC-graphs and fan-crossing / fan-planar graphs), we thereby obtain the first tight upper bounds on the edge density of the respective graph classes. In other cases, we give new streamlined and significantly shorter proofs for bounds that were already known in the literature. Thanks to the Density Formula, all of our proofs are mostly elementary counting and mostly circumvent the typical intricate case analysis found in earlier proofs. Further, in some cases (simple and non-homotopic quasiplanar graphs), our alternative proofs using the Density Formula lead to the first tight lower bound examples.

2012 ACM Subject Classification Mathematics of computing → Graph theory; Mathematics of computing → Graphs and surfaces

Keywords and phrases beyond-planar, density, fan-planar, fan-crossing, right-angle crossing, quasiplanar

Digital Object Identifier 10.4230/LIPIcs.GD.2024.7

Related Version *Full Version:* <https://arxiv.org/abs/2311.06193> [15]

Funding *Felix Schröder:* supported by the Czech Science Foundation grant GAČR 23-04949X.
Torsten Ueckerdt: supported by the Deutsche Forschungsgemeinschaft – 520723789.

1 Introduction

Topological Graph Theory is concerned with the analysis of graphs drawn in the plane \mathbb{R}^2 or the sphere \mathbb{S}^2 such that the drawing has a certain property often related to forbidden crossing configurations. The most prominent example is the class of planar graphs, which admit drawings without any crossings. Other well-studied examples include k -planar graphs where every edge can have up to k crossings, RAC-graphs where edges are straight line



© Michael Kaufmann, Boris Klemz, Kristin Knorr, Meghana M. Reddy, Felix Schröder, and Torsten Ueckerdt;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 7; pp. 7:1–7:17



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

segments and every crossing happens at a right angle, or quasiplanar graphs where no three edges are allowed to pairwise cross each other. As all these include planar graphs as a special case, they are commonly known as *beyond-planar* graph classes. See [14] for a recent survey.

When studying a beyond-planar graph class \mathcal{G} , one of the most natural and important questions is to determine how many edges a graph in \mathcal{G} can have. The *edge density* of \mathcal{G} is the function giving the maximum number of edges over all n -vertex graphs in \mathcal{G} . For example, planar graphs with at least three vertices have edge density $3n - 6$. All the beyond-planar graph classes mentioned above have linear edge density, i.e., their edge density is in $\mathcal{O}(n)$. Proofs of precise linear upper bounds for the edge density of a specific class \mathcal{G} are often times involved and very tailored to the specific drawing style that defines \mathcal{G} . In particular, getting a *tight* bound (even only up to an additive constant) was achieved only in a couple of cases. A particularly simple case is the class of planar graphs, whose edge density of $3n - 6$ can be easily derived from Euler’s Formula. However, a comparable formula for general drawings (with crossings) that can be used to easily derive tight upper bounds for the edge density of beyond-planar graph classes was not known – until now.

Our Contribution. In this paper, we introduce a new tool, called the **Density Formula** (Lemma 3.1), which can be used to derive upper bounds on edge densities for many beyond-planar graph classes. It is an equation that relates the number of edges, vertices, crossings, and sizes¹ of cells¹ in a connected drawing of a graph and is parameterized by a real-valued parameter t . Intuitively, the Density Formula allows us to obtain density bounds by counting the cells of small size in a drawing, which is often times quite an elementary task. The parameter t is chosen in accordance with the desired density bound, e.g., when aiming for a bound of roughly $5n$, where n is the number of vertices, we might set $t = 5$, in which case the Density Formula states that the number of edges in the drawing is $5n - 10 - \sum_{c \in \mathcal{C}} (\|c\| - 5) - x$, where x is the number of crossings, \mathcal{C} is the set of cells, and $\|c\|$ denotes the size of a cell c . Thus, any upper bound on $-(\sum_{c \in \mathcal{C}} (\|c\| - 5) + x)$ yields an upper bound on the number of edges. Since the quantity $(\|c\| - 5)$ is non-negative for cells of size at least 5, such a bound can indeed be obtained by counting the cells of small sizes (here, at most 4) and cross-charging them with the crossings.

We give the precise, more general, statement of the Density Formula in Section 3, where we also develop some general tools that help with the required counting / charging arguments. Before that, in Section 2, we formally define some basic notions, such as (connected) drawings, cells, cell sizes, etc., and discuss some further preliminaries. We demonstrate the capabilities of the Density Formula by providing several applications, which are discussed next.

Applications

k -Bend RAC-Drawings. For an integer $k \geq 0$, a drawing Γ in the plane \mathbb{R}^2 of some graph G is *k -bend RAC*, which stands for right-angle crossing, if every edge of Γ is a polyline with at most k bends and every crossing in Γ happens at a right angle, and in this case G is called a *k -bend RAC-graph*. The k -bend RAC-graphs were introduced by Didimo, Eades, and Liotta [13], who prove that n -vertex 0-bend RAC-graphs have at most $4n - 10$ edges (and this is tight), while every graph is a 3-bend RAC-graph. The best known upper bound

¹ Loosely speaking, a cell of a drawing is a connected region of the plane (or sphere) after removing the drawing; its size is the number of vertex and edge segment occurrences along its boundary, see Figure 3 for examples.

■ **Table 1** Overview of edge density bounds, i.e., the maximum number of edges in connected n -vertex (n large enough) graphs in that graph class. In particular, the third column lists previous work on upper bounds and the fourth column lists the upper bounds we obtain using the Density Formula. **Previously unknown bounds** are highlighted with boxes. Results from the literature that are written in light red rely on **incomplete proofs** as they use an incorrect statement from [16, 17], as we discuss in more detail in Section 5.1.

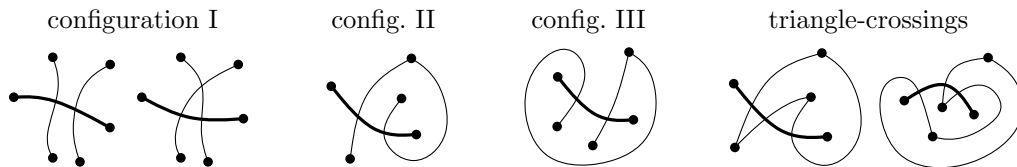
beyond-planar graph class	variant	previous work	Density Formula	lower bound
0-bend RAC	no constraint	$4n - 10$ [13]	$4n - 8$ full version [15]	$4n - 10$ [13]
1-bend RAC	non-homotopic	$5.4n - 10.8$ [5]	$5n - 10$ Theorem 4.2	$5n - 10$ [5]
2-bend RAC	non-homotopic	$20n - 24$ [21]	$10n - 19$ Theorem 4.2	$10n - 54$ Theorem 4.3
fan-crossing / fan-planar	simple	$5n - 10$ [11, 12, 16, 17]	$5n - 10$ Theorem 5.2	$5n - 10$ [16, 17]
fan-cr. / fan-pl. + bipartite	simple	$4n - 12$ [7, 12]	$4n - 10$ full version [15]	$4n - 16$ [7]
quasiplanar	simple	$6.5n - 20$ [4]	$6.5n - 20$ full version [15]	$6.5n - 20$ Theorem 6.2
	non-homotopic	$8n - 20$ [4]	$8n - 20$ full version [15]	$8n - 20$ Theorem 6.1
1 ⁺ -real face	non-homotopic	$5n - 10$ [10]	$5n - 10$ full version [15]	$5n - 10$ [10]
2 ⁺ -real face	non-homotopic	$4n - 8$ [10]	$4n - 8$ full version [15]	$4n - 8$ [10]
k^+ -real face $k \geq 3$	no constraint	$\frac{k}{k-2}(n-2)$ [10]	$\frac{k}{k-2}(n-2)$ full version [15]	$\frac{k}{k-2}(n-2)$ [10]
1-planar	non-homotopic	$4n - 8$ [20]	$4n - 8$ full version [15]	$4n - 8$ [20]
2-planar	non-homotopic	$5n - 10$ [20]	$5n - 10$ full version [15]	$5n - 10$ [20]

for simple² 1-bend RAC-drawings is $5.4n - 10.8$ [5], while the lower bound is $5n - 10$ [5]. By means of the Density Formula, we give an improved upper bound of $5n - 10$ for the connected case, which is best-possible. Very recently, Tóth [21] established an upper bound of $20n - 24$ for simple graphs admitting 2-bend RAC-drawings, thereby improving the long

² Loosely speaking, in a simple drawing, every pair of edges intersects in at most one point, thereby forbidding digons formed by segments of two edges. In non-homotopic drawings, such digons are allowed as long as both regions bounded by a digon contain at least one vertex or crossing.

standing previous best upper bound of $74.2n$ [8]. Using the Density Formula, we derive a significantly improved upper bound of $10n - 19$ for simple drawings. We also show that this bound is tight up to an additive constant by constructing an infinite family of simple 2-bend RAC-drawings with $10n - 54$ edges. (A similar construction of 2-bend RAC-drawings with $10n - 46$ edges was presented by Angelini et al. [6], but their drawings are not simple.) Both of our upper bound results in fact apply even to the non-homotopic² case. We prove these results in Section 4. For completeness, in the full version [15], we also apply the Density Formula to reprove the known upper bound for 0-bend RAC-graphs.

Fan-Crossing Drawings. A drawing Γ on the sphere \mathbb{S}^2 of some graph G is *fan-crossing* if for every edge e of G , the edges crossing e in Γ form a star in G , and in this case G is called a *fan-crossing graph*. A simple drawing is fan-crossing if and only if there is no configuration I and no triangle-crossing, as shown in Figure 1. Fan-crossing drawings generalize fan-planar drawings; but the story about fan-planar graphs is problematic and tricky. In a preprint from 2014, Kaufmann and Ueckerdt [16] introduced fan-planar drawings as the simple drawings in \mathbb{R}^2 without configuration I and II, as shown in Figure 1. These are today known as *weakly fan-planar* and they show that n -vertex weakly fan-planar graphs have at most $5n - 10$ edges [16]. However, recently, a first flaw in this proof was discovered [18]. It was fixed in the journal version [17] of [16] from 2022 by additionally forbidding configuration III, as shown in Figure 1. These, more restricted, graphs are today known as *strongly fan-planar* graphs, and it is known that this indeed is a different graph class [12]. However, for each n -vertex weakly fan-planar graph, there is a strongly fan-planar graph on the same number of vertices and edges [12]. So any density result could be lifted. As every triangle-crossing contains configuration II, weakly fan-planar graphs are also fan-crossing, while again these are indeed different graph classes [11]. However again, for each n -vertex fan-crossing graph, there is a weakly fan-planar graph on the same number of vertices and edges [11], and thus the density can be lifted to fan-crossing graphs.



■ **Figure 1** Simple fan-planar drawings have neither configuration I, nor II, nor III. Simple fan-crossing drawings have no configuration I and no triangle-crossings.

In Section 5, we prove an upper bound of $5n - 10$ for simple n -vertex connected fan-crossing drawings by applying the Density Formula. We also briefly describe in Section 5.1 another issue in the (updated) proof from [17] by providing a counterexample to one of their crucial statements. As all previous density results rest on [17], our result on fan-crossing drawings is the first complete proof for fan-crossing, weakly fan-planar, and strongly fan-planar drawings. Moreover, our proof is significantly simpler than the strategy used in [17]. In the full version [15], we discuss the special case of fan-crossing drawings of bipartite graphs, for which we obtain similar results.

We remark that, in a very recent preprint, Ackerman and Keszegh [3] also (independently of us) propose a new alternative proof for the $5n - 10$ upper bound for fan-crossing graphs. Moreover, Brandenburg [11] also considers *adjacency-crossing* graphs by just forbidding configuration I, but allowing triangle-crossings. He shows however that this class coincides with fan-crossing graphs, and hence our $5n - 10$ upper bound applies.

Quasiplanar Drawings. A drawing Γ on the sphere \mathbb{S}^2 of some graph G is *quasiplanar* if no three edges of G pairwise cross in Γ and in this case G is called a *quasiplanar graph*. Quasiplanar graphs were introduced by Pach [19]. It is known that simple n -vertex quasiplanar drawings have at most $6.5n - 20$ edges [4] and non-homotopic connected n -vertex quasiplanar drawings have at most $8n - 20$ edges [4]. However, the best known lower bounds [4] are just $6.5n - 29$ and $7n - 29$, respectively. In the full version [15], we reprove the known upper bounds using the Density Formula. In Section 6, inspired by insights gained in our upper bound proofs, we provide families of drawings showing that the previous upper bounds are actually best-possible.

Further applications. In the full version [15], we also use the Density Formula to reprove (and slightly generalize) the known upper bounds for so-called k^+ -real face graphs and 1-planar and 2-planar graphs. All results are summarized in Table 1.

Some previously known density proofs already contain ideas that are similar to (parts of) our strategy, and we discuss this further in Section 7. But with the Density Formula, whose proof is merely a straight-forward application of Euler’s Formula, we have a unified and simple approach that somewhat unveiled the essential tasks in this field of research. We believe it will serve as a useful tool for proving density bounds in the future. For example, very recently, the Density Formula was already applied by Bekos et al. [9] to give bounds on the density of k -planar graphs without short cycles. Moreover, given that the Density Formula behaves symmetrically when it comes to the number of edges and the number of crossings, it seems plausible that it can also be used to derive bounds on crossing numbers of beyond-planar graph classes.

2 Terminology, Conventions, and Notation

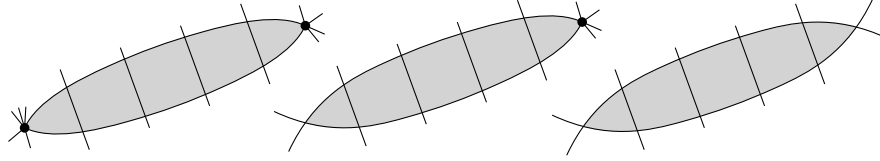
All graphs in this paper are finite and have no loops, but possibly parallel edges. We consider classic node-link drawings of graphs. More precisely, in a *drawing* Γ of a graph $G = (V, E)$ (in the plane \mathbb{R}^2 or on the sphere \mathbb{S}^2) the vertices are pairwise distinct points and each edge is a simple³ Jordan curve connecting the two vertices. In particular, no edge crosses itself. In order to avoid special treatment of the unbounded region, we mostly consider drawings on the sphere \mathbb{S}^2 . In one case (RAC-drawings), however, we consider drawings in the plane \mathbb{R}^2 , as the drawing style involves straight lines and angles. In any case, we require throughout the usual assumptions of no edge passing through a vertex, having only proper crossings and no touchings, only finitely many crossings, and no three edges crossing in the same point.

So-called *simple* drawings are a particularly important and well-studied type of drawing. In such a drawing, any two edges have at most one point in common. In particular, simple drawings contain no two edges crossing more than once, no crossing adjacent edges, and no parallel edges. However, there is increasing interest in generalizations of simple drawings that allow these types of configurations, as long as the involved edge pairs are not just drawn in basically the same way within a narrow corridor. This notion is formalized as follows. A *lens* in a drawing Γ is a region whose boundary is described by a simple³ closed Jordan curve γ such that γ is comprised of exactly two contiguous parts, each being formed by (a part of) one edge. So the curve γ consists of either two non-crossing parallel edges, or parts of two crossing adjacent edges, or parts of two edges crossing more than once; see Figure 2 for

³ with no self-intersection

7:6 The Density Formula

illustrations. Be aware that for drawings in \mathbb{R}^2 , a lens might be an unbounded region. Now let us call a drawing Γ *non-homotopic*⁴ if every lens contains a vertex or a crossing in its interior. This is indeed a generalization of simple drawings, as these cannot contain any lens.



■ **Figure 2** Lenses with no vertex and no crossing in their interior. Such configurations are forbidden in non-homotopic drawings.

Beyond-planar graph classes are implicitly defined as all graphs G that admit a drawing Γ with specific properties, such as all edges of G having at most one crossing in Γ . These for example are called *1-planar drawings*⁵ and the corresponding graphs are called *1-planar graphs*. We extend this policy to the properties “simple” and “non-homotopic” in the same way, e.g., a non-homotopic 1-planar graph is a graph that admits a non-homotopic 1-planar drawing. Observe that this aligns with a simple graph being a graph with no loops (which we rule out entirely) and no parallel edges.

Fix a drawing Γ of some graph $G = (V, E)$. Setting up some notation, let $E_x \subseteq E$ be the set of all *crossed* edges of G , i.e., with at least one crossing in Γ , and $E_p = E \setminus E_x$ be the set of all *planar* edges (without crossings). Further, let \mathcal{X} denote the set of all crossings in Γ . Each edge e is split into one or more *edge-segments* by the crossings along e . That is, an edge with exactly k crossings, $k \geq 0$, is split into exactly $k + 1$ edge-segments. An *outer* edge-segment of Γ is incident to some vertex, while an *inner* edge-segment is not. The set of all edge-segments of Γ is denoted by \mathcal{S} and the set of all inner edge-segments by \mathcal{S}_{in} .

► **Observation 2.1.** *Let Γ be any drawing of some graph $G = (V, E)$. Then*

$$|\mathcal{S}| = 2|\mathcal{X}| + |E| \quad \text{and} \quad |\mathcal{S}_{\text{in}}| = |\mathcal{S}| - 2|E_x| - |E_p| = 2|\mathcal{X}| - |E_x|.$$

The *planarization* Λ of the drawing Γ is the planar drawing obtained from Γ by replacing each crossing by a new vertex and replacing each edge by its edge-segments. We call the drawing Γ *connected* if the graph underlying its planarization Λ is connected. Let us remark that most density results in this paper assume for brevity the considered graphs to be connected, while our proofs actually only require the respective drawings to be connected.

The connected components of \mathbb{S}^2 or \mathbb{R}^2 after removing all edges and vertices in Γ are called the *cells* of Γ . The set of all cells is denoted by \mathcal{C} . The *boundary* ∂c of each cell c consists of a cyclic sequence alternating between $V \cup \mathcal{X}$ and \mathcal{S} , i.e., vertices/crossings and edge-segments of Γ . If Γ is not connected, ∂c might consist of multiple such sequences. Be aware that an edge-segment might appear twice on ∂c , a crossing might appear up to four times on ∂c , and a vertex v may appear up to $\deg(v)$ times on ∂c . Each appearance of an edge-segment / vertex / crossing on ∂c is called an *edge-segment-incidence* / *vertex-incidence* / *crossing-incidence* of c . The total number of edge-segment-incidences and vertex-incidences

⁴ Usually, non-homotopic drawings require a vertex in each lens, but we only need our weaker requirement.

⁵ In literature, planar drawings are also referred to as *plane drawings*, and a planar graph with a fixed plane drawing is called a *plane graph*. And there is a similar distinction for each beyond-planar graph class (e.g., 1-planar vs. 1-plane graphs). But for simplicity, we treat *planar* and *plane* as equivalent here.

of c is called the *size* of c and denoted by $\|c\|$. Note that $\|c\|$ does not take the number of crossings on ∂c into account, while edge-segments and vertices are counted with multiplicities; see Figure 3 for several examples. For an integer i , let \mathcal{C}_i and $\mathcal{C}_{\geq i}$ denote the set of all cells of size exactly i and the set of all cells of size at least i , respectively.

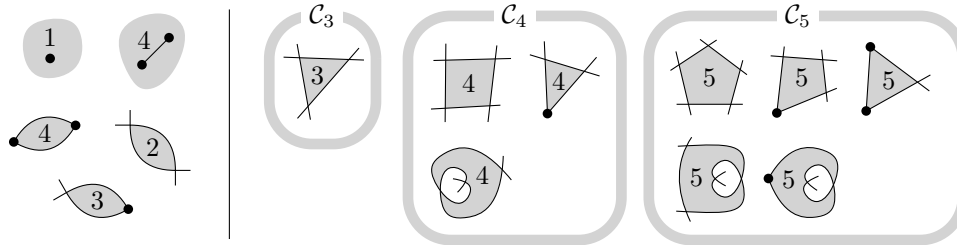


Figure 3 Left: Cells (and their sizes) that do not appear in simple or non-homotopic drawings on at least three vertices. Right: All types of cells c of size $\|c\| \leq 5$ in a non-homotopic connected drawing on at least three vertices (cf. Observation 2.2). The bottom row shows degenerate $\overline{4}$ -cells, $\overline{5}$ -cells, and $\overline{5}$ -cells.

Figure 3(right) shows all possible types of cells of size at most 5 that can occur in connected non-homotopic drawings on \mathbb{S}^2 with at least three vertices; the bottom row shows the *degenerate* cells, i.e., those cells c with a crossing or vertex appearing repeated in ∂c . When proving edge density bounds by means of the Density Formula, our main task will be to count these “small cells”; we denote the different types for convenience with little pictograms, such as $\overline{3}$ -cells, $\overline{4}$ -cells, and $\overline{5}$ -cells. More precisely, each pictogram describes a type of cell c in terms of the sequence of types of incidences found along its connected boundary ∂c . E.g., the boundary of a $\overline{3}$ -cell consists of a vertex-incidence, an edge-segment-incidence, a crossing-incidence, an edge-segment-incidence, a crossing-incidence, and an edge-segment-incidence, in this order.

► **Observation 2.2.** Let Γ be any non-homotopic connected drawing of some graph G with at least three vertices. Then

- \mathcal{C}_3 is the set of all $\overline{3}$ -cells,
- \mathcal{C}_4 is the set of all $\overline{4}$ -cells and $\overline{4}$ -cells, and
- \mathcal{C}_5 is the set of all $\overline{5}$ -cells, $\overline{5}$ -cells, and $\overline{5}$ -cells. ┘

3 The Density Formula

In this section, we first state and prove the Density Formula, then derive some immediate consequences, and finally develop some general tools that are useful for its application.

► **Lemma 3.1** (Density Formula). Let t be a real number. Let Γ be a connected drawing of a graph $G = (V, E)$ with at least one edge. Then

$$|E| = t(|V| - 2) - \sum_{c \in \mathcal{C}} \left(\frac{t-1}{4} \|c\| - t \right) - |\mathcal{X}|.$$

Proof. First recall that, by Observation 2.1,

$$|\mathcal{S}| = |E| + 2|\mathcal{X}|. \tag{1}$$

7:8 The Density Formula

Considering the total sum of $\|c\|$ over all cells $c \in \mathcal{C}$, we count every vertex $v \in V$ exactly $\deg(v)$ times and every edge-segment exactly twice. (Here we use that $|E| \geq 1$ and, thus, $\deg(v) \geq 1$ for each $v \in V$ as Γ is connected.) Thus,

$$\sum_{c \in \mathcal{C}} \frac{1}{4} \|c\| = \frac{1}{4} \left(\sum_{v \in V} \deg(v) + 2|\mathcal{S}| \right) = \frac{1}{4} (2|E| + 2|\mathcal{S}|) \stackrel{(1)}{=} \frac{1}{4} (4|E| + 4|\mathcal{X}|) = |E| + |\mathcal{X}|. \quad (2)$$

Let $\Lambda = (V_\Lambda, E_\Lambda)$ be the planarization of G . It has exactly $|V_\Lambda| = |V| + |\mathcal{X}|$ vertices, $|E_\Lambda| = |\mathcal{S}|$ edges, and $|\mathcal{C}|$ faces. As Λ is connected, we can apply Euler's Formula (*):

$$|E| + 2|\mathcal{X}| \stackrel{(1)}{=} |\mathcal{S}| = |E_\Lambda| \stackrel{(*)}{=} |V_\Lambda| + |\mathcal{C}| - 2 = |V| - 2 + |\mathcal{C}| + |\mathcal{X}|,$$

which gives the following two equations:

$$|E| = |V| - 2 + |\mathcal{C}| - |\mathcal{X}| = (|V| - 2) - \sum_{c \in \mathcal{C}} (-1) - |\mathcal{X}| \quad (3)$$

$$0 = |V| - 2 - (|E| + |\mathcal{X}|) + |\mathcal{C}| \stackrel{(2)}{=} (|V| - 2) - \sum_{c \in \mathcal{C}} \left(\frac{1}{4} \|c\| - 1 \right) \quad (4)$$

Adding (3) and $(t - 1)$ times (4) gives the result. \blacktriangleleft

The Density Formula can be used to find upper bounds on edge densities by counting cells of small size. To see how this works, let us plug in two specific values for t ($t = 4$ and $t = 5$, which we use quite often throughout the paper) resulting in the following statements:

► **Corollary 3.2.** *For any connected drawing Γ of a graph $G = (V, E)$ with $|E| \geq 1$ we have*

$$|E| = 4|V| - 8 - \sum_{c \in \mathcal{C}} \left(\frac{3}{4} \|c\| - 4 \right) - |\mathcal{X}| \leq 4|V| - 8 + \frac{7}{4} |\mathcal{C}_3| + |\mathcal{C}_4| + \frac{1}{4} |\mathcal{C}_5| - |\mathcal{X}|.$$

► **Corollary 3.3.** *For any connected drawing Γ of a graph $G = (V, E)$ with $|E| \geq 1$ we have*

$$|E| = 5|V| - 10 - \sum_{c \in \mathcal{C}} (\|c\| - 5) - |\mathcal{X}| = 5|V| - 10 + 2|\mathcal{C}_3| + |\mathcal{C}_4| - |\mathcal{X}| - \sum_{c \in \mathcal{C}_{\geq 5}} (\|c\| - 5).$$

So indeed, Corollary 3.2 allows us to derive upper bounds on $|E|$ by proving upper bounds on $\frac{7}{4} |\mathcal{C}_3| + |\mathcal{C}_4| + \frac{1}{4} |\mathcal{C}_5| - |\mathcal{X}|$, which can be done by counting cells of sizes 3, 4, and 5 and cross-charging them with the crossings. Similarly, noting that $\sum_{c \in \mathcal{C}_{\geq 5}} (\|c\| - 5)$ is non-negative, Corollary 3.3 allows us to derive upper bounds on $|E|$ by proving upper bounds on $2|\mathcal{C}_3| + |\mathcal{C}_4| - |\mathcal{X}|$. In fact, by taking into account the cells of larger sizes, one can sometimes obtain more precise bounds. Thus, in the remainder of the section, we will devise some general tools that help with the required counting / charging arguments. Moreover, we give a first concrete example of such an argument by proving Lemma 3.4, which is a simple but very general statement – in fact, it immediately gives two bounds of $4n - 8$ in Table 1.

► **Lemma 3.4.** *Let Γ be a non-homotopic connected drawing of a graph $G = (V, E)$ with $|V| \geq 3$ and with no ∇ -cells, no \square -cells, no \bowtie -cells, no \triangleleft -cells, and no \searrow -cells. Then $|E| \leq 4|V| - 8$.*

Proof. By assumption and Observation 2.2, we have $|\mathcal{C}_3| = 0$ and $|\mathcal{C}_4| = 0$ and $|\mathcal{C}_5| = \# \searrow$ -cells. Clearly, every crossing is incident to at most four \searrow -cells and every \searrow -cell has one incident crossing. In particular, it follows that $\# \searrow$ -cells $\leq 4|\mathcal{X}|$. Therefore, the Density Formula with $t = 4$ (Corollary 3.2) immediately gives

$$|E| \leq 4|V| - 8 + \frac{7}{4} |\mathcal{C}_3| + |\mathcal{C}_4| + \frac{1}{4} |\mathcal{C}_5| - |\mathcal{X}| = 4|V| - 8 + \frac{1}{4} \# \searrow$$

► **Lemma 3.5.** *Let Γ be any non-homotopic drawing. Then $\#\triangleleft$ -cells $\leq |\mathcal{X}|$. Moreover, one can assign each \triangleleft -cell c a crossing in ∂c such that each crossing is assigned at most once.*

Proof. At every crossing incident to a \triangleleft -cell there is one inner edge-segment and one outer edge-segment. As Γ is non-homotopic, every inner edge-segment is incident to at most one \triangleleft -cell. This implies that every crossing is incident to at most two \triangleleft -cells, while every \triangleleft -cell has two distinct incident crossings, which implies the claim. ◀

► **Lemma 3.6.** *Let Γ be a connected non-homotopic drawing of some graph G with at least three vertices. Then*

$$|\mathcal{S}_{\text{in}}| \geq \#\triangleleft\text{-cells} + 2 \cdot \#\square\text{-cells} + 3 \cdot \#\nabla\text{-cells} \quad \text{and} \quad |\mathcal{S}_{\text{in}}| + \#\triangleleft\text{-cells} \geq 2|\mathcal{C}_4| + 3|\mathcal{C}_3|.$$

Proof. The second inequality follows by combining the first inequality with Observation 2.2. To prove the first inequality, let us call an inner edge-segment *bad* if it is incident to a \triangleleft -cell or ∇ -cell in Γ . As Γ is non-homotopic, every bad edge-segment is incident to only one \triangleleft -cell or ∇ -cell. Hence, for the set $\mathcal{B} \subseteq \mathcal{S}_{\text{in}}$ of all bad edge-segments we have $|\mathcal{B}| = \#\triangleleft\text{-cells} + 3 \cdot \#\nabla\text{-cells}$. Define an auxiliary graph $J = (V_J, E_J)$ with vertex set $V_J = \mathcal{S}_{\text{in}}$ and with two edge-segments being adjacent in J if and only if they are an opposite pair of edge-segments for some \square -cell. Note that this and the following is true whether the \square -cells are degenerate or not. Then $|V_J| = |\mathcal{S}_{\text{in}}|$ and $|E_J| = 2 \cdot \#\square\text{-cells}$, and the maximum degree in J is at most two. Observe that J contains no cycle, as such a cycle would correspond to a cyclic arrangement of \square -cells and therefore two edges in G with no endpoints. Hence, J is a disjoint union of paths (possibly of length 0) and every bad edge-segment is an endpoint of one such path. Further, no path in J on two or more vertices can have two bad endpoints, as such a path would correspond to a lens in Γ containing no vertex and no crossing (as illustrated in Figure 2), contradicting the fact that Γ is non-homotopic. Note that this implies $|V_J| \geq |E_J| + |\mathcal{B}|$. Recalling that $|\mathcal{B}| = \#\triangleleft\text{-cells} + 3 \cdot \#\nabla\text{-cells}$, $|V_J| = |\mathcal{S}_{\text{in}}|$ and $|E_J| = 2 \cdot \#\square\text{-cells}$, we obtain the first inequality of the lemma. ◀

4 k -Bend RAC-Graphs

In this section, we present our results for 1-bend and 2-bend RAC-graphs. We begin with the upper bounds, for which we only require the following lemma.

► **Lemma 4.1.** *Let $k \in \{1, 2\}$ and Γ be a non-homotopic drawing of a connected graph $G = (V, E)$ such that every crossed edge $e \in E_x$ is a polyline with at most k bends, and every crossing is a right-angle crossing. Then $2|\mathcal{C}_3| + |\mathcal{C}_4| \leq |\mathcal{X}| + \frac{k-1}{2}(|E_x| + 1)$.*

Proof. Lemma 3.6 gives

$$|\mathcal{S}_{\text{in}}| \geq \#\triangleleft\text{-cells} + 2 \cdot \#\square\text{-cells} + 3 \cdot \#\nabla\text{-cells}. \quad (5)$$

Now, each \triangleleft -cell and each ∇ -cell c is a polygon, and as all crossings have right angles, c has at least one convex corner that is a bend, except when c is the unbounded cell. As every bend is a convex corner for only one cell, we have

$$k|E_x| \geq \#\triangleleft\text{-cells} + \#\nabla\text{-cells} - 1. \quad (6)$$

Together this gives the desired

$$4|\mathcal{C}_3| + 2|\mathcal{C}_4| \stackrel{(5),(6)}{\leq} |\mathcal{S}_{\text{in}}| + k|E_x| + 1 = 2|\mathcal{X}| + (k-1)|E_x| + 1,$$

where the last equality uses $|\mathcal{S}_{\text{in}}| = 2|\mathcal{X}| - |E_x|$ from Observation 2.1. Dividing by 2 and realizing that $2|\mathcal{C}_3| + |\mathcal{C}_4|$ is an integer, concludes the proof. ◀

7:10 The Density Formula

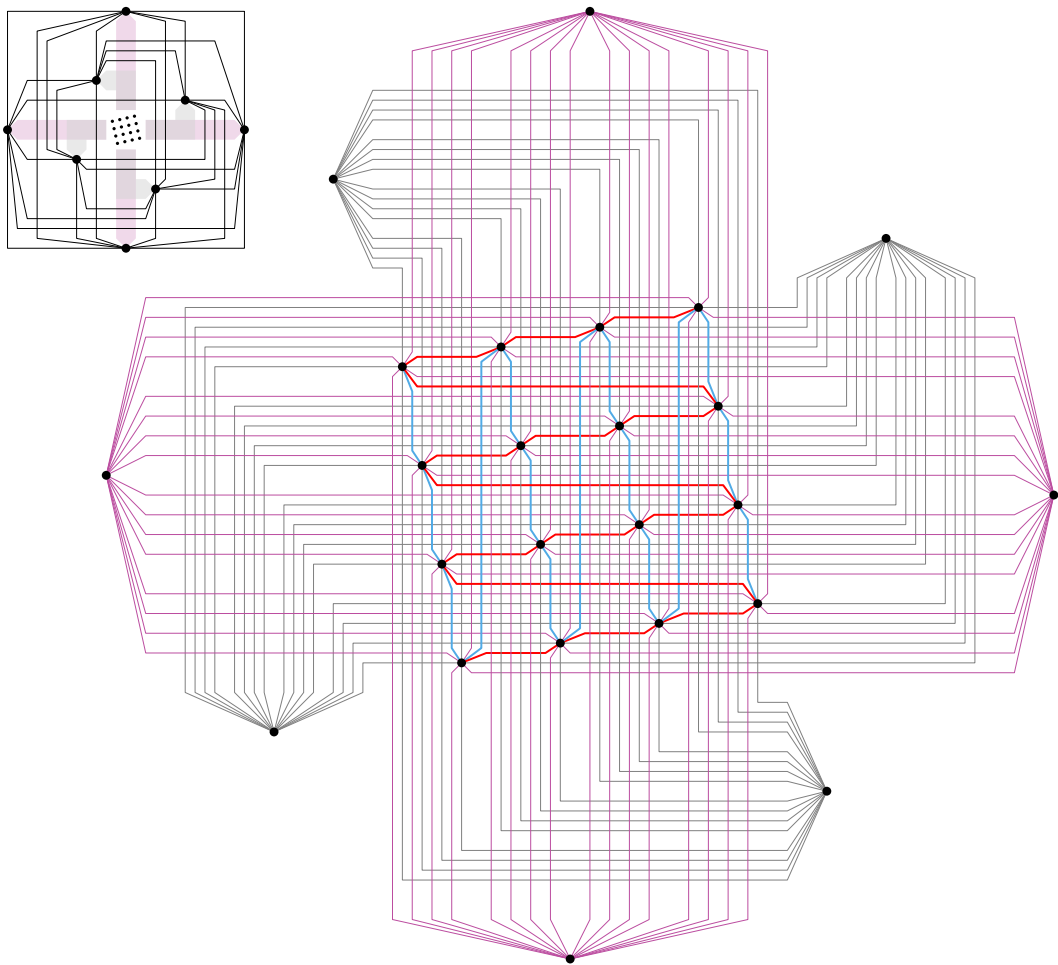
► **Theorem 4.2.** *For every $k \in \{1, 2\}$ and every $n \geq 3$, every connected non-homotopic n -vertex k -bend RAC-graph G has at most $k(5n - 10) + (k - 1)$ edges.*

Proof. Let Γ be a non-homotopic k -bend RAC-drawing of $G = (V, E)$. As G is connected, so is Γ . The Density Formula with $t = 5$ (Corollary 3.3) and Lemma 4.1 immediately give

$$|E| \leq 5|V| - 10 + 2|\mathcal{C}_3| + |\mathcal{C}_4| - |\mathcal{A}| \leq 5|V| - 10 + \frac{k-1}{2}(|E_x| + 1),$$

which implies the desired $|E| \leq |E| + (k-1)|E_p| \leq k(5|V| - 10) + (k-1)$. ◀

The lower bound construction in [6, Theorem 6] gives 2-bend RAC-graphs with n vertices and $10n - 46$ edges, but the provided drawings are not simple (not even non-homotopic). We modify it giving simple 2-bend RAC-graphs with n vertices and $10n - 54$ edges.



■ **Figure 4** (Illustration of) a simple 2-bend RAC-drawing of G_4 from Theorem 4.3.

► **Theorem 4.3.** *For every integer $k \geq 1$ there exists a simple connected 2-bend RAC-graph G_k with $n = k^2 + 8$ vertices and $10n - 54$ edges.*

Proof. For $k \geq 1$, a simple 2-bend RAC-drawing of the graph G_k (Figure 4) consists of

- a set Q of $k^2 = n - 8$ vertices in a regular but slightly rotated $k \times k$ grid,
- an x -monotone 2-bend edge between any two vertices of Q with consecutive y -coordinates (red), ($n - 9$ edges)
- a y -monotone 2-bend edge between any two vertices of Q with consecutive x -coordinates (blue), ($n - 9$ edges)
- a set P of eight vertices around Q , each connected to all vertices of Q with either all (weakly) x -monotone 2-bend edges or all (weakly) y -monotone 2-bend edges (gray and purple), ($8(n - 8)$ edges)
- a 2-bend edge between any two vertices of P (black). (28 edges)

The routing of the edges is illustrated in Figure 4. ◀

5 Fan-Crossing Graphs

Here, we present our upper bound for fan-crossing graphs, starting with the key lemma.

► **Lemma 5.1.** *Let Γ be a simple connected fan-crossing drawing of a graph with at least three vertices. Then $|\mathcal{C}_4| \leq |\mathcal{X}|$.*

Proof. First, observe that there are no degenerate \square -cells since Γ is simple. We shall map each cell $c \in \mathcal{C}_4$ onto one of its incident crossings $\phi(c)$ in such a way that no crossing is used more than once, i.e., the mapping $\phi: \mathcal{C}_4 \rightarrow \mathcal{X}$ is injective.

As an auxiliary structure, we orient edge-segments incident to \square -cells as follows. Let c be a \square -cell and s, s' be a pair of *opposite* edge-segments in ∂c (that do not share a crossing). As Γ is simple, the corresponding edges e, e' are distinct. Now orient s and s' , each towards the (unique) common endpoint of e and e' , which exists as Γ is fan-crossing. Doing this for every \square -cell and every pair of opposite edge-segments, we obtain a well-defined orientation:

▷ **Claim.** An edge-segment s shared by two \square -cells c_1, c_2 has the same orientation in both.

Proof. Observe that the six crossings incident to c_1 and c_2 are pairwise distinct since Γ is a simple drawing. Let $e = uv$ be the edge containing s and e_1, e_2 be the two (distinct) edges crossing e at the endpoints of s (which are crossings in Γ). Further, let f_1, f_2 be the two edges containing the edge-segment opposite to s in c_1, c_2 , respectively. In particular, e, f_1, f_2 all cross e_1 and all cross e_2 . As Γ is fan-crossing⁶, e, f_1, f_2 have a common endpoint, say u . But then s is oriented consistently towards u according to both incident \square -cells c_1, c_2 . ◀

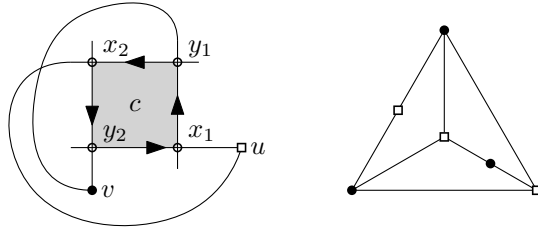
▷ **Claim.** For each \square -cell c , there is at least one crossing x incident to c such that both edge-segments incident to c and x are oriented outgoing from x .

Proof. Assuming otherwise, the edge-segments would be oriented cyclically around ∂c . Consider two crossings x_1, x_2 that are *opposite* along c (do not belong to the same edge segment of c). The edges of the two (distinct) edge-segments of c that are outgoing from x_1, x_2 have a common endpoint u , as Γ is fan-crossing; see Figure 5. The edges of the two edge-segments of c that are outgoing from the remaining two opposite crossings y_1, y_2 behave symmetrically and share an endpoint v , which is distinct from u , as Γ is simple. The four parts of the mentioned edges that join the vertices u, v with the crossings x_1, x_2, y_1, y_2 are pairwise crossing-free since Γ is simple. Hence, using these edge parts, we can obtain a planar drawing of the bipartite

⁶ Here it is crucial that e, f_1 and f_2 do not form a triangle-crossing.

7:12 The Density Formula

graph $K_{3,3} - e$ (obtained from $K_{3,3}$ by removing an edge) so that the bipartition classes are $\{x_1, x_2, v\}$ and $\{y_1, y_2, u\}$ and where the four degree-3 vertices form a face. However, the unique⁷ planar embedding of $K_{3,3} - e$ has no such face; see again Figure 5. \triangleleft

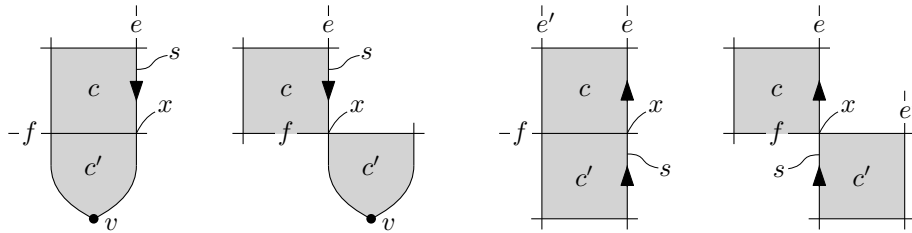


■ **Figure 5** A cyclic orientation of a \square -cell leading to a double-crossing (left) or the unique⁷ planar embedding of $K_{3,3} - e$ (right).

Now for every \square -cell c , we set $\phi(c)$ to be a crossing x in ∂c whose two edge-segments in ∂c are oriented outgoing from x . Moreover, by Lemma 3.5 for every \triangle -cell c , we can set $\phi(c)$ to be a crossing in ∂c such that $\phi(c) \neq \phi(c')$ for any distinct \triangle -cells c, c' .

▷ **Claim.** The mapping $\phi: \mathcal{C}_4 \rightarrow \mathcal{X}$ is injective.

Proof. For a \square -cell c and a \triangle -cell or \square -cell c' with $\phi(c) = x \in \partial c'$, we shall show $\phi(c') \neq x$.



■ **Figure 6** The four cases of a \square -cell c sharing a crossing x with a \triangle -cell or \square -cell c' .

If c' is a \triangle -cell, let e be the edge that is incident to the vertex $v \in \partial c'$ and contains x . Further, let f be the other edge at x (containing the inner edge-segment of c') and let s be the edge-segment of e in ∂c ; see Figure 6. Evidently, v is the common endpoint of all edges crossing f . In particular, s is oriented inwards at x , which is a contradiction to $x = \phi(c)$.

If c' is a \square -cell, let s be an edge-segment that ends at x and belongs to $\partial c'$, but not to ∂c . Let e be the edge containing s , let f be the other edge at x , and let e' be the edge containing the edge-segment opposite of s in $\partial c'$; see Figure 6. As $\phi(c) = x$, the edge-segment of e in ∂c is oriented outwards at x and towards the common endpoint of all edges crossing f . As e and e' cross f , edge-segment s is oriented inwards at x and thus $\phi(c') \neq x$. \triangleleft

Clearly, the last claim implies the desired $|\mathcal{C}_4| \leq |\mathcal{X}|$. \blacktriangleleft

Let us prove the edge density of $5n - 10$ for connected simple fan-crossing graphs in a slightly stronger form.

⁷ All planar embeddings of $K_{3,3} - e$ are combinatorially isomorphic since it is a subdivision of the 3-connected complete graph K_4 .

► **Theorem 5.2.** *Let Γ be a simple connected fan-crossing drawing of some graph $G = (V, E)$ with $|V| \geq 3$. Then*

$$|E| \leq 5|V| - 10 - \sum_{c \in \mathcal{C}_{\geq 5}} (\|c\| - 5).$$

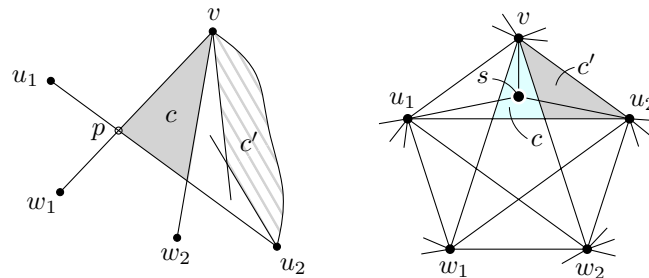
Proof. As every edge is crossed only by adjacent edges and adjacent edges do not cross (Γ is simple), there are no ∇ -cells in Γ and, hence, $|\mathcal{C}_3| = 0$. Therefore Corollary 3.3 (i.e., the Density Formula with $t = 5$) immediately gives

$$|E| = 5|V| - 10 + 2|\mathcal{C}_3| + |\mathcal{C}_4| - |\mathcal{X}| - \sum_{c \in \mathcal{C}_{\geq 5}} (\|c\| - 5) \leq 5|V| - 10 - \sum_{c \in \mathcal{C}_{\geq 5}} (\|c\| - 5),$$

where the last inequality uses Lemma 5.1. ◀

5.1 Flaws in the Original Proofs from Related Work

Recall that fan-planar graphs are a special case of fan-crossing graphs, defined by admitting drawings in \mathbb{R}^2 without configuration I and II (original definition [16]), respectively without configurations I, II, and III (revised definition [17]); cf. Figure 1. The proofs in [16, 17] involve a number of statements, each carefully analysing the possible routing of edges in a fan-planar drawing. In the past decade, many papers on (generalizations of) fan-planar graphs appeared and many rely (implicitly or explicitly) on said statements. As mentioned above, a flaw in one of the statements from [16] was discovered [18]. In this section, we will describe additional issues existing in both [16] and [17], thereby outlining why the previous proofs of the density bounds for fan-crossing, weakly fan-planar, and strongly fan-planar graphs are indeed incomplete.



■ **Figure 7** Left: Illustration of [17, Corollary 5] taken from the paper. Right: A counterexample.

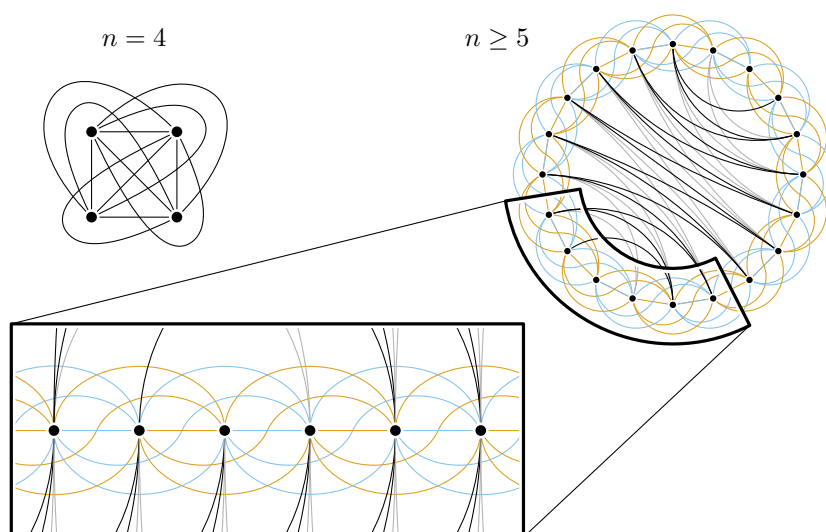
- The authors try to guarantee [17, Corollary 5] that no cell of size 4 of any *subdrawing* of a fan-planar drawing Γ contains vertices of G . In fact, if c is a \triangleleft -cell with incident vertex v and inner edge-segment of an edge u_1u_2 , and some set S of vertices lies inside c , it is suggested to move the drawing of $G[S]$ to a cell c' incident to an uncrossed edge vu_1 or vu_2 as illustrated in Figure 7(left). However, in the particular situation of Figure 7(right) with S just being a single vertex s , moving s into c' would cause edge u_1s to cross edge vw_2 , which is already crossed by the independent edge u_2w_1 ; thus losing fan-planarity.
- In a later proof [17, Lemma 11], induction is applied to the induced subdrawing of an induced subgraph G' of G . However throughout, the drawing Γ was chosen to satisfy (i) having the maximum number of planar edges, and (ii) being inclusionwise edge-maximal with that property [17, Section 3]. It is not shown or clear why the subdrawing for the induction still satisfies (i) and (ii).

6 Quasiplanar Graphs

The lower bounds for simple and non-homotopic quasiplanar graphs presented in this section are based on properties that tight examples must have that arise from a thorough reading of our upper bound proof, as provided in the full version [15]. For instance, the removal of any vertex leaves a cell of size 2 in the non-homotopic case, while in the simple case, the uncrossed edges must form a matching.

► **Theorem 6.1.** *For every $n \geq 4$, there exists a non-homotopic n -vertex connected quasiplanar graph with $8n - 20$ edges.*

Proof. For $n = 4$, let us simply refer to the construction illustrated in Figure 8(top-left).



■ **Figure 8** Illustrations of non-homotopic quasiplanar drawings with n vertices and $8n - 20$ edges. For better readability, the two zig-zag paths outside the cycle are omitted. The edge-coloring (works only for even n) just indicates four crossing-free sub-drawings, which helps to verify quasiplanarity.

- For $n \geq 5$, the desired graph G_n consists of (for illustrations refer to Figure 8(right))
- an n -vertex cycle C drawn in a non-crossing way, (n edges)
 - an edge between any two vertices at distance 2 on C drawn inside C , (n edges)
 - an edge between any two vertices at distance 2 on C drawn outside C , (n edges)
 - an edge between any two vertices at distance 3 on C , starting inside C , crossing C at distance 1.5, and ending outside C , (n edges)
 - a zig-zag path of edges drawn inside C where the endpoints of each edge have distance at least 3 on C , ($n - 5$ edges)
 - another (different) zig-zag path of edges drawn inside C where the endpoints of each edge have distance at least 3 on C , ($n - 5$ edges)
 - a zig-zag path of edges drawn outside C where the endpoints of each edge have distance at least 3 on C , ($n - 5$ edges)
 - another (different) zig-zag path of edges drawn outside C where the endpoints of each edge have distance at least 3 on C , ($n - 5$ edges)

Thereby, all edges are drawn without unnecessary crossings. For example, two edges drawn inside C cross only if the respective endpoints appear in alternating order around C .

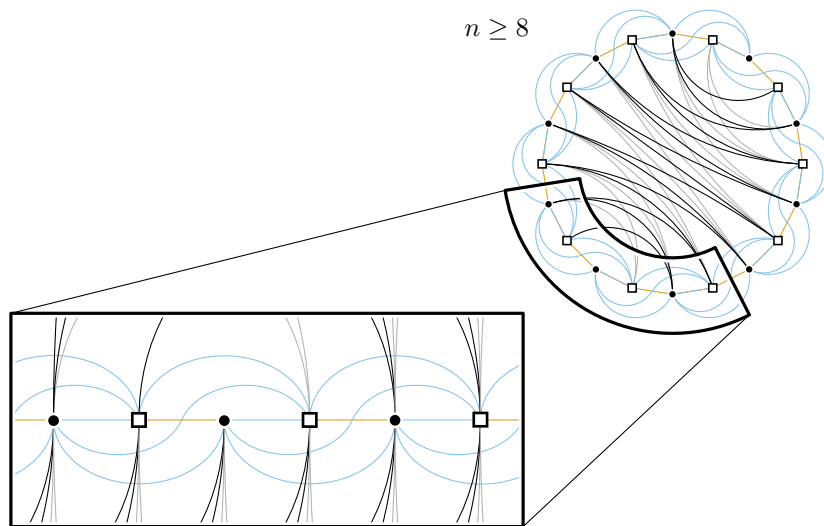
Evidently, G_n has n vertices and $8n - 20$ edges, and it is straightforward to check that the described drawing of G_n is non-homotopic and quasiplanar. ◀

► **Theorem 6.2.** *For every even $n \geq 8$, there exists a simple n -vertex quasiplanar graph with $6.5n - 20$ edges.*

Proof. Our construction is a subgraph of the corresponding graph in the proof of Theorem 6.1; see Figure 9 for an illustration. For every even $n \geq 8$, the desired simple quasiplanar graph G_n is missing all the orange edges depicted in Figure 8 except the ones at distance 1, i.e.

- the edges between any two black vertices at distance 2 on C drawn inside C , ($n/2$ edges)
- the edges between any two white vertices at distance 2 on C drawn outside C , ($n/2$ edges)
- the edges from each black vertex to its white vertex clockwise at distance 3 on C , starting inside C , crossing C at distance 1.5, and ending outside C , ($n/2$ edges)

As $n \geq 8$, the four zig-zag paths can be chosen without introducing parallel edges. Again, all edges are drawn without unnecessary crossings. For example, two edges drawn inside C cross only if the respective endpoints appear in alternating order around C .



■ **Figure 9** Illustration of simple quasiplanar drawings with n vertices and $6.5n - 20$ edges, for even $n \geq 8$. For better readability, the two zig-zag paths outside the cycle are omitted. The edge-coloring just indicates four crossing-free sub-drawings, which helps to verify quasiplanarity.

Evidently, G_n has n vertices and $6.5n - 20$ edges and it is straightforward to check that the described drawing of G_n is simple and quasiplanar. ◀

7 Concluding Remarks

Some previously known proofs already contain ideas that are similar to (parts of) our approach. Often times, this is phrased in terms of a discharging argument, instead of a direct counting. For example, some discharging steps in [4], [1], [2], and [3] (dealing with k -planar, so-called k -quasiplanar graphs, and fan-crossing graphs) directly correspond to our proof of Lemma 3.6. In these four cases, but also in [10], the total sum of all charges is $\sum_{c \in \mathcal{C}} (\|c\| - 4)$ (although stated a bit differently). In [5], which concerns 1-bend RAC-graphs, there is a charging involving the convex bends. Further, the concept of the size of a cell and the quantity $\sum_{c \in \mathcal{C}} (\|c\| - 5)$ already appear in the papers [16, 17] on fan-planar graphs. But

with the Density Formula, we have a unified approach that somewhat unveiled the essential tasks in this field of research. A valuable asset of our approach are very streamlined and clean combinatorial arguments, as well as substantially shorter proofs, as certified by the number of beyond-planar graph classes that we can treat in about 20 pages (referring to the full version [15]).

Additionally, it is straightforward to derive from the particular application of the Density Formula properties that must be fulfilled by all tight examples. For example, from our proof for k^+ -real face graphs (in the full version [15]), we immediately see that all tight examples of k^+ -real face graphs with $k \geq 3$ are planar. Similarly, from our proof for 2-planar graphs (in the full version [15]), we see that no tight example of a 2-planar graph has a ∇ -cell or \square -cell. And (together with a short calculation) our proof for quasiplanar graphs (in the full version [15]) implies that in all tight examples of simple quasiplanar graphs the planar edges form a perfect matching. Specifically, this approach of analyzing the situation in which the proof with the Density Formula is tight, allowed us to find the first tight examples for simple and non-homotopic quasiplanar graphs (cf. Theorems 6.1 and 6.2).

The only cases presented here in which upper and lower bounds still differ by an absolute constant are k -bend RAC-graphs; cf. Table 1. This is due to the fact that these are drawings in the plane \mathbb{R}^2 and the unbounded cell behaves crucially different from all other cells. It is possible to reduce our upper bounds by an absolute constant by a separate analysis of the unbounded cell, but we did not pursue this here. On the other hand, it may well be that our bounds are already optimal for k -bends RAC-drawings on the sphere \mathbb{S}^2 (for the natural definition of this concept) – they definitely are for $k = 0$.



Finally for open problems, there is a number of beyond-planar graph classes for which the exact asymptotics of their edge density is not known yet. This includes for example non-homotopic fan-crossing graphs, k -quasiplanar graphs for $k \geq 4$, and k -planar graphs for $k \geq 4$. Let us refer again to the survey [14] from 2019 for more such cases and more beyond-planar graph classes in general. Additionally, each class could be considered in a “bipartite variant” (as we do for fan-crossing graphs in the full version [15]) and/or an “outer variant” where one additionally requires that there is one cell that is incident to every vertex; see for example [7].

References



- 1 Eyal Ackerman. On the maximum number of edges in topological graphs with no four pairwise crossing edges. *Discrete & Computational Geometry*, 41(3):365–375, 2009. doi:10.1007/s00454-009-9143-9.
- 2 Eyal Ackerman. On topological graphs with at most four crossings per edge. *Computational Geometry*, 85:101574, 2019. doi:10.1016/j.comgeo.2019.101574.
- 3 Eyal Ackerman and Balázs Keszegh. The maximum size of adjacency-crossing graphs. arXiv preprint, 2023. doi:10.48550/arXiv.2309.06507.
- 4 Eyal Ackerman and Gábor Tardos. On the maximum number of edges in quasi-planar graphs. *Journal of Combinatorial Theory, Series A*, 114(3):563–571, 2007. doi:10.1016/j.jcta.2006.08.002.
- 5 Patrizio Angelini, Michael A. Bekos, Henry Förster, and Michael Kaufmann. On RAC drawings of graphs with one bend per edge. *Theoretical Computer Science*, 828-829:42–54, 2020. doi:10.1016/j.tcs.2020.04.018.
- 6 Patrizio Angelini, Michael A. Bekos, Julia Katheder, Michael Kaufmann, Maximilian Pfister, and Torsten Ueckerdt. Axis-parallel right angle crossing graphs. In Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman, editors, *31st Annual European Symposium on Algorithms (ESA 2023)*, volume 274 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 9:1–9:15, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ESA.2023.9.

- 7 Patrizio Angelini, Michael A. Bekos, Michael Kaufmann, Maximilian Pfister, and Torsten Ueckerdt. Beyond-planarity: Turán-type results for non-planar bipartite graphs. In *29th International Symposium on Algorithms and Computation (ISAAC 2018)*, volume 123, pages 28:1–28:13, 2018. doi:10.4230/LIPIcs.ISAAC.2018.28.
- 8 Karin Arikushi, Radoslav Fulek, Balázs Keszegh, Filip Moric, and Csaba D. Tóth. Graphs that admit right angle crossing drawings. *Comput. Geom.*, 45(4):169–177, 2012. doi:10.1016/J.COMGEO.2011.11.008.
- 9 Michael A. Bekos, Prosenjit Bose, Aaron Büngener, Vida Dujmović, Michael Hoffmann, Michael Kaufmann, Pat Morin, Saeed Odak, and Alexandra Weinberger. On k -planar graphs without short cycles. arXiv preprint, 2024. arXiv:2408.16085.
- 10 Carla Binucci, Giuseppe Di Battista, Walter Didimo, Seok-Hee Hong, Michael Kaufmann, Giuseppe Liotta, Pat Morin, and Alessandra Tappini. Nonplanar graph drawings with k vertices per face. In Daniël Paulusma and Bernard Ries, editors, *Graph-Theoretic Concepts in Computer Science - 49th International Workshop, WG 2023, Fribourg, Switzerland, June 28-30, 2023, Revised Selected Papers*, volume 14093 of *Lecture Notes in Computer Science*, pages 86–100. Springer, 2023. doi:10.1007/978-3-031-43380-1_7.
- 11 Franz J. Brandenburg. On fan-crossing graphs. *Theoretical Computer Science*, 841:39–49, 2020. doi:10.1016/j.tcs.2020.07.002.
- 12 Otfried Cheong, Henry Förster, Julia Katheder, Maximilian Pfister, and Lena Schlipf. Weakly and strongly fan-planar graphs. In *Graph Drawing and Network Visualization*, pages 53–68. Springer Nature Switzerland, 2023. doi:10.1007/978-3-031-49272-3_4.
- 13 Walter Didimo, Peter Eades, and Giuseppe Liotta. Drawing graphs with right angle crossings. *Theoretical Computer Science*, 412(39):5156–5166, 2011. doi:10.1016/j.tcs.2011.05.025.
- 14 Walter Didimo, Giuseppe Liotta, and Fabrizio Montecchiani. A survey on graph drawing beyond planarity. *ACM Computing Surveys*, 52(1), 2019. doi:10.1145/3301281.
- 15 Michael Kaufmann, Boris Klemz, Kristin Knorr, Meghana M. Reddy, Felix Schröder, and Torsten Ueckerdt. The density formula: One lemma to bound them all. arXiv preprint, 2024. doi:10.48550/arXiv.2311.06193.
- 16 Michael Kaufmann and Torsten Ueckerdt. The density of fan-planar graphs. arXiv preprint, 2014. arXiv:1403.6184.
- 17 Michael Kaufmann and Torsten Ueckerdt. The density of fan-planar graphs. *Electronic Journal of Combinatorics*, 29(1):P1.29, 2022. doi:10.37236/10521.
- 18 Boris Klemz, Kristin Knorr, Meghana M. Reddy, and Felix Schröder. Simplifying non-simple fan-planar drawings. *J. Graph Algorithms Appl.*, 27(2):147–172, 2023. Conference version in Proc. GD 2021 (doi:10.1007/978-3-030-92931-2_4). doi:10.7155/JGAA.00618.
- 19 János Pach. *Notes on geometric graph theory*, volume 6 of *DIMACS Series*, pages 273–285. American Mathematical Society, Providence, RI, 1991. doi:10.1090/DIMACS/006/19.
- 20 János Pach and Géza Tóth. Graphs drawn with few crossings per edge. *Combinatorica*, 17(3):427–439, 1997. doi:10.1007/BF01215922.
- 21 Csaba D. Tóth. On RAC drawings of graphs with two bends per edge. In *Graph Drawing and Network Visualization*, pages 69–77. Springer Nature Switzerland, 2023. doi:10.1007/978-3-031-49272-3_5.

Note on Min- k -Planar Drawings of Graphs

Petr Hliněný  

Masaryk University, Brno, Czech Republic

Lili Ködmön  

Eötvös Loránd University, Budapest, Hungary

Abstract

The k -planar graphs, which are (usually with small values of k such as 1, 2, 3) subject to recent intense research, admit a drawing in which edges are allowed to cross, but each one edge is allowed to carry at most k crossings. In recently introduced [Binucci et al., GD 2023] min- k -planar drawings of graphs, edges may possibly carry more than k crossings, but in any two crossing edges, at least one of the two must have at most k crossings. In both concepts, one may consider general drawings or a popular restricted concept of drawings called *simple*. In a simple drawing, every two edges are allowed to cross at most once, and any two edges which share a vertex are forbidden to cross.

While, regarding the former concept, it is for $k \leq 3$ known (but perhaps not widely known) that every general k -planar graph admits a simple k -planar drawing and this ceases to be true for any $k \geq 4$, the difference between general and simple drawings in the latter concept is more striking. We prove that there exist graphs with a min-2-planar drawing, or with a min-3-planar drawing avoiding crossings of adjacent edges, which have *no simple* min- k -planar drawings for arbitrarily large fixed k .

2012 ACM Subject Classification Mathematics of computing \rightarrow Graph theory; Theory of computation \rightarrow Computational geometry

Keywords and phrases Crossing Number, Planarity, k -Planar Graph, Min- k -Planar Graph

Digital Object Identifier 10.4230/LIPIcs.GD.2024.8

Related Version *Same Version*: <https://arxiv.org/abs/2401.11610>

Funding *Lili Ködmön*: Supported by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the ELTE TKP 2021-NKTA-62 funding scheme.

1 Introduction

One of the most popular current research directions in graph drawing is going “beyond planarity” [7, 10]. This somewhat broad direction can be described as considering drawings of graphs in which edges may cross, but the overall pattern of edge crossings is restricted, usually in a local setting. Some of the earliest examples are 1-planar graphs (every edge may have at most 1 crossing), and popular extensions nowadays include many families among which we, for example, mention k -planar- and fan-planar graphs, or k -quasiplanar graphs. These diverse classes often share some nice properties of planar graphs, such as having few edges (e.g., for 1-planar graphs [18], for k -planar graphs [14], and for 3-quasiplanar graphs [2]). However, they may differ greatly from planar graphs in other respects; for instance, recognizing 1-planar graphs is NP-complete [9, 12].

We refer to Section 2 for a precise definition of a drawing of a graph, and of simple and min- k -planar drawings. Briefly, by a *drawing* of a graph (here exclusively in the plane) we mean a topological representation in which edges (as arcs) join their end vertices (as points) and avoid passing through other vertices. Furthermore, every two distinct edges intersect finitely many times, and at most two edges intersect in one point except when it is their common end vertex. In a *simple drawing*, we additionally require that every two distinct



© Petr Hliněný and Lili Ködmön;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 8; pp. 8:1–8:10

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

edges intersect in at most one point – a crossing or a common end. (However, one has to be careful with this simplified definition when considering a simple drawing of a non-simple graph; then two parallel edges share two common ends.)

It is very common that research papers assume only simple drawings, for convenience, but it is sometimes not quite clear whether this assumption is made “without loss of generality,” or whether it is a significant restriction on the kinds of drawings considered. For instance, when studying the *crossing number* of a graph (the minimum total number of crossings over all drawings), one can quite easily restrict attention to simple drawings without loss of generality, but such restriction is not possible when studying the odd crossing number [16]. On a different note, so-called fan-planar graphs can be assumed to have a simple fan-planar drawing without loss of generality, but the proof [11] is highly nontrivial.

Consider *k-planar graphs*, which are graphs admitting a drawing in which no edge carries more than k crossings. (The same concept is also known as the local crossing number [17] or the crossing parameter [9].) The seminal paper of Pach and Tóth [14] explicitly requires simple k -planar drawings, while, e.g., “algorithmic” papers Grigoriev and Bodlaender [9] and Korzhik and Mohar [12] deal with general k -planar drawings (in fact, [12] mentions that any 1-planar drawing can be turned into a simple one) and Ackerman [1] distinguishes the cases. A recent survey on beyond planarity [7] unfortunately does not explicitly address this issue, and it mostly only implicitly restricts to results about simple drawings in this respect.

To illustrate the potential problem points (of unwary mixing general and simple k -planar drawings in research), we mention, e.g., [3] which formulates results about general 3-planar drawings, but importantly uses a lemma of [14] which, in unmentioned fact, relies on the assumption of a simple drawing. In this particular case of [3], as well as in other papers which deal with k -planar drawings for only $k \leq 3$, there is no real problem since every general k -planar graph for $k \leq 3$ admits a simple k -planar drawing, as shown already by Pach et al. [13]. On the other hand, for every $k \geq 4$ there exist k -planar graphs which have no simple k -planar drawing, e.g., Schaefer [17, Chapter 7].

Concerning the new related concept of *min- k -planar graphs*, which are graphs admitting a drawing in which every pair of crossing edges has one of the two edges with at most k crossings; the introductory paper by Binucci et al. [4, 5] requires simple drawings by the definition. However, min- k -planar drawings may also be understood in the general (non-simple) setting, and the difference between the general and the simple settings is much more striking than in the case of k -planar graphs.

Namely, we prove (Theorem 2.1) that for arbitrarily large fixed k there exist graphs that are min-2-planar without restricting to simple drawings, but which have *no simple* min- k -planar drawing. Alternatively, counterexample graphs with a min-3-planar drawing in which no two edges sharing a common vertex cross can also be constructed. In other words, the concepts of simple and general min- k -planar drawings *always significantly differ*, except in the trivial case of $k = 1$ (in which we can easily simplify any min-1-planar drawing, cf. Proposition 2.2), and they differ for $k \geq 3$ even if we forbid general drawings in which two edges with a common vertex cross.

In the course of proving this result, we develop a technical tool (Lemma 3.2) which suitably constrains possible min- k -planar drawings of graphs within a rigid “frame”, and we suggest this tool can be useful in further research of the properties of min- k -planar graphs.

2 Min- k -planar Drawings and Graphs

We consider general finite undirected graphs (with possible parallel edges or loops), and say that a graph is *simple* if it has no parallel edges and no loops.

A *drawing* \mathcal{G} of a graph G in the Euclidean plane \mathbb{R}^2 is a function that maps each vertex $v \in V(G)$ to a distinct point $\mathcal{G}(v) \in \mathbb{R}^2$ and each edge $e = uv \in E(G)$ to a simple curve (non-self-intersecting Jordan arc) $\mathcal{G}(e) \subset \mathbb{R}^2$ with the ends $\mathcal{G}(u)$ and $\mathcal{G}(v)$. We require that $\mathcal{G}(e)$ is disjoint from $\mathcal{G}(w)$ for all $w \in V(G) \setminus \{u, v\}$ and that $\mathcal{G}(e) \cap \mathcal{G}(f)$ is finite for all $e \neq f \in E(G)$. In a slight abuse of notation we identify a vertex v with its image $\mathcal{G}(v)$ and an edge e with $\mathcal{G}(e)$. An intersection (possibly tangential) of two edges e and f other than a common end vertex, in other words a point $x \in \mathcal{G}(e) \cap \mathcal{G}(f)$, is called a *crossing* (of e and f), and the pair e, f is said to *cross* in x . A *planar drawing* is a drawing with no crossings.

For further concepts, the definition of a “crossing” needs a careful clarification. If $k > 2$ edges together cross in one point, then this situation, strictly, counts as $\binom{k}{2}$ crossings of pairs of these edges (see, e.g., [6]). Since a slight local perturbation of the involved edges avoids common crossings of triples without adding any new crossing, for our purposes, we may simply discard the possibility that $k > 2$ edges together cross in one point, which is a common approach in literature. Therefore, without loss of generality, we require that, in every drawing, *no three edges cross* in the same point, unless stated otherwise.

One may, likewise, deal with possible tangential crossings in a drawing, which can be discarded by a slight local perturbation as well, and so exclude tangential crossings by the definition (as many papers do for convenience), but here we stay on the more general side and do not exclude them.

In a *simple drawing* \mathcal{G} (otherwise known as a *good drawing* [8, 17], but we wish to adhere to the recent terminology), crossings are allowed, but (again) no three edges cross in the same point, no two edges have more than one crossing in common, and no two adjacent edges (i.e., with a common end vertex) cross. Hence, in our setting, a simple drawing \mathcal{G} is defined such that, for every $e \neq f \in E(G)$, we have $|\mathcal{G}(e) \cap \mathcal{G}(f)| \leq 1$, except that $|\mathcal{G}(e) \cap \mathcal{G}(f)| = 2$ when e and f are parallel edges (the latter is irrelevant for simple graphs).

A drawing \mathcal{G} is *k-planar* if no edge contains more than k crossings. A drawing \mathcal{G} is *min-k-planar* if, for every two crossing edges e and f in \mathcal{G} , one (or both) of e, f has no more than k crossings. If an edge e has more than k crossings in a min- k -planar drawing \mathcal{G} , then e is called *heavy* in \mathcal{G} (hence two heavy edges cannot cross each other in a min- k -planar drawing).

A graph G is *min-k-planar* if G admits a min- k -planar drawing. Moreover, a graph G is *simply min-k-planar* if G admits a min- k -planar drawing \mathcal{G} such that \mathcal{G} is a simple drawing. Since some papers shortly speak about min- k -planar graphs while requiring simple drawings, to avoid further confusion, we will call min- k -planar graph without the additional requirement of a simple drawing as *general min-k-planar*.

A careful distinction between general min- k -planar graphs and simply min- k -planar graphs is indeed necessary for all $k > 1$, as we are going to prove here.

► **Theorem 2.1** (Proof in Section 3).

- a) For every $k \geq 2$, there exists a simple graph H_k which is general min-2-planar, but H_k has no simple min- k -planar drawing.
- b) Moreover, for all $k \geq 3$, there exists a graph H'_k which has a general min-3-planar drawing in which no two adjacent edges cross, but, again, H'_k has no simple min- k -planar drawing.

To complement Theorem 2.1, we resolve the remaining trivial cases in Proposition 2.2.

► **Proposition 2.2.**

- a) Every general min-1-planar graph admits a simple min-1-planar drawing (hence is simply min-1-planar).
- b) Every graph with a min-2-planar drawing in which no two adjacent edges cross also admits a simple min-2-planar drawing.



■ **Figure 1** An illustration of Proposition 2.2 a); up to symmetry between e and f (with common end vertex x), the edge e carries no other crossing than the point y , and so one can draw an uncrossed arc e'_1 tightly along the segment $e_1 \subseteq e$ from x to y . When redrawing from e to e'' and from f to f'' (using e'_1), the crossing at y is eliminated and no new crossings are added to any edge in the picture.



■ **Figure 2** An illustration of Proposition 2.2 b); now we have two crossings x and y of the same pair e and f of edges, and there are no more crossings on e . Similarly to Figure 1, when redrawing from e to e'' and from f to f'' (using the uncrossed arc e'_1), at least one of the crossings at x or y is eliminated and no new crossings are added to any edge in the picture.

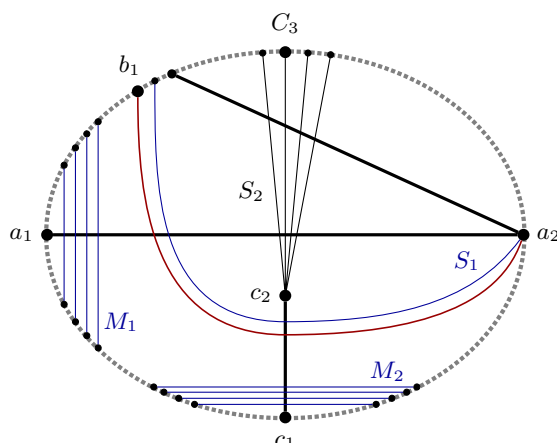
Proof.

a) Let G be a general min-1-planar graph and take a min-1-planar drawing \mathcal{G} of G . For simplicity, in this proof, we write e (an edge) also for the point set $\mathcal{G}(e)$ in the drawing. We may assume that \mathcal{G} minimizes the number of edge pairs which violate the simplicity of the drawing, i.e., edge pairs which share a vertex and cross, since a pair cannot cross twice in a min-1-planar drawing. Let $e, f \in E(G)$ be such a violating edge pair in \mathcal{G} ; hence the intersection of e and f contains two distinct points $x, y \in e \cap f$ where x is a vertex and y a crossing. Moreover, we may assume up to symmetry that e has no crossing other than with f .

Let us denote by e_1 and f_1 the subarcs of e and f , respectively, in \mathcal{G} with the ends x, y . Since e_1 is internally crossing-free, there exists an arc e'_1 from x to y drawn sufficiently close to e_1 such that e'_1 is disjoint from \mathcal{G} except in x, y . We replace f in \mathcal{G} with $f' := (f \setminus f_1) \cup e'_1$; this new drawing \mathcal{G}' of G is again min-1-planar since no crossings of \mathcal{G} have been affected. If y is a tangent of the edges e and f' (not crossing transversely at y), then a local perturbation of e around y simply removes this tangential crossing, and so we eliminate one violating edge pair from \mathcal{G} in \mathcal{G}' . Otherwise, if e and f' cross transversely at y , we instead replace e with $e'' := (e \setminus e_1) \cup e'_1$ and f with $f'' := (f \setminus f_1) \cup e_1$ in \mathcal{G} . See Figure 1. The new drawing \mathcal{G}'' of G is min-1-planar, too, and y is now a tangent of the edges e'' and f'' , which can be eliminated as previously.

b) We let \mathcal{G} be a min-2-planar drawing of a graph G in which no two adjacent edges cross. Again, we may assume that \mathcal{G} minimizes the number of edge pairs which violate the simplicity of the drawing – these are now the pairs which mutually cross exactly twice. Let $e, f \in E(G)$ be such a violating edge pair in \mathcal{G} ; hence the intersection of e and f contains two distinct points $x, y \in e \cap f$ which are crossings.

Since \mathcal{G} is min-2-planar, up to symmetry, the edge e has no other crossings than x, y . Denote by e_1 and f_1 the subarcs of e and f , respectively, in \mathcal{G} with the ends x, y . Since e_1 is internally crossing-free, there exists an arc e'_1 from x to y drawn sufficiently close to e_1



■ **Figure 3** An anchored general min-2-planar drawing of the anchored graph (G_k, A_k) (here for $k = 3$, with the heavy edges drawn thick) as used in the proof of Lemma 2.3. With general $k \geq 2$, at least $k - 1$ of the edges of the star S_1 with center a_2 follow the depicted route of the red edge a_2b_1 , while crossing the edge a_2a_1 .

such that e'_1 is disjoint from \mathcal{G} except in x, y . We define new arcs of the edges e and f in \mathcal{G} as follows; $e' = e$ and $f' := (f \setminus f_1) \cup e'_1$, and $e'' := (e \setminus e_1) \cup e'_1$ and $f'' := (f \setminus f_1) \cup e_1$. Clearly, for at least one of the pairs e', f' or e'', f'' , some of the points x, y is now a tangential crossing, which can be eliminated as previously. See Figure 2. The new drawing \mathcal{G}' of G is min-2-planar, too, and no new crossings have been created between any edge pairs.

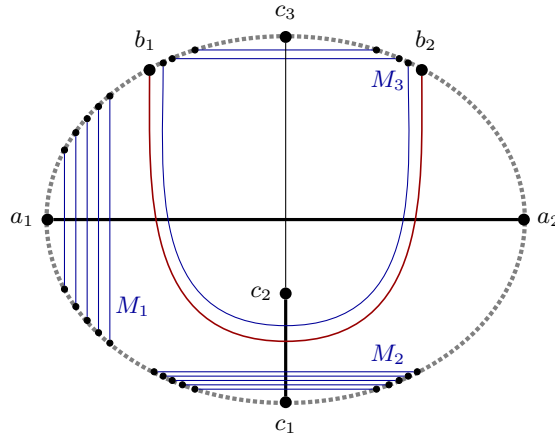
Altogether, we have in each case decreased the number of violating edge pairs, contradicting our minimal choice of the drawing \mathcal{G} . ◀

To prove Theorem 2.1, we use the following intermediate result formulated for so-called anchored graphs, which captures the essence of Theorem 2.1. An *anchored graph* is a pair (G, A) where $A \subseteq V(G)$ is an ordered tuple of vertices. An *anchored drawing* of (G, A) in the unit disk $D \subseteq \mathbb{R}^2$ is a drawing $\mathcal{G} \subseteq D$ of G such that \mathcal{G} intersects the boundary of D precisely in the points of A (the *anchors*) in this clockwise order. We naturally extend the adjective *anchored* to min- k -planar drawings. We prove:

▶ **Lemma 2.3.** *For every $k \geq 2$, there exists a simple anchored graph (G_k, A_k) which has an anchored general min-2-planar drawing, but (G_k, A_k) has no anchored simple min- k -planar drawing. Furthermore, for any $k \geq 3$, there exists a simple anchored graph (G'_k, A'_k) which has an anchored general min-3-planar drawing in which no pair of adjacent edges cross, but (G'_k, A'_k) has no anchored simple min- k -planar drawing.*

Proof. We define the anchored graph (G_k, A_k) as depicted in Figure 3. G_k is a disjoint union of two induced matchings M_1 and M_2 of $k + 1$ edges each, and of two induced stars S_1 and S_2 , where S_1 has center a_2 and $k + 1$ leaves including a_1, b_1 and S_2 has center c_2 and $k + 2$ leaves in the set $C_3 \cup \{c_1\}$ where $|C_3| = k + 1$. The anchor set is $A_k = V(G_k) \setminus \{c_2\}$ ordered as in Figure 3. Figure 3 also shows an anchored min-2-planar drawing of (G_k, A_k) (however, a_1a_2 crosses b_1a_2 there).

Assume, for a contradiction, that there exists an anchored simple min- k -planar drawing \mathcal{G} of (G_k, A_k) . By Jordan Curve Theorem, the edge a_1a_2 has to cross all $k + 1$ edges of M_1 , and so a_1a_2 is heavy in \mathcal{G} . If the edge c_1c_2 was crossing a_1a_2 , then, again using Jordan



■ **Figure 4** An anchored general min-3-planar drawing of the anchored graph (G'_k, A'_k) (here for $k = 4$, with the heavy edges drawn thick) as used in the proof of Lemma 2.3. For general $k \geq 3$, at least $k - 2$ of the edges of the matching M_3 follow the route of the red edge b_1b_2 . These edges cross the heavy edge a_1a_2 twice, and so the drawing is neither simple nor general min-2-planar.

Curve Theorem, some edge of M_2 would cross twice with a_1a_2 – which is not simple, or c_1c_2 would cross all edges of M_2 and be heavy as well – which contradicts \mathcal{G} being min-2-planar. Therefore, all $k + 1$ edges from c_2 to C_3 have to cross a_1a_2 and are non-heavy.

Consequently, none of the edges between c_2 and C_3 can cross all $k + 1$ edges of S_1 . By Jordan Curve Theorem, hence, some edge of S_1 , say b_1a_2 (as in the picture) crosses c_1c_2 . However, again by Jordan Curve Theorem, from this crossing point to b_1 , the edge b_1a_2 has to cross the edge a_1a_2 , contradicting the assumption that \mathcal{G} is simple.

For $k \geq 3$, we define the anchored graph (G'_k, A'_k) as depicted (with a min-3-planar drawing) in Figure 4. The definition is analogous to that of (G_k, A_k) , except that the star S_1 is now replaced with an induced matching M_3 of $k + 1$ edges (including the edge a_1a_2), and the star S_2 is replaced with a path (c_1, c_2, c_3) of length two.

A proof that (G'_k, A'_k) has no anchored simple min- k -planar drawing starts with the same steps as the previous one. We get that the edge a_1a_2 is heavy and not crossing c_1c_2 , and so the edge c_2c_3 has to cross a_1a_2 and is non-heavy. Then each of the k edges of M_3 has to cross c_2c_3 or c_1c_2 , but not all may cross c_2c_3 which would become heavy. Consequently, some edge, say $b_1b_2 \in E(M_3)$, crosses the edge c_1c_2 . Since c_1c_2 is separated from $V(M_3)$ by a_1a_2 , by Jordan Curve Theorem, the edge b_1b_2 has to cross a_1a_2 twice, again a contradiction. ◀

3 Technical Proofs

In this section we give the technical details leading to the proof of Theorem 2.1. In a nutshell, we are going to construct a “frame graph” which enforces a predefined anchored subdrawing of a given anchored graph, and then we apply this construction to the graphs of Lemma 2.3.

A t -amplification of a graph G is the graph obtained from G by replacing every edge $e = xy \in E(G)$ with a new collection of t pairwise internally disjoint paths of length 2 from x to y . Observe that any t -amplification of a planar graph G is again planar.

► **Lemma 3.1.** *For every planar graph G , there exists an integer $t = t(G, k, w)$, depending on G and integers k and w , such that the following holds. Every general min- k -planar drawing \mathcal{G} of the t -amplification G^t of G contains a planar subdrawing \mathcal{G}' which is isomorphic to the w -amplification G^w of G .*

Proof. Let the length-2 paths (with ends in $V(G)$) from the definition of the t -amplification G^t be called *double edges* of G^t . We start with an easy claim:

- (*) Let $a_1, a_2, b_1, b_2 \in V(G)$ (possibly $a_1 = a_2$ or $b_1 = b_2$), and let D_1 and D_2 be sets of double edges of G^t where those of D_i have ends a_i and b_i , such that each double edge of D_1 crosses each one of D_2 in the drawing \mathcal{G} . If $|D_1|, |D_2| \geq 2k + 1$, then \mathcal{G} is not min- k -planar.

Indeed, since every double edge $P \in D_1$ crosses all of D_2 , one of the two edges of P has at least $k + 1$ crossings and is heavy. The same holds for every $P' \in D_2$. Without loss of generality, assume now $|D_1| = |D_2| = 2k + 1$. If every crossing occurring between D_1 and D_2 involved a non-heavy edge, then the total number of these crossings would be bounded (counted along all non-heavy edges) by at most $|D_1| \cdot k + |D_2| \cdot k = k(4k + 2) < (2k + 1)^2 = |D_1| \cdot |D_2|$, which is impossible. Hence some two heavy edges cross and \mathcal{G} is not min- k -planar.

We continue with a Ramsey-type argument.¹ For any $f \in E(G)$ and any t_1 , by Ramsey Theorem, there is a sufficiently large t such that the following holds. Among the t double edges replacing f in the t -amplification G^t , there exist $2(2k + 1)$ pairwise crossing ones or t_1 pairwise non-crossing ones. If the former (crossing) case happens, then by (*) we get a contradiction to the assumption of this lemma, that \mathcal{G} is min- k -planar. Therefore, the non-crossing case happens, and we apply the same argument concurrently to all edges of G . This way we get a subdrawing $\mathcal{G}_1 \subseteq \mathcal{G}$ of a t_1 -amplification G^{t_1} of G such that the collection of paths replacing any edge of G is alone crossing-free in \mathcal{G}_1 .

Likewise, for any pair of edges $f, f' \in E(G)$ and any t_2 , by the bipartite Ramsey Theorem, there is a sufficiently large t_1 such that we get t_2 double edges replacing f and another t_2 replacing f' in G^{t_1} , which are pairwise noncrossing in \mathcal{G}_1 , or we again get a contradiction via (*). Applying this argument concurrently to disjoint pairs of edges of G , we obtain a subdrawing $\mathcal{G}_2 \subseteq \mathcal{G}_1$ of a t_2 -amplification of G . We iterate this argument until we exhaust all pairs of edges of G . Starting from a sufficiently large t , the resulting drawing \mathcal{G}' of this iterative process is a w -amplification of G , and no two double edges cross in \mathcal{G}' , as desired. ◀

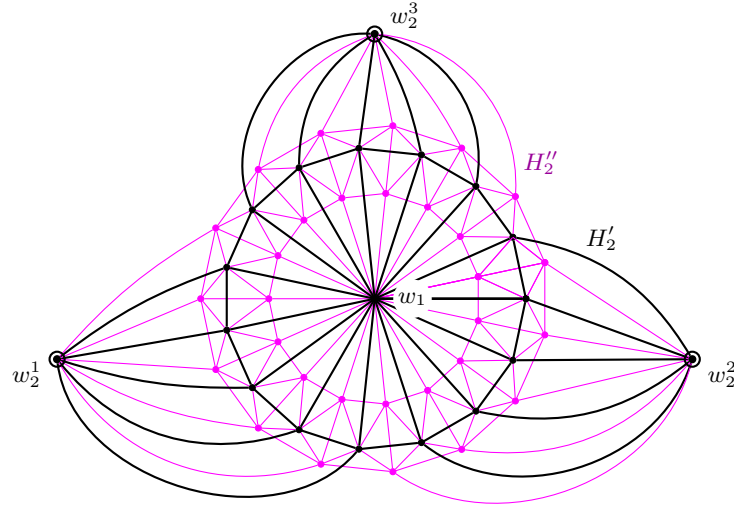
► **Lemma 3.2.** *For any integers a, k and simple graph G with an ordered subset $A \subseteq V(G)$, $|A| = a$, there exists a simple anchored graph (H, A) disjoint from G except in the anchors A , such that the following hold:*

- a) (H, A) has an anchored simple min-1-planar drawing.
- b) In every general min- k -planar drawing \mathcal{H} of $H \cup G$, the subdrawing $\mathcal{G} \subseteq \mathcal{H}$ of G is (spherically) homeomorphic to an anchored drawing of (G, A) or its mirror image.

Proof. For a start, we ignore all components of $G - A$ which attach to A in at most one vertex; their possible subdrawings can always be added homeomorphically and without further crossings to the rest of an anchored drawing of (G, A) . Let ℓ be the maximum finite(!) distance in G between a vertex of A and a (reachable) vertex of $V(G) \setminus A$. Let $d = a(2k\ell + 2k + 1)$.

We begin with the graph H_0 which is a double wheel of d spokes, i.e., a graph made from the cycle C_d by adding two central vertices w_1, w_2 adjacent to all cycle vertices. Let H_0^* be the planar dual of H_0 , and H_1 be constructed from $H_0 \cup H_0^*$ by adding extra edges $uv \in E(H_1)$ for every pair $u \in V(H_0)$ and $v \in V(H_0^*)$ such that u is a vertex incident to the face of (the unique planar drawing of) H_0 represented by v . We construct the graph H_2 by

¹ One can get better estimates of t using homotopy-based arguments, but that would not make our result stronger and we prefer simplicity of brute-force Ramsey here.



■ **Figure 5** An illustration of the graph H_2 from the proof of Lemma 3.2. The three black circled vertices are the designated anchors A (here $A = \{w_1^1, w_2^2, w_2^3\}$). The subgraph H'_2 is in thick black and H''_2 in magenta colour. All magenta edges get t -amplified in the construction of H , and consequently, the black edges forced to cross them will be heavy in any min- k -planar drawing of H by Lemma 3.1.

splitting the central vertex w_2 into a vertices, each incident with $2k\ell + 2k + 1$ consecutive spokes of H_0 and with naturally corresponding extra edges of H_1 . See Figure 5. The vertices split from w_2 , in their clockwise order, define the anchor set A .

Furthermore, let $H'_2 \subseteq H_2$ be the subgraph formed by the original edges of H_0 , and $H''_2 = H_2 \setminus E(H'_2)$. Observe one important property of planar H''_2 ; for every planar drawing of H''_2 and each edge $uv \in E(H''_2)$, the points u and v are separated by a cycle (e.g., the dual cycle of $u \neq w_2$) in H''_2 . Let now $t = t(H''_2, k, k + 1)$ be as in Lemma 3.1. We construct $H := H'_2 \cup H_2^t$ where H_2^t is the t -amplification of H''_2 . Obviously, (H, A) has an anchored simple min-1-planar drawing, e.g., one following Figure 5.

Consider now a min- k -planar drawing \mathcal{H} of $H \cup G$, and denote by $\mathcal{H}_0 \subseteq \mathcal{H}$ the subdrawing of H_2^t . By Lemma 3.1, \mathcal{H}_0 contains a planar subdrawing of a $(k + 1)$ -amplification of H''_2 . By the mentioned property of H''_2 , every edge of H'_2 thus has to cross at least $k + 1$ edges of H_2^t and is heavy. In particular, the subdrawing of H'_2 within \mathcal{H} is planar (and so has to look as in Figure 5).

We pick an anchor vertex $a \in A$ and observe that a has $2k\ell + 2k + 1$ disjoint length-2 paths to w_1 in H'_2 . Let P_a denote the “middle” one of them. Assume that an edge $f = xy \in E(G)$ crosses P_a in the drawing \mathcal{H} . Then each end x or y has distance at most ℓ in G to a vertex $b \in A \setminus \{a\}$ by our choice of ℓ , and so $\leq \ell + 1$ edges of a path from f (and including f) to b have to cross together $k\ell + k + 1$ paths of H'_2 between a and w_1 , by Jordan Curve Theorem. Therefore, some edge of this path of G must cross at least $k + 1$ edges of H'_2 , and we get pair(s) of crossing heavy edges, which is a contradiction.

Therefore, in \mathcal{H} there are H'_2 -paths from each anchor vertex $a \in A$ to central w_1 which are not crossed by any edge of the subdrawing of G . This means that the subdrawing of G within \mathcal{H} can be homeomorphically deformed in the sphere so that the vertices of A will be drawn on a disk boundary and the rest inside, as required by an anchored drawing. The correct cyclic order (up to a mirror image) of the anchors A on the disk boundary is ensured by the cycle of H'_2 on the neighbours of w_1 (the rim cycle of the starting wheel). ◀

We are now ready to finish the proof of the main result.

Proof of Theorem 2.1. We take the anchored graph (G_k, A_k) , $k \geq 2$, and the anchored graph (G'_k, A'_k) , $k \geq 3$, from Lemma 2.3 for (a) and (b), respectively, and plug them into Lemma 3.2 as G . The rest is an immediate consequence of the previous statements. ◀

4 Conclusions

Many papers in the graph drawing area deal with only simple drawings, either as a convenient simplification of the general case, or as a strict condition in the definition. However, some works are not clear in distinguishing between the two situations and this can bring troublesome problems in the future. In this regard, we would like to mention, for instance, a similar past confusion thoroughly studied in the remarkable paper of Pach and Tóth [15] (entitled “Which crossing number is it anyway?”).

We have demonstrated that a careful distinction (between simple / non-simple drawings) is surely necessary when considering the recent min- k -planar graphs. Our note brings a natural open question about which of the published results of [4] concerning simply min- k -planar graphs with $k \geq 2$ remain valid also for general min- k -planar graphs.

References

- 1 Eyal Ackerman. On topological graphs with at most four crossings per edge. *Comput. Geom.*, 85:101574, 2019. doi:10.1016/J.COMGEO.2019.101574.
- 2 Pankaj K. Agarwal, Boris Aronov, János Pach, Richard Pollack, and Micha Sharir. Quasi-planar graphs have a linear number of edges. *Comb.*, 17(1):1–9, 1997. doi:10.1007/BF01196127.
- 3 Michael A. Bekos, Michael Kaufmann, and Chrysanthi N. Raftopoulou. On the density of non-simple 3-planar graphs. In *GD*, volume 9801 of *Lecture Notes in Computer Science*, pages 344–356. Springer, 2016. doi:10.1007/978-3-319-50106-2_27.
- 4 Carla Binucci, Aaron Büngener, Giuseppe Di Battista, Walter Didimo, Vida Dujmovic, Seok-Hee Hong, Michael Kaufmann, Giuseppe Liotta, Pat Morin, and Alessandra Tappini. Min- k -planar drawings of graphs. In *GD (1)*, volume 14465 of *Lecture Notes in Computer Science*, pages 39–52. Springer, 2023. doi:10.1007/978-3-031-49272-3_3.
- 5 Carla Binucci, Aaron Büngener, Giuseppe Di Battista, Walter Didimo, Vida Dujmovic, Seok-Hee Hong, Michael Kaufmann, Giuseppe Liotta, Pat Morin, and Alessandra Tappini. Min- k -planar drawings of graphs. *CoRR*, abs/2308.13401v3, 2024. doi:10.48550/arXiv.2308.13401.
- 6 Sergio Cabello and Bojan Mohar. Adding one edge to planar graphs makes crossing number and 1-planarity hard. *SIAM J. Comput.*, 42(5):1803–1829, 2013. doi:10.1137/120872310.
- 7 Walter Didimo, Giuseppe Liotta, and Fabrizio Montecchiani. A survey on graph drawing beyond planarity. *ACM Comput. Surv.*, 52(1):4:1–4:37, 2019. doi:10.1145/3301281.
- 8 P. Erdős and R. K. Guy. Crossing number problems. *The American Mathematical Monthly*, 80(1):52–58, 1973. doi:10.2307/2319261.
- 9 Alexander Grigoriev and Hans L. Bodlaender. Algorithms for graphs embeddable with few crossings per edge. *Algorithmica*, 49(1):1–11, 2007. doi:10.1007/S00453-007-0010-X.
- 10 Seok-Hee Hong and Takeshi Tokuyama, editors. *Beyond Planar Graphs, Communications of NII Shonan Meetings*. Springer, 2020. doi:10.1007/978-981-15-6533-5.
- 11 Boris Klemz, Kristin Knorr, Meghana M. Reddy, and Felix Schröder. Simplifying non-simple fan-planar drawings. *J. Graph Algorithms Appl.*, 27(2):147–172, 2023. doi:10.7155/JGAA.00618.
- 12 Vladimir P. Korzhik and Bojan Mohar. Minimal obstructions for 1-immersions and hardness of 1-planarity testing. In *GD*, volume 5417 of *Lecture Notes in Computer Science*, pages 302–312. Springer, 2008. doi:10.1007/978-3-642-00219-9_29.

8:10 Note on Min- k -Planar Drawings of Graphs

- 13 János Pach, Rados Radoicic, Gábor Tardos, and Géza Tóth. Improving the crossing lemma by finding more crossings in sparse graphs. *Discret. Comput. Geom.*, 36(4):527–552, 2006. doi:10.1007/S00454-006-1264-9.
- 14 János Pach and Géza Tóth. Graphs drawn with few crossings per edge. *Comb.*, 17(3):427–439, 1997. doi:10.1007/BF01215922.
- 15 János Pach and Géza Tóth. Which crossing number is it anyway? *J. Comb. Theory, Ser. B*, 80(2):225–246, 2000. doi:10.1006/JCTB.2000.1978.
- 16 Michael J. Pelsmajer, Marcus Schaefer, and Daniel Stefankovic. Odd crossing number and crossing number are not the same. *Discret. Comput. Geom.*, 39(1-3):442–454, 2008. doi:10.1007/S00454-008-9058-X.
- 17 M. Schaefer. *Crossing Numbers of Graphs*. Discrete mathematics and its applications. CRC Press, Taylor & Francis Group, 2017.
- 18 Von H. Schumacher. Zur struktur 1-planarer graphen. *Mathematische Nachrichten*, 125(1):291–300, 1986. doi:10.1002/mana.19861250122.

Partitioning Complete Geometric Graphs on Dense Point Sets into Plane Subgraphs

Adrian Dumitrescu  

AlgoResearch L.L.C., Milwaukee, WI, USA

János Pach  

Alfréd Rényi Institute of Mathematics, Budapest, Hungary

EPFL, Lausanne, Switzerland

Abstract

A *complete geometric graph* consists of a set P of n points in the plane, in general position, and all segments (edges) connecting them. It is a well known question of Bose, Hurtado, Rivera-Campo, and Wood, whether there exists a positive constant $c < 1$, such that every complete geometric graph on n points can be partitioned into at most cn plane graphs (that is, noncrossing subgraphs). We answer this question in the affirmative in the special case where the underlying point set P is *dense*, which means that the ratio between the maximum and the minimum distances in P is of the order of $\Theta(\sqrt{n})$.

2012 ACM Subject Classification Mathematics of computing → Discrete mathematics; Theory of computation → Randomness, geometry and discrete structures

Keywords and phrases Convexity, complete geometric Graph, crossing Family, plane Subgraph

Digital Object Identifier 10.4230/LIPIcs.GD.2024.9

Funding Work on this paper was supported by ERC Advanced Grant no. 882971 “GeoScape.” Work by the second named author was partially supported by NKFIH (National Research, Development and Innovation Office) grant K-131529.

Acknowledgements We thank the anonymous reviewers for carefully reading the manuscript.

1 Introduction

A set of points in the plane is said to be: (i) in *general position* if no 3 points are collinear; and (ii) in *convex position* if none of the points lies in the convex hull of the other points. For a set A of n points in the plane, consider the ratio

$$D(A) = \frac{\max\{\text{dist}(a, b) : a, b \in A, a \neq b\}}{\min\{\text{dist}(a, b) : a, b \in A, a \neq b\}},$$

where $\text{dist}(a, b)$ is the Euclidean distance between points a and b . We assume throughout this paper and without loss of generality that the minimum pairwise distance is 1. In this case $D(A)$ is the diameter of A . A standard volume argument shows that if A has n points, then $D(A) \geq \alpha_0 n^{1/2}$, with

$$\alpha_0 := 2^{1/2} 3^{1/4} \pi^{-1/2} \approx 1.05, \tag{1}$$

provided that n is large enough; see [26, Prop. 4.10]. On the other hand, a $\sqrt{n} \times \sqrt{n}$ section of the integer lattice shows that this bound is tight up to a constant factor.

Given n points in the plane, in general position, the graph obtained by connecting certain point-pairs by straight-line segments is called a *geometric graph* G . If no two segments (edges) of G cross each other, then G is a *plane graph*. A graph of the form K_{1s} , where $s \geq 0$, is a special plane graph, called a *star*; in particular, a single vertex is a star with no leaves. A graph in which every connected component is a star is called a *star-forest*; see, e.g., [3].



© Adrian Dumitrescu and János Pach;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 9; pp. 9:1–9:10

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

9:2 Partitioning Complete Geometric Graphs into Plane Subgraphs

Obviously, every complete geometric graph of n vertices can be decomposed into $n-1$ plane stars. In the present note, we address the following problem of Bose, Hurtado, Rivera-Campo, and Wood [7], raised almost 20 years ago.

► **Problem 1** ([7]). *Does there exist a positive constant $c < 1$ such that every complete geometric graph on n vertices can be partitioned into at most cn plane subgraphs?*

An n -element point set A satisfying the condition $D(A) \leq \alpha n^{1/2}$, for some constant $\alpha \geq \alpha_0$, is said to be α -dense; see the works of Edelsbrunner, Valtr, and Welzl [10], Kovács and Tóth [15], and Valtr [27]. (Note, the larger α becomes, the “less dense” the set gets.)

Here, we solve Problem 1 for dense point sets.

► **Theorem 2.** *Let A be an α -dense point set of n points in general position in the plane, and let $K_n = K_n[A]$ denote the complete geometric graph induced by A . Then (the edge set of) K_n can be decomposed into at most cn plane subgraphs, where $c = c(\alpha) < 1$ is a constant. Specifically, we have*

$$c(\alpha) \leq 1 - \Omega(\alpha^{-12}). \quad (2)$$

Each of these plane graphs is either a star or a plane union of two stars.

Let A be a randomly and uniformly distributed set of n points in the unit square. With probability tending to 1, as $n \rightarrow \infty$, the order of magnitude of the minimum distance in A will be much smaller than $n^{-1/2}$. Therefore, $D(A)$ will be larger than $\alpha n^{1/2}$, for every α , provided that n is sufficiently large, and Theorem 2 cannot be applied to $K_n[A]$. Nevertheless, A almost surely contains a linear-size α' -dense subset, for a suitable constant α' , and we can easily deduce the following statement, which is also implied by a result of Valtr [28, Thm. 14] in conjunction with Lemma 5 below. In Section 3 we provide an alternative proof of Corollary 3.

► **Corollary 3.** *Let A be a set of n random points uniformly distributed in $[0, 1]^2$, and let $n \rightarrow \infty$. There exists an absolute constant $c < 1$ such that, with probability tending to 1, the complete geometric graph induced by A can be decomposed into at most cn plane subgraphs.*

There is an intimate relationship between the above problem and another old question in combinatorial geometry, due to Aronov, Erdős, Goddard, Kleitman, Klugerman, Pach, and Schulman [5]. Two segments are said to *cross* each other if they do not share an endpoint and they have an interior point in common.

► **Problem 4** ([5]). *Does there exist a positive constant $c < 1/2$ such that every complete geometric graph on n vertices has cn pairwise crossing edges?*

In the general case, Pach, Rubin, and Tardos [19] established the existence of at least $n/2^{O(\sqrt{\log n})} = n^{1-o(1)}$ pairwise crossing edges. For dense point sets, a better, but still sublinear, lower bound was established by Valtr[28]. From the other direction, Aichholzer, Kynčl, Scheucher, Vogtenhuber, and Valtr [1] constructed n -element point sets that do not contain more than $8\lceil \frac{n}{41} \rceil$ pairwise crossing edges; see also [13].

Problems 1 and 4 are connected by the following simple, but important finding of Bose et al. [7].

► **Lemma 5** ([7]). *If a complete geometric graph K_n of n vertices has p pairwise crossing edges, then K_n can be partitioned into $n-p$ plane trees and, hence, into $n-p$ plane subgraphs.*

In view of this statement, a positive answer to Problem 4 would immediately imply our Theorem 2. Lacking such an answer, we need to take a different approach, which is described in the next section.

All point sets appearing in this note are in general position, and the logarithms are in base 2. For any triple of points a, b, c , let Δabc denote the triangle with vertices a, b, c .

2 Proof of Theorem 2

In this section we prove Theorem 2. We start with a basic observation.

► **Lemma 6.** *Let A be a set of n points in general position in the plane, and let $B \subseteq A$ where $|B| = b$. Suppose that the complete geometric graph $K_b[B]$ induced by B can be decomposed into $b - p$ plane subgraphs, for some $p \geq 1$. Then $K_n[A]$, the complete geometric graph induced by A , can be decomposed into $n - p$ plane subgraphs.*

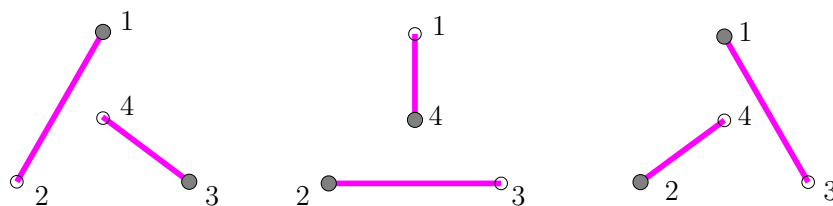
Proof. Consider the $n - b$ stars centered at points in $A \setminus B$ together with the $b - p$ plane subgraphs in the decomposition of $K_b[B]$, and delete duplicate edges. ◀

In view of Lemma 6, to establish Theorem 2, it is enough to find a large subset $B \subseteq A$ that can be decomposed into relatively few plane graphs. Instead of Lemma 5, we use the following result, whose proof is included for completeness.

► **Lemma 7** (Pach, Saghafian, and Schnider [20]). *Let $B = \bigcup_{i=1}^4 B_i$ be a set of $4m$ points in general position in the plane, where $|B_1| = |B_2| = |B_3| = |B_4| = m$, such that for every choice $p_i \in B_i$, for $i = 1, 2, 3, 4$, we have that p_4 lies inside the convex hull of $\{p_1, p_2, p_3\}$. Then the complete geometric graph $K_{4m}[B]$ induced by B can be decomposed into at most $3m$ plane subgraphs.*

Proof. We decompose the complete geometric graph $K_{4m}[B]$ into $3m$ plane star-forests, which come in three families; see Fig. 1:

1. all stars emanating from points in B_1 connecting to all points in B_1 and B_2 together with all stars emanating from points in B_3 connecting to all points in B_3 and B_4
2. all stars emanating from points in B_2 connecting to all points in B_2 and B_3 together with all stars emanating from points in B_4 connecting to all points in B_4 and B_1
3. all stars emanating from points in B_1 connecting to all points in B_1 and B_3 together with all stars emanating from points in B_2 connecting to all points in B_2 and B_4



■ **Figure 1** Schematic representation of the $3m$ plane subgraphs in Lemma 7: $i \in \{1, 2, 3, 4\}$ represents cluster B_i .

Observe that these stars cover all edges of $K_{4m}[B]$. The first family is the union of m plane subgraphs: indeed, no star connecting a point in B_1 to every point in B_1 and B_2 crosses any star connecting a point in B_3 to every point in B_3 and B_4 by the assumption. Therefore, these stars can be grouped in pairs such that each pair forms a plane *star forest*.

9:4 Partitioning Complete Geometric Graphs into Plane Subgraphs

Similarly, the second and third families also consist of m plane subgraphs, each. Removing duplicate edges, we obtain a decomposition of $K_{4m}[B]$ into $3m$ plane star forests. ◀

We show that every α -dense n -element point set A contains a subset B satisfying the conditions in Lemma 7 with $m = \Omega(n)$. The overall idea is to find four large (linear-size) subsets of A clustered around four points *not* in convex position, as depicted in Fig. 3 (right). Once this favorable configuration is found, it yields a partition of the corresponding complete geometric graph into a small number of plane subgraphs. This partition is extended to a partition of the complete geometric graph of the original set into a small number of plane subgraphs. We next provide the details.

Let $k(\alpha) \geq 3\alpha^2$ be an increasing function of α , and set

$$k = k(\alpha), \text{ and } n_0 = \lceil 12k^2/\alpha^2 \rceil. \quad (3)$$

The function $k(\alpha)$ will be specified in (4) and (8), in two different ways, as required by the proof of Lemma 10. We may assume without loss of generality that $k(\alpha)$ takes integer values (by applying the ceiling function if needed). We distinguish between two cases: $n \leq n_0$, and $n \geq n_0$. Suppose first, that $n \leq n_0$. Recall that there is a decomposition of $K_n[A]$ into $n - 1$ plane subgraphs that are stars. Note that $n - 1 \leq cn$ for $n \leq n_0$ provided that $c < 1$ is large enough: indeed, $n(1 - c) \leq n_0(1 - c) \leq 1$ for $c \geq 1 - 1/n_0$.

Suppose next, that $n \geq n_0$. Let A be an n -element α -dense set. Since $D(A) \leq \alpha\sqrt{n}$, we may assume that A is contained in an axis-aligned square Q of side-length $\alpha\sqrt{n}$. Subdivide Q into k^2 axis-parallel squares, called *cells*, each of side-length $\alpha\sqrt{n}/k$. Let Σ be the set of all k^2 cells in Q . We may assume without loss of generality that no point in A lies on a cell boundary.

► **Lemma 8.** *Each cell in Σ contains at most $\frac{2\alpha^2}{k^2} n$ points of A .*

Proof. Let $\sigma \subset Q$ be any cell and σ' be the axis-aligned square concentric with σ and whose side-length is $\frac{\alpha\sqrt{n}}{k} + 1$. Obviously, σ' contains all disks of radius $1/2$ centered at the points of $A \cap \sigma$. Moreover, since A is α -dense, these n disks are interior disjoint. Moreover, σ' is a so-called *tiling domain*, i.e., a domain that can be used to tile the whole plane. Let m denote the number of points of $A \cap \sigma$. A packing of m congruent disks of radius $1/2$ in σ' requires [25, Ch. 3.4] that $m \frac{\pi}{4} \leq \frac{\pi}{\sqrt{12}} \text{Area}(\sigma')$, which yields (by using that $n \geq n_0$):

$$m \leq \frac{2}{\sqrt{3}} \left(\frac{\alpha\sqrt{n}}{k} + 1 \right)^2 \leq \frac{2}{\sqrt{3}} \left(1 + \frac{1}{\sqrt{12}} \right)^2 \frac{\alpha^2}{k^2} n \leq \frac{2\alpha^2}{k^2} n. \quad \blacktriangleleft$$

A cell $\sigma \in \Sigma$ is said to be *rich* if it contains at least $n/(3k^2)$ points of A , and *poor* otherwise. Let $\mathcal{R} \subset \Sigma$ denote the set of rich cells.

► **Lemma 9.** *There are at least $\frac{k^2}{3\alpha^2}$ rich cells; that is, $|\mathcal{R}| \geq \frac{k^2}{3\alpha^2}$.*

Proof. Let $r = |\mathcal{R}|$ denote the number of rich cells. Assume for contradiction that $r < \frac{k^2}{3\alpha^2}$. By Lemma 8 the total number of points of rich cells is at most

$$r \cdot \frac{2\alpha^2}{k^2} n < \frac{k^2}{3\alpha^2} \cdot \frac{2\alpha^2}{k^2} n = \frac{2}{3} n.$$

The total number of points of poor cells is less than

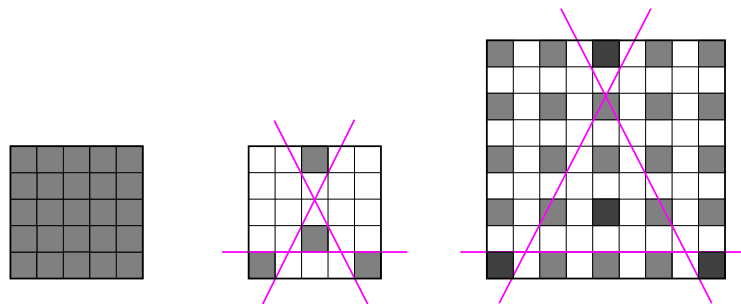
$$k^2 \cdot \frac{n}{3k^2} = \frac{n}{3}.$$

Thus the total number of points of A is strictly less than n , a contradiction that completes the proof. ◀

- **Lemma 10.** *There exist four rich cells $\sigma'_1, \sigma'_2, \sigma'_3, \sigma'_4$, such that:*
 - *for any four points $a_i \in \sigma'_i \cap A$, $i = 1, 2, 3, 4$, we have $a_4 \in \Delta a_1 a_2 a_3$.*

Before presenting our proof of Lemma 10, we sketch a simple alternative proof using a very powerful tool: the Furstenberg-Katznelson theorem, also called *Density Hales-Jewett theorem*. However, it is not strong enough to yield the quantitative statement in Theorem 2. In a sufficiently large dense subset of a grid in \mathbb{Z}^d , for any fixed d and s , one can always find a $s \times s$ grid as a subgrid. The case $d = 1$ corresponds to a classical result of Szemerédi [24]. A higher dimensional generalization of Szemerédi’s density theorem was obtained by Furstenberg and Katznelson [14]; see also [17]. Their proof uses infinitary methods in ergodic theory. A more recent combinatorial proof of this statement can be found in [22], but the resulting bound is huge (a tower of 2’s of polynomial height).

- **Theorem 11** (Furstenberg–Katznelson [14]). *For all positive integers d, s and every $c > 0$, there exists a positive integer $N = N(d, s, c)$ with the following property: every subset X of $\{1, 2, \dots, N\}^d$ of size at least cN^d contains a homothetic copy of $\{1, 2, \dots, s\}^d$.*



■ **Figure 2** Left: A 5×5 subgrid of rich cells (shaded) and a relevant subset of four cells. Right: A 5×5 subgrid of rich cells with some separation.

To deduce Lemma 10, apply Theorem 11 with $d = 2$, $s = 5$, and $c = 1/(3\alpha^2)$, to the set Σ of cells in Q and its subset \mathcal{R} of rich cells. That is, let

$$k(\alpha) = N(2, 5, 1/(3\alpha^2)), \tag{4}$$

and recall that we have set $k = k(\alpha)$ in the beginning of the proof. By (3), if n is large enough, this setting ensures the existence of a 5×5 subgrid of rich cells. Fig. 2 shows the four selected rich cells satisfying the requirements in Lemma 10. Note that a separation between subgrid cells, if any, does not interfere with the result.

Proof of Lemma 10. Let $P = \text{conv}(\mathcal{R})$ and $v(P)$ denote the number of vertices of P . Note that P is a lattice polygon whose vertices are in the $(k + 1) \times (k + 1)$ grid \mathcal{G} subdividing Q . Let $\mathcal{C} \subset \mathcal{R}$ denote the set of rich cells incident to vertices of P : we have $|\mathcal{C}| \leq v(P)$. By a well-known result, P has $v(P) \leq c'k^{2/3}$ vertices in \mathcal{G} , where $c' > 0$ is a constant [16, Exercise 2, p. 34]. Further, we may take $c' = (8\pi^2)^{1/3}$, see [6], and we will use this value later in (8).

Choose an arbitrary element of \mathcal{C} , say a leftmost one, and denote it by σ_0 . Label the remaining elements of \mathcal{C} in clockwise order around the boundary of P as $\sigma_1, \sigma_2, \dots, \sigma_{|\mathcal{C}|-1}$. Consider the convex sets $\tau_1, \tau_2, \dots, \tau_{2|\mathcal{C}|-3}$ defined as follows:

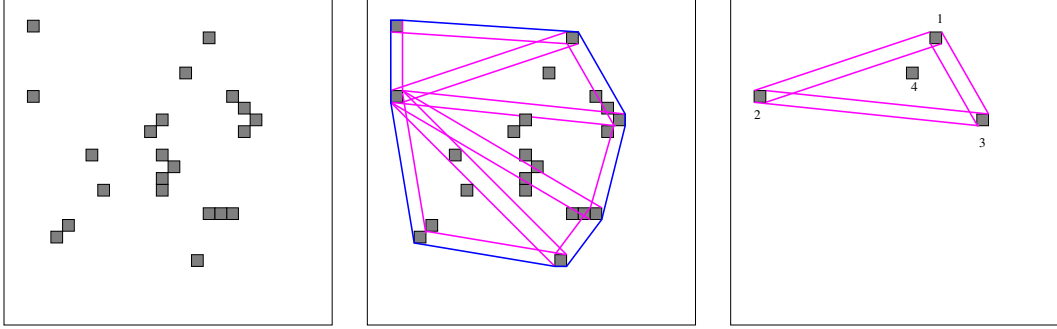
9:6 Partitioning Complete Geometric Graphs into Plane Subgraphs

$$\tau_j = \text{conv}(\sigma_0 \cup \sigma_j), \quad j = 1, 2, \dots, |\mathcal{C}| - 1, \quad (5)$$

$$\tau_{|\mathcal{C}|+j-1} = \text{conv}(\sigma_j \cup \sigma_{j+1}), \quad j = 1, \dots, |\mathcal{C}| - 2, \text{ and let} \quad (6)$$

$$K = \bigcup_{j=1}^{2|\mathcal{C}|-3} \tau_j. \quad (7)$$

We refer to K as the *star triangulation* from the boundary cell σ_0 . Let \mathcal{S} denote the set of segments that appear on the boundaries of $\tau_1, \tau_2, \dots, \tau_{2|\mathcal{C}|-3}$. See Fig. 3 for an example.



■ **Figure 3** Left: The set of rich cells in Q (each rich cell is shaded). Center: the star triangulation K from a boundary cell in \mathcal{C} . Here $|\mathcal{R}| = 22$ and $|\mathcal{C}| = 7$. Segments in \mathcal{S} are in bold lines. Right: a set of four rich cells as in Lemma 10.

▷ **Claim 12.** The segments in \mathcal{S} intersect at most $8c'k^{5/3}$ cells in Σ .

We verify the claim. Observe that for each $i = 1, \dots, 2|\mathcal{C}| - 3$, the segments in \mathcal{S} associated with τ_i intersect at most $4k$ cells in Σ (recall that Σ consists of k^2 cells). Indeed, if the translation vector corresponding to the pair of cells in τ_i makes an angle of at most 45° with the x -axis, τ_i can intersect at most four cells in each column. Otherwise τ_i can intersect at most four cells in each row. Since no point of A lies on a cell boundary and $|\mathcal{C}| \leq c'k^{2/3}$, the claim follows.

Since $P = \text{conv}(\mathcal{R})$, there are no rich cells in the exterior of P . Moreover, every rich cell intersecting some τ_i intersects a segment in \mathcal{S} . Note that for $k > (24c' \cdot \alpha^2)^3$ we have $\frac{k^2}{3\alpha^2} - 8c'k^{5/3} > 0$. To this end, let

$$k(\alpha) = \lceil (24c' \cdot \alpha^2)^3 \rceil + 1, \quad (8)$$

and recall that we have set $k = k(\alpha)$ in the beginning of the proof. It follows that there exists at least one rich cell completely inside one of the triangles of the star triangulation from σ_0 . More precisely, if $\sigma_0, \sigma_j, \sigma_{j+1}$, is such a triangle (triple) of rich cells and σ' is a rich cell inside the triangle, then

$$\text{conv}(\sigma_0 \cup \sigma_j) \cap \sigma' = \emptyset, \quad \text{conv}(\sigma_0 \cup \sigma_{j+1}) \cap \sigma' = \emptyset, \quad \text{and} \quad \text{conv}(\sigma_j \cup \sigma_{j+1}) \cap \sigma' = \emptyset. \quad (9)$$

Setting $\sigma'_1 := \sigma_0$, $\sigma'_2 := \sigma_j$, $\sigma'_3 := \sigma_{j+1}$, and $\sigma'_4 := \sigma'$, it is now easily verified that for any four points $a_i \in \sigma'_i \cap A$, $i = 1, 2, 3, 4$, we have $a_4 \in \Delta a_1 a_2 a_3$, as required. ◀

Final argument. We use the point set structure guaranteed by Lemma 10. A very similar structure is highlighted and implicitly used in [20]. For completeness we include the proof tailored for our structure.

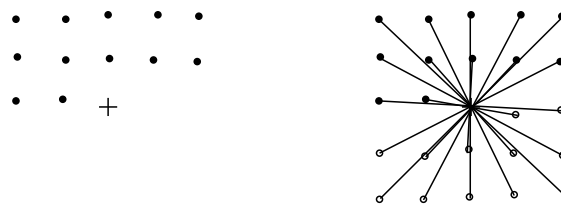
Recall that a cell $\sigma \in \Sigma$ is *rich* if it contains at least $n/(3k^2)$ points in A . Consider four rich cells $\sigma_1, \sigma_2, \sigma_3, \sigma_4$, such that for any four points $a_i \in \sigma_i \cap A$, $i = 1, 2, 3, 4$, we have $a_4 \in \Delta a_1 a_2 a_3$. Let $A_i = A \cap \sigma_i$, for $i = 1, 2, 3, 4$. Remove points from each of the four cells, if needed, until there are exactly $\lceil n/(3k^2) \rceil$ points in A in each of these cells. Let us denote the resulting sets as $B_i \subseteq A_i$, for $i = 1, 2, 3, 4$, where $|B_1| = |B_2| = |B_3| = |B_4| = m = \lceil n/(3k^2) \rceil$.

Recall that we are in the case $n \geq n_0$ and that $m = \lceil n/(3k^2) \rceil$, where $k = k(\alpha)$ is a fixed integer. Applying Lemma 6 with $P := A$ and $Q := B$ and Lemma 7 yields that the edge-set of the complete geometric graph $K_n[A]$ can be decomposed into at most

$$n - 4m + 3m = n - m \leq \left(1 - \frac{1}{3k^2}\right)n \tag{10}$$

plane subgraphs. Setting $c(\alpha) = 1 - \frac{1}{3k^2(\alpha)}$ and recalling (8) completes the proof of inequality (2). ◀

Note. Next, we show that $c(\alpha) \geq 1/2$ in Theorem 2, i.e., some sets require at least $n/2$ plane subgraphs in the partition. For simplicity, we give a grid example, where $\alpha \leq \sqrt{2} + \varepsilon$, for a small $\varepsilon > 0$. (A suitable example can be found for every $\alpha > \alpha_0$.) Let $n = k^2 - 1$, where $k = 2a + 1$. Consider the $n/2$ integer points with positive y -coordinates or zero y -coordinate and negative x -coordinate in the lattice section $\{-a, \dots, a\}^2$, suitably perturbed to avoid collinearities. Refer to Fig. 4. Add to these points the $n/2$ reflections with respect to the



■ **Figure 4** A dense set of 24 points with a crossing family of size 12. The origin (marked with a cross) is not part of the set.

origin, suitably perturbed to avoid collinearities. Observe that the resulting point set has n points and admits a crossing family of size $n/2$ consisting of $n/2$ edges connecting the $n/2$ initial points with their reflections. Consequently, any partition of the corresponding complete geometric graphs into plane subgraphs consists of at least $n/2$ such subgraphs.

3 Concluding remarks

A. There are many geometric results for finite point sets that can be strengthened under the assumption that the set is dense, for instance in the case of crossing families or the classic Erdős–Szekeres problem on points in convex position, as explained next; see also [8, Ch. 10]. In 1935, Erdős and Szekeres [11] proved, as one of the first Ramsey-type results in combinatorial geometry, that every set of n points in general position in the plane contains $\Omega(\log n)$ points in convex position, and some 25 years later showed [12] that this bound is tight up to the multiplicative constant. According to the current best (asymptotic) upper bound, due to Suk [23], every set of n points in general position in the plane contains $(1 - o(1)) \log n$ points in convex position, and this bound is tight up to lower-order terms.

In contrast, a classic result of Valtr given below specifies a much larger threshold for the maximum size of a subset in convex position in a density-restricted point set [26]: For every $\alpha \geq \alpha_0$ there exists $\beta = \beta(\alpha) > 0$ such that any set of n points in general position in the

plane satisfying $D(A) \leq \alpha n^{1/2}$, contains a subset of $\beta n^{1/3}$ points in convex position. On the other hand, for every $n \in \mathbb{N}$ there exists an n -element point set $A \subset \mathbb{R}^2$ in general position, satisfying $D(A) = O(n^{1/2})$, in which every subset in convex position has at most $O(n^{1/3})$ points. In particular, a suitable small perturbation of a piece of the integer lattice has this property.

B. Pach and Solymosi [21] gave a concise characterization of point sets admitting a crossing family of size $n/2$. They showed that a set P of n points in general position in the plane (n even) admits a perfect matching with pairwise crossing segments if and only if P has precisely n halving lines. A *halving line* for such a set is a line incident to two points of the set and leaving exactly $n/2 - 1$ points in each of the two open halfplanes it determines [8, Ch. 8.3].

As defined by Dillencourt, Eppstein, and Hirschberg [9], the *geometric thickness* of an abstract graph G is the minimum $k \in \mathbb{N}$ such that G has a drawing as a geometric graph whose edges can be partitioned into k plane subgraphs. The authors proved that the geometric thickness of K_n is between $\lceil (n/5.646) + 0.342 \rceil$ and $\lceil n/4 \rceil$. As pointed out in [7], the difference between Problem 1 and determining the geometric thickness of K_n is that Problem 1 deals with all possible drawings of K_n whereas geometric thickness asks for the best drawing.

Decompositions of the edge-set of a complete geometric graph on n points into the minimum number of families of pairwise disjoint edges (resp., pairwise intersecting edges), have been studied among others, by Araujo, Dumitrescu, Hurtado, Noy, and Urrutia [4].

Recently, Obenaus and Orthaber [18] gave a negative answer to the question of whether every complete geometric graph on n vertices (n even) can be partitioned into $n/2$ plane subgraphs. See also [2]. As such, $n/2 + 1$ is a lower bound in some instances on the number of such subgraphs in Problem 1.

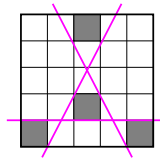
C. If X is a finite point set in the plane, every point of $\text{conv}(X)$ can be expressed as a convex combination of at most 3 points in X . This implies that every point set in general position that is not in convex position contains a subset of 4 points that are not in convex position, i.e., a four-tuple $a, b, c, d \in X$ such that $d \in \Delta abc$.

Our Theorem 2 gives the following quantitative version of Carathéodory’s Theorem for α -dense sets.

► **Corollary 13.** *Let A be a set of n points in the plane, with $D(A) \leq \alpha n^{1/2}$, for some $\alpha \geq \alpha_0$. Then there exist at least cn^4 four-tuples $a_1, a_2, a_3, a_4 \in A$ such that $a_4 \in \Delta a_1 a_2 a_3$, where $c = c(\alpha) > 0$ is a constant.*

Proof. We may assume that n is large enough. Recall that $m = \lceil n/(3k^2) \rceil$, where $k = k(\alpha)$ is a fixed integer. By Lemma 10, there exist four rich cells $\sigma'_1, \sigma'_2, \sigma'_3, \sigma'_4$, such that for any four points $a_i \in \sigma'_i \cap A$, $i = 1, 2, 3, 4$, we have $a_4 \in \Delta a_1 a_2 a_3$. Consequently, the number of (ordered) 4-tuples with this property is at least $m^4 \geq n^4/(81k^8)$. Setting $c(\alpha) = 3^{-4}k^{-8}(\alpha)$ completes the proof of the lower bound. On the other hand, the total number of such 4-tuples is clearly less than n^4 . ◀

D. The proof of Corollary 3 is straightforward. Subdivide $U = [0, 1]^2$ into 25 smaller axis-parallel squares as in Fig. 5. Consider the four subsquares: $\sigma_1 = [0, 1/5]^2$, $\sigma_2 = [4/5, 1] \times [0, 1/5]$, $\sigma_3 = [2/5, 3/5] \times [4/5, 1]$, and $\sigma_4 = [2/5, 3/5] \times [1/5, 2/5]$.



■ **Figure 5** The four distinguished subsquares are shaded.

The expected number of points in each subsquare is $n/25$. With probability tending to 1 as $n \rightarrow \infty$, each of the four subsquares contains at least $n/50$ points in A .

Observe that any line connecting a point in σ_1 with a point in σ_3 leaves σ_4 below. By symmetry, any line connecting a point in σ_2 with a point in σ_3 leaves σ_4 below. Third, any line connecting a point in σ_1 with a point in σ_2 leaves σ_4 above. As such, a structure analogous to that in Lemma 10 is obtained, and the corollary follows. ◀

E. Can the dependency of $c(\alpha)$ on α in (2) be improved? Or completely eliminated?

References

- 1 Oswin Aichholzer, Jan Kyncl, Manfred Scheucher, Birgit Vogtenhuber, and Pavel Valtr. On crossing-families in planar point sets. *Comput. Geom.*, 107:101899, 2022. doi:10.1016/J.COMGEO.2022.101899.
- 2 Oswin Aichholzer, Johannes Obenaus, Joachim Orthaber, Rosna Paul, Patrick Schnider, Raphael Steiner, Tim Taubner, and Birgit Vogtenhuber. Edge partitions of complete geometric graphs. In Xavier Goaoc and Michael Kerber, editors, *38th International Symposium on Computational Geometry, SoCG 2022, June 7-10, 2022, Berlin, Germany*, volume 224 of *LIPICs*, pages 6:1–6:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICs.SOCG.2022.6.
- 3 Jin Akiyama and Mikio Kano. Path factors of a graph. In Frank Harary and John S. Maybee, editors, *Graph and Applications, Proceedings of the First Colorado Symposium on Graph Theory*, pages 1–21. Wiley, New York, 1985.
- 4 Gabriela Araujo, Adrian Dumitrescu, Ferran Hurtado, Marc Noy, and Jorge Urrutia. On the chromatic number of some geometric type kneser graphs. *Comput. Geom.*, 32(1):59–69, 2005. doi:10.1016/J.COMGEO.2004.10.003.
- 5 Boris Aronov, Paul Erdős, Wayne Goddard, Daniel J. Kleitman, Michael Klugerman, János Pach, and Leonard J. Schulman. Crossing families. *Comb.*, 14(2):127–134, 1994. doi:10.1007/BF01215345.
- 6 Imre Bárány and Maria Prodromou. On maximal convex lattice polygons inscribed in a plane convex set. *Israel Journal of Mathematics*, 154(1):337–360, 2006.
- 7 Prosenjit Bose, Ferran Hurtado, Eduardo Rivera-Campo, and David R. Wood. Partitions of complete geometric graphs into plane trees. *Comput. Geom.*, 34(2):116–125, 2006. doi:10.1016/J.COMGEO.2005.08.006.
- 8 Peter Brass, William O. J. Moser, and János Pach. *Research Problems in Discrete Geometry*. Springer, 2005.
- 9 Michael B. Dillencourt, David Eppstein, and Daniel S. Hirschberg. Geometric thickness of complete graphs. *J. Graph Algorithms Appl.*, 4(3):5–17, 2000. doi:10.7155/JGAA.00023.
- 10 Herbert Edelsbrunner, Pavel Valtr, and Emo Welzl. Cutting dense point sets in half. *Discret. Comput. Geom.*, 17(3):243–255, 1997. doi:10.1007/PL00009291.
- 11 Paul Erdős and George Szekeres. A combinatorial problem in geometry. *Compositio mathematica*, 2:463–470, 1935.

- 12 Paul Erdős and George Szekeres. On some extremum problems in elementary geometry. *Ann. Univ. Sci. Budapest. Eötvös Sect. Math.*, 3(4):53–62, 1960.
- 13 William S. Evans and Noushin Saeedi. On problems related to crossing families. *CoRR*, abs/1906.00191, 2019. [arXiv:1906.00191](https://arxiv.org/abs/1906.00191).
- 14 Hillel Furstenberg and Yitzhak Katznelson. An ergodic Szemerédi theorem for commuting transformations. *Journal d'Analyse Mathématique*, 34(1):275–291, 1978.
- 15 István Kovács and Géza Tóth. Dense point sets with many halving lines. *Discret. Comput. Geom.*, 64(3):965–984, 2020. [doi:10.1007/S00454-019-00080-3](https://doi.org/10.1007/S00454-019-00080-3).
- 16 Jirí Matousek. *Lectures on Discrete Geometry*, volume 212 of *Graduate texts in mathematics*. Springer, 2002.
- 17 Jaroslav Nešetřil. Ramsey theory. In Ron Graham, Martin Grötschel, and László Lovász, editors, *Handbook of Combinatorics, Vol. II*, chapter 25, pages 1331–1403. Elsevier, Amsterdam, first edition, 1995.
- 18 Johannes Obenaus and Joachim Orthaber. Edge partitions of complete geometric graphs (part 1). *CoRR*, abs/2108.05159, 2021. [arXiv:2108.05159](https://arxiv.org/abs/2108.05159).
- 19 János Pach, Natan Rubin, and Gábor Tardos. Planar point sets determine many pairwise crossing segments. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 1158–1166. ACM, 2019. [doi:10.1145/3313276.3316328](https://doi.org/10.1145/3313276.3316328).
- 20 János Pach, Morteza Saghafian, and Patrick Schnider. Decomposition of geometric graphs into star-forests. In Michael A. Bekos and Markus Chimani, editors, *Graph Drawing and Network Visualization - 31st International Symposium, GD 2023, Isola delle Femmine, Palermo, Italy, September 20-22, 2023, Revised Selected Papers, Part I*, volume 14465 of *Lecture Notes in Computer Science*, pages 339–346. Springer, 2023. [doi:10.1007/978-3-031-49272-3_23](https://doi.org/10.1007/978-3-031-49272-3_23).
- 21 János Pach and József Solymosi. Halving lines and perfect cross-matchings. *Contemporary Mathematics*, 223:245–250, 1999.
- 22 DHJ Polymath. A new proof of the density Hales-Jewett theorem. *Annals of Mathematics*, pages 1283–1327, 2012.
- 23 Andrew Suk. On the Erdős-Szekeres convex polygon problem. *Journal of the American Mathematical Society*, 30:1047–1053, 2017. [doi:10.1090/jams/869](https://doi.org/10.1090/jams/869).
- 24 Endre Szemerédi. On sets of integers containing no k elements in arithmetic progression. *Acta Arithmetica*, 27(1):199–245, 1975.
- 25 László Fejes Tóth, Gábor Fejes Tóth, and Włodzimierz Kuperberg. *Lagerungen*. Springer, 2023.
- 26 Pavel Valtr. Convex independent sets and 7-holes in restricted planar point sets. *Discret. Comput. Geom.*, 7:135–152, 1992. [doi:10.1007/BF02187831](https://doi.org/10.1007/BF02187831).
- 27 Pavel Valtr. Planar point sets with bounded ratios of distances. *Dissertation, Freie Univ.*, 1994.
- 28 Pavel Valtr. Lines, line-point incidences and crossing families in dense sets. *Comb.*, 16(2):269–294, 1996. [doi:10.1007/BF01844852](https://doi.org/10.1007/BF01844852).

Constrained Outer-String Representations

Therese Biedl  

David R. Cheriton School of Computer Science, University of Waterloo, Canada

Sabine Cornelsen  

Department of Computer and Information Science, University of Konstanz, Germany

Jan Kratochvíl  

Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic

Ignaz Rutter  

Faculty of Computer Science and Mathematics, University of Passau, Germany

Abstract

An *outer-string representation* of a graph is an intersection representation in which each vertex is represented by a curve that is contained in the unit disk and has at least one endpoint on the boundary of the unit disk. In an *outer-1-string representation* the curves representing any two vertices are in addition allowed to intersect at most once.

In this paper, we consider the following constrained version: Given a graph G plus a cyclic order v_1, \dots, v_n of the vertices in G , test whether G has an outer-string or an outer-1-string representation in which the curves representing v_1, \dots, v_n intersect the boundary of the unit disk in this order. We first show that a graph has an outer-string representation for all possible cyclic orders of the vertices if and only if the graph is the complement of a chordal graph. Then we turn towards the situation where one particular cyclic order of the vertices is fixed.

We characterize the chordal graphs admitting a constrained outer-string representation and the trees and cycles admitting a constrained outer-1-string representation. The characterizations yield polynomial-time recognition and construction algorithms; in the case of outer-1-string representations the run time is linear. We also show how to decide in polynomial time whether an arbitrary graph admits a constrained *L-shaped* outer-1-string representation. In an L-shaped representation the curves are 1-bend orthogonal polylines anchored on a horizontal line, and they are contained in the half-plane below that line. However, not even all paths with a constrained outer-1-string representation admit one with L-shapes. We show that 2-bend orthogonal polylines are sufficient for trees and cycles with a constrained outer-1-string representation.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases String representation, outer-string, outer-1-string, chordal graphs, trees, polynomial-time algorithms, computational complexity

Digital Object Identifier 10.4230/LIPIcs.GD.2024.10

Funding *Therese Biedl*: Research supported by NSERC FRN RGPIN-2020-03958.

Jan Kratochvíl: Research supported by Czech Science Foundation grant No. GAČR 23-04949X.

Acknowledgements This work was initiated at the Dagstuhl Seminar 24062 on Beyond-Planar Graphs: Models, Structures and Geometric Representations, Schloss Dagstuhl, Germany, February 2024. The authors would like to thank the other participants (and especially Stefan Felsner) for stimulating discussions.

1 Introduction

In a *string representation* [6, 15] of a graph $G = (V, E)$, each vertex v is drawn as a simple curve $\partial(v)$ such that the curves of two vertices intersect if and only if the two vertices are adjacent. We study here only *outer-string representations* where all curves reside within a disk or simple closed region D , and the curve of every vertex has at least one endpoint



© Therese Biedl, Sabine Cornelsen, Jan Kratochvíl, and Ignaz Rutter;
licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 10; pp. 10:1–10:18



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

on the boundary of D , the so-called *anchor* of the vertex. See Figure 1a. Outer-string representations were named as such in 1982 [10], though they were implicitly defined and first results obtained already in 1966 [15]. It follows from a construction of Middendorf and Pfeiffer [12] that testing whether a graph admits an outer-string representation is NP-hard; see [14] for a sketch of the proof. One key result about outer-string graphs is that they are χ -*bounded*, i.e., their chromatic number is bounded by a function of the clique number [14]. By contrast, the chromatic number even of triangle-free segment graphs, which are a subclass of string graphs, can be $\Omega(\log n)$ [13]. A graph is a *chordal graph*, if it does not contain an induced cycle of length greater than three. By its tree representation [7, 16], every chordal graph admits an outer-string representation, and so do the complements of chordal graphs. Unfortunately, outer-string representations sometimes need exponentially many crossings [1]. So it is interesting to investigate which graphs allow an outer-string representation with a restricted number of crossings. In an *outer-1-string representation*, it is additionally required that the curves of two vertices intersect at most once. This is similar to the intersection graph of pseudosegments [6], however, with the additional constraint that the anchors still have to be on the boundary of a simple closed region containing all pseudosegments. Representing chordal graphs as intersections of pseudosegments was considered in [3].

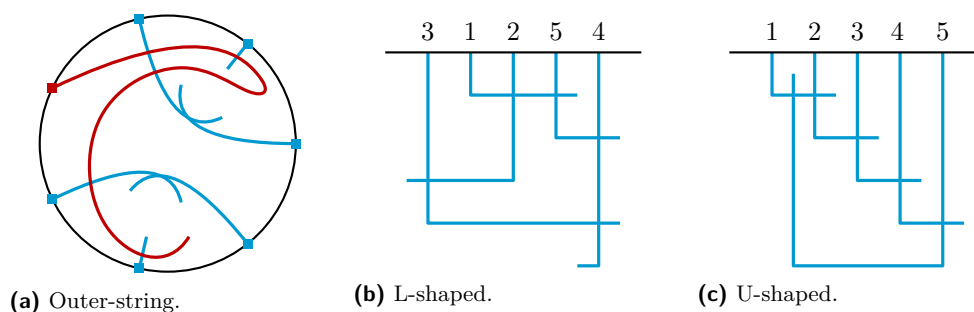
Biedl and Derka [2] considered outer-string representation where the order of crossings along a curve was constrained. In this paper, we study outer-string representations that are *constrained* in the sense that the cyclic order of the anchors is fixed, i.e., we consider as input *cyclically ordered graphs*¹ (that is, graphs together with a cyclic order of the vertices) and we ask whether there is an outer-string, or an outer-1-string representation within a disk D in which the anchors occur on the boundary of D in the given cyclic order. Constrained outer-string representations were called the *constrained case* in [15]. Sinden [15] showed that the constrained case with n vertices can be reduced to the unconstrained case with $2n$ additional vertices and $4n$ additional edges.

One can restrict the shapes of the curves further. In particular, we also consider *L-shaped* [11, 9, 4] and *U-shaped representations* in which the anchors are on a horizontal line ℓ and the vertices are 1- or 2-bend orthogonal polylines below that line; see Figures 1b and 1c. More precisely, in the case of L-shaped representations, the curves are required to consist of a vertical segment going downward from its anchor on ℓ followed (optionally) by a single horizontal segment. I.e., in particular, we also allow Js. In the case of U-shaped representations, there may be an additional final vertical segment pointing upward. In the *constrained* version the input are *ordered graphs*, i.e., graphs with a linear order of the vertices and we require that the anchors on ℓ appear in this specific order.

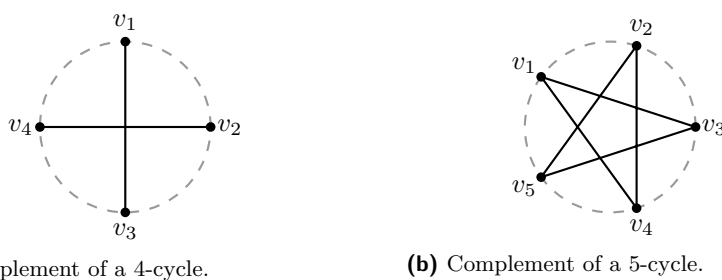
Besides some sufficient conditions for constrained outer-string representations, Sinden [15] also observed the following necessary condition: The *complement of an anchor-ordered cycle* with at least four vertices does not have a constrained outer-string representation, i.e., if the cyclic order is v_1, \dots, v_n then the graph with edge set $E = \{\{v_i, v_j\}; |i - j| \notin \{1, n - 1\}\}$ does not have a constrained outer-string representation; see Figure 2.

Our Results. We show that a graph admits a constrained outer-string representation for every circular order of the vertices if and only if its complement is chordal (Theorem 2 in Section 2). In Section 3 we show that a cyclically ordered chordal graph admits a constrained outer-string representation if and only if it does not contain the complement of an anchor-ordered 4-cycle. The proof is constructive and yields a construction algorithm as

¹ Sinden [15] used the term *constrained graphs*.



■ **Figure 1** (a) An outer-string representation of a tree that is not an outer-1-string representation (b+c) two outer-1-string representations of a 5-cycle with special shapes.



■ **Figure 2** Complements of anchor-ordered cycles; vertices at anchor positions.

well as a polynomial time testing algorithm. In order to characterize the cyclically ordered trees that admit a constrained outer-1-string representation, we need two more forbidden substructures, which we define in Section 4. We then provide a linear-time algorithm that either constructs a constrained outer-1-string representation of a cyclically ordered tree, or returns a forbidden substructure. In Section 5, we show how to test in time quadratic in the number of vertices whether any ordered graph admits a constrained L-shaped outer-1-string representation. In Section 6, we characterize cyclically ordered simple cycles that admit a constrained outer-1-string representation. The characterization yields a linear time testing algorithm. We further show that every cyclically ordered tree (Corollary 18) or simple cycle (Corollary 23) that admits a constrained outer-1-string representation already admits one with U-shapes for every induced linear order.² Full proofs of statements marked with (★) will appear in the forthcoming full version of the paper.

2 Preliminaries

Let $G = (V, E)$ be a simple graph. For $e \in E$, let $G - e = (V, E \setminus \{e\})$. For $V' \subseteq V$, let $G - V'$ be the graph obtained from G by removing V' and all edges incident to a vertex in V' ; we write $G - v$ for $G - \{v\}$. A set $A \subseteq V$ is *connected* if A induces a connected subgraph in G . The degree $\deg(v)$ of a vertex v is the number of edges that are incident to v . A *bridge* of a graph G is an edge e of G such that $G - e$ has more connected components than G . If G is connected, then the *bridge components* of a bridge $e = \{x, y\}$ are the vertex sets X and Y of the two connected components of $G - e$, named such that $x \in X$ and $y \in Y$.

² A cyclic order $\langle v_1, \dots, v_n \rangle$ induces n linear orders $\langle v_{k+1}, \dots, v_n, v_1, \dots, v_k \rangle$, $1 \leq k \leq n$.

2.1 Input and Output

An instance (G, \circlearrowleft) of the problem of testing for a constrained outer-string or outer-1-string representation consists of a graph G and a cyclic order \circlearrowleft of the anchors around the disk D . During one of our algorithms, for some curves we need to fix both endpoints to the boundary of the disk D at specific positions. We call such vertices *doubly-anchored*, and they occur twice in \circlearrowleft . For our algorithms we assume that the graph G is given as an adjacency list and \circlearrowleft is given as a doubly-linked circular list of vertex-references. Moreover, each vertex is equipped with pointers to its one or two entries in \circlearrowleft .

A representation is stored as a plane graph H . Every anchor corresponds to an *anchor-vertex* in H , and these are connected in an *anchor-cycle* according to \circlearrowleft with doubly-anchored vertices appearing twice in the anchor-cycle. Every crossing of two curves corresponds to a *crossing-vertex*. In an outer-1-string representation this means that each edge of G corresponds to a crossing-vertex. Every vertex-curve $\partial(v)$ gives rise to edges in H that correspond to maximal sub-curves of $\partial(v)$ between its anchor(s) and crossings or between two crossings, connecting the corresponding vertices. Finally, H comes with a fixed circular order of the edges around each vertex that corresponds to the representation and in which the anchor-cycle bounds the outer face. Any embedding-preserving planar drawing of H yields then the desired representation of the instance.

2.2 A Necessary Condition for Constrained Outer-String Representations

Two sets V_1 and V_2 of vertices are *independent* if they have no vertex in common, and there is no edge with one endvertex in V_1 and the other in V_2 . In an instance of constrained outer-string representation, we call two disjoint sets A_1 and A_2 of anchors *interleaved* if the cyclic order \circlearrowleft of anchors contains a subsequence a_1, a_2, a'_1, a'_2 where $a_i, a'_i \in A_i$ for $i = 1, 2$. Note that a_i and a'_i can be different anchors of the same doubly-anchored vertex. Two sets V_1 and V_2 of vertices are *interleaved* if their anchors in \circlearrowleft are interleaved. Observe that the complement of an anchor-ordered 4-cycle is a pair of interleaved independent edges.

► **Lemma 1** (interleaved independent pairs \star). *If (G, \circlearrowleft) has a constrained outer-string representation, then there are no two independent connected vertex sets that are interleaved.*

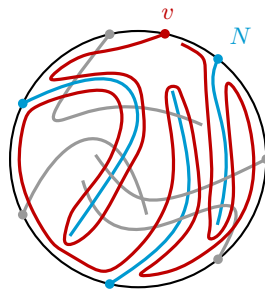
2.3 Complements of Chordal Graphs

The necessary condition of Sinden [15] implies that, if the complement of an input graph contains an induced cycle of length at least 4, then there exists a cyclic order for which it does not admit a constrained outer-string representation. This yields the necessity of the following characterization.

► **Theorem 2.** *A graph admits a constrained outer-string representation for any cyclic ordering of its vertices if and only if its complement is chordal.*

Proof. If the complement of a graph G is not chordal, then it contains an induced cycle C_k of length $k > 3$. Let u_1, \dots, u_k be the vertices of this cycle in the natural order. Then no circular order of the vertices of G which extends this order allows an outer-string representation of G , because it contains the complement of C_k with the natural order of its vertices as an induced subgraph.

We prove the opposite implication by induction on the number of vertices of G . Clearly, the one-vertex graph allows an outer-string representation for any (i.e., just one) circular ordering of its vertices (i.e., vertex). Suppose $G = (V, E)$ has more than one vertex and that



■ **Figure 3** How to construct a constrained outer-string representation for a complement of a chordal graph.

the complement \overline{G} of G is a chordal graph. Let a circular order \circlearrowleft of V be given. Let v be a *simplicial vertex* of \overline{G} , i.e., a vertex v whose neighborhood $N = N_{\overline{G}}(v)$ in \overline{G} induces a clique in \overline{G} , and hence an independent set in G . Observe that every chordal graph contains a simplicial vertex [5]. Consider $G' = G - v$ and its complement $\overline{G'} = \overline{G} - v = \overline{G} - v$. By the induction hypothesis, $\overline{G'}$ has an outer-string representation that respects the circular order \circlearrowleft' of $V \setminus \{v\}$ induced by \circlearrowleft . In this representation, the neighbors of v (in \overline{G}) are represented by disjoint curves; see Figure 3. We add a curve $\partial(v)$ starting at the anchor of v and contouring the boundary of the region $D \setminus \bigcup_{x \in N} \partial(x)$. In this way $\partial(v)$ intersects all curves $\partial(y)$ for $y \in V \setminus (N \cup \{v\})$ and avoids crossing all curves $\partial(x)$ for $x \in N$. Thus we constructed a constrained outer-string representation of G that respects \circlearrowleft . ◀

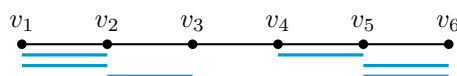
3 Chordal Graphs and Constrained Outer-String Representations

We characterize chordal graphs with a cyclic order of the vertices that admit a constrained outer-string representation. The proof is by induction on the number of pairs of independent edges. For example, a path of length five has three pairs of independent edges; see Figure 4.

► **Theorem 3.** *A chordal graph $G = (V, E)$ with a cyclic order \circlearrowleft of V has a constrained outer-string representation if and only if no two independent edges are interleaved.*

Proof. By Lemma 1, (G, \circlearrowleft) has no constrained outer-string representation if there are two independent edges that are interleaved. So assume that there is no pair of interleaved independent edges. We show by induction on the number of pairs of independent edges that (G, \circlearrowleft) has a constrained outer-string representation within a simple connected region D . We may assume that there are no isolated vertices.

In the base case, G has no pair of independent edges. Thus [8, Theorem 6.3], G is a *split graph*, i.e., it consists of a clique $K = \{k_1, \dots, k_r\}$ and a set S of independent vertices, with an arbitrary set of edges between K and S . To obtain an outer-string representation of G , add $|K|$ concentric circles inside D , and assign them to k_1, \dots, k_r . For every clique-vertex k_i , go perpendicular from the anchor to the circle assigned to k_i , then along this circle until we almost touch the curve $\partial(k_i)$. This creates an intersection for each edge $\{k_i, k_j\}$: Assume



■ **Figure 4** A path of length five contains three pairs of independent edges.

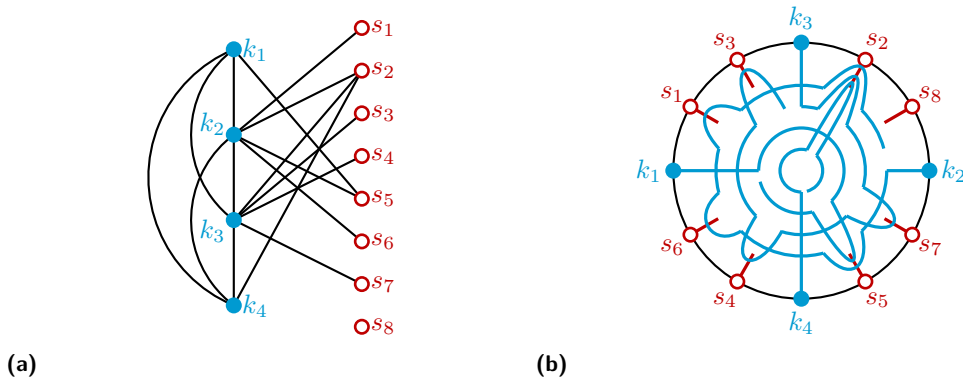


Figure 5 Illustration of the base case in the proof of Theorem 3 on chordal graphs.

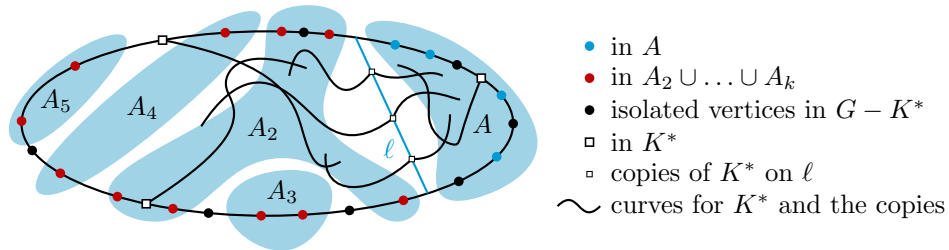


Figure 6 Splitting a chordal graph into smaller instances. If G is chordal but not a split graph, we find a minimal set K^* of vertices such that $G - K^*$ contains at least two non-trivial components. Let A be such a component for which the anchors are consecutive up to isolated vertices of $G - K^*$. Split D by a curve ℓ separating the anchors of A from the anchors of the other non-trivial components A_2, \dots, A_k of $G - K^*$. Insert copies of K^* on ℓ .

that the circle for k_i has greater radius. Then $\partial(k_j)$ intersects this circle when connecting from the anchor of k_j to its circle. This represents the clique K . Now for every vertex $s \in S$, add a short segment $\partial(s)$ from the anchor of s perpendicular to the boundary of D , and for all $k \in K$ with $\{k, s\} \in E$, add a detour to $\partial(k)$ to intersect $\partial(s)$; see Figure 5.

Now assume that G contains at least one pair of independent edges. Let K^* be a minimal set of vertices such that $G - K^*$ contains at least two *non-trivial components*, i.e., connected components that contain an edge. The following claim is an implication of [8, Theorem 4.1].

▷ Claim 4 (★). K^* exists and is a clique.

Let A_1, \dots, A_k be the non-trivial components of $G - K^*$. Since there is no pair of interleaved independent edges, it follows that the anchors of A_1, \dots, A_k on the boundary of D are nested; see Figure 6. In particular, there must be a component, say $A = A_1$, whose anchors are consecutive, except for perhaps some isolated vertices of $G - K^*$. Split D along a line ℓ that separates the anchors of A from the anchors of A_2, \dots, A_k . Place $|K^*|$ anchor points along ℓ , one per vertex of K^* in arbitrary order.

Now we get two instances, an instance I_A and an instance $I_{\bar{A}}$, by cutting along ℓ . The instance I_A contains (1) all vertices of A , (2) all vertices whose anchor were on the same side of ℓ as the anchors of A ; these might be anchors of isolated vertices of $G - K^*$ or anchors of vertices in K^* , and (3) copies of vertices in K^* with an anchor on ℓ . Here the point assigned to $k \in K^*$ is taken as the endpoint for k if the actual endpoint of k is not in this part of D , and it gets taken as endpoint for a new vertex k' otherwise, where k' is adjacent only to k . The instance $I_{\bar{A}}$ is defined analogously.

▷ **Claim 5.** I_A and $I_{\bar{A}}$ are both chordal.

Proof. I_A and $I_{\bar{A}}$ are obtained from chordal graphs by removing vertices and adding leaves. This neither creates new cycles, nor does it remove chords from remaining cycles. ◁

▷ **Claim 6 (★).** If I_A and $I_{\bar{A}}$ have constrained outer-string representations then so does (G, \circlearrowleft) .

Sketch of Proof. A curve for a vertex k in K^* can be obtained by starting at the original anchor of k , following $\partial(k)$ until its end and back to an intersection point with a curve ∂'_1 anchored at a copy k' of k on ℓ , following ∂'_1 up to ℓ and finally along the curve ∂'_2 of k' in the other subinstance until its end. ◁

▷ **Claim 7.** If I_A or $I_{\bar{A}}$ contains a pair of interleaved independent edges then so does (G, \circlearrowleft) .

Proof. Assume I_A contains a pair $\{v, w\}, \{a, b\}$ of interleaved independent edges, the case for $I_{\bar{A}}$ is symmetric. Unless this involves vertices with an anchor on the curve ℓ , the same pair is already contained in (G, \circlearrowleft) . If both $\{v, w\}$ and $\{a, b\}$ contain a vertex of K^* , then the pair is not independent. Since I_A contains only vertices of A, K^* , and degree-one neighbours of K^* , we may assume that $\{a, b\} \subseteq A$. If the anchors of both v and w are on ℓ , then $\{v, w\}$ and $\{a, b\}$ are not interleaved. So, we may assume that the anchor of v is on ℓ and the anchor of w is not. We distinguish two cases based on whether $w \in K^*$ or not.

If $w \in K^*$ then, by the minimality of K^* , vertex w had a neighbour v' in component A_2 . It follows that $\{a, b\}$ and $\{w, v'\}$ are interleaved and independent. If $w \in A$ or w is an isolated vertex in $G - K^*$, then v is the copy of a vertex $v' \in K^*$ whose anchor is on the other side of ℓ than A . Thus, $\{a, b\}$ and $\{w, v'\}$ are interleaved and independent. ◁

▷ **Claim 8 (★).** The number of pairs of independent edges in I_A and $I_{\bar{A}}$, respectively, is smaller than in G .

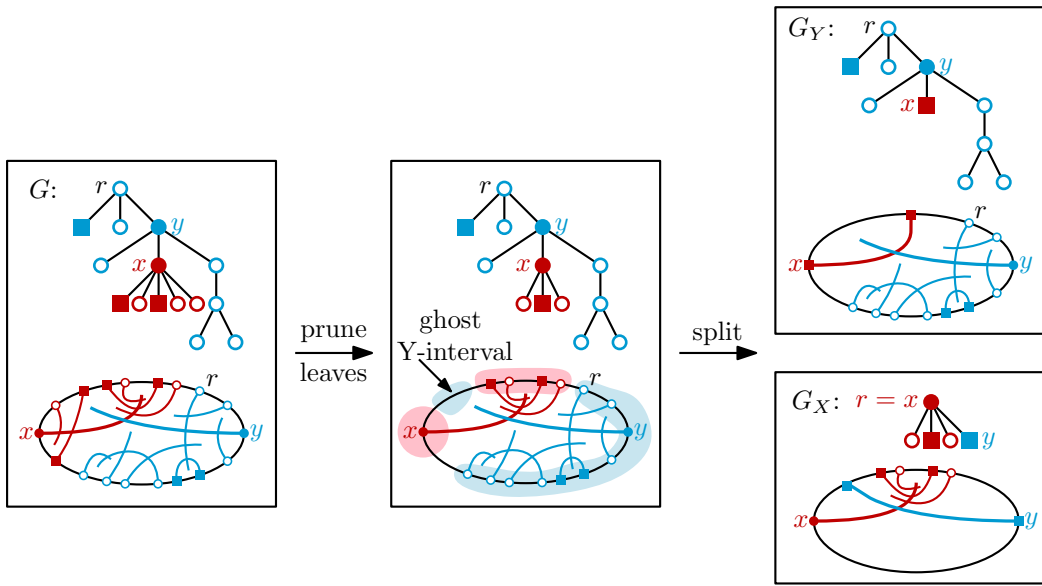
Sketch of Proof. By construction, there is a pair of independent edges $e_1 \subseteq A_1 = A, e_2 \subseteq A_2$ in G which is neither contained in I_A nor $I_{\bar{A}}$. On the other hand the pairs of independent edges of I_A ($I_{\bar{A}}$, respectively), can be mapped into the pairs of independent edges of G with at least one endvertex in K^* or A ($A_2 \cup \dots \cup A_k$, respectively). ◁

This concludes the proof: If (G, \circlearrowleft) contains no pair of interleaved independent edges, then, by Claim 7, none of the sub-instances has one. By Claim 8, they have fewer independent edge pairs than (G, \circlearrowleft) and they hence have a constrained outer-string representation by the inductive hypothesis. By Claim 6 these representations can be combined to a constrained outer-string representation for the original instance (G, \circlearrowleft) . ◀

► **Corollary 9.** *It can be tested in polynomial time whether a chordal graph with a given cyclic order of the vertices admits a constrained outer-string representation.*

4 Constrained Outer-1-String Representations for Trees

In this section, we show how to test for a constrained outer-1-string representation if the graph is a tree. We first give an outline of our approach. See Figure 7. Let (G, \circlearrowleft) be the given instance where G is a tree. If G is a single vertex, then the answer is always true. Otherwise, we root G , preferably at a vertex that has at least two neighbors, and process the vertices in post-order, i.e., children are processed before their parents. Whenever we encounter a vertex x that is not a leaf, we either find an obstruction, i.e., a sub-instance that makes a constrained outer-1-string representation impossible, or we remove the children of



■ **Figure 7** We solve constrained outer-1-string on trees in a postorder traversal. Some leaves might be doubly-anchored (squared vertices). When processing a vertex x then all its children are leaves. We first prune some of x 's children (Rule 1). We then make sure that the sequence of anchors contains exactly two X -intervals (pink regions), one of which contains only x . If this is impossible, we reject the instance (Rule 2, Rule 4, and Rule 5). Finally, we split the instance along the edge between x and its parent y , keeping y and x as a doubly-anchored vertex in the opposite component (Rule 6). The base case is reached after pruning the leaves of a star (Rule 2 and Rule 3).

x until x is a leaf. Thus at the end only a single vertex remains and we are done. For the recursions, we will sometimes have doubly-anchored vertices, but we maintain as invariant that only leaves of the rooted tree can be doubly-anchored.

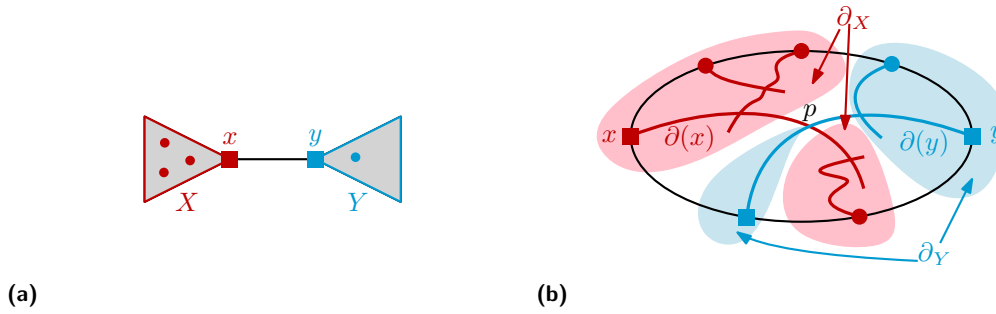
4.1 Obstructions

By Lemma 1, there cannot be a constrained outer-string representation if there is a pair of independent connected vertex-sets that are interleaved. We call such an interleaved independent pair a *pair-obstruction* if each of the two vertex-sets contains at most two vertices, i.e., it is an edge or a set containing a doubly-anchored vertex. We will have two other obstructions for constrained outer-1-string representations. Recall that a bridge $e = \{x, y\}$ defines the bridge-components X and Y of $G - e$ with $x \in X$ and $y \in Y$. An X -interval is a maximal sub-sequence of \circlearrowleft that only contains anchors of X . We define Y -interval analogously. See Figure 8 for the following lemma.

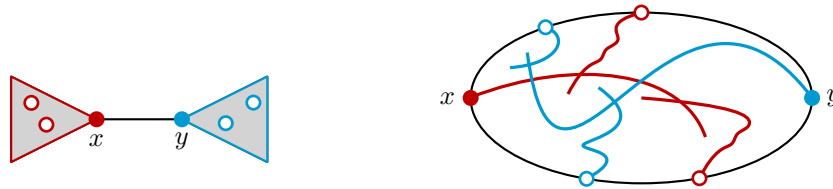
► **Lemma 10** (bridge-obstruction \star). *If (G, \circlearrowleft) has a constrained outer-1-string representation, then no bridge $\{x, y\}$ has three or more X -intervals.*

We use the term *bridge-obstruction* for a bridge that has three or more X -intervals and hence prevents a constrained outer-1-string representation. See Figure 9.

Finally, observe that two adjacent doubly-anchored vertices must be interleaved. The third kind of obstruction generalizes this observation and is based on a *central path* Π with $\ell \geq 0$ edges and hence will be called Π_ℓ -obstruction, or simply *path-obstruction*. See Figure 10. Let $\Pi = \langle v_0, v_1, \dots, v_{\ell-1}, v_\ell \rangle$ be a path, and note that $\ell = 0$ is specifically allowed. For the ends of the central path, there are three variants. In the first variant, there are additionally



■ **Figure 8** Illustration of the proof of Lemma 10. Red curves represent vertices in X , blue curves represent vertices in Y . The union ∂_X and ∂_Y of all red and all blue curves, respectively, is connected and the two sets intersect in $p = \partial(x) \cap \partial(y)$. Removing p from $\partial_X \cup \partial_Y$ yields four connected components (pink and light-blue areas) the anchor of which form X - or Y -intervals in \odot .



■ **Figure 9** An outer-string representation of a bridge-obstruction.

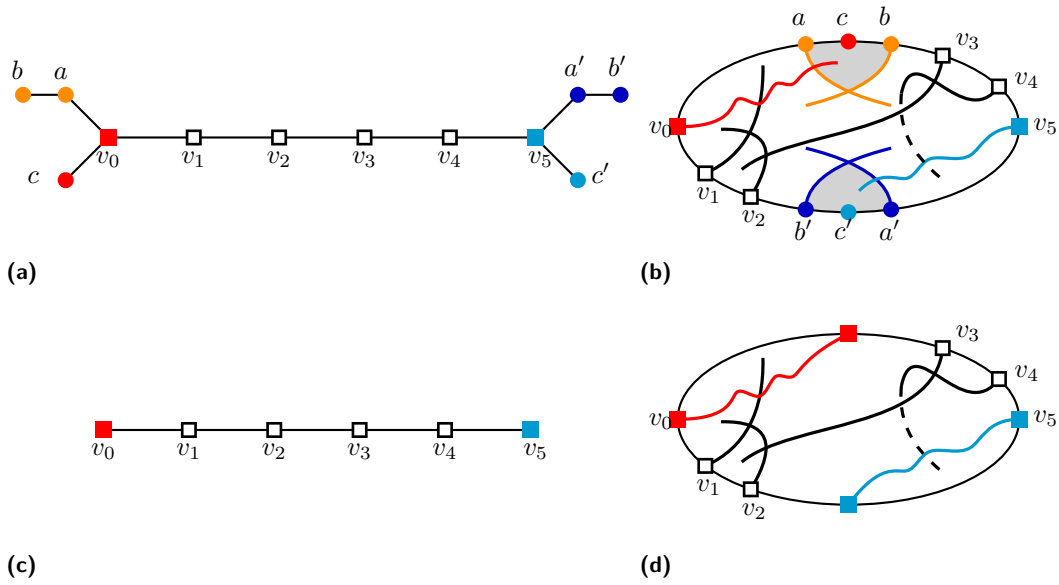
two *bounding paths* $Q = \langle c, v_0, b, a \rangle$ and $Q' = \langle c', v_\ell, b', a' \rangle$ that are disjoint from Π and each other except at v_0 and v_ℓ . The anchor-order \odot is such that in the order induced by the vertices in Π , Q , and Q' satisfies two things: (i) the anchors of the sets $\{a, b, c\}$, $\{a, b, c, v_0\}, \dots$, and $\{a, b, c, v_0, \dots, v_\ell\}$, respectively, appear consecutive and (ii) the pair $\{a, b\}$ and $\{v_0, c\}$ as well as the pair $\{a', b'\}$ and $\{v_\ell, c'\}$ are interleaved. In the second variant, one of the bounding paths, say Q , is replaced by the condition that v_0 is doubly-anchored and that the anchors of v_0 are consecutive in the induced anchor-order. The other conditions on the anchor order remain. The third variant is defined only for $\ell \geq 1$ and is obtained from the second variant by similarly replacing Q' with the requirement that v_ℓ be doubly-anchored with consecutive anchors.

► **Lemma 11** (path-obstruction \star). *If (G, \odot) has a constrained outer-1-string representation, then there is no path-obstruction.*

Clearly, no instance with a constrained outer-1-string representation can contain any of the three obstructions. As our main result for trees, we show that this necessary condition is also sufficient, and furthermore an efficient constructive testing algorithm exists. We prove the following theorem in the next section.

► **Theorem 12.** *An instance (G, \odot) where G is a tree admits a constrained outer-1-string representation if and only if it contains no pair-obstruction, no bridge-obstruction, and no path-obstruction. Furthermore, there is a linear-time algorithm that either finds such an obstruction or returns a constrained outer-1-string representation for (G, \odot) .*

10:10 Constrained Outer-String Representations



■ **Figure 10** (a) The graph of a path-obstruction (b) and a forbidden order of the anchors along with an outer-string representation. (c,d) A doubly-anchored version of a path-obstruction.

4.2 Reduction-Rules

As outlined, we root the tree, preferably at a vertex of degree at least two, and process the vertices in post-order. Furthermore, we maintain that all vertices that have been processed are leaves in the tree or have been deleted altogether. Finally only leaves that are not the root may be doubly-anchored. Let x be the currently processed vertex. If x is already a leaf, then we proceed to the next vertex. So assume that x has children. These children have been processed already, so they are leaves. Let X be the set consisting of x and all its children. When processing x , we will apply a number of reduction rules, each of which yielding one or two smaller instances. In particular, all children of x are deleted eventually.

For the reduction rules we have to argue that they are *correct*, which means two things. First, if the smaller instances have solutions, then so does (G, \circlearrowleft) . Second, if one of the smaller instances contains an obstruction, then so does (G, \circlearrowleft) . The second one implies that if (G, \circlearrowleft) has a solution, then so do the smaller instances: If (G, \circlearrowleft) has a solution, then it has no obstruction, so the smaller instances have no obstructions; by the inductive hypothesis, this implies that the smaller instances have a constrained outer-1-string representation. Our arguments for this will be constructive, which means that there will be an easy algorithm to retrieve the solution or the obstruction from the ones for the smaller instances. A special type of these rules are *obstruction-rules*, where the returned instance is an obstruction that is contained in the instance. In that case the instance is a no-instance and we show how to exhibit the obstruction in the proof of correctness.

► **Rule 1** (leaves). *If x is adjacent to a leaf v that either (i) is singly-anchored and x, v are consecutive in \circlearrowleft or (ii) is doubly-anchored and v, x, v are consecutive in \circlearrowleft , then remove v and its anchors from (G, \circlearrowleft) .*

► **Lemma 13.** *Rule 1 is correct.*

Proof. Given a constrained outer-1-string representation of the smaller instance (G', \circlearrowleft') , we can add a curve for v that is anchored on the correct side of x in case (i) or anchored on both sides of x in case (ii) to obtain a constrained outer-1-string representation of (G, \circlearrowleft) .

If (G', \odot') contains an obstruction, then the exact same subgraph is also an obstruction for (G, \odot) because (G', \odot') is an induced sub-instance of (G, \odot) , and adding more vertices and anchors does not destroy an obstruction. ◀

► **Rule 2** (anchor of x surrounded by X). *If Rule 1 cannot be applied, but there are at least three anchors in \odot , and both anchors immediately before and after the anchor of x in \odot belong to vertices of X , then report that the instance is a no-instance.*

► **Lemma 14.** *Rule 2 is correct.*

Proof. Let u and v be the vertices in X whose anchors are next to the anchor of x in \odot . Since Rule 1 cannot be applied, both u and v are doubly-anchored and $u \neq v$. We have $u \neq x \neq v$ since otherwise x would be doubly-anchored, but x is not a leaf and so cannot be doubly-anchored. We distinguish two cases: If the anchors of u , v , and x are in the cyclic order u, u, x, v, v then this is a Π_2 -obstruction for path $\langle u, x, v \rangle$. If the cyclic order is v, u, x, v, u then u and v are two independent doubly-anchored vertices that are interleaved, so this is a pair-obstruction. ◀

If x is the root and neither Rule 1 nor Rule 2 applies then x is the only vertex that is left. Hence, we get the following rule.

► **Rule 3** (base case). *If x is the root and neither Rule 1 nor Rule 2 applies, then the current (sub-)instance is a yes-instance.*

So for the following rules, we assume that x has a parent y . Vertex y in turn either has a parent, or it is the root and, by the choice of the root, has at least one other child, so y has neighbors other than x . Consider the bridge $e = \{x, y\}$. Let X and Y be the respective bridge components of $G - e$ with $x \in X$ and $y \in Y$. This is consistent with our earlier definition of X . The following rule is clearly correct since it directly exhibits a bridge-obstruction.

► **Rule 4** (three X -intervals). *If the bridge $\{x, y\}$ has three or more X -intervals, then report that the instance is a no-instance.*

So from now on we assume that there are one or two X -intervals. Actually both these cases can be handled at once. We first identify another obvious no-instance.

► **Rule 5** (two X -intervals, x not alone). *If there are two X -intervals, Rule 1 and Rule 2 cannot be applied, and the X -interval containing the anchor of x contains at least two anchors, then report that the instance is a no-instance.*

► **Lemma 15.** *Rule 5 is correct.*

Proof. Assume that the X -interval containing the anchor of x contains at least two anchors. The anchor of x is the first or last anchor in the X -interval, otherwise Rule 2 would apply. Up to symmetry assume that it is the first. Let x' be the vertex of X whose anchor follows x in this X -interval. Since Rule 1 cannot be applied, x' is doubly-anchored. Note that $x' \neq x$ since otherwise x would be doubly-anchored, but x is not a leaf and so not doubly-anchored.

Consider one anchor in each of the two Y -intervals such that the respective vertices $y', y'' \in Y$ are either adjacent or identical. If the two anchors of x' are in the two X -intervals, then x' and $\{y', y''\}$ are independent and interleaved, so a pair-obstruction. If the two anchors of x' are in the same X -interval, then let x'' be any vertex of X whose anchor is in the other X -interval. This is a child of x . If one of y', y'' , say y'' , is y , then we obtain a Π_1 -obstruction with central path $\langle x', x \rangle$, the doubly-anchored vertex x' and the bounding path $\langle x'', x, y, y' \rangle$. Observe that the anchors appear in a suitable order. If neither of y', y'' is y , then $\{y', y''\}$ forms an independent interleaved pair with edge $\{x, x''\}$, so we have a pair-obstruction. ◀

10:12 Constrained Outer-String Representations

If none of the above rules apply, we are in one of two possible situations. Either (a) there is only one X -interval, and x is the first or last vertex in it, or (b) there are two X -intervals, but one of them contains only x . Both situations can be handled as one if in situation (a) we view the range between x and the rest of the X -interval as a “ghost Y -interval” that has no anchors of Y in it. So we always have two X -intervals (one of them contains only the anchor of x) and two Y -intervals (one of which may be a ghost-interval).

► **Rule 6** (split the instance). *If none of the previous rules can be applied, then we split the instance into the graphs G_X and G_Y induced by $X \cup \{y\}$ and $Y \cup \{x\}$, respectively. In G_X , we doubly-anchor y in place of the two Y -intervals. In G_Y , we doubly-anchor x in place of the two X -intervals. All other vertices use the same anchors as in \circ .*

► **Lemma 16** (\star). *Rule 6 is correct.*

Sketch of Proof. A constrained outer-1-string representation for G can be constructed by combining constrained outer-1-string representation for G_X and G_Y at the intersection point of $\partial(x)$ and $\partial(y)$ such that the anchors are in the correct order.

It remains to show how to reconstruct obstructions of (G, \circ) from obstructions of the reduced instances. To this end we have to show that if the new anchor of the vertex x or y is contained in an obstruction of a reduced instance, then we can use omitted vertices to find an obstruction in (G, \circ) . Since G_X is a star centered at x and xyx is a subsequence of the anchor order, it follows that G_X cannot contain an obstruction that uses y .

So, let \mathcal{O} be an obstruction of G_Y that contains x . Let $x' \in X \setminus \{x\}$ be some child of x , preferably a doubly-anchored one. Any independent interleaved pair of G_Y that uses x can be expanded into one of G by adding x' to the set that contains x . Any bridge-obstruction at some bridge e of G_Y is also one in G . In both cases, an anchor of x' can take the place of the second anchor of x in G_Y . It remains to consider the case that \mathcal{O} is a path-obstruction.

If the two anchors of x are consecutive among the anchors of \mathcal{O} , we may assume that x is the endvertex of the central path. If x has a doubly-anchored child, then we obtain a path-obstruction of G by appending x' to x . Otherwise there is no ghost Y -interval. Let $\{y', y''\}$ be an edge with anchors in different Y -intervals. Depending on whether $y' \neq y \neq y''$ or not either $\{x, x'\}$ and $\{y', y''\}$ is a pair-obstruction or G contains a path-obstruction with bounding path $\langle x', x, y, y' \rangle$. See Figure 11a.

If the two anchors of x are not consecutive among the anchors of \mathcal{O} , then x is the endvertex of a bounding path and G contains an interleaved independent pair. See Figure 11. ◀

Note that Rule 6 can *always* be applied if none of the previous rules apply. Observe that G_X is a star and so directly brings us to the base case after rooting G_X at x and applying Rule 1, as well as Rule 2 or Rule 3. Observe further that G_Y is obtained from G by removing the children of x and by doubly-anchoring x . Hence, in G_Y , vertex x has become a leaf as desired and we continue processing the rest of G_Y in post-order. This proves the characterization stated in Theorem 12. For the linear run time, we refer to the full version of the paper.

► **Corollary 17.** *A cyclically ordered path has a constrained outer-1-string representation if and only if there are no two independent edges that are interleaved.*

Proof. A path cannot have a path-obstruction and if a path has a bridge-obstruction then this already implies two independent edges that are interleaved. ◀

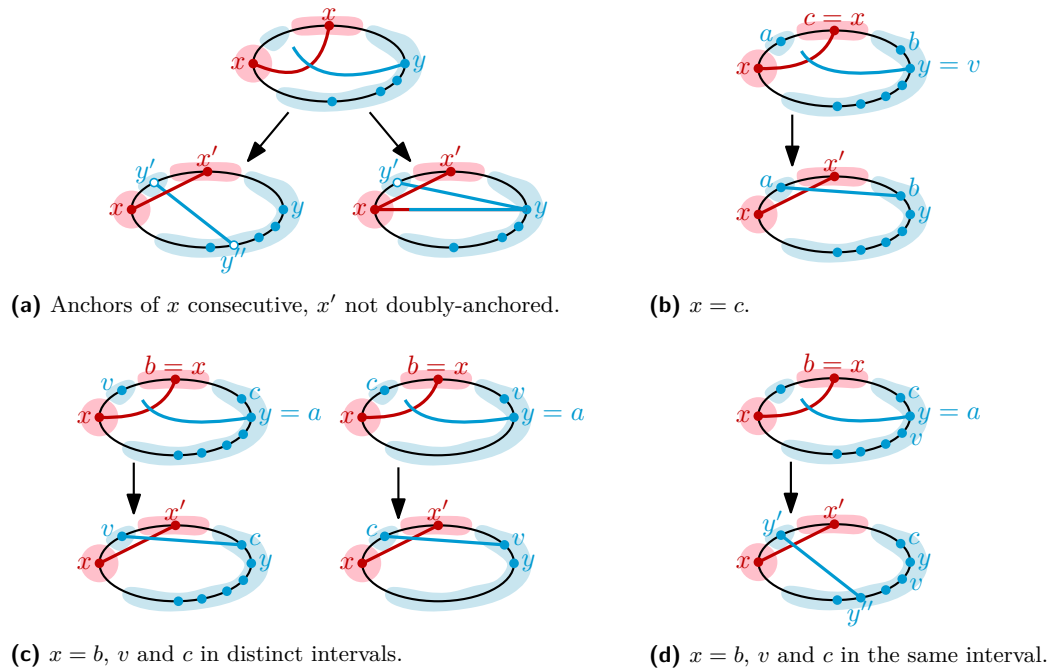


Figure 11 How to reconstruct an obstruction of G from a path-obstruction \mathcal{O} of G_Y . In each case the topmost drawing is a string-representation while in the bottommost drawing the edges are represented as segments. (a) If x is the endvertex of the central path and no child x' of x is doubly-anchored then G contains an interleaved independent pair or $\{x, y\}$ is replaced by the bounding path $\langle x', x, y, y' \rangle$ in \mathcal{O} . (b-d) If x is the endvertex of a bounding path $\langle c, v, a, b \rangle$ and the two anchors of x are not consecutive then G contains an interleaved independent pair.

► **Corollary 18** (*). *A tree with a given cyclic order \circlearrowleft of the vertices that admits a constrained outer-1-string representation also has a constrained U-shaped outer-1-string representation with respect to any linear order induced by \circlearrowleft .*

Sketch of Proof. We follow the construction for constrained outer-1-string representations. Whenever Rule 1 or Rule 6 yields yes-instances, we show how to obtain a constrained U-shaped outer-1-string representation for the original instance. See Figures 12 and 13. We maintain the property that nothing is drawn to the left of the left-most or to the right of the right-most anchor. If the linear order is such that an X -interval is split into a right-most and a left-most part and one of the two parts contains both anchors of a doubly-anchored vertex then the second anchor of x in G_Y is put in this sub-interval. ◀

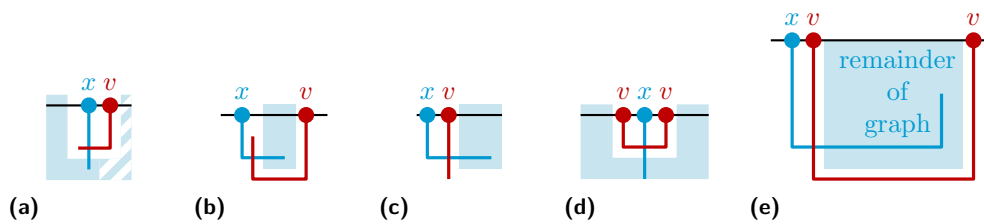
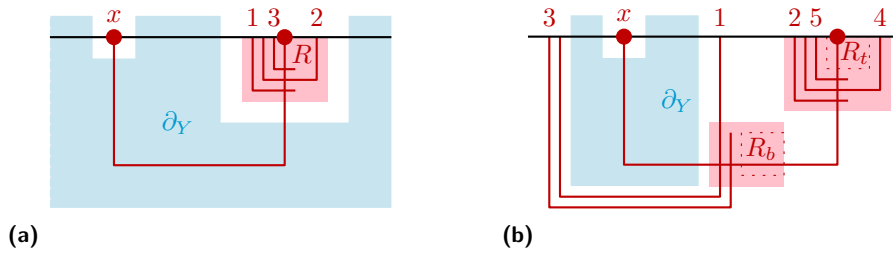
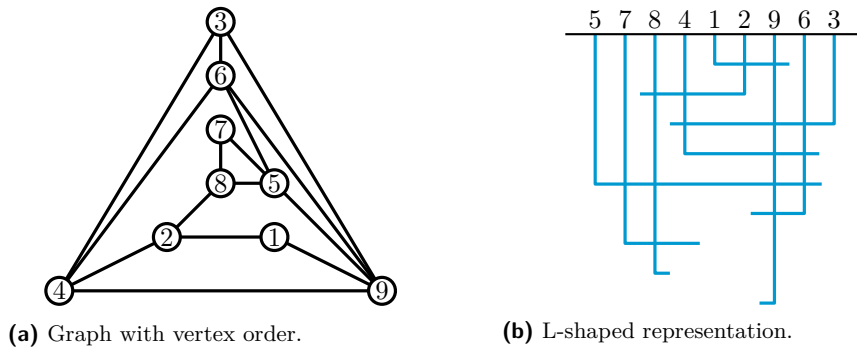


Figure 12 Constructing a U-shaped representation of a tree after the application of Rule 1.

10:14 Constrained Outer-String Representations



■ **Figure 13** Constructing a U-shaped representation of a tree after the application of Rule 6, by inserting $X \setminus \{x\}$ into a representation of G_Y in the order in which we would apply Rule 1 in G_X .



■ **Figure 14** How to construct a constrained L-shaped outer-1-string representation. First order the vertices such that the anchors of the future neighbors of any vertex v are consecutive and next to v . Then draw the L-shaped curves in this order with decreasing y -coordinates of the horizontal part.

5 Constrained L-Shaped Outer-1-String Representations

We now show how to test in quadratic time whether an ordered graph admits a constrained L-shaped outer-1-string representation. See Figure 14.

► **Lemma 19.** *Let $G = (V, E)$ be a graph and let \prec be an order of the vertices. Then (G, \prec) admits a constrained L-shaped outer-1-string representation if and only if the vertices of G can be ordered v_1, \dots, v_n such that for $i = 1, \dots, n$ the set of future neighbors $V_i = \{v_j; j > i \text{ and } \{v_i, v_j\} \in E\}$ of v_i as well as $V_i \cup \{v_i\}$ are consecutive in \prec .*

Proof. Assume that v_1, \dots, v_n is such an order. Let the horizontal line from which the vertices hang have y -coordinate 0. For $i = 1, \dots, n$ we draw the vertical part of $\partial(v_i)$ from 0 to $-i$. The future neighbors V_i are all directly to the left or all directly to the right of v_i . Draw the horizontal part of $\partial(v_i)$ in that direction until the last future vertex is met.

Assume now that there is a constrained L-shaped outer-1-string representation of (G, \prec) . Order the vertices v_1, \dots, v_n according to the y -coordinate of the horizontal part of their curve from top to bottom. Then the curve of all future neighbors of v_i must intersect the horizontal part of $\partial(v_i)$ and all vertical segments of all vertices $v_j, j > i$ must be at least as long as the one of v_i . It follows that V_i must be consecutive and directly next to v_i . ◀

An example of an ordered graph without a constrained L-shaped outer-1-string representation is the path $\langle 1234 \rangle$ with vertex ordering $\langle 2413 \rangle$.

► **Corollary 20.** *It can be tested in $\mathcal{O}(n^2)$ time whether an ordered graph with n vertices admits a constrained L-shaped outer-1-string representation.*

Proof. Let $G = (V, E)$ be a graph with a linear order \prec of its n vertices. For $i = 1, \dots, n$, iteratively check whether there is a vertex, such that the set of its neighbors is directly to the left or to the right of v_i . If so remove v_i and continue. Otherwise report that there is no constrained L-shaped outer-1-string representation.

This can be tested in $\mathcal{O}(n^2)$ time. For a vertex v , let $N(v)$ be the set of neighbors of v . An $N(v)$ -interval is a maximal subsequence of \prec that contains only anchors of vertices of $N(v)$. Let $k(v)$ be the number of $N(v)$ -intervals. We first compute $k(v)$ for each vertex v . This can be done in linear time for each vertex. Observe that we can choose v as a next vertex if and only if $k(v) = 0$ or $k(v) = 1$ and at least one of the neighbors of v in \prec is in $N(v)$. Each time we remove a vertex w , we update $k(v)$ as follows. Decrease $k(v)$ by one if and only if either $w \in N(v)$ and both neighbors of w in \prec are not in $N(v)$ or $w \notin N(v)$ and both neighbors of w in \prec are in $N(v)$. Otherwise do not change $k(v)$. This update can be done in constant time per removed vertex w and remaining vertex v . ◀

6 Constrained Outer-1-String Representations for Simple Cycles

An *extended complement of a 5-cycle* (Figure 15a) is the complement of an anchor-ordered 5-cycle or a subpath $w_1v'vuu'w_2$ of a cycle whose anchors are in the order $w_1uv'u'vw_2$. A cyclically ordered cycle admits a constrained outer-1-string representation if and only if it neither contains a pair of interleaved independent edges nor an extended complement of a 5-cycle:

► **Theorem 21.** *Let $G = (V, E)$ be a simple cycle and let \circlearrowleft be a cyclic order of V . Then the following are equivalent.*

1. (G, \circlearrowleft) has a constrained outer-1-string representation.
2. For every path $\langle u'uvv' \rangle$ of G , at least one among the sequences uv , $uu'v'v$, $uu'v$, or $uv'v$, or their reverse is a subsequence of \circlearrowleft .
3. (G, \circlearrowleft) does not contain two interleaved independent edges nor an extended complement of a 5-cycle.

Proof. We show equivalence of (1) and (3) to (2).

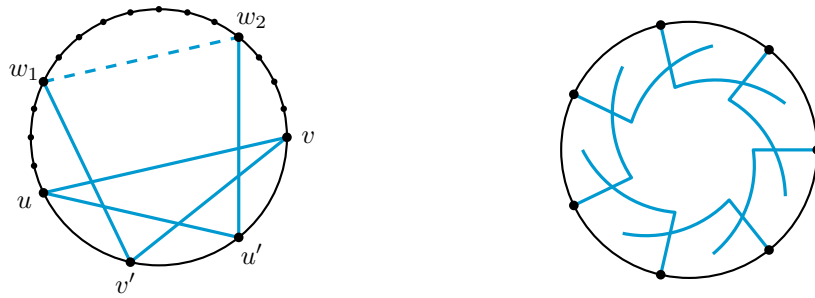
1 \Rightarrow 2: Let P be the path obtained from G after removing u , v , u' , and v' . Let \circlearrowleft_1 and \circlearrowleft_2 be the subsequences obtained by splitting \circlearrowleft at the anchors of u and v . Then the anchors of P are either all in \circlearrowleft_1 or all in \circlearrowleft_2 : If there were two adjacent vertices v_1 and v_2 in P such that the anchor of v_i is in \circlearrowleft_i , $i = 1, 2$ Then $\{u, v\}$ and $\{v_1, v_2\}$ would be two interleaved independent edges.

So assume that the anchors of P are in \circlearrowleft_1 , i.e., \circlearrowleft_2 contains either nothing, or the anchor of u' , of v' , or of both. If \circlearrowleft_1 is not empty and \circlearrowleft_2 contains the anchor of u' and of v' , then u' must be next to u : The curve $\partial(u')$ must intersect $\partial(u)$ and reach the curve of the neighbor of u' in P in \circlearrowleft_1 . Similarly, the curve $\partial(v')$ must intersect $\partial(v)$ and reach the curve of the neighbor of v' in P in \circlearrowleft_1 . This is impossible if the order is $uv'u'v$.

3 \Rightarrow 2: Let again P , \circlearrowleft_1 , and \circlearrowleft_2 be defined as above. As above the anchors of P are either all in \circlearrowleft_1 or all in \circlearrowleft_2 . The fact that there is no extended complement of a 5-cycle forbids the sequence $wuv'u'v$ for any neighbor w of v' or u' other than v or u .

2 \Rightarrow 3: Assume there were two independent interleaved edges $\{u, v\}$ and $\{x, y\}$. Then one subsequence of \circlearrowleft between u and v would contain x and the other y . But neither x nor y is a neighbor of u or v . Assume now that there is an extended complement of a 5-cycle. Then there is the sequence $wuv'u'v$ for some neighbor $w \neq v$ of v' .

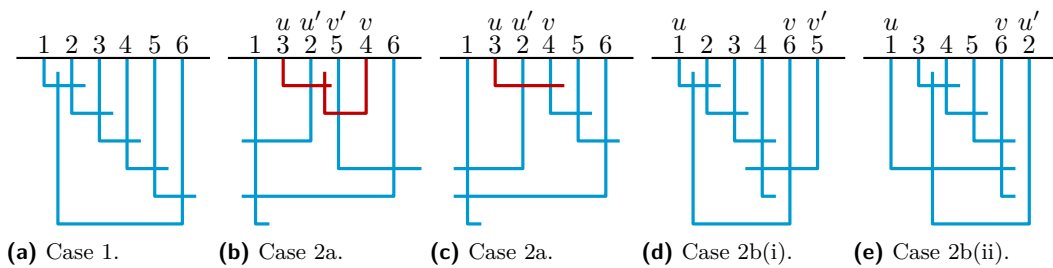
10:16 Constrained Outer-String Representations



(a) Extended complement of a 5-cycle.

(b) Representation of a simple cycle.

■ **Figure 15** a) An obstruction for simple cycles. $w_1 = w_2$ is also possible. b) How to draw a simple cycle if for any edge the anchors of the two endvertices are next to each other.



(a) Case 1.

(b) Case 2a.

(c) Case 2a.

(d) Case 2b(i).

(e) Case 2b(ii).

■ **Figure 16** How to draw a simple cycle as a constrained U-shaped outer-1-string representation. If not all vertices of the cycle are consecutive in the cyclic order of the anchors, we remove one or two vertices (indicated in red), draw the resulting path and reinsert the vertices.

$2 \Rightarrow 1$: If the anchors for any pair of adjacent vertices u, v of G form a subsequence of \odot then there is a constrained outer-1-string representation. See Figure 15b.

Assume now that G contains a path $u'uvv'$ such that \odot contains $uu'v$ as a subsequence (the case $uv'v$ or their reverse being symmetric). Let P be the path obtained from G by removing u . Since Item 2 implies Item 3, there are no two interleaved independent edges. Thus, P has a constrained outer-1-string representation by Corollary 17. Route $\partial(u)$ closely along the border of the disk until it intersects first $\partial(u')$ and then $\partial(v)$.

Assume now that G contains a path $u'uvv'$ such that the anchors are in the order $uu'v'v$. Let P be the path obtained from G by removing u and v . Again by Corollary 17, it follows that P has a constrained outer-1-string representation. Route $\partial(u)$ and $\partial(v)$ closely along the border of the disk until they intersect between the anchor of u' and v' . ◀

The second condition of Theorem 21 can be tested in constant time per edge.

► **Corollary 22.** *It can be tested in linear time, whether a simple cycle with a given cyclic order of the vertices admits a constrained outer-1-string representation.*

Simple cycles do not necessarily have a constrained L-shaped outer-1-string representation even if the respective cyclic order of the anchors admits a constrained outer-1-string representation, consider for example 12345, 13452, or 34127856. The existence of a constrained outer-1-string representation follows from Theorem 21 and the non-existence of an L-shaped outer-1-string representation follows from Lemma 19. However, U-shapes suffice.

► **Corollary 23** (★). *Each simple cycle with a given cyclic order \odot of the vertices that admits a constrained outer-1-string representation also has a constrained U-shaped outer-1-string representation with respect to any linear order induced by \odot .*

Sketch of Proof. We follow “ $2 \Rightarrow 1$ ” in the proof of Theorem 21. We distinguish whether any two adjacent vertices are consecutive in \circlearrowleft (Figure 16a), some adjacent vertices contain some neighbors between them (Figures 16b and 16c), or an adjacent pair contains everything but a neighbor between them (Figures 16d and 16e). ◀

7 Conclusion

We considered outer-string and outer-1-string representations of graphs in which the sequence of the anchors of the vertices was fixed. In particular, we considered outer-string representations of chordal graphs, outer-1-string representations of trees and cycles, as well as L-shaped representations of general graphs. We leave some interesting open problems.

What is the complexity of testing whether a graph has an outer-1-string, a constrained outer-1-string, or a constrained outer-string representation? Can these problems be efficiently solved for cacti or graphs with constant treewidth? Can it be tested efficiently whether an ordered graph admits a constrained U-shaped outer-1-string representation?

A variant of the problem would be to specify for each vertex a set of anchors and to require that these points are within its curve. What can be said about this variant?




References

- 1 Therese Biedl, Ahmad Biniaz, and Martin Derka. On the size of outer-string representations. In *16th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT*, volume 101 of *LIPICs*, pages 10:1–10:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICs.SWAT.2018.10.
- 2 Therese Biedl and Martin Derka. Order-preserving 1-string representations of planar graphs. In *SOFSEM 2017: Theory and Practice of Computer Science*, volume 10139 of *Lecture Notes in Computer Science*, pages 283–294. Springer, 2017. doi:10.1007/978-3-319-51963-0_22.
- 3 Cornelia Dangelmayr, Stefan Felsner, and William T. Trotter. Intersection graphs of pseudosegments: Chordal graphs. *J. Graph Algorithms Appl.*, 14(2):199–220, 2010. doi:10.7155/JGAA.00204.
- 4 James Davies, Tomasz Krawczyk, Rose McCarty, and Bartosz Walczak. Grounded L-graphs are polynomially χ -bounded. *Discret. Comput. Geom.*, 70(4):1523–1550, 2023. doi:10.1007/S00454-023-00592-Z.
- 5 G.A. Dirac. On rigid circuit graphs. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 25:71–76, 1961. doi:doi.org/10.1007/BF02992776.
- 6 G. Ehrlich, S. Even, and R.E. Tarjan. Intersection graphs of curves in the plane. *Journal of Combinatorial Theory, Series B*, 21(1):8–20, 1976. doi:10.1016/0095-8956(76)90022-8.
- 7 F. Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *Journal of Combinatorial Theory Ser. B*, 16:47–56, 1974. doi:10.1016/0095-8956(74)90094-X.
- 8 Martin Charles Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Elsevier, 1980.
- 9 Vít Jelínek and Martin Töpfer. On grounded L-graphs and their relatives. *Electron. J. Comb.*, 26(3):P3.17, 2019. doi:10.37236/8096.
- 10 Jan Kratochvíl. String graphs. In *Graphs and Other Combinatorial Topics, Proceedings Third Czechoslovak Symposium on Graph Theory, Prague*, volume 59 of *Teubner Texte zur Mathematik*, pages 168–172. Teubner, Berlin, 1982.
- 11 Sean McGuinness. On bounding the chromatic number of L-graphs. *Discret. Math.*, 154(1–3):179–187, 1996. doi:10.1016/0012-365X(95)00316-0.
- 12 M. Middendorf and F. Pfeiffer. The max clique problem in classes of string graphs. *Discret. Math.*, 108(1–3):365–372, 1992. doi:10.1016/0012-365X(92)90688-C.

10:18 Constrained Outer-String Representations

- 13 Arkadiusz Pawlik, Jakub Kozik, Tomasz Krawczyk, Michal Lason, Piotr Micek, William T. Trotter, and Bartosz Walczak. Triangle-free intersection graphs of line segments with large chromatic number. *J. Comb. Theory, Ser. B*, 105:6–10, 2014. doi:10.1016/J.JCTB.2013.11.001.
- 14 Alexandre Rok and Bartosz Walczak. Outerstring graphs are χ -bounded. *SIAM J. Discret. Math.*, 33(4):2181–2199, 2019. doi:10.1137/17M1157374.
- 15 F. W. Sinden. Topology of thin film RC circuits. *Bell System Tech. J.*, 45:1639–1662, 1966. doi:10.1002/j.1538-7305.1966.tb01713.x.
- 16 James R. Walter. Representations of chordal graphs as subtrees of a tree. *Journal of Graph Theory*, 2(3):265–267, 1978.

Monotone Arc Diagrams with Few Biarcs

Steven Chaplick   




Maastricht University, The Netherlands

Henry Förster   

Universität Tübingen, Germany

Michael Hoffmann   

Department of Computer Science, ETH Zürich, Switzerland

Michael Kaufmann   

Universität Tübingen, Germany

Abstract

We show that every planar graph has a monotone topological 2-page book embedding where at most $(4n - 10)/5$ (of potentially $3n - 6$) edges cross the spine, and every edge crosses the spine at most once; such an edge is called a *biarc*. We can also guarantee that all edges that cross the spine cross it in the same direction (e.g., from bottom to top). For planar 3-trees we can further improve the bound to $(3n - 9)/4$, and for so-called Kleetopes we obtain a bound of at most $(n - 8)/3$ edges that cross the spine. The bound for Kleetopes is tight, even if the drawing is not required to be monotone. A *Kleetope* is a plane triangulation that is derived from another plane triangulation T by inserting a new vertex v_f into each face f of T and then connecting v_f to the three vertices of f .

2012 ACM Subject Classification Mathematics of computing → Combinatorics; Mathematics of computing → Graph theory; Human-centered computing → Graph drawings

Keywords and phrases planar graph, topological book embedding, monotone drawing, linear layout

Digital Object Identifier 10.4230/LIPIcs.GD.2024.11

Related Version *Full Version:* <https://doi.org/10.48550/arXiv.2408.14299>

Funding *Steven Chaplick:* supported by DFG grant WO 758/11-1.

Michael Hoffmann: supported by the Swiss National Science Foundation within the collaborative D-A-CH project *Arrangements and Drawings* as SNSF project 200021E-171681.

Acknowledgements This work started at the workshop on *Graph and Network Visualization* (GNV 2017) in Heiligkreuztal, Germany. Preliminary results were presented at the 36th European Workshop on Computational Geometry (EuroCG 2020). We thank Stefan Felsner and Stephen Kobourov for useful discussions.

1 Introduction

Arc diagrams (Figure 1) are drawings of graphs that represent vertices as points on a horizontal line, called *spine*, and edges as *arcs*, consisting of a sequence of halfcircles centered on the spine. A *proper arc* consists of one halfcircle. In *proper arc diagrams* all arcs are proper (see Figure 1a). In *plane arc diagrams* no two edges cross. Note that proper plane arc diagrams are also known as *2-page book embeddings*. Bernhard and Kainen [1] characterized the graphs that admit proper plane arc diagrams: subhamiltonian planar graphs, i.e., subgraphs of planar graphs with a Hamiltonian cycle. In particular, non-Hamiltonian maximal planar graphs do not admit proper plane arc diagrams.

To represent all planar graphs as a plane arc diagram, it suffices to allow each edge to cross the spine once [12, 13]. The resulting arcs composed of two halfcircles are called *biarcs* (see Figure 1b). Additionally, all edges can be drawn as *monotone* curves w.r.t. the spine [7, 10]; such a drawing is called a *monotone topological (2-page) book embedding* (see



© Steven Chaplick, Henry Förster, Michael Hoffmann, and Michael Kaufmann; licensed under Creative Commons License CC-BY 4.0

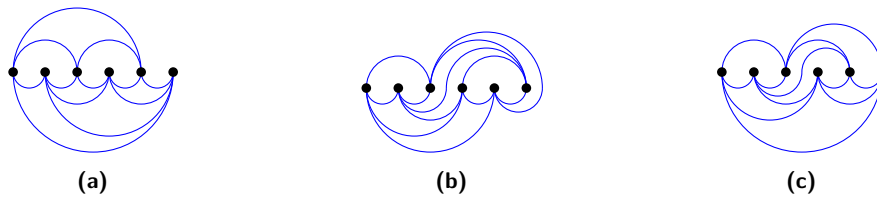
32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 11; pp. 11:1–11:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** Arc diagrams of the octahedron: (a) proper, (b) general, and (c) monotone.

Figure 1c). A monotone biarc is either *up-down* or *down-up*, depending on whether the left halfcircle is drawn above or below the spine, respectively. Note that a *monotone topological book embedding* is not necessarily a book embedding, even though the terminology suggests it.

In general, biarcs are needed, but *many* edges can be drawn as proper arcs. Cardinal, Hoffmann, Kusters, Tóth, and Wettstein [2, 3] gave bounds on the required number of biarcs by showing that every planar graph on $n \geq 3$ vertices admits a plane arc diagram with at most $\lfloor (n-3)/2 \rfloor$ biarcs and how this quantity is related to the diameter of the so-called combinatorial flip graph of triangulations. However, they allow general, not necessarily monotone biarcs. When requiring biarcs to be monotone, Di Giacomo, Didimo, Liotta, and Wismath [7, 10] gave an algorithm to construct a monotone plane arc diagram that may create close to $2n$ biarcs for an n -vertex planar graph. Cardinal, Hoffmann, Kusters, Tóth, and Wettstein [2, 3] improved this bound to at most $n-4$ biarcs.

As a main result, we improve the upper bound on the number of monotone biarcs.

► **Theorem 1.** *Every n -vertex planar graph admits a plane arc diagram with at most $\lfloor \frac{4}{5}n \rfloor - 2$ biarcs that are all down-up monotone.*

It is an intriguing open question if there is a *monotonicity penalty*, that is, is there a graph G and a plane arc diagram of G with k biarcs such that every monotone plane arc diagram of G has strictly more than k biarcs? No such graph is known, even if for the stronger condition that all biarcs are monotone of the same type, such as down-up.

For general plane arc diagrams, in some cases $\lfloor (n-8)/3 \rfloor$ biarcs are required [2, 3]. The (only) graphs for which this lower bound is known to be tight belong to the class of Kleetopes. A *Kleetope* is a plane triangulation¹ that is derived from another plane triangulation T by inserting a new vertex v_f into each face f of T and then connecting v_f to the three vertices of f . One might think that Kleetopes are good candidates to exhibit a monotonicity penalty. However, we show that this is not the case, but instead the known lower bound is tight.

► **Theorem 2.** *Every Kleetope on n vertices admits a monotone plane arc diagram with at most $\lfloor (n-8)/3 \rfloor$ biarcs, where every biarc is down-up.*

So, to discover a monotonicity penalty we have to look beyond Kleetopes. We investigate another class of planar graphs: planar 3-trees. A *planar 3-tree* is built by starting from a (combinatorial) triangle. At each step we insert a new vertex v into a (triangular) face f of the graph built so far, and connect v to the three vertices of f . As a third result we give an improved upper bound on the number of monotone biarcs needed for planar 3-trees.

► **Theorem 3.** *Every planar 3-tree admits a plane arc diagram with at most $\lfloor \frac{3}{4}(n-3) \rfloor$ biarcs that are all down-up monotone.*

¹ A *plane triangulation* is a triangulation associated with a combinatorial embedding. For the scope of this paper, we also consider the outer face to be fixed.

Related work. Giordano, Liotta, Mchedlidze, Symvonis, and Whitesides [11] showed that every upward planar graph admits an *upward topological book embedding* in which all edges are either proper arcs or biarcs. These embeddings are also monotone arc diagrams that respect the orientations of the edges and use at most one spine crossing per edge. One of their directions for future work is to minimize the number of spine crossings. We believe that our approach for undirected graphs may provide some insights. Everett, Lazard, Liotta, and Wismath [8, 9] used monotone arc diagrams to construct small universal point sets for 1-bend drawings of planar graphs, heavily using the property that all biarcs have the same *shape* (e.g., all are down-up biarcs). This result has been extended by Löffler and Tóth [14] by restricting the set of possible bend positions. They use the existence of monotone arc diagrams with at most $n - 4$ biarcs to build universal point sets of size $6n - 10$ (vertices and bend points) for 1-bend drawings of planar graphs on n vertices. Using Theorem 1, we can decrease the number of points by about $n/5$.

Outline. We sketch the proof of Theorem 1 in Sections 2–4, then in Section 5 the proof of Theorem 2, and finally, in Section 6 the proof of Theorem 3. Due to space constraints, some proofs are provided in the full version only.

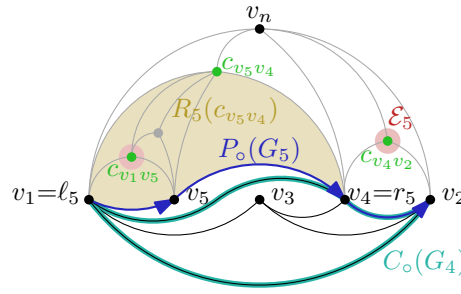
2 Overview of our Algorithm

To prove Theorem 1 we describe an algorithm to incrementally construct an arc diagram for a given planar graph $G = (V, E)$ on $n \geq 4$ vertices. Without loss of generality we assume that G is a combinatorial *triangulation*, that is, a maximal planar graph. Further, we consider G to be embedded, that is, G is a *plane* graph. As every triangulation on $n \geq 4$ vertices is 3-connected, by Whitney’s Theorem selecting one facial triangle as the *outer face* embeds it into the plane. This choice also determines a unique outer face for every biconnected subgraph. For a biconnected plane graph G denote the outer face (an open subset of \mathbb{R}^2) by $F_o(G)$ and denote by $C_o(G)$ the cycle that bounds $F_o(G)$. A plane graph is *internally triangulated* if it is biconnected and every inner face is a triangle. A central tool for our algorithm is the notion of a canonical ordering [5, 6]. Consider an internally triangulated plane graph G on the vertices v_1, \dots, v_n , and let $V_k = \{v_j : 1 \leq j \leq k\}$. The sequence v_1, \dots, v_n forms a *canonical ordering* for G if the following conditions hold for every $i \in \{3, \dots, n\}$:

- (C1) the induced subgraph $G_i = G[V_i]$ is internally triangulated;
- (C2) the edge v_1v_2 is an edge of $C_o(G_i)$; and
- (C3) for all j with $i < j \leq n$, we have $v_j \in F_o(G_i)$.

Every internally triangulated plane graph admits a canonical ordering, for any starting pair v_1, v_2 where v_1v_2 is an edge of $C_o(G)$ [5, 6]. Moreover, such an ordering can be computed by iteratively selecting v_i , for $i = n, \dots, 3$, to be a vertex of $C_o(G_i) \setminus \{v_1, v_2\}$ that is not incident to a chord of $C_o(G_i)$. This computation can be done in $O(n)$ time [4]. In general, a triangulation may admit many canonical orderings. We will use this freedom to adapt the canonical ordering we work with to our needs. To this end, we compute a canonical ordering for G incrementally, starting with v_1, v_2, v_3 , where v_1v_2 is an arbitrary edge of $C_o(G)$, and v_3 is the unique vertex of G such that $v_1v_2v_3$ bounds a triangular face of G and v_3 is not a vertex of $C_o(G)$. A canonical ordering v_1, \dots, v_i for G_i , where $3 \leq i \leq n$, is *extensible* if there exists a sequence v_{i+1}, \dots, v_n such that v_1, \dots, v_n is a canonical ordering for G .

► **Lemma 4.** *A canonical ordering v_1, \dots, v_i for G_i is extensible $\iff V \setminus V_i \subset F_o(G_i)$.*



■ **Figure 2** Overview of notation used throughout the paper.

We set up some terminology used throughout the paper; refer also to Figure 2. Consider an extensible canonical ordering v_1, \dots, v_i for G_i and some vertex $v \in V \setminus V_i$. Let $P_o(G_i)$ denote the path $C_o(G_i) - v_1 v_2$ and direct it from v_1 to v_2 . As G_i is an induced subgraph of the plane graph G and $v \in F_o(G_i)$ (by extensibility), all neighbors of v in G_i are on $P_o(G_i)$. We associate a planar region $R_i(v)$ to v as follows. If $d_i(v) = \deg_{G_i}(v) \leq 1$, then $R_i(v) = F_o(G_i)$; else, let $R_i(v)$ be the open bounded region bounded by the simple closed curve formed by the part of $P_o(G_i)$ between ℓ and r together with the edges ℓv and $r v$ of G , where ℓ and r are the first and last, respectively, neighbor (in G) of v on $P_o(G_i)$. We partially order the vertices in $V \setminus V_i$ by defining $v < v'$ if $R_i(v) \subseteq R_i(v')$.

A vertex $v \in V \setminus V_i$ is *eligible* (for G_i) if setting $v_{i+1} = v$ yields an extensible canonical ordering v_1, \dots, v_{i+1} for G_{i+1} . Denote the set of vertices eligible for G_i by \mathcal{E}_i . Let $e = uv$ be an arbitrary edge of $P_o(G_i)$, for $i < n$. As G is a triangulation, there exists a unique vertex $c_e \in V \setminus V_i$ such that uwc_e bounds a triangular face of G ; we say that c_e *covers* e . Given a canonical ordering v_1, \dots, v_n , vertex v_i covers exactly the edges of $P_o(G_{i-1})$ that are not on $P_o(G_i)$. Similarly, we say that v_i covers a vertex v of $P_o(G_{i-1})$ if v is not part of $P_o(G_i)$. The following observations are direct consequences of these definitions and Lemma 4.

► **Corollary 5.** *A vertex $v \in V \setminus V_i$ is eligible $\iff R_i(v) \cap V = \emptyset \iff R_i(v) \cap \mathcal{E}_i = \emptyset$.*

While computing a canonical ordering v_1, \dots, v_n , we also maintain an arc diagram, for short, *diagram* of G_i . This diagram must satisfy certain properties to be considered valid, as detailed below. In some cases we apply induction to handle a whole induced subgraph of G , for instance, within a (separating) triangle, at once. As a result, in certain steps, subgraph G_i may not correspond to a valid arc diagram.

Every vertex v_i arrives with $1 - \chi$ credits, for some constant $\chi \geq 0$.² For these credits we can either create biarcs (at a cost of one credit per biarc), or we place them on edges of the outer face of the diagram for later use. The *costs* $\text{cost}(D)$ of a diagram D is the sum of credits on its edges. An edge in the diagram can be one of three different types: *mountain* (proper arc above the spine), *pocket* (proper arc below the spine), or *down-up biarc*. So the diagram is determined by (1) the spine order (left-to-right) of the vertices and crossings along with (2) for every edge, its type and number of credits. The *lower envelope* of a diagram consists of all vertices and edges that are vertically visible from below, that is, there is no other vertex or edge of the diagram vertically below it. Analogously, the *upper envelope* consists of all vertices and edges that are vertically visible from above.

² For Theorem 1 we will set $\chi = 1/5$. But we think it is instructive to keep χ as a general constant in our argumentation. For instance, this way it is easier to see in which cases our analysis is tight.

A diagram for v_1, \dots, v_i and $i \in \{3, \dots, n\}$, is *valid* if it satisfies the following invariants:

- (X1) Every edge is either a proper arc or a down-up biarc. Every edge on the upper envelope is a proper arc.
- (X2) Every mountain whose left endpoint is on $C_o(G_i) \setminus \{v_2\}$ carries one credit.
- (X3) Every biarc carries (that is, is paid for with) one credit.
- (X4) Every pocket on $P_o(G_i)$ carries χ credits³.

Moreover, a valid drawing is *extensible* if it also satisfies

- (X5) Vertex v_1 is the leftmost and v_2 is the rightmost vertex on the spine. Edge v_1v_2 forms the lower envelope of $C_o(G_i)$. The edges of $P_o(G_i)$ form the upper envelope.

To prove Theorem 1 it suffices to prove the following.

► **Lemma 6.** *Let G be a maximal plane graph on $n \geq 3$ vertices, let v_1, \dots, v_i be an extensible canonical ordering for G_i , for some $3 \leq i < n$, and let D be an extensible arc diagram for G_i . Then, for any $\chi \leq \frac{1}{5}$, D can be extended to an extensible arc diagram D' for G with $\text{cost}(D') \leq \text{cost}(D) + (n-i)(1-\chi) + \xi$, for some $\xi \leq 2\chi$.*

Proof of Theorem 1 assuming Lemma 6. We may assume $n \geq 4$, as the statement is trivial for $n \leq 3$. Let $C_o(G) = v_1v_2v_n$, and let v_3 be the other (than v_n) vertex that forms a triangle with v_1v_2 in G . Then v_1, v_2, v_3 is an extensible canonical ordering for G_3 in G . To obtain an extensible diagram D for G_3 , place $v_1v_3v_2$ on the spine in this order from left to right. All three edges are drawn as pockets so that v_1v_2 is below v_1v_3 and v_3v_2 . On the latter two edges we put χ credits each. It is easily verified that D is extensible and $\text{cost}(D) = 2\chi$. By Lemma 6 we obtain an extensible diagram D' for G with $\text{cost}(D') \leq 2\chi + (n-3)(1-\chi) + 2\chi = n(1-\chi) + 7\chi - 3$. Setting $\chi = 1/5$ yields $\text{cost}(D') \leq \frac{4}{5}n - \frac{8}{5}$. As v_n is incident to a mountain on the outer face by (X5) which carries a credit by (X2), $\text{cost}(D') - 1$ is an upper bound for the number of biarcs in D' and the theorem follows. ◀

We can avoid the additive term ξ in Lemma 6 by dropping (X5) for D' :

► **Lemma 7.** *Let G be a maximal plane graph on $n \geq 4$ vertices, let v_1, \dots, v_i be an extensible canonical ordering for G_i , for $3 \leq i < n$, and let D be an extensible arc diagram for G_i . Then, for any $\chi \leq \frac{1}{5}$, D can be extended to a valid arc diagram D' for G such that (1) $\text{cost}(D') \leq \text{cost}(D) + (n-i)(1-\chi)$, (2) Vertex v_1 is the leftmost and v_n is the rightmost vertex on the spine. The mountain v_1v_n forms the upper envelope, and the pocket v_1v_2 along with edge v_2v_n forms the lower envelope of D' , and (3) v_2v_n is not a pocket.*

3 Default vertex insertion

We prove both Lemma 6 and Lemma 7 together by induction on n . For Lemma 6, the base case $n = 3$ is trivial, with $D' = D$. For Lemma 7, the base case is $n = 4$ and $i = 3$. We place v_4 as required, to the right of v_2 , and draw all edges incident to v_4 as mountains. To establish (X2) it suffices to put one credit on v_1v_4 because v_3 is covered by v_4 and mountains with left endpoint v_2 are excluded in (X2). The edge of D with left endpoint v_3 is covered by v_4 ; thus, we can take the at least χ credits on it. The invariants (X1), (X3), and (X4) are easily checked to hold, as well as the statements in Lemma 7.

³ As in the Greek word for pocket money: $\chi\alpha\rho\tau\zeta\upsilon\lambda\acute{\iota}\alpha\iota$.

11:6 Monotone Arc Diagrams with Few Biarcs

In order to describe a generic step of our algorithm, assume that we already have an extensible arc diagram for G_{i-1} , for $i = 4, \dots, n$. We have to select an eligible vertex $V_i \in V \setminus V_{i-1}$ and add it using at most $1 - \chi$ credits obtaining an extensible diagram for G_i . In this section we discuss some cases where a suitable vertex exists that can easily be added to the arc diagram, using what we call a *default insertion*. Let v_i be any vertex in \mathcal{E}_{i-1} .

We call the sequence of (at least one) edges of $P_o(G_{i-1})$ between the leftmost neighbor ℓ_i of v_i and the rightmost neighbor r_i of v_i the *profile* $\text{pr}(v_i)$ of v_i . By (X1) each edge on the profile is a pocket or a mountain, i.e., writing \smile and \frown for pocket and mountain, respectively, each profile can be described by a string over $\{\smile, \frown\}$. For a set A of characters, let A^* , A^k and A^+ denote the set of all strings, all strings of length exactly k and all strings of length at least one, respectively, formed by characters from A . Let d_i denote the degree of v_i in G_i .

► **Lemma 8.** *If $\text{pr}(v_i) \in \{\smile, \frown\}^* \smile \smile^*$, then we can insert v_i and use ≤ 1 credit to obtain an extensible arc diagram for G_i . At most $1 - \chi$ credits suffice, unless $\text{pr}(v_i) = \smile \smile$.*

Proof Sketch. We place v_i into the rightmost pocket $p_\ell p_r$ it covers, draw $p_\ell v_i$ and $v_i p_r$ as pockets and all other new edges as mountains; see Figure 3. We take the χ credits from $p_\ell p_r$. If $d_i = 2$, then we place χ credits on each of the two pockets incident to v_i so as to establish (X4), for a cost of $\chi \leq 1 - \chi$, assuming $\chi \leq 1/2$.

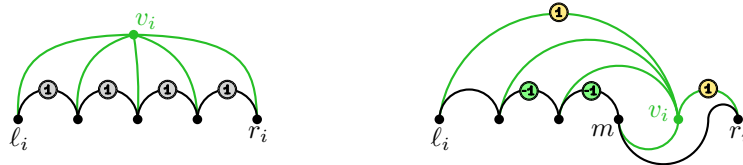


■ **Figure 3** Inserting a vertex v_i into a pocket, using $1 - \chi$ credits (Lemma 8).

For $d_i \geq 3$ each new mountain m from v_i to the right covers a mountain m' of $P_o(G_{i-1})$ whose left endpoint is covered by v_i . Thus, we can take the credit from m' and place it on m . Among all mountains from v_i to the left, a credit is needed for the leftmost one only. If there is such a mountain, then we do not need the χ credits on $p_\ell v_i$. And if v_i covers two or more edges to the left of p_ℓ , we gain at least χ credits from the rightmost such edge. ◀

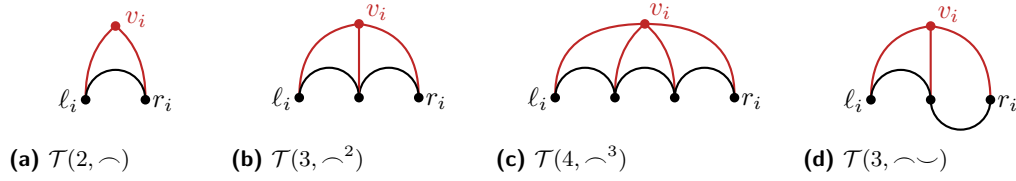
It is more difficult to insert v_i if it covers mountains only, at least if d_i is small. But if the degree of v_i is large, then we can actually gain credits by inserting v_i (see Figure 4).

► **Lemma 9.** *If $\text{pr}(v_i) \in \frown^+$ and $d_i \geq 5$, then we can insert v_i and gain at least $d_i - 5$ credits to obtain an extensible arc diagram for G_i .*



■ **Figure 4** Inserting a vertex v_i into mountains, using $5 - d_i$ credits (Lemma 9).

An eligible vertex is *problematic* if it is of one of the four specific types depicted in Figure 5. Using Lemmas 8 and 9 we insert vertices using at most $1 - \chi$ credits per vertex, unless all eligible vertices are problematic. This specific situation is discussed in the next section.



■ **Figure 5** The four types of problematic vertices where default insertion fails.

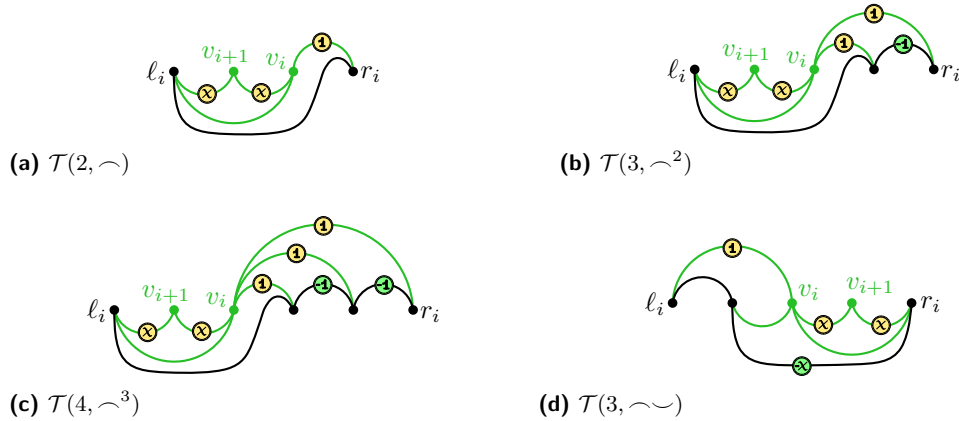
4 When default insertion fails

In this section we discuss how to handle the case where all eligible vertices are problematic, that is, they cannot be handled by our default insertion. Let v be an arbitrary vertex in \mathcal{E}_{i-1} , and let ℓ and r denote the leftmost and rightmost neighbor of v on $P_o(G_{i-1})$, respectively.

A special case arises if $v = v_n$ is the last vertex of the canonical ordering. This case is easy to resolve, see Appendix C in the full version for details. Otherwise, we have $i < n$ and pick a *pivot vertex* $p(v)$ as follows: If v is $\mathcal{T}(3, -\sim)$ we set $p(v) = r$ and say that v has *right pivot type*, in the three remaining cases we set $p(v) = \ell$ and say that v has *left pivot type*. Let $pc(v) \in V \setminus V_i$ denote the unique vertex that covers the *pivot edge* $vp(v)$.

► **Lemma 10.** *Assume there is a vertex $v \in \mathcal{E}_{i-1}$ such that $pc(v)$ has only one neighbor on $P_o(G_{i-1})$. Then we can set $v_i = v$ and $v_{i+1} = p(v)$ and spend at most $1 + 2\chi$ credits to obtain an extensible arc diagram for G_{i+1} .*

Proof. The resulting diagram is shown in Figure 6. The costs to establish are $1 + \chi$ for $\mathcal{T}(3, -\sim)$ and $1 + 2\chi$ for the other types. Note that $1 + 2\chi \leq 2(1 - \chi)$, for $\chi \leq 1/4$. ◀

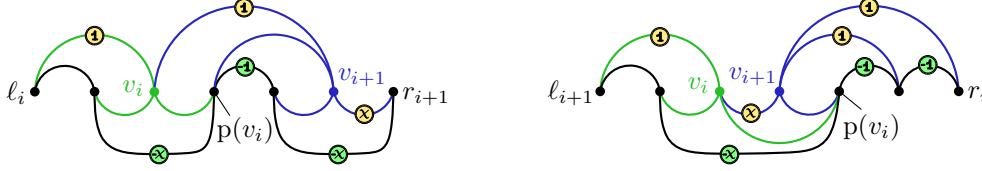


■ **Figure 6** Insertion of v_i and v_{i+1} if $v_{i+1} = pc(v_i)$ has degree two in G_{i+1} .

► **Lemma 11.** *Assume that there are $v, v' \in \mathcal{E}_{i-1}$ such that $pc(v) = v'$ and at least one of v, v' has right pivot type. Then we can set $v_i = v$ and $v_{i+1} = v'$ and spend at most one credit to obtain an extensible arc diagram for G_{i+1} .*

Proof. If both v and v' have right pivot type, then we use the diagram shown in Figure 7 (left). The costs are $1 - \chi \leq 2(1 - \chi)$, for $\chi \leq 1$. Otherwise, one of v, v' has left pivot type and the other has right pivot type, then $p(v) = p(v')$ and $pc(v') = v$. As the roles of v and v' are

symmetric, we may assume w.l.o.g. that v has right pivot type and v' has left pivot type. We use the diagram shown in Figure 7 (right) for the case where v' is $\mathcal{T}(3, \curvearrowright^2)$; other types are handled analogously. The costs to establish the invariants are $1 \leq 2(1 - \chi)$, for $\chi \leq 1/2$. ◀



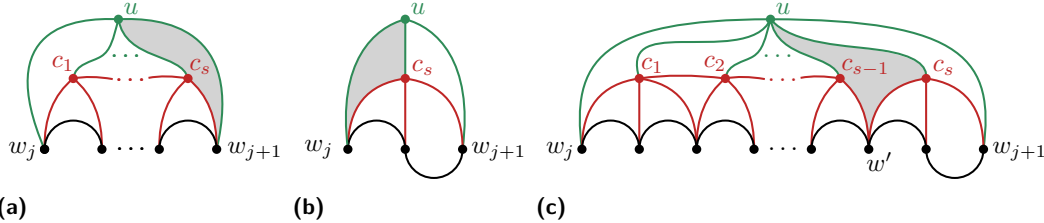
■ **Figure 7** Insertion of v_i and $v_{i+1} = pc(v_i) \in \mathcal{E}_{i-1}$ if v_i has right pivot type.

If we can apply one of Lemmas 10 and 11, we make progress by inserting two vertices v_i and v_{i+1} . Hence, from now on, we assume that neither of Lemmas 10 and 11 can be applied. Our goal in the remainder of this section is to show that in this case we can find a vertex u that is not eligible but sufficiently close to being eligible – in a way described in the following – that we can aim to insert u next, along with some other vertices.

More specifically, the vertex u has neighbors w_1, \dots, w_k on $P_o(G_{i-1})$, for $k \geq 2$, and each subregion X_j of $R_{i-1}(u)$ bounded by the edges uw_j and uw_{j+1} has a particularly simple structure. First of all, there exists an integer $s = s(X_j)$ such that we have $X_j \cap \mathcal{E}_{i-1} = \{c_1, \dots, c_s\}$, and every c_ℓ , for $1 \leq \ell \leq s$, is adjacent to u in G . We distinguish three types of regions, depending on whether X_j contains eligible vertices of left, right, or both pivot types.

Left-pivot region. (see Figure 8a)

- Every c_ℓ , for $1 \leq \ell \leq s$, has left pivot type.
- We have $pc(c_1) = u$ and $pc(c_\ell) = c_{\ell-1}$, for all $2 \leq \ell \leq s$.
- All vertices in $(V \setminus \mathcal{E}_{i-1}) \cap X_j$ lie inside the face bounded by $uc_s w_{j+1}$.



■ **Figure 8** Structure of regions that our to-be-inserted-next vertex u spans with $P_o(G_{i-1})$. All eligible vertices (shown red) are adjacent to u , all other vertices lie inside the shaded region.

Right-pivot region. (see Figure 8b)

- We have $s = 1$, the vertex c_1 has right pivot type, and $pc(c_1) = u$.
- All vertices in $(V \setminus \mathcal{E}_{i-1}) \cap X_j$ lie inside the face bounded by $uw_j c_1$.

Both-pivot region. (see Figure 8c)

- Every c_ℓ , for $1 \leq \ell \leq s - 1$, has left pivot type and c_s has right pivot type.
- We have $pc(c_1) = pc(c_s) = u$ and $pc(c_\ell) = c_{\ell-1}$, for all $2 \leq \ell \leq s - 1$.
- The rightmost neighbor of c_{s-1} on $P_o(G_{i-1})$ is the same as the leftmost neighbor of c_s on $P_o(G_{i-1})$; denote this vertex by w' .
- All vertices in $(V \setminus \mathcal{E}_{i-1}) \cap X_j$ lie inside the quadrilateral $uc_{s-1}w'c_s$.

How to select u . In the remainder of this section we will sketch how to select a suitable vertex u such that all regions spanned by u and $P_o(G_{i-1})$ have the nice structure explained above. The first part of the story is easy to tell: We select u to be a minimal (w.r.t. \prec) element of the set $\mathcal{U} := \{\text{pc}(v) : v \in \mathcal{E}_{i-1}\} \setminus \mathcal{E}_{i-1}$. Such a vertex always exists because

► **Lemma 12.** *We have $\mathcal{U} \neq \emptyset$.*

As there is a vertex $v \in \mathcal{E}_{i-1}$ with $u = \text{pc}(v)$, we know that $u \in \mathcal{U}$ has at least one neighbor on $P_o(G_{i-1})$, which is $p(v)$. By Lemma 10 we may assume $d_{i-1}(u) \geq 2$. Let w_1, \dots, w_k denote the sequence of neighbors of u along $P_o(G_{i-1})$. The edges uw_j , for $2 \leq j \leq k-1$, split $R_{i-1}(u)$ into $k-1$ subregions; let X_j denote the (open) region bounded by w_juw_{j+1} and the part of $P_o(G_{i-1})$ between w_j and w_{j+1} , for $1 \leq j < k$.

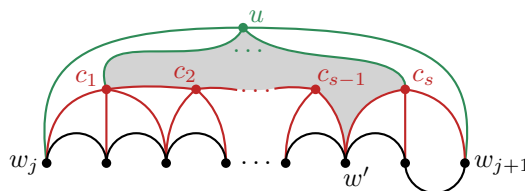
► **Lemma 13.** *In every region X_j , for $1 \leq j < k$, there is at most one eligible vertex v of each pivot type for which $\text{pc}(v) = u$.*

► **Lemma 14.** *In every region X_j , at most one eligible vertex has right pivot type. If there exists a vertex $v \in X_j \cap \mathcal{E}_{i-1}$ that has right pivot type, then $\text{pc}(v) = u$.*

► **Lemma 15.** *Let Q denote the set of vertices in $X_j \cap \mathcal{E}_{i-1}$ that have left pivot type. If $Q \neq \emptyset$, then the vertices in Q form a sequence x_1, \dots, x_q , for some $q \geq 0$, such that $x_j = \text{pc}(x_{j+1})$, for $1 \leq j \leq q-1$, and $\text{pc}(x_1) = u$.*

► **Lemma 16.** *Let $e \in P_o(G_{i-1}) \cap \partial X_j$, for some $1 \leq j < k$, and let $c_e \in V \setminus V_{i-1}$ denote the vertex that covers e . Then either $c_e = u$ or $c_e \in \mathcal{E}_{i-1}$.*

We process the regions X_1, \dots, X_{k-1} together with u . Consider region X_j such that $X_j \cap V \neq \emptyset$, and denote $E_j = P_o(G_{i-1}) \cap \partial X_j$. By Lemma 16 the vertices that cover one or more edges of E_j are exactly the vertices in $\mathcal{E}_{i-1} \cap X_j$. Thus, we can order these vertices from left to right, according to the edge(s) in E_j they cover. Denote this sequence by c_1, \dots, c_s . By Lemma 14 the only vertex in $X_j \cap V$ that may have right pivot type is c_s . Denote $s' = s-1$ if c_s has right pivot type, and $s' = s$, otherwise; i.e., $c_{s'}$ is the rightmost vertex of the sequence that has left pivot type. By Lemma 15 we have $c_h = \text{pc}(c_{h+1})$, for $1 \leq h \leq s'-1$, and $\text{pc}(c_1) = u$. It follows that the rightmost vertex w' of $P_o(G_{i-1})$ that is adjacent to $c_{s'}$ is the only vertex of $P_o(G_{i-1})$ that can be adjacent to a vertex in $(X_j \cap V) \setminus \mathcal{E}_{i-1}$. So the general situation inside X_j can be summarized as depicted in Figure 9. Neither the sequence of left pivot vertices nor the right pivot vertex may exist, but if neither is present, then $X_j \cap V = \emptyset$.



■ **Figure 9** The structure of eligible vertices within a region X_j . All triangular faces here are empty, only the central face (shaded) may contain other vertices or edges uc_h , for $2 \leq h < s$. The left pivot vertices could be of any type $\mathcal{T}(z, \prec^{z-1})$.

The following lemma allows us to assume that the central face in each region X_j is subdivided into empty (of vertices) triangles and at most one – not necessarily empty – triangle or quadrilateral (the latter if X_j contains eligible vertices of both pivot types).

► **Lemma 17.** *Let X_j be a region s.t. there exist $v, v' \in \mathcal{E}_{i-1} \cap X_j$ with $\text{pc}(v) = v'$, let v'' be the vertex that covers vv' . If $v'' \neq u$ and $\chi \leq 1/5$, there exist v_i, \dots, v_{i+h-1} with $h \geq 3$ s.t. a valid diagram for G_{i+h-1} can be obtained by spending at most $(1 - \chi)h$ credits.*

Proof. By Lemma 14 both v and v' have left pivot type. In particular, if $c_s \neq c_{s'}$, this implies that we have $v, v' \neq c_s$ (see also Figure 9). By planarity and as $v'' \neq u$, we have $v'' \in X_j$. If v'' is not adjacent to w' , then v'' is eligible after adding v and v' and we can set $v_i = v$, $v_{i+1} = v'$, and $v_{i+2} = v''$ and use the diagram for G_{i+2} shown in Figure 10 (left), for a cost of $2 + 2\chi \leq 3 - 3\chi$, for $\chi \leq 1/5$. The figure shows the drawing where both v and v' are $\mathcal{T}(2, \curvearrowright)$; it easily extends to the types $\mathcal{T}(3, \curvearrowright^2)$ and $\mathcal{T}(4, \curvearrowright^3)$ because more mountains to the right of v can be paid for by the corresponding mountains whose left endpoint is covered by v and for more mountains to the left of v' their left endpoint is covered by v' .

Otherwise, v'' is adjacent to w' . We claim that we may assume $v = c_{s'}$ and $v' = c_{s'-1}$. To see this let $\tilde{v} \neq v''$ be the vertex that covers $c_{s'-1}c_{s'}$ and observe that \tilde{v} is enclosed by a cycle formed by $vv''w'$ and the part of $P_o(G_{i-1})$ between the right neighbor of v and w' . In particular, we have $\tilde{v} \neq u$ and so $c_{s'-1}, c_{s'}, \tilde{v}$ satisfy the conditions of the lemma, as claimed. We set $v_i = v$ and $v_{i+1} = v'$, and use the diagram shown in Figure 10 (right). If v'' is eligible in G_{i+1} , that is, the triangle $vv''w'$ is empty of vertices, then we set $v_{i+2} = v''$ and have a diagram for G_{i+2} for a cost of $2 + \chi \leq 3 - 3\chi$, for $\chi \leq 1/4$.

Otherwise, by Lemma 7 we inductively obtain a valid diagram D for the subgraph of G induced by taking $vv''w'$ as an outer triangle together with all vertices inside, with $v''v$ as a starting edge and w' as a last vertex. Then we plug D into the triangle $vv''w'$ as shown in Figure 10 (right). All mountains of D with left endpoint v'' carry a credit by (X2) for D . Thus, the resulting diagram is extensible. For the costs we have to account for the fact that w' is considered to contribute $1 - \chi$ credits to D , whereas we had already accounted for w' in the diagram for G_{i-1} . On the other hand, the edge $v''w'$ is paid for as a part of D . Thus, the additional costs to handle v, v', v'' are $(1 - \chi) + 1 + \chi = 2 \leq 3 - 3\chi$, for $\chi \leq 1/3$. ◀



■ **Figure 10** Two vertices v, v' that have left pivot type and $v'' \neq u$ covers the edge vv' .

To complete the proof of Lemmas 6 and 7 it remains to insert u along with the set $\mathcal{V}_u := V \cap R_{i-1}(u)$ of all vertices inside X_1, \dots, X_{k-1} , at a cost of $1 - \chi$ credits per vertex. We process these regions from right to left in two phases: In Phase 1, we select a suitable collection X_j, \dots, X_{k-1} of regions, for some $j \in \{1, \dots, k - 1\}$, so that we can insert u together with all the vertices inside these regions. Then in Phase 2, we process the remaining regions, assuming that u is already placed on the spine, somewhere to the right. To achieve this we do a case analysis, depending on the four types of regions: left, right, both pivot, or empty. In Appendix E of the full version, we show that in all cases $u \cup \mathcal{V}_u$ can be inserted as required.

5 Triangulations with many degree three vertices

► **Theorem 18.** *Let G be a triangulation with n vertices, and let d denote the number of degree three vertices in G . Then G admits a monotone plane arc diagram with at most $n - d - 4$ biarcs, where every biarc is down-up.*

Proof Sketch. Let T denote the triangulation that results from removing all degree-3 vertices from G , i.e., T has $k = n - d$ vertices. We proceed in two steps; see Appendix F of the full version for details.



■ **Figure 11** Insert a vertex using at most one credit and make every triangle cross the spine.

First step. We draw T while maintaining Invariants (X1)–(X3) and (X5) using the following modifications of our default insertion rules; see Figure 11. First, if we insert v_i into a pocket, we always ensure that the leftmost edge incident to v_i is a mountain. Second, if all edges covered by v_i are mountains, we *push down* the leftmost such mountain m , that is, we redraw m and all mountains having the same left endpoint as m into down-up biarcs. Third, instead of assigning credits to covered mountains whose left endpoint remains on the outer face, we immediately transform them into biarcs. Fourth, each vertex aside from v_1, v_2, v_3, v_n contributes 1 credit to the charging scheme. As a result, the arc diagram of T has at most $n - d - 4$ biarcs and all created faces have a non-empty intersection with the spine – note that the latter property does not follow from the result by Cardinal et al. [2, 3].

Second step. We insert each degree-three vertex v in its containing face f of T . Using that f crosses the spine we can place v there and then realize each edge to a vertex of f as a proper arcs. Thus, no new biarcs are created in the second step. ◀

▶ **Theorem 2.** *Every Kleetope on n vertices admits a monotone plane arc diagram with at most $\lfloor (n - 8)/3 \rfloor$ biarcs, where every biarc is down-up.*

Proof. Let G be a Kleetope on n vertices, and let d denote the number of degree three vertices in G . By Theorem 18 the graph G admits a monotone plane arc diagram with at most $n - d - 4$ biarcs, where every biarc is down-up. Removing the degree three vertices from G we obtain a triangulation T on $n - d$ vertices, which by Euler's formula has $2(n - d) - 4$ triangular faces. As G is a Kleetope, it is obtained by inserting a vertex into each of these faces, that is, we have $n = (n - d) + 2(n - d) - 4$ and thus $d = (2n - 4)/3$. So there are at most $n - d - 4 = (n - 8)/3$ biarcs in the diagram. ◀

6 Planar 3-Trees

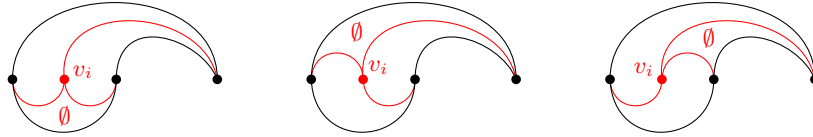
For 3-trees it is natural to follow their recursive construction sequence and build a corresponding diagram incrementally. A planar 3-tree G is built by starting from a (combinatorial) triangle. At each step we insert a new vertex v into a (triangular) face f of the graph built so far, and connect v to the three vertices of f . Every planar 3-tree G on at least four vertices is 3-connected. So its combinatorial embedding is unique, and for each triangle of the abstract graph we know whether it is facial or separating. In the former case, there is exactly one vertex of G that is adjacent to all vertices of the triangle, in the latter case there are exactly two such vertices. In particular, we can pick any facial triangle to be the starting triangle of our construction sequence for G and become the outer face of our diagram.

11:12 Monotone Arc Diagrams with Few Biarcs

Let v_1, \dots, v_n be such a construction sequence for G . For $i \in \{3, \dots, n\}$, let $V_i = \{v_1, \dots, v_i\}$ and $G_i = G[V_i]$. Each vertex v_i , for $i \in \{4, \dots, n\}$, is inserted into a face $F(v_i) = uvw$ of G_{i-1} , creating three *child faces* uvv_i , vvv_i and wvv_i of uvw in G_i . We also say that v_i is the *face vertex* $v(uvw)$ of face uvw . We call a face f of G_i *active* if it has a face vertex in $V \setminus V_i$; otherwise, it is *inactive*. The *grand-degree* $\text{gd}(f)$ is the maximum number of active child faces of f in all of G_3, \dots, G_n . Observe that by construction $\text{gd}(f) \in \{0, \dots, 3\}$ and that f is active for some G_i if and only if $\text{gd}(f) > 0$. Similarly, a vertex is a *gd- i* vertex, for $i \in \{0, 1, 2, 3\}$, if it is the face vertex of a face f with $\text{gd}(f) = i$. For a construction sequence we define its dual *face tree* \mathcal{T} on the faces of all G_i such that the root of \mathcal{T} is $v_1v_2v_3$, and each active face uvw has three children: the faces uvz , vwz , and wuz , where $z = v(uvw)$. Note that the leaves of \mathcal{T} are inactive for all G_i . Let us first observe that no biarcs are needed if all faces have small grand-degree. To this end, also recall that G admits a plane proper arc diagram if and only if it is subhamiltonian and planar.

► **Theorem 19.** *Let G be a planar 3-tree that has a construction sequence v_1, \dots, v_n such that for each face f in its dual tree $\text{gd}(f) \leq 2$. Then G admits a plane proper arc diagram.*

Proof. We start by drawing the face $v_1v_2v_3$ as a *drop*, that is, a face where the two short edges are proper arcs on different sides of the spine; see Figure 12. Then we iteratively insert the vertices v_i , for $i = 4, \dots, n$, such that every face that corresponds to an internal vertex of the dual tree \mathcal{T} is a drop in the diagram D_i for G_i . This can be achieved because by assumption at least one of the three faces of D_i created by inserting v_i is a leaf of \mathcal{T} , which need not be realized as a drop. But we can always realize the two other faces as drops, as shown in Figure 12. In this way we obtain a diagram for G without any biarc. ◀



■ **Figure 12** Insert a vertex v_i into a drop s.t. any chosen two of the faces created are drops.

As \mathcal{T} is a tree, we can relate the number of internal vertices to the number of leaves.

► **Lemma 20.** *Let f_d denote the number of faces in \mathcal{T} with grand-degree exactly d , and let n_{inact} denote the number of face vertices that create inactive faces only. Then $n_{\text{inact}} \geq 2f_3 + f_2$.*

Proof. Consider the rooted tree \mathcal{T}' obtained by removing all leaves of \mathcal{T} , and observe that the grand-degree in \mathcal{T} corresponds to the vertex degree in \mathcal{T}' . ◀

We are now ready to describe our drawing algorithm for general planar 3-trees.

► **Theorem 3.** *Every planar 3-tree admits a plane arc diagram with at most $\lfloor \frac{3}{4}(n-3) \rfloor$ biarcs that are all down-up monotone.*

Proof. Our algorithm is iterative and draws G in the sequence prescribed by \mathcal{T} . Namely, at each step of our algorithm, we select an arbitrary already drawn face uvw and insert its face vertex $v(uvw)$, possibly together with the face vertex of a child face. We will consider faces of a particular shape mostly. Consider a face $f = uvw$ such that u, v, w appear in this order along the spine and uw forms the upper envelope of f . (There is a symmetric configuration, obtained by a rotation by an angle of π where uw forms the lower envelope of f .) We say

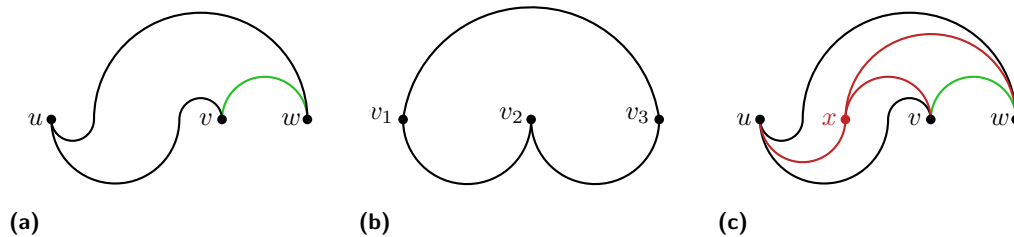
that f is *ottifant-shaped*⁴ if it contains a region bounded by a down-up biarc between u and w , a down-up biarc between u and v and a mountain between v and w ; see Figure 13a. Note the word “contains” in the definition of ottifant-shaped, which allows the actual face to be larger. For instance, the top boundary could be a mountain, but we treat it as if it was a biarc for the purposes of drawing edges; that is, we only connect to u from below the spine.

To control the number of biarcs drawn we maintain a charge $\text{ch}(v)$ for each vertex v . We require additional flexibility from the edge vw of an ottifant-shaped face $f = uvw$, which we call the *belly* of f . To this end, we call a mountain vw *transformable* if it can be redrawn as a down-up biarc for at most $3/2$ units of charge. (Note that every edge can be drawn as a biarc for only one credit. But in some cases redrawing an edge as a biarc requires another adjacent edge to be redrawn as a biarc as well. Having an extra reserve of half a credit turns out sufficient to cover these additional costs, as shown in the analysis below.)

More specifically, we maintain the following invariants:

- (O1) Each internal active face is ottifant-shaped.
- (O2) If the belly of an active face is a mountain, it is transformable.
- (O3) The sum of the charges of all vertices is at least the number of biarcs drawn.
- (O4) For each vertex v we have $\text{ch}(v) \leq \frac{3}{4}$.

It is easy to see that a drawing D of G has at most $\lfloor \frac{3}{4}n \rfloor$ biarcs if the invariants hold for D .



■ **Figure 13** (a) An ottifant-shaped face uvw , where the long edge is on the top page (green edges are transformable). (b) Drawing of the initial face $v_1v_2v_3$. (c) Insertion of a gd-1 vertex $x = v(uvw)$.

Initialization. We put $v_1v_2v_3$ on the spine in this order and draw the edges v_1v_2 and v_2v_3 as pockets and v_1v_3 as a mountain; see Figure 13b. The invariants (O1)–(O4) hold.

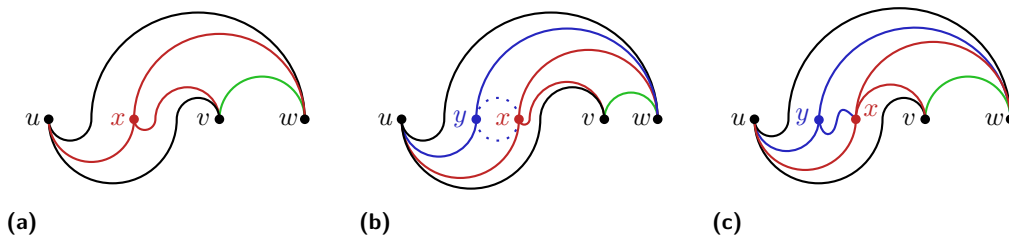
Charging rights. Typically we charge a vertex when it is added to the drawing. But different vertices have different needs. Specifically, we will see that no biarc/charge is used when inserting a gd-0 vertex. Therefore, for each gd-0 vertex v we distribute the rights to use the charge of v among two targets: (1) the *parent* of v (i.e., the vertex $v(f)$ of the parent f of $F(v)$ in \mathcal{T}) – if it exists – may assign a charge of $\leq 1/4$ to v and (2) the so-called *preferred ancestor* $p(v)$ may assign a charge of $\leq 1/2$ to v . Preferred ancestors are determined by selecting an arbitrary surjective map p from the set of gd-0 vertices to the set of gd-2 and gd-3 vertices. According to Lemma 20 there exists such a map such that every gd-2 is selected at least once and every gd-3 vertex is selected at least twice as a preferred ancestor.

⁴ An *ottifant* is a cartoon abstraction of an elephant designed and popularized by the artist Otto Waalkes. Use of the term *ottifant* with kind permission of Ottifant Productions GmbH.

Iterative step. We select an arbitrary active face $f = uvw$, which is ottifant-shaped by (O1), and insert its face vertex $x := v(f)$ into f . Assume w.l.o.g. (up to rotation by an angle of π) that uw forms the top boundary of f . We make a case distinction based on $\text{gd}(f)$.

Case 1: $\text{gd}(f) = 0$. Then all child faces of f are inactive so that (O1) and (O2) hold trivially. We insert x inside f between u and v on the spine, draw the edge ux as a pocket and xv and xw as mountains; see Figure 13c. No biarcs are created, so (O3)–(O4) hold.

Case 2: $\text{gd}(f) \geq 2$. We insert x as in Case 1, except that xv is drawn as a biarc rather than as a mountain; see Figure 14a. All created child faces are ottifant-shaped (O1) and all bellies are transformable (O2). We created one biarc. So to establish (O3)–(O4) it suffices to set $\text{ch}(x) = \frac{3}{4}$ and add a charge of $\frac{1}{4}$ to one of the (at least one) $\text{gd}-0$ vertices in $p^{-1}(x)$.



■ **Figure 14** Insertion of (a) a $\text{gd}-2$ vertex x ; (b) a $\text{gd}-1$ vertex y ; (c) a $\text{gd}-2$ vertex y .

Case 3: $\text{gd}(f) = 1$. Then only one of the three child faces of f is active. If uvx is the active child face, then we use the same drawing as for a $\text{gd}-0$ vertex (see Figure 13c) and all invariants hold. However, if one of the other child faces is active, then we cannot use this drawing because xw is not transformable and xvw is not ottifant-shaped.

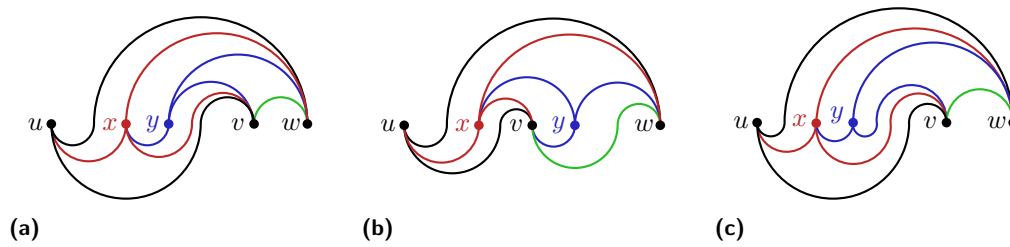
So we also consider the face vertex y of the unique child face f' of x and insert both x and y into the drawing together. We consider two subcases, according to f' .

Case 3A: $f' = uxw$. If $\text{gd}(f') = 0$, then we can once again use the drawing for a $\text{gd}-0$ vertex (see Figure 13c) because f' is ottifant-shaped and none of its child faces are active.

If $\text{gd}(f') = 1$, then we add first x as described for a $\text{gd}-2$ vertex above (see Figure 14a). Then we add y into f' and draw all incident edges as proper arcs; the edge yx can be drawn either as a mountain (if uxy is the active child face of f') or as a pocket (otherwise); see Figure 14b. In either case, invariants (O1)–(O2) hold. We added one biarc (xv). To establish (O3)–(O4) we set $\text{ch}(x) = \text{ch}(y) = \frac{1}{2} < \frac{3}{4}$.

Otherwise, we have $\text{gd}(f') \geq 2$. We first add x as described above for a $\text{gd}-0$ vertex and then y as a $\text{gd}-2$ vertex; see Figure 14c. Invariant (O1) holds. To establish (O2) we have to make the bellies xw and uy of yxw and uyx , respectively, transformable. To this end, we put $1/2$ units of charge aside so that both xv and xw could be redrawn as biarcs for $3/2$ units of charge, as required. Moreover, we observe that uy can be transformed into a biarc for 1 units of charge if necessary as there is no other edge that must be transformed in this scenario. We also added a biarc, namely, yx . To establish (O3)–(O4) we set $\text{ch}(x) = \text{ch}(y) = \frac{3}{4}$.

Case 3B: $f' = xv w$. We consider several subcases according to $\text{gd}(f')$. If $\text{gd}(f') = 0$, we first insert x as described above for a $\text{gd}-2$ vertex and then y as a $\text{gd}-0$ vertex; see Figure 15a. Invariants (O1)–(O2) hold trivially. We used one biarc (xv). To establish (O3)–(O4), we set $\text{ch}(x) = \frac{3}{4}$ and increase $\text{ch}(y)$ by $\frac{1}{4}$. The latter is allowed because x is the parent of y .



■ **Figure 15** Insertion of (a) a gd-2 vertex x ; (b) a gd-1 vertex y ; (c) a gd-2 vertex y .

We use the same drawing if $\text{gd}(f') = 1$ and the (only) active child face of f' is xvy or xyw . If xvy is active, then we set $\text{ch}(x) = \text{ch}(y) = \frac{1}{2} < \frac{3}{4}$ to establish (O3)–(O4). If xyw is active, then we put $1/2$ units of charge aside to make yw transformable and establish (O2). Then we set $\text{ch}(x) = \text{ch}(y) = \frac{3}{4}$ to establish (O3)–(O4).

If $\text{gd}(f') = 1$, then it remains to consider the case that the (only) active child face of f' is yvw . We transform vw into a biarc, then insert x between u and v , and finally insert y between v and w on the spine inside f . All edges incident to x and y are drawn as proper arcs; see Figure 15b. The only active (grand)child face of f is yvw , and (O1)–(O2) hold. We have spent $3/2$ units of charge to transform vw , and we did not create any biarc. Thus, it suffices to set $\text{ch}(x) = \text{ch}(y) = \frac{3}{4}$ to establish (O3)–(O4).

If $\text{gd}(f') \geq 2$, then we first insert x between u and v and then y between x and v on the spine inside f . Then we draw xv and yv as biarcs and the remaining edges as proper arcs such that xy is a pocket; see Figure 15c. Invariants (O1)–(O2) hold. We created two biarcs (xv and yv). To establish (O3)–(O4), we set $\text{ch}(x) = \text{ch}(y) = \frac{3}{4}$ and we increase the charge of a vertex in $p^{-1}(y)$ by $1/2$.

It follows that (O1)–(O4) hold after each step. ◀

7 Conclusions

We proved the first upper bound of the form $c \cdot n$, with $c < 1$, for the number of monotone biarcs in arc diagrams of planar graphs. In our analysis, only some cases require $\chi \leq 1/5$, indicating a possibility to further refine the analysis to achieve an even better bound. It remains open whether there exists a “monotonicity penalty” in this problem, but we ruled out the probably most prominent class of non-Hamiltonian maximal planar graphs, the Kleetopes, as candidates to exhibit such a phenomenon. It would be very interesting to close the gap between upper and lower bounds, both in the monotone and in the general settings.

References

- 1 Frank Bernhart and Paul C. Kainen. The book thickness of a graph. *J. Combin. Theory Ser. B*, 27:320–331, 1979. doi:10.1016/0095-8956(79)90021-2.
- 2 Jean Cardinal, Michael Hoffmann, Vincent Kusters, Csaba D. Tóth, and Manuel Wettstein. Arc diagrams, flip distances, and Hamiltonian triangulations. In *Proc. 32nd Int. Sympos. Theoret. Aspects Comput. Sci. (STACS 2015)*, volume 30 of *LIPICs*, pages 197–210, 2015. doi:10.4230/LIPICs.STACS.2015.197.
- 3 Jean Cardinal, Michael Hoffmann, Vincent Kusters, Csaba D. Tóth, and Manuel Wettstein. Arc diagrams, flip distances, and Hamiltonian triangulations. *Comput. Geom. Theory Appl.*, 68:206–225, 2018. doi:10.1016/j.comgeo.2017.06.001.
- 4 Marek Chrobak and Thomas H. Payne. A linear-time algorithm for drawing a planar graph on a grid. *Inform. Process. Lett.*, 54:241–246, 1995. doi:10.1016/0020-0190(95)00020-D.

- 5 Hubert de Fraysseix, János Pach, and Richard Pollack. Small sets supporting Fary embeddings of planar graphs. In *Proc. 20th Annu. ACM Sympos. Theory Comput. (STOC 1988)*, pages 426–433, 1988. doi:10.1145/62212.62254.
- 6 Hubert de Fraysseix, János Pach, and Richard Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990. doi:10.1007/BF02122694.
- 7 Emilio Di Giacomo, Walter Didimo, Giuseppe Liotta, and Stephen K. Wismath. Curve-constrained drawings of planar graphs. *Comput. Geom. Theory Appl.*, 30(1):1–23, 2005. doi:10.1016/j.comgeo.2004.04.002.
- 8 Hazel Everett, Sylvain Lazard, Giuseppe Liotta, and Stephen K. Wismath. Universal sets of n points for 1-bend drawings of planar graphs with n vertices. In *Proc. 15th Int. Sympos. Graph Drawing (GD 2007)*, volume 4875 of *LNCS*, pages 345–351, 2007. doi:10.1007/978-3-540-77537-9_34.
- 9 Hazel Everett, Sylvain Lazard, Giuseppe Liotta, and Stephen K. Wismath. Universal sets of n points for one-bend drawings of planar graphs with n vertices. *Discrete Comput. Geom.*, 43(2):272–288, 2010. doi:10.1007/s00454-009-9149-3.
- 10 Emilio Di Giacomo, Walter Didimo, Giuseppe Liotta, and Stephen K. Wismath. Drawing planar graphs on a curve. In *Proc. 13th Int. Workshop Graph-Theoret. Concepts Comput. Sci. (WG 2003)*, volume 2880 of *LNCS*, pages 192–204, 2003. doi:10.1007/978-3-540-39890-5_17.
- 11 Francesco Giordano, Giuseppe Liotta, Tamara Mchedlidze, Antonios Symvonis, and Sue Whitesides. Computing upward topological book embeddings of upward planar digraphs. *J. Discrete Algorithms*, 30:45–69, 2015. doi:10.1016/j.jda.2014.11.006.
- 12 Michael Kaufmann and Roland Wiese. Embedding vertices at points: Few bends suffice for planar graphs. In *Proc. 6th Int. Sympos. Graph Drawing (GD 1998)*, volume 1547 of *LNCS*, pages 462–463, 1998. doi:10.1007/3-540-37623-2_47.
- 13 Michael Kaufmann and Roland Wiese. Embedding vertices at points: Few bends suffice for planar graphs. *J. Graph Algorithms Appl.*, 6(1):115–129, 2002. doi:10.7155/jgaa.00046.
- 14 Maarten Löffler and Csaba D. Tóth. Linear-size universal point sets for one-bend drawings. In *Proc. 23rd Int. Sympos. Graph Drawing Network Visualization (GD 2015)*, volume 9411 of *LNCS*, pages 423–429, 2015. doi:10.1007/978-3-319-27261-0_35.

The Parameterized Complexity Of Extending Stack Layouts

Thomas Depian ✉ 

Algorithms and Complexity Group, TU Wien, Austria

Simon D. Fink ✉ 

Algorithms and Complexity Group, TU Wien, Austria

Robert Ganian ✉ 

Algorithms and Complexity Group, TU Wien, Austria

Martin Nöllenburg ✉ 

Algorithms and Complexity Group, TU Wien, Austria

Abstract

An ℓ -page stack layout (also known as an ℓ -page book embedding) of a graph is a linear order of the vertex set together with a partition of the edge set into ℓ stacks (or pages), such that the endpoints of no two edges on the same stack alternate. We study the problem of extending a given partial ℓ -page stack layout into a complete one, which can be seen as a natural generalization of the classical NP-hard problem of computing a stack layout of an input graph from scratch. Given the inherent intractability of the problem, we focus on identifying tractable fragments through the refined lens of parameterized complexity analysis. Our results paint a detailed and surprisingly rich complexity-theoretic landscape of the problem which includes the identification of paraNP-hard, W[1]-hard and XP-tractable, as well as fixed-parameter tractable fragments of stack layout extension via a natural sequence of parameterizations.

2012 ACM Subject Classification Theory of computation \rightarrow Fixed parameter tractability; Mathematics of computing \rightarrow Graphs and surfaces

Keywords and phrases Stack Layout, Drawing Extension, Parameterized Complexity, Book Embedding

Digital Object Identifier 10.4230/LIPIcs.GD.2024.12

Related Version *Full Version:* <https://arxiv.org/abs/2409.02833> [17]

Funding All authors acknowledge support from the Vienna Science and Technology Fund (WWTF) [10.47379/ICT22029]. Robert Ganian and Thomas Depian furthermore acknowledge support from the Austrian Science Fund (FWF) [10.55776/Y1329].

1 Introduction

An ℓ -page stack layout (or ℓ -page book embedding) of a graph G consists, combinatorially speaking, of (i) a linear order \prec of its vertex set $V(G)$ and (ii) a partition σ of its edge set $E(G)$ into $\ell \geq 1$ (*stack-)*pages such that for no two edges (with distinct endpoints) uv and wx with $u \prec v$ and $w \prec x$ that are assigned to the same page their endpoints alternate in \prec , i.e., we have $u \prec w \prec v \prec x$. When drawing a stack layout, the vertices are placed on a line called the *spine* in the order given by \prec and the edges of each page are drawn as pairwise non-crossing arcs in a separate half-plane bounded by the spine, see Figure 1a. Stack layouts are a classic and well-studied topic in graph drawing and graph theory [6, 12, 30]. They have immediate applications in graph visualization [4, 25, 38] as well as in bioinformatics, VLSI design, and parallel computing [14, 27]; see also the overview by Dujmović and Wood [20].

The minimum number ℓ such that a given graph G admits an ℓ -page stack layout is known as the *stack number*, *page number*, or *book thickness* of G . While the graphs with stack number $\ell = 1$ are the outerplanar graphs, which can be recognized in linear time, the



© Thomas Depian, Simon D. Fink, Robert Ganian, and Martin Nöllenburg; licensed under Creative Commons License CC-BY 4.0

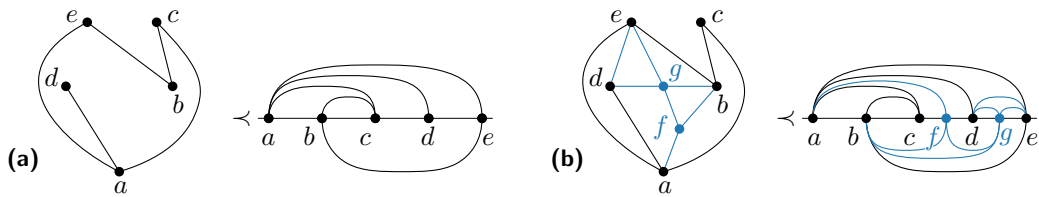
32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 12; pp. 12:1–12:17



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** (a) A graph H and a two-page stack layout of it. In (b), the graph H and its two-page stack layout are extended by the new vertices and edges marked in blue.

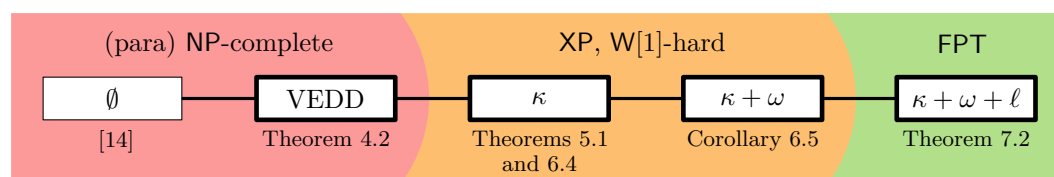
problem of computing the stack number is NP-complete in general. Indeed, the class of graphs with stack number $\ell \leq 2$ are precisely the subhamiltonian graphs (i.e., the subgraphs of planar Hamiltonian graphs) and recognizing them is NP-complete [6, 14, 39]. Computing the stack number is known to also remain NP-complete if the vertex order is provided as part of the input and $\ell = 4$ [36], and overcoming the intractability of these problems has been the target of several recent works in the field [10, 11, 24, 29]. Many other results on stack layouts are known – for instance, every planar graph has a 4-page stack layout and this bound is tight [5, 40]. For a comprehensive list of known upper and lower bounds for the stack number of different graph classes, we refer to the collection by Pupyrev [33].

In this paper, we take a new perspective on stack layouts, namely the perspective of drawing extensions. In drawing extension problems, the input consists of a graph G together with a partial drawing of G , i.e., a drawing of a subgraph H of G . The task is to insert the vertices and edges of G which are missing in H in such a way that a desired property of the drawing is maintained; see Figure 1b for an example. Such drawing extension problems occur, e.g., when visualizing dynamic graphs in a streaming setting, where additional vertices and edges arrive over time and need to be inserted into the existing partial drawing. Drawing extension problems have been investigated for many types of drawings in recent years – including planar drawings [1, 28, 31, 32], upward planar drawings [16], level planar drawings [13], 1-planar drawings [21, 22], and planar orthogonal drawings [2, 3, 9] – but until now, essentially nothing was known about the extension of stack layouts/book embeddings.

Since it is NP-complete to determine whether a graph admits an ℓ -page stack layout (even when ℓ is a small fixed integer), the extension problem for ℓ -page stack layouts is NP-complete as well – after all, setting H to be empty in the latter problem fully captures the former one. In fact, the extension setting can seamlessly also capture the previously studied NP-complete problem of computing an ℓ -stack layout with a prescribed vertex order [10, 11, 14, 36, 37]; indeed, this corresponds to the special case where $V(H) = V(G)$ and $E(H) = \emptyset$. Given the intractability of extending ℓ -page stack layouts in the classical complexity setting, we focus on identifying tractable fragments of the problem through the more refined lens of parameterized complexity analysis [15, 19], which considers both the input size of the graph and some additional parameter k of the instance¹.

Contributions. A natural parameter in any drawing extension problem is the size of the missing part of the graph, i.e., the missing number of vertices and/or edges. We start our investigation by showing that the STACK LAYOUT EXTENSION problem (SLE) for instances without any missing vertices, i.e., $V(G) = V(H)$, is fixed-parameter tractable when parameterized by the number of missing edges $|E(G) \setminus E(H)|$ (Section 3).

¹ We assume familiarity with the basic foundations of *parameterized complexity theory*, notably including the notions of *fixed-parameter tractability*, XP, W[1]-, and paraNP-hardness [15].



■ **Figure 2** The complexity landscape of STACK LAYOUT EXTENSION. VEDD denotes the vertex+edge deletion distance, ω denotes the page width of the ℓ -page stack layout of H , and $\kappa = |V(G) \setminus V(H)| + |E(G) \setminus E(H)|$. Boxes outlined in bold represent new results that we show in the linked theorems and corollaries. The only result that is not depicted is Theorem 3.2.

The above result, however, only applies in the highly restrictive setting where no vertices are missing – generally, we would like to solve instances with missing vertices as well as edges. A parameterization that has been successfully used in this setting is the *vertex+edge deletion distance*, i.e., the number of vertex and edge deletion operations² required to obtain H from G . But while this parameter has yielded parameterized algorithms when extending, e.g., 1-planar drawings [21, 22] and orthogonal planar drawings [9], we rule out any analogous result for SLE by establishing its NP-completeness even if H can be obtained from G by deleting only two vertices (Section 4). This means that more “restrictive” parameterizations are necessary to achieve tractability for the problem of extending ℓ -page stack layouts.

Since the missing vertices in our hardness reduction have a high degree, we then consider parameterizations by the combined number of missing vertices and edges $\kappa = |V(G) \setminus V(H)| + |E(G) \setminus E(H)|$. We show that SLE belongs to the class XP when parameterized by κ (Section 5) while being W[1]-hard (Section 6), which rules out the existence of a fixed-parameter tractable algorithm under standard complexity assumptions. The latter result holds even if we additionally bound the page width ω of the stack layout of H , which measures the maximum number of edges that are crossed on a single page by a line perpendicular to the spine [14]. On our quest towards a fixed-parameter tractable fragment of the problem, we thus need to include another restriction, namely the number ℓ of pages of the stack layout. So finally, when parameterizing SLE by the combined parameter $\kappa + \omega + \ell$, we show that it becomes fixed parameter tractable (Section 7). Our results are summarized in Figure 2.

Full proofs of statements marked by ★ can be found in the full version [17].

2 Preliminaries

We assume the reader to be familiar with standard graph terminology [18]. Throughout this paper, we assume standard graph representations, e.g., as double-linked adjacency list, that allow for efficient graph modifications. For two integers $p \leq q$ we denote with $[p, q]$ the set $\{p, p + 1, \dots, q\}$ and use $[p]_0$ and $[p]$ as abbreviations for $[0, p]$ and $[1, p]$, respectively. Let G be a graph that is, unless stated otherwise, simple and undirected, with vertex set $V(G)$ and edge set $E(G)$. For $X \subseteq V(G)$, we denote by $G[X]$ the subgraph of G induced on X .

Stack Layouts. For an integer $\ell \geq 1$, an ℓ -page stack layout of G is a tuple $\langle \prec_G, \sigma_G \rangle$ where \prec_G is a linear order of $V(G)$ and $\sigma_G: E(G) \rightarrow [\ell]$ is a function that assigns each edge to a page $p \in [\ell]$ such that for each pair of edges u_1v_1 and u_2v_2 with $\sigma(e_1) = \sigma(e_2)$ it does not hold $u_1 \prec u_2 \prec v_1 \prec v_2$. For the remainder of the paper, we write \prec and σ if the graph G is

² As usual, we assume that deleting a vertex automatically also deletes all of its incident edges.

12:4 The Parameterized Complexity Of Extending Stack Layouts

clear from context. We call \prec the *spine (order)* and σ the *page assignment*. Observe that we can interpret a stack layout as a drawing of G on different planar half-planes, one per page $p \in [\ell]$, each of which is bounded by the straight-line spine delimiting all half-planes. One fundamental property of a stack layout is its *page width* – denoted as $\omega(\langle \prec, \sigma \rangle)$ or simply ω if $\langle \prec, \sigma \rangle$ is clear from context – which is the maximum number of edges that are crossed on a single page by a line perpendicular to the spine [14]. The properties of stack layouts with small page width have been studied, e.g., by Stöhr [34, 35].

We say that two vertices u and v are *consecutive on the spine* if they occur consecutively in \prec . A vertex $u \in V(G)$ *sees* a vertex $v \in V(G)$ on a page $p \in [\ell]$ if there does not exist an edge $e = xy \in E(G)$ with $\sigma(e) = p$ and $x \prec u \prec y \prec v$ or $u \prec x \prec v \prec y$. Note that if u sees v , then v also sees u . For two vertices u and v which are consecutive in \prec , we refer to the segment on the spine between u and v as the *interval* between u and v , denoted as $[u, v]$.

Problem Statement. Let $H \subseteq G$ be a subgraph of a graph G . We say that $\langle \prec_G, \sigma_G \rangle$ is an *extension* of $\langle \prec_H, \sigma_H \rangle$ if $\sigma_H \subseteq \sigma_G$ and $\prec_H \subseteq \prec_G$. We now formalize our problem of interest:

STACK LAYOUT EXTENSION (SLE)

Given Integer $\ell \geq 1$, graph G , subgraph H of G , and ℓ -page stack layout $\langle \prec_H, \sigma_H \rangle$.

Question Does there exist an ℓ -page stack layout $\langle \prec_G, \sigma_G \rangle$ of G that extends $\langle \prec_H, \sigma_H \rangle$?

We remark that while SLE is defined as a decision problem for complexity-theoretic reasons, every algorithm presented in this article is constructive and can be trivially adapted to also output a layout $\langle \prec_G, \sigma_G \rangle$ as a witness (also called a *solution*) for positive instances. For an instance \mathcal{I} of SLE, we use $|\mathcal{I}|$ as shorthand for $|V(G)| + |E(G)| + \ell$.

In line with the terminology previously used for drawing extension problems [21], we refer to the vertices and edges in $V(H) \cup E(H)$ as *old* and call all other vertices and edges of G *new*. Let V_{add} and E_{add} denote the sets of all new vertices and edges, respectively, and set $n_{\text{add}} := |V_{\text{add}}|$ and $m_{\text{add}} := |E_{\text{add}}|$. Furthermore, we denote with E_{add}^H the set of new edges incident to two old vertices, i.e., $E_{\text{add}}^H := \{e = uv \in E_{\text{add}} \mid u, v \in V(H)\}$. We consider the parameterized complexity of our extension problem by measuring how “incomplete” the provided partial solution is using the following natural parameters that have also been used in this setting before [7, 8, 21–23]: the *vertex+edge deletion distance*, which is $n_{\text{add}} + |E_{\text{add}}^H|$, and the total *number of missing vertices and edges*, i.e., $n_{\text{add}} + m_{\text{add}}$.

3 SLE With Only Missing Edges is FPT

We begin our investigation by first analyzing the special case where $V(G) = V(H)$, i.e., when only edges are missing from H . We recall that the problem remains NP-complete even in this setting, as it generalizes the problem of computing the stack number of a graph with a prescribed vertex order [10, 11, 14, 36, 37]. Furthermore, both of the aforementioned measures of the incompleteness of $\langle \prec_H, \sigma_H \rangle$ are the same and equal $m_{\text{add}} = |E_{\text{add}}^H|$. As a “warm-up” result, we show that in this setting SLE is fixed-parameter tractable parameterized by m_{add} .

Towards this, consider the set $S(e) \subseteq [\ell]$ of pages on which we could place a new edge e without introducing a crossing with edges from H ; formally, $p \in S(e)$ if and only if $\langle \prec_H, \sigma_H \cup (e, p) \rangle$ is an ℓ -page stack layout of $H \cup \{e\}$. Intuitively, if $|S(e)|$ is large enough, then we are always able to find a “free” page to place e independent of the placement of the remaining new edges. Formally, one can easily show:

► **Lemma 3.1 (★).** *Let $\mathcal{I} = (\ell, G, H, \langle \prec, \sigma \rangle)$ be an instance of SLE with $V_{\text{add}} = \emptyset$ that contains an edge $e \in E_{\text{add}}$ with $|S(e)| \geq m_{\text{add}}$. The instance $\mathcal{I}' = (\ell, H, G', \langle \prec, \sigma \rangle)$ with $G' = G \setminus \{e\}$ is a positive instance if and only if \mathcal{I} is a positive instance.*

With Lemma 3.1 in hand, we can establish the desired result:

► **Theorem 3.2 (★).** *Let $\mathcal{I} = (\ell, G, H, \langle \prec, \sigma \rangle)$ be an instance of SLE with $V_{\text{add}} = \emptyset$. We can find an ℓ -page stack layout of G that extends $\langle \prec, \sigma \rangle$ or report that none exists in $\mathcal{O}(m_{\text{add}}^{m_{\text{add}}} \cdot |\mathcal{I}|)$ time.*

Proof sketch. We compute for a single edge $e \in E_{\text{add}}$ the set $S(e)$ in linear time by checking with which of the old edges e would cross. If $|S(e)| \geq m_{\text{add}}$, then following Lemma 3.1, we remove e from G . Overall, this takes $\mathcal{O}(m_{\text{add}} \cdot |\mathcal{I}|)$ time and results in a graph G' with $H \subseteq G' \subseteq G$. Furthermore, each remaining new edge $e' \in E(G') \setminus E(H)$ can be put in fewer than m_{add} different pages. Hence, we can brute-force over all the at most $\mathcal{O}(m_{\text{add}}^{m_{\text{add}}})$ page assignments σ' that extend σ_H for all edges in $E(G') \setminus E(H)$, and for each such assignment we check in linear time whether no pair of edges $e', e'' \in E(G') \setminus E(H)$ cross each other. ◀

4 SLE With Two Missing Vertices is NP-complete

Adding only edges to a given linear layout is arguably quite restrictive. Therefore, we now lift this restriction and consider SLE in its full generality, i.e., also allow adding vertices. Somewhat surprisingly, as our first result in the general setting we show that SLE is NP-complete even if the task is to merely add two vertices, i.e., for $n_{\text{add}} = 2$ and $E_{\text{add}}^H = \emptyset$. This rules out not only fixed-parameter but also XP tractability when parameterizing by the vertex+edge deletion distance, and represents – to the best of our knowledge – the first example of a drawing extension problem with this behavior.

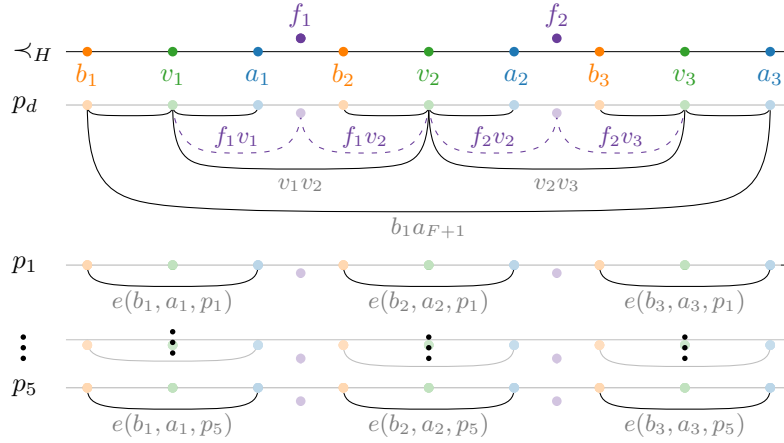
To establish the result, we devise a reduction from 3-SAT [26]. In our reduction, we insert two new vertices into a partial layout derived from the given formula, and use the page assignment of their incident edges to encode a truth assignment and validate that it satisfies all clauses. For this, we will need to restrict the positions of the new vertices to a certain range along the spine. In Section 4.1, we introduce the *fixation gadget* that ensures this. We also reuse this gadget in the reduction shown in Section 6. But first, we use it in this section to perform our reduction and prove NP-completeness in Section 4.2.

The graph H that we construct will have multi-edges to facilitate the presentation of the reduction. The procedure for removing multi-edges is detailed in the full version [17].

4.1 Restricting the Placement of New Vertices: The Fixation Gadget

The purpose of the so-called *fixation gadget* is to restrict the possible positions of new vertices to given intervals. As this gadget will also find applications outside this reduction, we describe in the following in detail its general construction for $F > 1$ new vertices $\mathcal{F} = \{f_1, \dots, f_F\}$.

First, we introduce $3(F + 1)$ new vertices v_1, \dots, v_{F+1} , b_1, \dots, b_{F+1} , and a_1, \dots, a_{F+1} . We fix the spine order \prec_H among these vertices to $b_1 \prec v_1 \prec a_1 \prec b_2 \prec v_2 \prec a_2 \prec \dots \prec b_{F+1} \prec v_{F+1} \prec a_{F+1}$; see also Figure 3. Then, every new vertex f_i is made adjacent to v_i and v_{i+1} and we aim to allow these new edges to be placed only in a dedicated further page p_d . To achieve this, we first introduce for every $i \in [F + 1]$ and every page $p \neq p_d$ an edge $e(b_i, a_i, p) = b_i a_i$ in H and set $\sigma(e(b_i, a_i, p)) = p$; see Figure 3. Furthermore, we also introduce the edges $b_i v_i$ and $v_i a_i$ and set $\sigma(b_i v_i) = \sigma(v_i a_i) = p_d$ for all $i \in [F + 1]$. For every $i \in [F]$, we add the edge $v_i v_{i+1}$ and place it on the page p_d , i.e., we have $\sigma(v_i v_{i+1}) = p_d$ as in



■ **Figure 3** A fixation gadget for $F = 2$ with five other pages in the stack layout. We also highlight the intended position for f_1 and f_2 on the spine and the page assignment for their incident edges.

Figure 3. Finally, we also create the edge $b_1 a_{F+1}$ and set $\sigma(b_1 a_{F+1}) = p_d$. To complete the construction of the fixation gadget, we add the new edges $f_i v_i$ and $f_i v_{i+1}$ for every $i \in [F]$ to G . Figure 3 shows an example of the fixation gadget for $F = 2$.

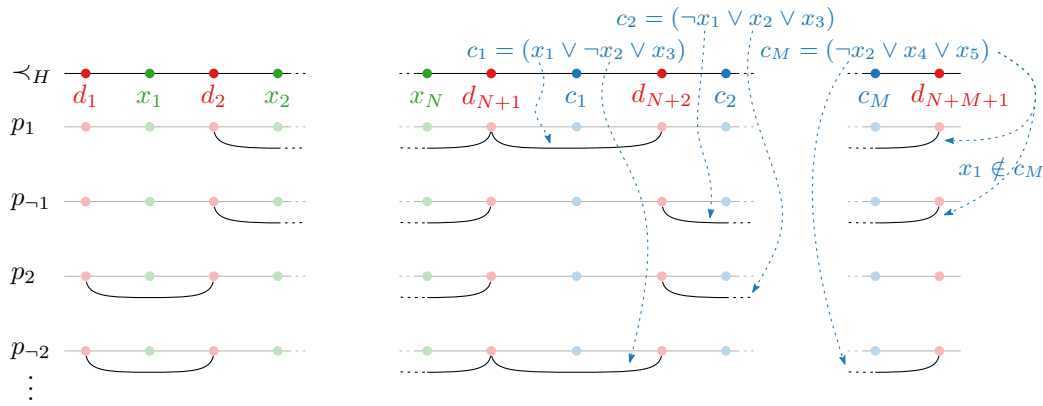
Next, we show that the fixation gadget forces f_i to lie between v_i and v_{i+1} on the spine and the edges $f_i v_i$ and $f_i v_{i+1}$ to be on the page p_d for every $i \in [F]$.

► **Lemma 4.1 (★).** *Let $\mathcal{I} = (\ell, G, H, \langle \prec, \sigma \rangle)$ be an instance of SLE that contains a fixation gadget on F vertices $\{f_1, \dots, f_F\}$. In any solution $\langle \prec_G, \sigma_G \rangle$ to \mathcal{I} and for every $i \in [F]$, we have $v_i \prec f_i \prec v_{i+1}$ and $\sigma(f_i v_i) = \sigma(f_i v_{i+1}) = p_d$. Furthermore, the fixation gadget contributes $4F + 3$ vertices and $(\ell + 4)F + \ell + 2$ edges to the size of \mathcal{I} .*

Proof sketch. Towards establishing $v_i \prec f_i \prec v_{i+1}$, one can show that $f_i \prec v_i$ would prevent f_i from seeing v_{i+1} on any page: As $f_i \prec v_i$ implies $f_i \prec b_{i+1} \prec v_{i+1} \prec a_{i+1}$ and we have the edge $b_{i+1} a_{i+1}$ on any page except p_d , only visibility on page p_d would still be possible. However, the edges on the page p_d prevent visibility to v_{i+1} for any spine position left of v_i . By symmetric arguments, we can obtain that $v_{i+1} \prec f_i$ would prevent v_i from seeing f_i . Using again the fact that we have the edge $b_i a_i$ on any page except p_d , in concert with the relation $v_i \prec f_i \prec v_{i+1}$ shown above and the edges $v_i a_i$ and $b_{i+1} v_{i+1}$ on the page p_d , one can deduce that $\sigma(f_i v_i) = \sigma(f_i v_{i+1}) = p_d$ must hold. Finally, the bound on the size of the gadget can be obtained by a close analysis of the construction. ◀

Lemma 4.1 tells us that we can restrict the feasible positions for f_i to a pre-defined set of consecutive intervals by choosing suitable positions for v_i and v_{i+1} in the spine order \prec_H . As the fixation gadget requires an additional page p_d , we must ensure that the existence of the (otherwise mostly empty) page p_d does not violate the semantics of our reductions. In particular, we will (have to) ensure that our full constructions satisfy the following property.

► **Property 1.** *Let $\mathcal{I} = (\ell, G, H, \langle \prec, \sigma \rangle)$ be an instance of SLE that contains a fixation gadget on F vertices $\{f_1, \dots, f_F\}$. In any solution $\langle \prec_G, \sigma_G \rangle$ to \mathcal{I} and for every new edge $e \in E_{add}$ with $\sigma(e) = p_d$, we have $e \in \{f_i v_i, f_i v_{i+1} \mid i \in [F]\}$.*



■ **Figure 4** An overview of the created vertices and edges in our reduction. Green vertices represent variables, blue vertices clauses, and red vertices the dummy vertices d_q . Furthermore, we visualize some of the edges in H that are created for the variable-vertices (left) and clause-vertices (middle and right) to block visibility on the respective pages. If an edge is created due to the (non-)existence of a literal in the clauses c_1 , c_2 , or c_M it is indicated via a blue arc.

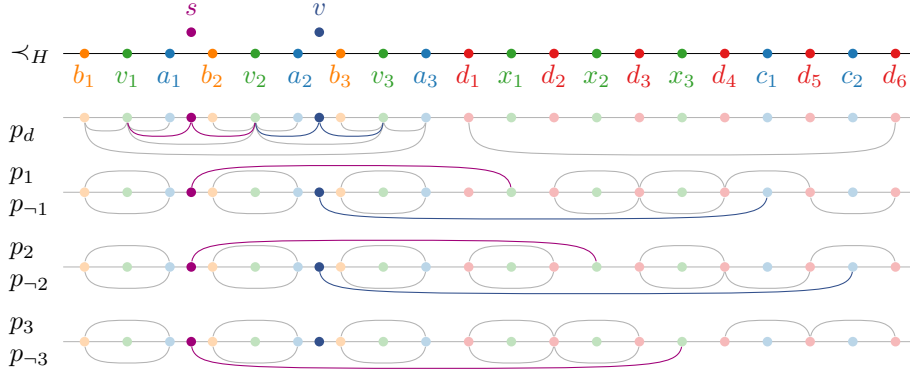
4.2 The Complete Reduction

Let $\varphi = (\mathcal{X}, \mathcal{C})$ be an instance of 3-SAT consisting of N variables $\mathcal{X} = \{x_1, \dots, x_N\}$ and M clauses $\mathcal{C} = \{c_1, \dots, c_M\}$, each consisting of three different and pairwise non-complementary literals. The reduction constructs an instance $\mathcal{I} = (\ell, G, H, \prec, \sigma)$ of SLE which represents each variable x_i and each clause c_j of φ , respectively, by a corresponding vertex in H . The linear order \prec_H has the form $x_1 \prec x_2 \prec \dots \prec x_N \prec c_1 \prec \dots \prec c_M$; see Figure 4. Furthermore, \mathcal{I} contains two new vertices s and v . The vertex s is adjacent to all variable-vertices and the construction will ensure that the page assignment for its incident edges represents, i.e., selects, a truth assignment Γ for φ . The vertex v is adjacent to all clause-vertices, and its purpose is to verify that the truth assignment satisfies all clauses. For the following description of how this is achieved, we assume $s \prec v \prec x_1$ as we will use a fixation gadget to ensure that every solution (\prec, σ) of \mathcal{I} has this property.

To each variable x_i , we associate two pages p_i and p_{-i} corresponding to its possible truth states. We ensure that s can see each variable-vertex only on its associated pages using edges incident to dummy vertices d_q with $q \in [N + M + 1]$. These dummy vertices are distributed as in Figure 4. Hence, a page assignment for the edges incident to s induces a truth assignment. Similar edges also ensure that v can see a clause-vertex c_j only on the pages that are associated to the negation of the literals the clause c_j is composed of, see the blue arcs in Figure 4 for an illustration. We defer the full construction to the full version [17].

We now ensure that $s \prec v \prec x_1$ holds in every solution of \mathcal{I} by using a fixation gadget on two vertices, i.e., for $F = 2$. In particular, we set $a_3 \prec d_1$, i.e., we place the fixation gadget at the beginning of the spine, and identify $s = f_1$ and $v = f_2$. The spine order \prec_H is then the transitive closure of all the partial orders stated until now; see Figure 4. Finally, we add the edge $d_1 d_{N+M+1}$ and set $\sigma(d_1 d_{N+M+1}) = p_d$ to ensure that our construction has Property 1.

Regarding the correctness of our reduction, we make the following observation. If an induced truth assignment does not satisfy a clause c_j , then it must use the pages associated to the negated literals of c_j . Thus, the new edge vc_j will cross another edge no matter which page we use. However, if a clause c_j is satisfied, we can find a page for the edge vc_j that does



■ **Figure 5** An example of our reduction for the formula φ consisting of the clauses $c_1 = (x_1 \vee \neg x_2 \vee x_3)$ and $c_2 = (\neg x_1 \vee x_2 \vee x_3)$. The extension indicated in saturated colors induces the truth assignment $\Gamma(x_1) = \Gamma(x_2) = 1$ and $\Gamma(x_3) = 0$, which satisfies φ .

not introduce a crossing: the page associated to the negation of the literal that satisfies c_j . Consequently, if φ is satisfiable, then there exists an extension $\langle \prec_G, \sigma_G \rangle$. Similarly, the page assignment of an extension $\langle \prec_G, \sigma_G \rangle$ induces a truth assignment Γ that satisfies φ . An intuitive example of the reduction is provided in Figure 5, and we obtain the following theorem.

► **Theorem 4.2 (★)**. *SLE is NP-complete even if we have just two new vertices and $E_{\text{add}}^H = \emptyset$.*

Finally, we want to remark that Theorem 4.2 is tight in the sense that SLE with only one new vertex v and $E_{\text{add}}^H = \emptyset$ can be solved in polynomial time. To that end, we can branch over all $\mathcal{O}(n)$ possible spine positions where v can be placed. For each of these, the observation that edges incident to the same vertex can never cross each other allows us to greedily assign a new edge uv to the first page p where v can see u . Recall that we only add one new vertex v . Hence, u is an old vertex whose spine position is known. Clearly, an extension exists if and only if there exists a spine position for v such that our greedy page assignment can find a page for all new edges.

► **Remark 4.3**. Let $\mathcal{I} = (\ell, G, H, \langle \prec, \sigma \rangle)$ be an instance of SLE with $n_{\text{add}} = 1$ and $E_{\text{add}}^H = \emptyset$. We can find an ℓ -page stack layout of G that extends $\langle \prec, \sigma \rangle$ or report that none exists in $\mathcal{O}(n \cdot m_{\text{add}} \cdot |\mathcal{I}|)$ time.

5 SLE Parameterized by Missing Vertices and Edges is in XP

In the light of Theorem 4.2, which excludes the use of the vertex+edge deletion distance as a pathway to tractability, we consider parameterizing by the total number of missing vertices and edges $\kappa := n_{\text{add}} + m_{\text{add}}$. As our first result in this direction, we show that parameterizing SLE by κ makes it XP-tractable. To this end, we combine a branching-procedure with the fixed parameter algorithm for the special case obtained in Theorem 3.2.

► **Theorem 5.1**. *Let $\mathcal{I} = (\ell, G, H, \langle \prec, \sigma \rangle)$ be an instance of SLE. We can find an ℓ -page stack layout of G that extends $\langle \prec, \sigma \rangle$ or report that none exists in $\mathcal{O}(|\mathcal{I}|^{n_{\text{add}}+1} m_{\text{add}}^{m_{\text{add}}})$ time.*

Proof. We branch over the possible assignments of new vertices to the intervals in \prec_H . As a solution could assign multiple vertices to the same interval, we also branch over the order in which the new vertices will appear in the spine order \prec_G . Observe that \prec_H induces

$|V(H)| + 1$ different intervals, out of which we have to choose n_{add} with repetition. Together with the possible orders of the new vertices, we can bound the number of branches by $n_{\text{add}}! \cdot \binom{|V(H)| + n_{\text{add}}}{n_{\text{add}}}$. We can simplify this expression to

$$\frac{n_{\text{add}}! \cdot (|V(H)| + n_{\text{add}})!}{n_{\text{add}}! \cdot ((|V(H)| + n_{\text{add}}) - n_{\text{add}})!} = \frac{(|V(H)| + n_{\text{add}})!}{|V(H)|!} = \prod_{i=1}^{n_{\text{add}}} (|V(H)| + i) = \mathcal{O}(|\mathcal{I}|^{n_{\text{add}}}).$$

In each branch, the spine order \prec_G is fixed and extends \prec_H . Hence, it only remains to check whether \prec_G allows for a valid page assignment σ_G . As each branch corresponds to an instance of SLE where only edges are missing, we use Theorem 3.2 to check in $\mathcal{O}(m_{\text{add}}^{m_{\text{add}}} \cdot |\mathcal{I}|)$ time whether such an assignment σ_G exists. The overall running time now follows readily. \blacktriangleleft

The running time stated in Theorem 5.1 not only proves that SLE is in XP when parameterized by κ , but also FPT when parameterized by m_{add} for constant n_{add} . However, common complexity assumptions rule out an efficient algorithm parameterized by κ , as we show next.

6 SLE Parameterized by Missing Vertices and Edges is W[1]-hard

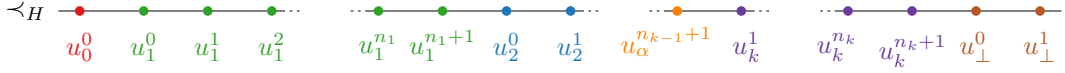
In this section, we show that SLE parameterized by the number κ of missing vertices and edges is W[1]-hard. To show W[1]-hardness, we reduce from the MULTI-COLORED CLIQUE (MCC) problem. Here, we are given a graph G_C , an integer $k > 0$, and a partition of $V(G_C)$ into k independent subsets V_1, \dots, V_k , and ask whether there exists a *colorful k -clique* $\mathcal{C} \subseteq V(G_C)$ in G_C , i.e., a clique on k vertices that contains exactly one vertex of every set V_i , $i \in [k]$. It is known that MCC is W[1]-hard when parameterized by k [15]. In the following, we will use Greek letters for the indices of the partition and denote with n_α the number of vertices in V_α , i.e., $n_\alpha = |V_\alpha|$. Observe that $\sum_{\alpha \in [k]} n_\alpha = N$ with $N = |V(G_C)|$. As we can interpret the partitioning of the vertices into V_1, \dots, V_k as assigning to them one of k colors, we will call a vertex v_α^i with $\alpha \in [k]$ and $i \in [n_\alpha]$ a vertex with *color* α . Our construction will heavily use the notion of a successor and predecessor of a vertex in a given spine order \prec . For a vertex u , the function $\text{succ}(\prec, u)$ returns the *successor* of u in the spine order \prec , i.e., the consecutive vertex in \prec after u . Note that $\text{succ}(\prec, u)$ is undefined if there is no vertex $v \in V(G)$ with $u \prec v$. We write $\text{succ}(u)$ if \prec is clear from context. The *predecessor* function $\text{pred}(\prec, u)$ is defined analogously. In the following, we first give an overview of and intuition behind our reduction in Section 6.1, before we show its correctness in Section 6.2. Note that the full details of the construction can be found in the full version [17]. Furthermore, as in the reduction from Section 4, we will allow multi-edges in the graph H to facilitate the presentation of the reduction. The procedure for removing multi-edges by distributing the individual edges over auxiliary vertices is also detailed in the full version [17].

6.1 An Overview of the Construction

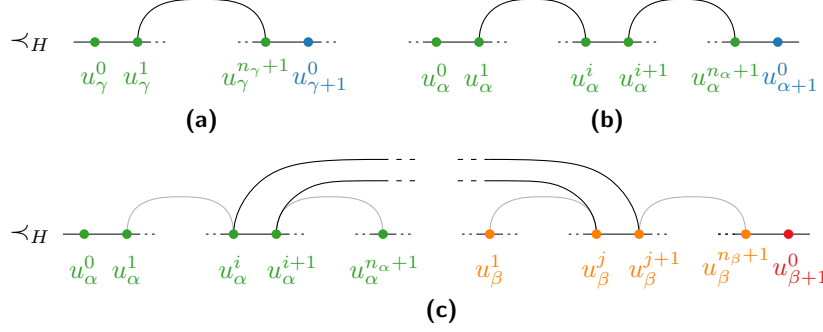
Let $(G_C, k, (V_1, \dots, V_k))$ be an instance of MCC. We will construct an SLE instance $(\ell, G, H, \langle \prec, \sigma \rangle)$ parameterized by κ that will fulfill two crucial properties to ensure its correctness. While, at the time of stating the property, our construction might not yet fulfill it, we show in Section 6.2 that in the end it indeed has the desired properties.

First, we define the base layout of our reduction. In the base layout, we create the $N + 2k + 3$ vertices $\{u_\alpha^j \mid \alpha \in [k], j \in [n_\alpha + 1]_0\} \cup \{u_0^0, u_\perp^0, u_\perp^1\}$ in H . Note that for each *original* vertex $v_\alpha^i \in V(G_C)$, we have a *copy* u_α^i . We will refer to the vertices u_0^0 , u_\perp^0 , and u_\perp^1 as *dummy vertices* and set, for ease of notation, $\perp = k + 1$ and $n_\perp = 1$. The vertices are placed on the spine based on their color α and index i ; see Figure 6. Finally, observe that $\text{succ}(u_\alpha^i) = u_\alpha^{i+1}$ for every $v_\alpha^i \in V(G_C)$. Furthermore, every vertex $v_\alpha^i \in V(G_C)$ induces the interval $[u_\alpha^i, u_\alpha^{i+1}]$

12:10 The Parameterized Complexity Of Extending Stack Layouts



■ **Figure 6** The base layout of our reduction. We use colors to additionally differentiate vertices that originate from different vertex sets V_α , for $\alpha \in [k]$, and the dummy vertices u_0^0 , u_{\perp}^0 and u_{\perp}^1 .



■ **Figure 7** Edges of H that model the adjacency given by the edge $e = v_\alpha^i v_\beta^j \in E(G_C)$. All of these edges are placed on the page p_e . Intuitively, we span the intervals induced **(a)** by all vertices for each color $\gamma \in [k] \setminus \{\alpha, \beta\}$ and **(b)** by vertices of the colors α and β that are not incident to e , here visualized for the color α . **(c)** Furthermore, we create a tunnel that connects $\Upsilon(v_\alpha^i)$ with $\Upsilon(v_\beta^j)$. The gray edges in **(c)** are from **(a)** and **(b)**.

in \prec_H , which we denote with $\Upsilon(v_\alpha^i)$. The equivalence between the two problems will be obtained by adding a k -clique to G that consists of the k new vertices $\mathcal{X} = \{x_1, \dots, x_k\}$. Placing $x_\alpha \in \mathcal{X}$ in $\Upsilon(v_\beta^j)$ indicates that v_β^j will be part of the colorful k -clique in G_C , i.e., we will have the equivalence $u_\alpha^i \prec x_\alpha \prec \text{succ}(\prec_H, u_\alpha^i) \stackrel{\text{p. d.}}{\iff} x_\alpha$ is placed in $\Upsilon(v_\alpha^i) \iff v_\alpha^i \in \mathcal{C}$ between a solution $\langle \prec_G, \sigma_G \rangle$ to SLE and a solution \mathcal{C} to MCC. To guarantee that \mathcal{C} is colorful, i.e., contains exactly one vertex from each color, we will ensure the following property with our construction.

► **Property 2.** *In a solution $\langle \prec, \sigma \rangle$ to SLE we have $u_\alpha^0 \prec x_\alpha \prec u_{\alpha+1}^0$ for every $\alpha \in [k]$.*

To establish the correctness of our reduction, we have to ensure two things. First, we have to model the adjacencies in G_C . In particular, two new vertices x_α and x_β , with $\alpha < \beta$, should only be placed in intervals induced by vertices adjacent in G_C . We enforce this by adding for every edge $e = v_\alpha^i v_\beta^j \in E(G_C)$ a page p_e . On this page p_e , we create the following edges in H ; see also Figure 7 for a visualization. Firstly, we create for every color $\gamma \in [k] \setminus \{\alpha, \beta\}$ an edge that spans exactly the intervals induced by vertices of color γ , thereby intuitively blocking visibility to any interval induced by a vertex of a color different to α and β ; see Figure 7a. Secondly, we create up to two edges that span all intervals induced by vertices of color α except $\Upsilon(v_\alpha^i)$; see Figure 7b. We do so similarly for color β . These edges in concert with a *tunnel* that we create on page p_e , see Figure 7c, allow us to place the edge $x_\alpha x_\beta \in E(G)$ in the page p_e if and only if x_α is placed in $\Upsilon(v_\alpha^i)$ and x_β in $\Upsilon(v_\beta^j)$. More formally, our construction will ensure the following property.

► **Property 3.** *Let $\langle \prec, \sigma \rangle$ be a solution to an instance of SLE that fulfills Property 2 and for which we have $e = v_\alpha^i v_\beta^j \in E(G_C)$, $1 \leq \alpha < \beta \leq k$, and $x_\alpha, x_\beta \in \mathcal{X}$. If $\sigma(x_\alpha x_\beta) = p_e$ then x_α is in $\Upsilon(v_\alpha^i)$ and x_β is in $\Upsilon(v_\beta^j)$.*

Second, we have to ensure that we select exactly one vertex $v_\alpha^i \in V_\alpha$ for every color $\alpha \in [k]$. In particular, the new vertex x_α should only be placed in intervals that are induced by vertices from V_α . To this end, we modify H to include a fixation gadget on $F = k$ vertices

by re-using some vertices of the base layout. Most importantly, we identify $v_\alpha = u_\alpha^0$ for every $\alpha \in [k+1]$ and $f_\alpha = x_\alpha$ for every $\alpha \in [k]$; see the full version [17] for details. As the whole base layout thereby forms the fixation gadget, our construction trivially satisfies Property 1.

6.2 Bringing It Together: Showing Correctness of the Reduction

With the overview of the construction and the intuition behind the reduction settled, we now proceed to show its correctness in Theorem 6.4. In the proof, we make use of Properties 2 and 3. Therefore, on our path to obtain Theorem 6.4, we first have to show that our construction fulfills them. Recall that Property 2 is defined as follows.

► **Property 2.** *In a solution $\langle \prec, \sigma \rangle$ to SLE we have $u_\alpha^0 \prec x_\alpha \prec u_{\alpha+1}^0$ for every $\alpha \in [k]$.*

When incorporating the fixation gadget on $F = k$ vertices in our construction, we identified $v_\alpha = u_\alpha^0$ and $f_\alpha = x_\alpha$ for every $\alpha \in [k]$. Similarly, we identified $v_{F+1} = u_{k+1}^0$. The fixation gadget now guarantees thanks to Lemma 4.1 that we have $v_\alpha \prec_G f_\alpha \prec_G v_{\alpha+1}$, i.e., $u_\alpha^0 \prec x_\alpha \prec u_{\alpha+1}^0$, in any solution $\langle \prec_G, \sigma_G \rangle$. Hence, we can observe the following.

► **Observation 6.1.** *Our instance \mathcal{I} of SLE fulfills Property 2.*

Recall that Lemma 4.1 furthermore tells us that we have in any solution $\langle \prec_G, \sigma_G \rangle$ the page assignment $\sigma(x_\alpha u_\alpha^0) = \sigma(x_\alpha u_{\alpha+1}^0) = p_d$ for every $\alpha \in [k]$. As we have by Property 2 $u_\alpha^0 \prec x_\alpha \prec u_{\alpha+1}^0$ and furthermore by the construction of the fixation gadget $\sigma_H(\text{pred}(u_\alpha^0)u_\alpha^0) = \sigma_H(u_\alpha^0 \text{succ}(u_\alpha^0)) = p_d$ for every $\alpha \in [k]$, we cannot have in $\langle \prec_G, \sigma_G \rangle$ $u_\alpha^0 \prec x_\alpha \prec \text{succ}(\prec_H, u_\alpha^0)$ or $\text{pred}(\prec_H, u_{\alpha+1}^0) \prec x_\alpha \prec u_{\alpha+1}^0$, as this would introduce a crossing on page p_d . As we have in \prec_H the equality $\text{succ}(u_\alpha^0) = u_\alpha^1$ and $\text{pred}(u_{\alpha+1}^0) = u_\alpha^{n_\alpha+1}$ for every $\alpha \in [k]$, we can strengthen Property 2 and obtain the following.

► **Corollary 6.2.** *In a solution $\langle \prec, \sigma \rangle$ to SLE we have $u_\alpha^1 \prec x_\alpha \prec u_\alpha^{n_\alpha+1}$ for every $\alpha \in [k]$.*

Finally, we now show that our construction fulfills Property 3, which was defined as follows.

► **Property 3.** *Let $\langle \prec, \sigma \rangle$ be a solution to an instance of SLE that fulfills Property 2 and for which we have $e = v_\alpha^i v_\beta^j \in E(G_C)$, $1 \leq \alpha < \beta \leq k$, and $x_\alpha, x_\beta \in \mathcal{X}$. If $\sigma(x_\alpha x_\beta) = p_e$ then x_α is in $\Upsilon(v_\alpha^i)$ and x_β is in $\Upsilon(v_\beta^j)$.*

► **Lemma 6.3.** *Our instance \mathcal{I} of SLE fulfills Property 3.*

Proof. First, recall that we made Observation 6.1, i.e., our construction fulfills Property 2. Let $\langle \prec_G, \sigma_G \rangle$ be a solution to SLE with $\sigma(x_\alpha x_\beta) = p_e$, for $e = v_\alpha^i v_\beta^j \in E(G_C)$, $1 \leq \alpha < \beta \leq k$. Corollary 6.2 tells us that $u_\alpha^1 \prec x_\alpha \prec u_\alpha^{n_\alpha+1}$ and $u_\beta^1 \prec x_\beta \prec u_\beta^{n_\beta+1}$ holds. Corollary 6.2 also holds for any new vertices x_γ and x_δ with $\gamma, \delta \in [k] \setminus \{\alpha, \beta\}$ and $\gamma \neq \delta$. Furthermore, we have the edges $u_\gamma^1 u_\gamma^{n_\gamma+1}$ and $u_\delta^1 u_\delta^{n_\delta+1}$ on page p_e . Hence, all new edges on page p_e must be among new vertices placed in intervals induced by vertices of color α or β .

Now assume that we have $u_\alpha^1 \preceq x_\alpha \preceq u_\alpha^i$. Using $\sigma_H(u_\alpha^1 u_\alpha^i) = p_e$ together with $u_\alpha^1 \preceq x_\alpha \preceq u_\alpha^i \prec x_\beta$, we derive that $u_\alpha^1 \preceq x_\alpha \preceq u_\alpha^i$ results in a crossing on page p_e . Hence, $u_\alpha^1 \preceq x_\alpha \preceq u_\alpha^i$ cannot hold. Now assume that we have $u_\alpha^{i+1} \preceq x_\alpha \preceq u_\alpha^{n_\alpha+1}$. From $\sigma_H(u_\alpha^{i+1} u_\alpha^{n_\alpha+1}) = p_e$ and $u_\alpha^{i+1} \preceq x_\alpha \preceq u_\alpha^{n_\alpha+1} \prec x_\beta$ we get that $u_\alpha^{i+1} \preceq x_\alpha \preceq u_\alpha^{n_\alpha+1}$ results in a crossing on page p_e . Hence, $u_\alpha^{i+1} \preceq x_\alpha \preceq u_\alpha^{n_\alpha+1}$ cannot hold. Since we can exclude $u_\alpha^1 \preceq x_\alpha \preceq u_\alpha^i$ and $u_\alpha^{i+1} \preceq x_\alpha \preceq u_\alpha^{n_\alpha+1}$ by the construction of the tunnel on page p_e , we can derive that x_α must be placed in $\Upsilon(v_\alpha^i)$. As similar arguments can be made for x_β , we can conclude that we get a crossing on page p_e unless x_α is placed in $\Upsilon(v_\alpha^i)$ and x_β in $\Upsilon(v_\beta^j)$. ◀

► **Theorem 6.4 (★).** *SLE parameterized by the number κ of missing vertices and edges is $W[1]$ -hard.*

Proof sketch. Let $(G_C, k, (V_1, \dots, V_k))$ be an instance of MCC with $N = |V(G_C)|$ and $M = |E(G_C)|$ and let $\mathcal{I} = (\ell, G, H, \langle \prec, \sigma \rangle)$ be the instance of SLE parameterized by the number κ of missing vertices and edges created by our construction described above. Closer analysis reveals that the size of \mathcal{I} is bounded by $\mathcal{O}(N + Mk + k^2)$, and we have $\kappa = 3k + \binom{k}{2}$ as $n_{\text{add}} = k$ and $m_{\text{add}} = \binom{k}{2} + 2k$; recall that the fixation gadget contributes $2k$ new edges.

Towards arguing correctness, assume that $(G_C, k, (V_1, \dots, V_k))$ contains a colorful k -clique \mathcal{C} . We construct a solution to \mathcal{I} by, for every new vertex $x_\alpha \in \mathcal{X}$, considering the vertex $v_\alpha^i \in \mathcal{C}$ and placing x_α immediately to the right of the copy u_α^i of v_α^i in H . The fact that \mathcal{C} is a clique then guarantees that, for each edge $e \in E(G_C[\mathcal{C}])$, there exists the page p_e in which the corresponding edge $e' \in E(G[\mathcal{X}])$ can be placed in. For the remaining edges from the fixation gadget, we can use the page assignment from Lemma 4.1.

For the converse (and more involved) direction, assume that SLE admits a solution $\langle \prec_G, \sigma_G \rangle$. By Property 2, we have that each $x_\alpha \in V_{\text{add}}$ must be placed between u_α^0 and $u_{\alpha+1}^0$. Moreover, our construction together with the page assignment forced by Lemma 4.1 guarantees that x_α is placed between precisely one pair of consecutive vertices $u_\alpha^{i_\alpha}$ and $u_\alpha^{i_\alpha+1}$, for some $i_\alpha \in [n_\alpha]$; recall Corollary 6.2. Our solution \mathcal{C} to the instance of MCC will consist of the vertices $v_\alpha^{i_\alpha}$, i.e., exactly one vertex per color α . Moreover, each new edge $x_\alpha x_\beta \in E(G[\mathcal{X}])$ must be placed by σ_G on some page, and as our construction satisfies Properties 1 and 3, this page must be one that is associated to one edge $e = v_\alpha^{i_\alpha} v_\beta^{i_\beta}$ of G_C . Property 3 now also guarantees that this page assignment enforces that x_α and x_β are placed precisely between the consecutive vertices $u_\alpha^{i_\alpha}$ and $u_\alpha^{i_\alpha+1}$ and $u_\beta^{i_\beta}$ and $u_\beta^{i_\beta+1}$ of H , respectively. This means that the vertices in \mathcal{C} are pairwise adjacent, which implies that \mathcal{C} is a colorful k -clique. ◀

Figure 8 shows an example of the reduction for a small graph G_C with three colors. Since in a stack layout constructed by our reduction each line perpendicular to the spine intersects a constant number of edges, see also Figure 8, we also obtain:

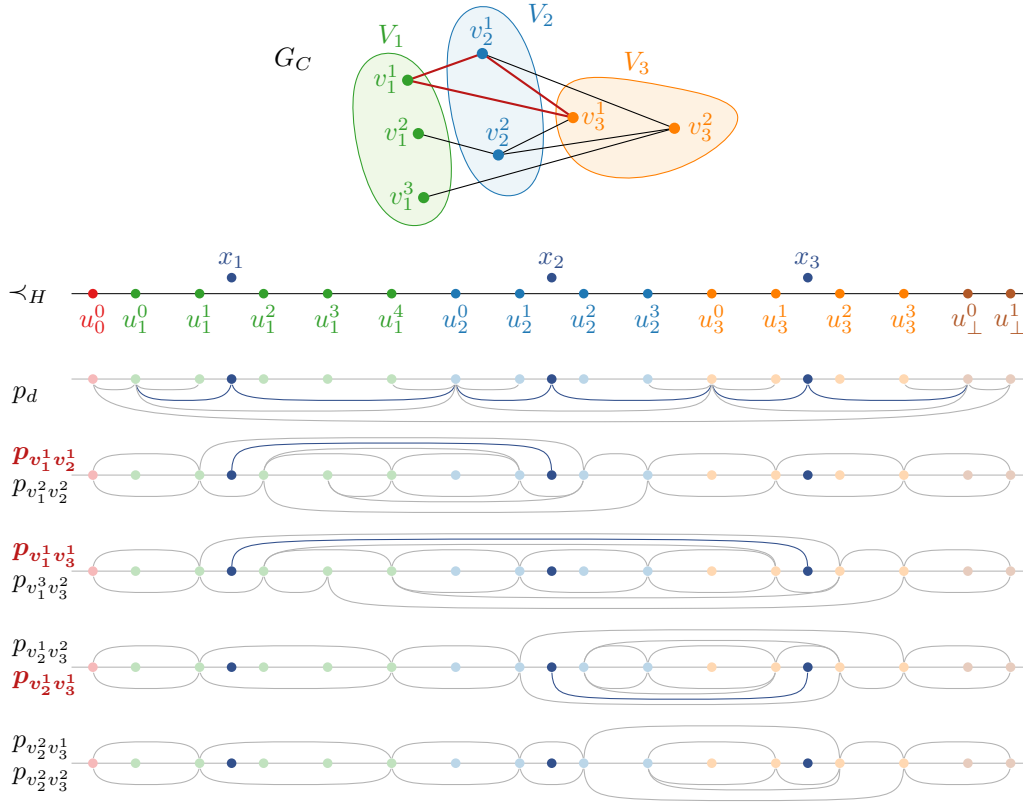
► **Corollary 6.5.** *SLE parameterized by the number κ of missing vertices and edges and the page width ω of the given layout, i.e., by $\kappa + \omega$, is $W[1]$ -hard.*

7 Adding the Number of Pages as Parameter for SLE

In this section, we complete the landscape of Figure 2 by showing that SLE becomes fixed-parameter tractable once we add ℓ to the parameterization considered by Corollary 6.5. We will make use of the following concepts.

Consider a page p of a stack layout $\langle \prec, \sigma \rangle$ of G and recall that we can interpret it as a plane drawing of the graph G' with $V(G') = V(G)$ and $E(G') = \{e \in E(G) \mid \sigma(e) = p\}$ on a half-plane, where the edges are drawn as (circular) arcs. A *face* on the page p in $\langle \prec, \sigma \rangle$ coincides with the notion of a face in the drawing (on the half-plane p) of G' . This also includes the definition of the *outer face*. See Figure 9 for a visualization of this concept and observe that we can identify every face, except the outer face, by the unique edge $e = uv \in E(G)$ with $u \prec v$ and $\sigma(e) = p$ that bounds it from upwards.

Let $V_{\text{inc}} \subseteq V(H)$ be the vertices of H that are incident to new edges, i.e., $V_{\text{inc}} := \{u \in V(H) \mid \text{there is an edge } e = uv \in E_{\text{add}}\}$. The size of V_{inc} is upper-bounded by $2m_{\text{add}}$. We will define an equivalence class on the intervals of \prec_H based on the location of the vertices from V_{inc} . Consider the two intervals $[u_1, v_1]$ and $[u_2, v_2]$ defined by the old vertices u_1, v_1, u_2 and v_2 , respectively. These two intervals are in the same equivalence class if and only if



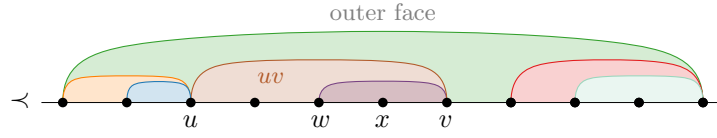
■ **Figure 8** An instance $(G_C, 3, (V_1, V_2, V_3))$ of MCC (top) and the SLE instance resulting from our construction (bottom). Colors indicate (correspondence to) the partition. The extension $\langle \prec, \sigma \rangle$ indicated in saturated colors induces the colorful 3-clique $\mathcal{C} = \{v_1^1, v_2^1, v_3^1\}$ in G_C . The edges in $G_C[\mathcal{C}]$ and their corresponding pages are highlighted in red.

$\{w \in V_{\text{inc}} \mid w \preceq u_1\} = \{w \in V_{\text{inc}} \mid w \preceq u_2\}$ and $\{w \in V_{\text{inc}} \mid v_1 \preceq w\} = \{w \in V_{\text{inc}} \mid v_2 \preceq w\}$ holds. Each equivalence class, which we call *super interval*, consists of a set of consecutive intervals delimited by (up to) two old vertices; see Figure 10. Note that the first and last super interval are defined by a single vertex $v \in V_{\text{inc}}$. The number of super intervals is bounded by $2m_{\text{add}} + 1$. Furthermore, for a given \prec_G , we define $\prec_{G \setminus H}$ to be its restriction to new vertices, i.e., for every two vertices $u, v \in V_{\text{add}}$ we have that $u \prec v$ implies $u \prec_{G \setminus H} v$.

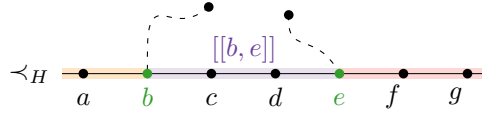
The Algorithm. With the above concepts at hand, we can now describe our algorithm. It consists of a branching step, where we consider all possible page assignments for the new edges, all relative orders among the new vertices, all their possible assignments to super intervals, and all distances new edges can have from the outer face. In the following, we show that we can verify in polynomial-time whether a branch can be extended to a solution $\langle \prec, \sigma \rangle$ or not. The core of our algorithm is a dynamic program that we apply in each branch.

► **Lemma 7.1 (★).** *Given an instance $\mathcal{I} = (\ell, G, H, \langle \prec, \sigma \rangle)$ of SLE, (i) a page assignment σ_G for all edges, (ii) an order $\prec_{G \setminus H}$ in which the new vertices will appear along the spine, (iii) for every new vertex $v \in V_{\text{add}}$ an assignment to a super interval, and (iv) for every new edge e an assigned distance ω_e to the outer face with respect to H and $\langle \prec, \sigma \rangle$. In $\mathcal{O}(n_{\text{add}} \cdot m_{\text{add}} \cdot |\mathcal{I}|)$ time we can compute an ℓ -page stack layout of G that extends $\langle \prec, \sigma \rangle$ and respects the given assignments (i)–(iv) or report that no such layout exists.*

12:14 The Parameterized Complexity Of Extending Stack Layouts



■ **Figure 9** A stack layout $\langle \prec, \sigma \rangle$ and the faces on page p . Note that each edge has the same color as the face it identifies.



■ **Figure 10** Visualization of super intervals. Each color represents one super interval. Vertices from V_{inc} are marked in green.

Proof sketch. We first observe that assignments (i)–(iv) fix everything except for the actual position of the new vertices within their super interval. Especially, assignment (i) allows us to check whether an edge $e \in E_{\text{add}}^H$ incident to two old vertices crosses any old edge or another new edge from E_{add}^H . Furthermore, assignments (i) and (ii) allow us to check whether two new edges $e = ua, e' = vb \in E_{\text{add}}$ with $u, a, v, b \in V_{\text{add}}$ will cross. Adding assignment (iii), we can also check this for new edges with some endpoints in $V(H)$, i.e., extend this to all $u, a, v, b \in V(G)$. If the assignments imply a crossing or contradict each other, we can directly return that no desired layout exists. These checks can be performed in $\mathcal{O}(n_{\text{add}}^2 + m_{\text{add}} \cdot |\mathcal{I}|)$ time. It remains to check whether there exists a stack layout in which no edge of $E_{\text{add}} \setminus E_{\text{add}}^H$ intersects an old edge. This depends on the exact intervals new vertices are placed in.

To do so, we need to assign new vertices to faces such that adjacent new vertices are in the exact same face and not two different faces with the same distance to the outer face. We will find this assignment using a dynamic program that models whether there is a solution that places the first j new vertices (according to $\prec_{G \setminus H}$) within the first i intervals in \prec_H . When placing vertex v_{j+1} in the i th interval, we check that all preceding neighbors are visible in the faces assigned by (iv). When advancing to the interval $i + 1$, we observe that when we leave a face, all edges with the same or a higher distance to the outer face need to have both endpoints placed or none. We thus ensure that for no edge only one endpoint has been placed; see also Figure 11. These checks require $\mathcal{O}(m_{\text{add}})$ time for each of the $\mathcal{O}(n_{\text{add}} \cdot |V(H)|)$ combinations of j and i . Once we reach the interval $|V(H)| + 1$ and have successfully placed all n_{add} new vertices, we know that there exists an ℓ -page stack layout of G that extends $\langle \prec, \sigma \rangle$ and respects the assignments. Finally, by applying standard backtracing techniques, we can extract the spine positions of the new vertices to also obtain the layout. ◀

We observe that there are $\mathcal{O}(\ell^{m_{\text{add}}} \cdot n_{\text{add}}! \cdot m_{\text{add}}^{n_{\text{add}}} \cdot \omega^{m_{\text{add}}})$ different possibilities for assignments (i)–(iv). Applying Lemma 7.1 to each of these, we get the following theorem.



■ **Figure 11** Illustration of advancing from the i th interval, marked in blue, to the interval $i + 1$. In (a) and (b), we leave the green face and there exists an edge $e \in E_{\text{add}}$, marked in orange, with the same distance to the outer face as the green face. However, in (a), both end points of the edge e have already been placed, whereas in (b) only one has, which implies a crossing.

► **Theorem 7.2 (★).** Let $\mathcal{I} = (\ell, G, H, \langle \prec, \sigma \rangle)$ be an instance of SLE. We can find an ℓ -page stack layout of G that extends $\langle \prec, \sigma \rangle$ or report that none exists in $\mathcal{O}(\ell^{m_{add}} \cdot n_{add}! \cdot m_{add}^{n_{add}} \cdot \omega^{m_{add}} \cdot (n_{add} \cdot m_{add} \cdot |\mathcal{I}|))$ time.

8 Concluding Remarks

Our results provide the first investigation of the drawing extension problem for stack layouts through the lens of parameterized algorithmics. We show that the complexity-theoretic behavior of the problem is surprisingly rich and differs from that of previously studied drawing extension problems. One prominent question left for future work is whether one can still achieve fixed-parameter tractability for SLE when parameterizing by $\kappa + \ell$, thus generalizing Theorem 7.2. As our final result, we show that this is indeed possible at least in the restricted case where no two missing vertices are adjacent, as we can then greedily assign the first “possible” interval to each vertex that complies with assignment (i)–(iii).

► **Theorem 8.1 (★).** Let $\mathcal{I} = (\ell, G, H, \langle \prec, \sigma \rangle)$ be an instance of SLE where $G[V_{add}]$ is an independent set. We can find an ℓ -page stack layout of G that extends $\langle \prec, \sigma \rangle$ or report that none exists in $\mathcal{O}(\ell^{m_{add}} \cdot n_{add}! \cdot m_{add}^{n_{add}} \cdot (m_{add}|\mathcal{I}|^2))$ time.

A further natural and promising direction for future work is to consider generalizing the presented techniques to other types of linear layouts, such as queue layouts. Finally, future work could also investigate the following generalized notion of extending linear layouts: Given a graph G , the spine order for some subset of its vertices and the page assignment for some subset of its edges, does there exist a linear layout of G that extends both simultaneously?

References

- 1 Patrizio Angelini, Giuseppe Di Battista, Fabrizio Frati, Vít Jelínek, Jan Kratochvíl, Maurizio Patrignani, and Ignaz Rutter. Testing Planarity of Partially Embedded Graphs. *ACM Transactions on Algorithms*, 11(4):1–42, 2015. doi:10.1145/2629341.
- 2 Patrizio Angelini, Ignaz Rutter, and T. P. Sandhya. Extending Partial Orthogonal Drawings. In David Auber and Pavel Valtr, editors, *Proc. 28th International Symposium on Graph Drawing and Network Visualization (GD’20)*, volume 12590 of *Lecture Notes in Computer Science*, pages 265–278. Springer, 2020. doi:10.1007/978-3-030-68766-3_21.
- 3 Patrizio Angelini, Ignaz Rutter, and T. P. Sandhya. Extending Partial Orthogonal Drawings. *Journal of Graph Algorithms and Applications*, 25(1):581–602, 2021. doi:10.7155/jgaa.00573.
- 4 Michael Baur and Ulrik Brandes. Crossing Reduction in Circular Layouts. In Juraž Hromkovič, Manfred Nagl, and Bernhard Westfechtel, editors, *Proc. 30th International Workshop on Graph-Theoretic Concepts in Computer Science (WG’04)*, volume 3353 of *Lecture Notes in Computer Science*, pages 332–343. Springer, 2004. doi:10.1007/978-3-540-30559-0_28.
- 5 Michael A. Bekos, Michael Kaufmann, Fabian Klute, Sergey Pupyrev, Chrysanthi N. Raftopoulou, and Torsten Ueckerdt. Four Pages Are Indeed Necessary for Planar Graphs. *Journal of Computational Geometry*, 11(1):332–353, 2020. doi:10.20382/JOCG.V11I1A12.
- 6 Frank Bernhart and Paul C. Kainen. The Book Thickness of a Graph. *Journal of Combinatorial Theory, Series B*, 27(3):320–331, 1979. doi:10.1016/0095-8956(79)90021-2.
- 7 Sujoy Bhore, Giordano Da Lozzo, Fabrizio Montecchiani, and Martin Nöllenburg. On the Upward Book Thickness Problem: Combinatorial and Complexity Results. In Helen C. Purchase and Ignaz Rutter, editors, *Proc. 29th International Symposium on Graph Drawing and Network Visualization (GD’21)*, volume 12868 of *Lecture Notes in Computer Science*, pages 242–256. Springer, 2021. doi:10.1007/978-3-030-92931-2_18.


- 8 Sujoy Bhore, Giordano Da Lozzo, Fabrizio Montecchiani, and Martin Nöllenburg. On the Upward Book Thickness Problem: Combinatorial and Complexity Results. *European Journal of Combinatorics*, 110:103662, 2023. doi:10.1016/j.ejc.2022.103662.
- 9 Sujoy Bhore, Robert Ganian, Liana Khazaliya, Fabrizio Montecchiani, and Martin Nöllenburg. Extending Orthogonal Planar Graph Drawings Is Fixed-Parameter Tractable. In Erin W. Chambers and Joachim Gudmundsson, editors, *Proc. 39th International Symposium on Computational Geometry (SoCG'23)*, volume 258 of *LIPICs*, pages 18:1–18:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICs.SOCG.2023.18.
- 10 Sujoy Bhore, Robert Ganian, Fabrizio Montecchiani, and Martin Nöllenburg. Parameterized Algorithms for Book Embedding Problems. In Daniel Archambault and Csaba D. Tóth, editors, *Proc. 27th International Symposium on Graph Drawing and Network Visualization (GD'19)*, volume 11904 of *Lecture Notes in Computer Science*, pages 365–378. Springer, 2019. doi:10.1007/978-3-030-35802-0_28.
- 11 Sujoy Bhore, Robert Ganian, Fabrizio Montecchiani, and Martin Nöllenburg. Parameterized Algorithms for Book Embedding Problems. *Journal of Graph Algorithms and Applications*, 24(4):603–620, 2020. doi:10.7155/jgaa.00526.
- 12 Tomasz Bilski. Embedding graphs in books: a survey. *IEE Proceedings E (Computers and Digital Techniques)*, 139(2):134, 1992. doi:10.1049/ip-e.1992.0021.
- 13 Guido Brückner and Ignaz Rutter. Partial and Constrained Level Planarity. In Philip N. Klein, editor, *Proc. 28th ACM-SIAM Symposium on Discrete Algorithms (SODA'17)*, pages 2000–2011. SIAM, 2017. doi:10.1137/1.9781611974782.130.
- 14 Fan R. K. Chung, Frank Thomson Leighton, and Arnold L. Rosenberg. Embedding Graphs in Books: A Layout Problem with Applications to VLSI Design. *SIAM Journal on Algebraic Discrete Methods*, 8(1):33–58, 1987. doi:10.1137/0608002.
- 15 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshantov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
- 16 Giordano Da Lozzo, Giuseppe Di Battista, and Fabrizio Frati. Extending Upward Planar Graph Drawings. *Computational Geometry Theory and Applications*, 91:101668, 2020. doi:10.1016/j.comgeo.2020.101668.
- 17 Thomas Depian, Simon D. Fink, Robert Ganian, and Martin Nöllenburg. The Parameterized Complexity of Extending Stack Layouts, 2024. doi:10.48550/arXiv.2409.02833.
- 18 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate Texts in Mathematics*. Springer, 2012.
- 19 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.
- 20 Vida Dujmović and David R. Wood. On Linear Layouts of Graphs. *Discrete Mathematics & Theoretical Computer Science*, Vol. 6 no. 2, 2004. doi:10.46298/dmtcs.317.
- 21 Eduard Eiben, Robert Ganian, Thekla Hamm, Fabian Klute, and Martin Nöllenburg. Extending Partial 1-Planar Drawings. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *Proc. 47th International Colloquium on Automata, Languages and Programming (ICALP'20)*, volume 168 of *LIPICs*, pages 43:1–43:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.ICALP.2020.43.
- 22 Eduard Eiben, Robert Ganian, Thekla Hamm, Fabian Klute, and Martin Nöllenburg. Extending Nearly Complete 1-Planar Drawings in Polynomial Time. In Javier Esparza and Daniel Král', editors, *Proc. 45th Mathematical Foundations of Computer Science (MFCS'20)*, volume 170 of *LIPICs*, pages 31:1–31:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.MFCS.2020.31.
- 23 Robert Ganian, Thekla Hamm, Fabian Klute, Irene Parada, and Birgit Vogtenhuber. Crossing-Optimal Extension of Simple Drawings. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *Proc. 48th International Colloquium on Automata, Languages and Programming (ICALP'21)*, volume 198 of *LIPICs*, pages 72:1–72:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICs.ICALP.2021.72.

- 24 Robert Ganian, Haiko Müller, Sebastian Ordyniak, Giacomo Paesani, and Mateusz Rychlicki. A Tight Subexponential-Time Algorithm for Two-Page Book Embedding. In Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson, editors, *Proc. 51st International Colloquium on Automata, Languages and Programming (ICALP'24)*, volume 297 of *LIPICs*, pages 68:1–68:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICs.ICALP.2024.68.
- 25 Emden R. Gansner and Yehuda Koren. Improved Circular Layouts. In Michael Kaufmann and Dorothea Wagner, editors, *Proc. 14th International Symposium on Graph Drawing and Network Visualization (GD'06)*, volume 4372 of *Lecture Notes in Computer Science*, pages 386–398. Springer, 2006. doi:10.1007/978-3-540-70904-6_37.
- 26 Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- 27 Christian Haslinger and Peter F. Stadler. RNA Structures with Pseudo-knots: Graph-theoretical, Combinatorial, and Statistical Properties. *Bulletin of Mathematical Biology*, 61(3):437–467, 1999. doi:10.1006/bulm.1998.0085.
- 28 Vít Jelínek, Jan Kratochvíl, and Ignaz Rutter. A Kuratowski-Type Theorem for Planarity of Partially Embedded Graphs. *Computational Geometry Theory and Applications*, 46(4):466–492, 2013. doi:10.1016/j.comgeo.2012.07.005.
- 29 Yunlong Liu, Jie Chen, Jingui Huang, and Jianxin Wang. On Parameterized Algorithms for Fixed-Order Book Thickness with respect to the Pathwidth of the Vertex Ordering. *Theoretical Computer Science*, 873:16–24, 2021. doi:10.1016/J.TCS.2021.04.021.
- 30 Taylor L. Ollmann. On the Book Thicknesses of Various Graphs. In *Proc. 4th Southeastern Conference on Combinatorics, Graph Theory and Computing*, volume 8, page 459, 1973.
- 31 Maurizio Patrignani. On Extending a Partial Straight-Line Drawing. In Patrick Healy and Nikola S. Nikolov, editors, *Proc. 13th International Symposium on Graph Drawing and Network Visualization (GD'05)*, volume 3843 of *Lecture Notes in Computer Science*, pages 380–385. Springer, 2005. doi:10.1007/11618058_34.
- 32 Maurizio Patrignani. On Extending a Partial Straight-Line Drawing. *International Journal of Foundations of Computer Science*, 17(5):1061–1070, 2006. doi:10.1142/S0129054106004261.
- 33 Sergey Pupyrev. A Collection of Existing Results on Stack and Queue Numbers, 2023. Last accessed: 2024-05-20. URL: <https://spupyrev.github.io/linearlayouts.html>.
- 34 Elena Stöhr. The pagewidth of trivalent planar graphs. *Discrete Mathematics*, 89(1):43–49, 1991. doi:10.1016/0012-365X(91)90398-L.
- 35 Elena Stöhr. A Trade-off between Page Number and Page Width of Book Embeddings of Graphs. *Information and Computation*, 79(2):155–162, 1988. doi:10.1016/0890-5401(88)90036-3.
- 36 Walter Unger. On the k -Colouring of Circle-Graphs. In Robert Cori and Martin Wirsing, editors, *Proc. 5th Symposium on Theoretical Aspects of Computer Science (STACS'88)*, volume 294 of *Lecture Notes in Computer Science*, pages 61–72. Springer, 1988. doi:10.1007/BFB0035832.
- 37 Walter Unger. The Complexity of Colouring Circle Graphs (Extended Abstract). In Alain Finkel and Matthias Jantzen, editors, *Proc. 9th Symposium on Theoretical Aspects of Computer Science (STACS'92)*, volume 577 of *Lecture Notes in Computer Science*, pages 389–400. Springer, 1992. doi:10.1007/3-540-55210-3_199.
- 38 Martin Wattenberg. Arc Diagrams: Visualizing Structure in Strings. In Pak Chung Wong and Keith Andrews, editors, *Proc. 8th IEEE Information Visualization Conference (InfoVis'02)*, pages 110–116. IEEE Computer Society, 2002. doi:10.1109/INFVIS.2002.1173155.
- 39 Avi Wigderson. The Complexity of the Hamiltonian Circuit Problem for Maximal Planar Graphs. Technical Report 298, Princeton University, 1982.
- 40 Mihalis Yannakakis. Embedding Planar Graphs in Four Pages. *Journal of Computer and System Sciences*, 38(1):36–67, 1989. doi:10.1016/0022-0000(89)90032-9.

The Price of Upwardness

Patrizio Angelini ✉ 

John Cabot University, Rome, Italy

Markus Chimani ✉ 

University of Osnabrück, Germany

Giordano Da Lozzo ✉ 

Roma Tre University, Italy

Giuseppe Liotta ✉ 

University of Perugia, Italy

Sergey Pupyrev ✉ 

Menlo Park, CA, USA

Alexander Wolff 

University of Würzburg, Germany

Therese Biedl ✉ 

University of Waterloo, Canada

Sabine Cornelsen ✉ 

University of Konstanz, Germany

Seok-Hee Hong ✉ 

University of Sydney, Australia

Maurizio Patrignani ✉ 

Roma Tre University, Italy

Ignaz Rutter ✉ 

University of Passau, Germany

Abstract

Not every directed acyclic graph (DAG) whose underlying undirected graph is planar admits an upward planar drawing. We are interested in pushing the notion of upward drawings beyond planarity by considering upward k -planar drawings of DAGs in which the edges are monotonically increasing in a common direction and every edge is crossed at most k times for some integer $k \geq 1$. We show that the number of crossings per edge in a monotone drawing is in general unbounded for the class of bipartite outerplanar, cubic, or bounded pathwidth DAGs. However, it is at most two for outerpaths and it is at most quadratic in the bandwidth in general. From the computational point of view, we prove that upward- k -planarity testing is NP-complete already for $k = 1$ and even for restricted instances for which upward planarity testing is polynomial. On the positive side, we can decide in linear time whether a single-source DAG admits an upward 1-planar drawing in which all vertices are incident to the outer face.

2012 ACM Subject Classification Theory of computation \rightarrow Design and analysis of algorithms; Mathematics of computing \rightarrow Graph theory

Keywords and phrases upward drawings, beyond planarity, upward k -planarity, upward outer-1-planarity

Digital Object Identifier 10.4230/LIPIcs.GD.2024.13

Related Version *Full Version*: <https://arxiv.org/abs/2409.01475> [1]

Funding Research by Da Lozzo, Liotta, and Patrignani was supported, in part, by MUR of Italy (PRIN Project no. 2022ME9Z78 – NextGRAAL and PRIN Project no. 2022TS4Y3N – EXPAND). Research by Liotta was supported in part by MUR PON Proj. ARS01_00540. Research by Rutter was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 541433306. Research by Biedl was supported by NSERC FRN RGPIN-2020-03958.

Acknowledgements This work was initiated at the Dagstuhl Seminar “Beyond-Planar Graphs: Models, Structures and Geometric Representations” (No. 24062), February 2024.

1 Introduction

Graph drawing “beyond planarity” studies the combinatorial and algorithmic questions related to representations of graphs where edges can cross but some crossing configurations are forbidden. Depending on the forbidden crossing configuration, different beyond-planar types of drawings can be defined including, for example, RAC, k -planar, fan planar, and quasi planar drawings. See [19, 30, 32] for surveys and books.



© Patrizio Angelini, Therese Biedl, Markus Chimani, Sabine Cornelsen, Giordano Da Lozzo, Seok-Hee Hong, Giuseppe Liotta, Maurizio Patrignani, Sergey Pupyrev, Ignaz Rutter, and Alexander Wolff;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

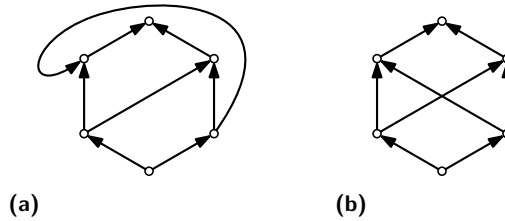
Editors: Stefan Felsner and Karsten Klein; Article No. 13; pp. 13:1–13:20



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

While most of the literature about beyond planar graph drawing has focused on undirected graphs (one of the few exceptions being [2,3] which studies RAC upward drawings), we study *upward k -planar drawings* of acyclic digraphs (DAGs), i.e., drawings of DAGs where the edges monotonically increase in y -direction and each edge can be crossed at most k times. The minimum k such that a DAG admits an upward k -planar drawing is called its *upward local crossing number*. We focus on values of $k = 1, 2$ and investigate both combinatorial properties and complexity questions. Our research is motivated by the observation that well-known DAGs that are not *upward-planar*, i.e., not upward 0-planar, do admit a drawing where every edge is crossed at most a constant number of times; see, e.g., Figure 1.



■ **Figure 1** A graph that is not upward planar but admits an upward 1-planar drawing.

Our contribution.

- A graph is an *outerpath* if it has a planar drawing in which each vertex is incident to the outer face and the internal faces induce a path in the dual graph. Papakostas [35] observed that there is a directed acyclic 8-vertex outerpath that is not upward-planar (see Figure 3a). We strengthen this observation by showing that there exists a directed acyclic fan (that is, a very specific outerpath) that has no upward-planar drawing (Proposition 1). On the other hand, we show that every directed acyclic outerpath is upward 2-planar (Theorem 9) and that the upward local crossing number is quadratic in the bandwidth (Theorem 6). However, the upward local crossing number of bipartite outerplanar DAGs (Theorem 2), bipartite DAGs with bounded pathwidth (Corollary 4), and cubic DAGs (Proposition 5) is in general unbounded.
- We show that upward 1-planarity testing is NP-complete, even for graph families where upward planarity testing can be solved in polynomial time. These include: single-source single-sink series-parallel DAGs with a fixed rotation system; single-source two-sink series-parallel DAGs where the rotation system is not fixed; and single-source single-sink DAGs without fixed rotation system that can be obtained from a K_4 by replacing the edges with series-parallel DAGs (Theorem 11).
- Finally, following a common trend in the study of beyond planar graph representations, we consider the *outer model*, in which all vertices are required to lie on a common face while maintaining the original requirements [19,30,32]. We prove that testing whether a single-source DAG admits an upward outer-1-planar drawing can be done in linear time (Theorem 13).

The details of omitted or sketched proofs can be found in the full version [1].

Related Work. A drawing of a graph is *monotone* if all edges are drawn monotone with respect to some direction, e.g., a drawing is *y -monotone* or *upward*, if each edge intersects each horizontal line at most once. The corresponding crossing number is introduced and studied in [24,37]. Schaefer [36] mentions the upward crossing number and the local crossing number

but not their combination. Schaefer [36, p. 64] also showed that a drawing with the minimum number of crossings per edge can require incident edges that cross. The edges of the provided 4-planar example graph can be oriented such that the resulting directed graph admits an upward 4-planar drawing. Thus, also an upward drawing that achieves the minimum local crossing number can require incident edges that cross. Also, the so-called strong Hanani–Tutte theorem carries over to directed graphs: Fulek et al. [24, Theorem 3.1] showed that every undirected graph that has a monotone drawing where any pair of independent edges crosses an even number of times also has a planar monotone drawing with the same vertex positions. This implies that in any upward drawing of a graph that is not upward-planar there must be a pair of independent edges that crosses an odd number of times.

Upward drawings of directed acyclic graphs have been studied in the context of (upward) book embeddings. In that model the vertices are drawn on a vertical line (a spine) following a topological order of the graph, while all edges are pointing upwards. To reduce the edge crossings, edges are partitioned into the fewest number of crossing-free subsets (pages). Studying upward book embeddings is a popular topic, which is usually centered around determining the smallest number of pages for various graph classes [22,23,29,31,34] or deciding whether a graph admits an upward drawing with a given number of pages [7,8,10,11,12]. Our model is equivalent to *topological book embeddings* [28,33], which are a relaxed version of book embeddings in which edges are allowed to cross the spine. To the best of our knowledge, earlier papers considered only the problem of minimizing the number of spine crossings, whereas we want to bound the maximum number of edge crossings per edge (ignoring the spine).

2 Preliminaries

A *drawing* Γ of a graph G maps the vertices of G to distinct points in the plane and the edges of G to Jordan arcs. For a vertex v of G and a drawing Γ of G , let $x_\Gamma(v)$ and $y_\Gamma(v)$ denote the x- and y-coordinates of v in Γ , respectively; when Γ is clear from the context, we may omit it and simply use the notation $x(v)$ and $y(v)$. A *face* of Γ is a region of the plane delimited by maximal uncrossed arc portions of the edges of G . The unique unbounded face of Γ is its *outer face*, the other faces are its *internal faces*. An *outer edge* is one incident to the outer face; all other edges are *inner edges*. The *rotation* of a vertex v in Γ is the counterclockwise cyclic order of the edges incident to v . The *rotation system* of Γ is the set of rotations of its vertices.

The drawing Γ is *planar* if no two of its edges cross; it is *k-planar* if each edge is crossed at most k times. A graph is *(k-)planar* if it admits a $(k-)$ planar drawing; it is *outer (k-)planar* if it admits a $(k-)$ planar drawing where all vertices are incident to the outer face.

A *planar embedding* \mathcal{E} of a planar graph G is an equivalence class of planar drawings of G , namely those that have the same set of faces. Each face can be described as a sequence of edges and vertices of G which bound the corresponding region in the plane; each such sequence is a face of G in the embedding \mathcal{E} . A planar embedding \mathcal{E} of a connected graph can also be described by specifying the rotation system and the outer face associated with any drawing of \mathcal{E} .

Let Γ be a non-planar drawing of a graph G ; the *planarization* of Γ is the planar drawing Γ' of the *planarized graph* G' obtained by replacing each crossing of Γ with a dummy vertex. If Γ is 1-planar, the planarization can be obtained as follows. Let uv and wz be any two edges that cross in Γ ; they are replaced in Γ' by the edges ux , xv , wx and xz , where x is the dummy vertex. Two non-planar drawings of a graph G have the same *embedding* if their

planarizations have the same planar embedding. An embedding \mathcal{E} of G can also be described by specifying the planarized graph G' and one of its planar embeddings. A planar graph with a given planar embedding is also called *plane graph*. An *outerplane graph* is a plane graph whose vertices are all incident to the outer face. A *fan* is a maximal outerpath that contains a vertex c that is adjacent to all other vertices; we call c the *central vertex* of the fan. A *2-tree* is a planar graph that can be reduced to an edge by iteratively removing a degree-two vertex that closes a 3-cycle. A *series-parallel graph* is a graph that can be augmented to a 2-tree by adding edges (and no vertices).

A *(simple, finite) directed graph* (*digraph* for short) G consists of a finite set $V(G)$ of *vertices* and a finite set $E(G) \subseteq \{(u, v) \mid u, v \in V(G), u \neq v\}$ of ordered pairs of vertices, which are called *edges*. A *source* (resp. *sink*) of G is a vertex with no incoming (resp. no outgoing) edges. A *single-source graph* is a digraph with a single source and, possibly, multiple sinks. A digraph G is an *st-graph* if: (i) it is acyclic and (ii) it has a single source s and a single sink t . An st-graph is a *planar st-graph* if it admits a planar embedding with s and t on the outer face. We say that a drawing of a digraph G is *upward* if every (directed) edge (u, v) of G is mapped to a y -monotone Jordan arc with $y(u) < y(v)$. Clearly, a digraph admits an upward drawing only if it does not contain a directed cycle. Therefore, we assume for the rest of the paper that the input graph is a *DAG*, a directed acyclic graph. Such a graph has a *linear extension*, i.e., a vertex order v_1, \dots, v_n such that, for any directed edge (v_i, v_j) , we have $i < j$. We say that a DAG is planar, outerplanar, or series-parallel if its underlying undirected graph is planar, outerplanar, or series-parallel, respectively.

Let Γ be an upward drawing of a DAG G . By the upwardness, the rotation system of Γ is such that for every vertex v of Γ the rotation of v has only one maximal subsequence of outgoing (incoming) edges. We call such a rotation system a *bimodal rotation system*. An *upward embedding* of a DAG G is an embedding of G with a bimodal rotation system. The minimum k such that a digraph G admits an upward k -planar drawing is called its *upward local crossing number* and denoted by $\text{lcr}^\uparrow(G)$.

For any positive integer k , we use $[k]$ as shorthand for $\{1, 2, \dots, k\}$. A *path-decomposition* of a graph $G = (V, E)$ is a sequence $P = \langle X_1, \dots, X_\ell \rangle$ of subsets of V , called *buckets*, such that (1) for each edge $e \in E$ there is a bucket that contains both end vertices of e , and (2) the set of buckets that contain a particular vertex $v \in V$ forms a contiguous subsequence of P . The *width* of a path-decomposition is one less than the size of the largest bucket. The *pathwidth* of the graph G is the width of a path decomposition of G with the smallest width.

3 Lower Bounds

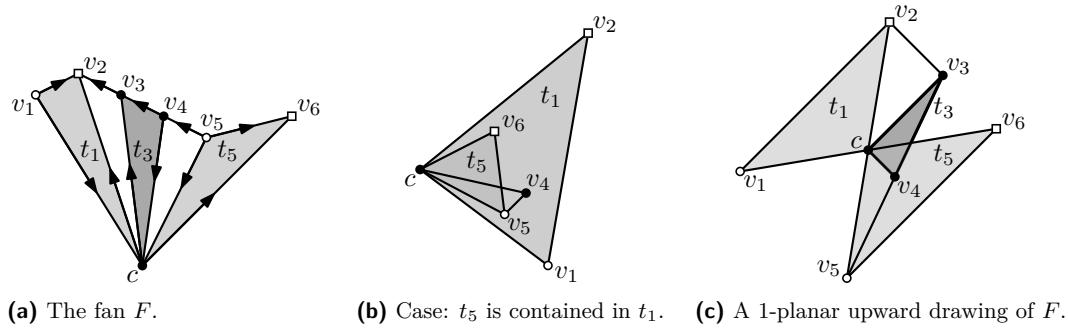
We start with a negative result that shows that even very special directed acyclic outerpaths may not admit upward-planar drawings, thus strengthening Papakostas' observation [35].

► **Proposition 1.** *Not every directed acyclic fan is upward-planar.*

Proof. Consider the 7-vertex fan F depicted in Figure 2a. Suppose for a contradiction that F is upward planar, that is, F admits an upward planar drawing Γ . Let c be the central vertex of F . We assume that c is placed at the origin. We say that a triangle of F is *positive* (*negative*, respectively) if the corresponding region of the plane in Γ contains the point $(\varepsilon, 0)$ ($(-\varepsilon, 0)$, respectively) for a sufficiently small value $\varepsilon > 0$. The triangles that have one vertex below c and one vertex above c (namely $t_1 = \Delta cv_1v_2$, $t_3 = \Delta cv_3v_4$, and $t_5 = \Delta cv_5v_6$) are either positive or negative.

If both t_1 and t_5 are positive, then one must contain the other in Γ , say, t_1 contains t_5 ; see Figure 2b. But then vertices v_3 and v_4 must also lie inside t_1 . If both lie inside t_5 , then the edge (v_3, v_2) intersects an edge of t_5 . So both must lie outside t_5 . But v_4 lies on one hand above v_5 and on the other hand below c and, thus, below v_6 . So the edge (v_4, c) intersects the edge (v_5, v_6) . (If t_5 is contained in t_1 , the edge (c, v_3) intersects the edge (v_1, v_2) .)

By symmetry, not both t_1 and t_5 can be negative, so exactly one of t_1 and t_5 must be negative, say, t_1 ; see Figure 2c. Now first assume that t_3 is positive. Due to edge (v_3, v_2) , vertex v_3 must be outside t_5 , so t_3 cannot be inside t_5 . On the other hand, t_3 cannot contain t_5 because v_4 is above v_5 . Hence t_3 intersects t_5 . Finally, assume that t_3 is negative. Due to edge (v_5, v_4) , vertex v_4 must be outside t_1 , so t_3 cannot be inside t_1 . On the other hand, t_3 cannot contain t_1 because v_3 is below v_2 . Hence t_3 intersects t_1 . ◀

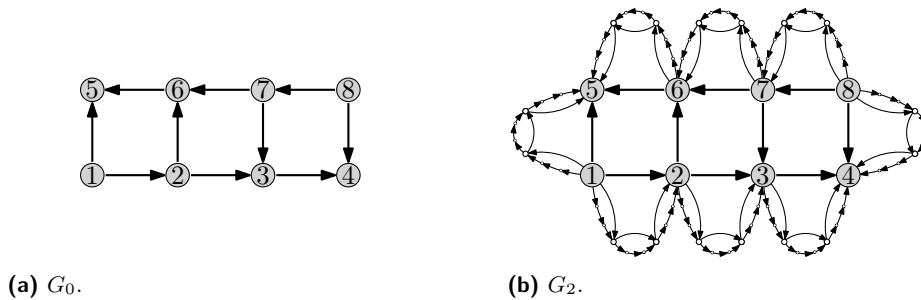


■ **Figure 2** A directed acyclic fan F that does not admit a planar upward drawing.

By iteratively adding paths on every outer edge of an outerplanar but not upward-planar DAG, we can construct outerplanar DAGs with an unbounded upward local crossing number.

► **Theorem 2.** For each $\ell \geq 0$, there is a bipartite outerplanar DAG G_ℓ with $n_\ell = 8 \cdot 3^\ell$ vertices, maximum degree $\Delta_\ell = 2\ell + 3$, and upward local crossing number greater than $\ell/6$, which is in $\Omega(\log n_\ell)$ and $\Omega(\Delta_\ell)$.

Proof. The bipartite graph G_0 in Figure 3a is not upward planar [35]. For $\ell \geq 1$, we construct G_ℓ from $G_{\ell-1}$ by adding a 3-edge path on every outer edge of the graph. Figure 3b shows G_2 . The maximum degree of G_ℓ is $\Delta_\ell = 2\ell + 3$. The number of vertices is $n_\ell = 8 + \sum_{i=1}^{\ell} 8 \cdot 3^{i-1} \cdot 2 = 8 \cdot 3^\ell$.



■ **Figure 3** There is a family $(G_\ell)_{\ell \geq 0}$ of bipartite outerplanar graphs such that G_ℓ has n_ℓ vertices, maximum degree Δ_ℓ , and upward local crossing number in $\Omega(\Delta_\ell) \cap \Omega(\log n_\ell)$.

Consider now an upward k -planar drawing Γ of G_ℓ for some k . Since G_0 is not upward planar, there must be a pair of independent edges of G_0 that crosses an odd number of times in Γ . Observe that G_0 has no upward planar drawing in which only the two inner edges

cross an odd number of times, for otherwise the two cycles $\langle 1, 2, 6, 5 \rangle$ and $\langle 3, 4, 8, 7 \rangle$ would intersect an odd number of times, which is impossible. Thus, in Γ there must be an outer edge e of G_0 that is crossed by an independent edge e' of G_0 an odd number of times. We choose e' to be an outer edge of G_0 , if possible.

We now determine a cycle C of G_ℓ that is crossed by e an odd number of times and does also not contain any end vertex of e . If e' is an inner edge, then we take the outer path P of G_0 that connects the ends of e' and does not contain e ; this is not intersected by e due to our choice of e' . Let C be the concatenation of P and e' . In this case C has length at most six.

If e' is an outer edge, we do the following: We start with the path P of length three that was added for e' . If P contains an edge that is crossed an odd number of times by e then we replace e' by such an edge and continue. More precisely, let $e_1 = e'$ and initialize $i = 1$. Let P_1 be the path of length three that was added for e_1 . While P_i contains an edge that is crossed an odd number of times by e , let e_{i+1} be such an edge, let P_{i+1} be the path of length three that was added for e_i , and increase i by one. Since e is crossed at most k times, this process stops at some $i < k$. Let C be the cycle that is composed of P_i and e_i . In this case C has length four.

Cycle C might cross itself. However, it divides the plane into cells. Since e crosses C an odd number of times, it follows that the end vertices of e must be in different cells of the plane. This means that not only e but also the ℓ edge-disjoint paths that were added on top of e have to cross C . But C contains at most six edges, each of which can be crossed at most k times. This is impossible if $\ell \geq 6k$. Hence, if there is an upward k -planar drawing then $\ell < 6k$, which means that $k > \ell/6$. ◀

We now show that if we expand the graph class beyond outerplanar graphs, then we get a lower bound on the upward local crossing number that is even linear in the number of vertices. The graphs in our construction have pathwidth 2, as opposed to the graphs in Theorem 2 whose pathwidth is logarithmic. Observe that a *caterpillar*, i.e., a tree that can be reduced to a path by removing all degree-1 vertices, has pathwidth 1, and that the pathwidth can increase by at most 1 if we add a vertex with some incident edges or subdivide some edges.

► **Theorem 3.** *For every $k \geq 1$, there exists a DAG with $\Theta(k)$ vertices, maximum degree in $\Theta(k)$, and pathwidth 2 that does not admit an upward k -planar drawing.*

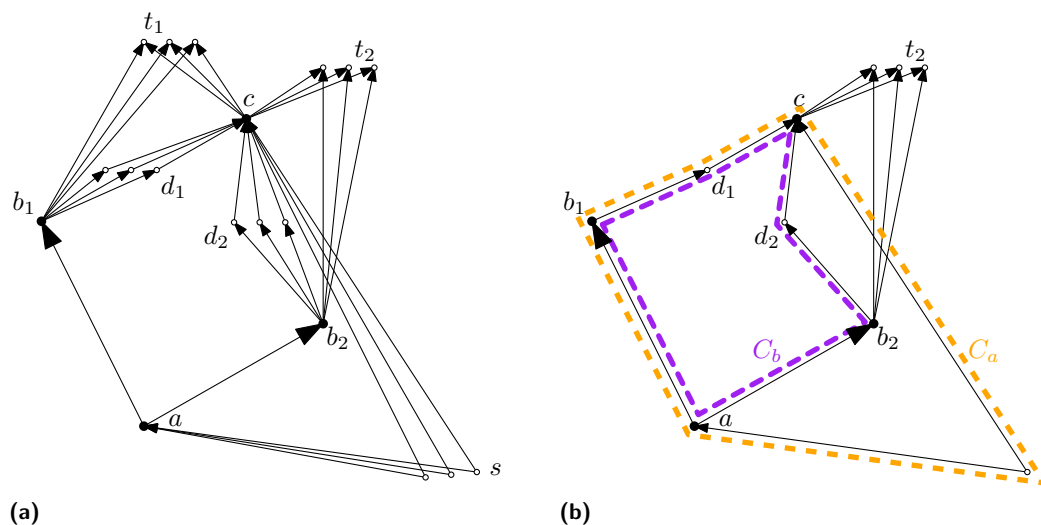
Proof. Let G_k be the graph consisting of the four vertices a , b_1 , b_2 , and c and the following set of edges and degree-2 vertices (see also Figure 4):

- edges (a, b_1) and (a, b_2) ;
- for $i \in [2]$ and $j \in [3k + 1]$, a *through-vertex at b_i* , i.e., a vertex $d_i^{(j)}$ and edges $(b_i, d_i^{(j)})$ and $(d_i^{(j)}, c)$;
- for $j \in [6k + 1]$, a *source below a* , i.e., a vertex $s^{(j)}$ and edges $(s^{(j)}, a)$ and $(s^{(j)}, c)$;
- for $i \in [2]$ and $j \in [4k + 1]$, a *sink above b_i* , i.e., a vertex $t_i^{(j)}$ and edges $(b_i, t_i^{(j)})$ and $(c, t_i^{(j)})$.

Clearly, G_k has $\mathcal{O}(k)$ vertices, and pathwidth 2, since $G - c$ is a caterpillar and has pathwidth 1.

Assume that there was an upward k -planar drawing Γ of G_k . Up to renaming, we may assume that $y(b_2) \leq y(b_1)$. Delete all but one of the through-vertices at b_1 from the drawing; in what follows, we write d_1 for the one that we keep (it does not matter which one).

Among the $3k + 1$ through-vertices $d_2^{(j)}$ at b_2 , there exists at least one for which the path $\langle b_2, d_2^{(j)}, c \rangle$ crosses none of the three edges in the path $\langle a, b_1, d_1, c \rangle$, for otherwise there would be an edge with more than k crossings. Delete all other through-vertices at b_2 ; in what



■ **Figure 4** A graph of pathwidth 2 (drawn upward) that does not have an upward k -planar drawing. (a) We only show three of the $\Theta(k)$ vertices of each group. (b) Cycles C_a and C_b .

follows we write d_2 for the one that we keep. Let a' be the topmost intersection point of (a, b_1) and (a, b_2) (possibly $a' = a$). Since $y(a) \leq y(a') < y(b_2) \leq y(b_1)$ the curve C_b formed by the two directed paths $\langle a', b_i, d_i, c \rangle$ (for $i \in [2]$) is drawn without crossing in Γ .

Curve C_b uses six edges, therefore among the $6k + 1$ sources below a , there exists one, call it s , for which edge (s, c) crosses no edge of C_b . Since $y(s) < y(a)$, vertex s is outside C_b , and so the entire edge (s, c) is outside C_b , except at the endpoint c . In particular, among the three edges (d_1, c) , (d_2, c) , and (s, c) that are incoming at c , edge (s, c) is either leftmost or rightmost (but cannot be the middle one). We assume here that (s, c) is rightmost, the other case is symmetric. Write $\{p, q\} = \{1, 2\}$ such that the left-to-right order of incoming edges at c is (d_p, c) , (d_q, c) , (s, c) . In Figure 4, we have $p = 1$ and $q = 2$.

Edge (s, a) is also outside C_b , except perhaps at endpoint a , since it uses smaller y -coordinates. Let s' be the topmost intersection point of (s, a) and (s, c) . Then there are no crossings in the curve C_a formed by the directed paths $\langle s', a, b_p, d_p, c \rangle$ and $\langle s', c \rangle$. By our choice of p and q , vertex d_q is *inside* C_a , and so is the entire path $\langle a', b_q, d_q, c \rangle$, except at the ends since it is part of C_b . In particular, b_q is inside C_a , whereas, for $j \in [4k + 1]$, $t_q^{(j)}$ is outside C_a due to $y(c) < y(t_q^{(j)})$. It follows that one of the four edges (a, b_p) , (b_p, d_p) , (d_p, c) and (s, c) must be crossed at least $k + 1$ times by edges from b_q to the sinks above it. Thus, the drawing was not k -planar, a contradiction. ◀

The graphs that we constructed in the proof of Theorem 3 are not bipartite, but one can make them bipartite by subdividing all edges once. This at best cuts the local crossing number in half, increases the pathwidth by at most 1, and yields the following result.

► **Corollary 4.** *There is a family of bipartite DAGs of constant pathwidth whose upward local crossing number is linear in the number of vertices.*

So far we needed graphs of unbounded maximum degree in order to enforce unbounded upward local crossing number. We now show that, intrinsically, this is not necessary.

► **Proposition 5.** *There are cubic DAGs whose upward local crossing number is at least linear in the number of vertices.*

Proof. The crossing number of a random cubic graph with n vertices is expected to be at least cn^2 for some absolute constant $c > 0$ [20], and thus there exist graphs yielding this bound. By the pigeon-hole principle, such a graph contains an edge with $\Omega(n)$ crossings among its $\Theta(n)$ edges. Impose arbitrary acyclic edge directions. ◀

4 Upper Bounds

The *bandwidth* $\text{bw}(G)$ of an undirected graph G is the smallest positive integer k such that there is a labeling of the vertices by distinct numbers $1, \dots, n$ for which the labels of every pair of adjacent vertices differ by at most k .

► **Theorem 6.** *The upward local crossing number of a DAG G with maximum degree Δ is at most $\Delta \cdot (2 \text{bw}(G) - 2) \leq 4 \text{bw}(G)(\text{bw}(G) - 1)$, so it is in $\mathcal{O}(\Delta \cdot \text{bw}(G)) \subseteq \mathcal{O}(\text{bw}(G)^2)$.*

Proof. Observe that the maximum degree Δ of a graph G is bounded in terms of the bandwidth of G ; namely, $\Delta \leq 2 \text{bw}(G)$. Consider a linear extension of G . For every vertex v of G , let $y(v)$ be its index in the extension. Now consider a labeling of G corresponding to the bandwidth. For every vertex v of G , let $x(v)$ be label. Construct a drawing of G by first placing every vertex v at the point $(x(v), y(v))$ and by then perturbing vertices slightly so that the points are in general position. Adjacent vertices are connected via straight-line segments.

It is easy to see that the drawing is upward since it is consistent with the linear extension. Consider an arbitrary edge (u, v) with $x(u) < x(v)$. Every edge that crosses (u, v) must have its left endpoint in the interval $[x(u) - \text{bw}(G) + 1, x(v) - 1]$. Since $x(v) - x(u) < \text{bw}(G)$, there are at most $2 \text{bw}(G) - 2$ such vertices distinct from u , each of which is incident to at most Δ edges. Hence, (u, v) has at most $\Delta \cdot (2 \text{bw}(G) - 2)$ crossings. ◀

For some graphs, a sublinear bound on the bandwidth is known, see [13, 21, 38]. This gives upper bounds on the local crossing number of many graph classes (e.g., interval graphs, co-comparability graphs, AT-free graphs, graphs of bounded treewidth); we list only a few:

- **Corollary 7.** *The following classes of DAGs have sublinear upward local crossing number:*
- *Square $k \times k$ grids have bandwidth $\Theta(k)$ and $\Delta = 4$, hence their upward local crossing number is in $\mathcal{O}(k) = \mathcal{O}(\sqrt{n})$.*
 - *Directed planar graphs with maximum degree Δ have bandwidth $\mathcal{O}(\frac{n}{\log_{\Delta} n})$ [13], hence their upward local crossing number is in $\mathcal{O}(\frac{n \cdot \Delta}{\log_{\Delta} n})$.*

We complement the negative result in Proposition 1 by showing that every directed acyclic outerpath allows an upward 2-planar drawing. We start with a technical lemma on fans.

► **Lemma 8.** *Let c be the central vertex of a directed acyclic fan G , and let $P = \langle v_1, \dots, v_{n-1} \rangle$ be the path of the remaining vertices in G . Let P_1, \dots, P_k be an ordered partition of P into maximal subpaths such that, for every $i \in [k]$, the edges between P_i and c either are all directed towards c or are all directed away from c . Then there is an upward 2-planar drawing of G with the following properties:*

1. *no edge incident to c is crossed;*
2. *vertex v_1 has x -coordinate 1, the central vertex c and v_{n-1} have x -coordinate $n - 1$, and the x -coordinates of v_2, \dots, v_{n-2} are distinct values within $\{2, \dots, n - 2\}$;*
3. *for all edges all x -coordinates of the curves are at most $n - 1$; all edges incident to c and all edges of the subpaths P_1, \dots, P_k are in the vertical strip between 1 and $n - 1$;*
4. *if P_1 is a directed path, then the edge between P_1 and P_2 is crossed at most once.*

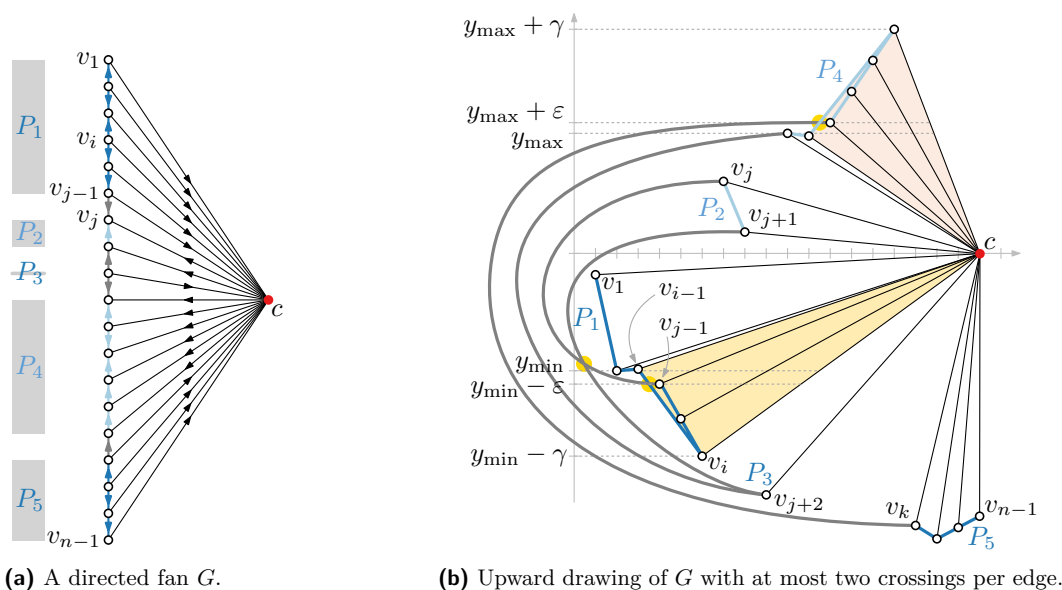


Figure 5 Upward 2-planar drawings of fans (Lemma 8). For $t \in [k]$, we add the path P_t below c (blue paths) or above c (green paths), going up and down as prescribed by the edge directions and such that no edge incident to c is crossed. We maintain the property that all vertices of P_t are on the outer face of the subgraph induced by P_t and c , except for possibly a last final part pointing upward if P_t is below c or pointing downward if P_t is above c . See the shaded areas, e.g., the final part $\langle v_i, \dots, v_{j-1} \rangle$ of P_1 . The edge connecting P_t and P_{t+1} (red edges) might either cross the last edge of P_t on the outer face (e.g., the edge of P_1 between v_{i-1} and v_i) or the edge connecting P_{t-1} to P_t in order to reach the outer face. The latter may have been crossed once before (as (v_{j-1}, v_j)).

Proof. We place c at $(n - 1, 0)$; then we place v_1, v_2, \dots, v_{n-1} above or below c depending on the direction of the edges that connect them to c ; see Figure 5 for an example.

For $i \in [n - 2]$, we keep the invariant that, when we place v_i , the leftward ray from v_i reaches the outer face of the current drawing after crossing at most one other edge, and that this edge had been crossed at most once.

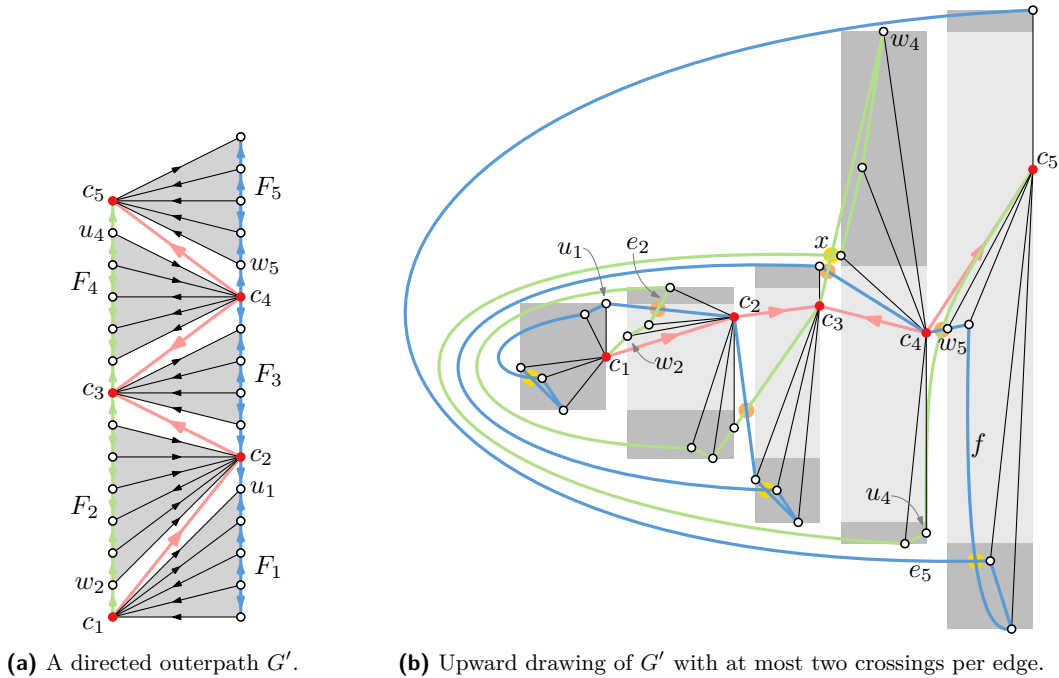
In order to choose appropriate y-coordinates, we maintain two values y_{\min} and y_{\max} indicating the minimum and maximum y-coordinate of any so far drawn vertex. Consider now a subpath $P' \in \{P_1, \dots, P_k\}$. Let v_h be the first and let v_{j-1} be the last vertex of P' , i.e., $P' = \langle v_h, v_{h+1}, \dots, v_{j-1} \rangle$. We describe in detail the case that the edge from v_h, \dots, v_{j-1} to c are directed towards c that is, v_h must lie below c . The other case is symmetric. We place v_h at x-coordinate h and with a y-coordinate sufficiently below y_{\min} . If $h = j - 1$ we are done.

We now consider the cases $j = n$ or $(v_{j-1}, v_{j-2}) \in E$. In that case, we place v_{h+1}, \dots, v_{j-1} using x-coordinates $h + 1, \dots, j - 1$, going up and down as needed but remaining below the x-axis. The edges are drawn such that all vertices of P' remain on the outer face of the drawing. I.e., if we use straight-line edges, then, for $i \in [n - 2]$, the slope of $v_i v_{i+1}$ must be less than the slope of $v_i c$. Since we go towards c , we can draw P' and the edges that connect v_1, v_2, \dots, v_{n-1} to c without any crossings.

If $j \neq n$ and $(v_{j-2}, v_{j-1}) \in E$, then let $i \in \{h, \dots, j - 1\}$ be the smallest index such that the subpath $\langle v_i, v_{i+1}, \dots, v_{j-1} \rangle$ is directed. In that case, we place v_{h+1}, \dots, v_{i-1} at x-coordinates $h + 1, \dots, i - 1$, going slightly up and down as in the case described above. Let y_{\min} be the smallest among the y-coordinates of all points placed so far.

Then we place $v_i, v_{i+1}, \dots, v_{j-1}$ in reverse order, i.e., at x-coordinates $j-1, j-2, \dots, i$. Set $y(v_i) = y_{\min} - \gamma$ and $y(v_{j-1}) = y_{\min} - \varepsilon$ for some (large) $\gamma > 0$ and (small) $\varepsilon > 0$ such that v_{j-1} lies inside the triangle $\Delta v_{i-1}v_i c$ (pale yellow in Figure 5b) if $i > h$ and within the triangle $\Delta ov_i c$, with $o = (0, 0)$ otherwise. (Observe that in the case $i = h$, we already required that v_i is sufficiently below y_{\min} ; this is now further specified here.) Draw v_{i+1}, \dots, v_{j-2} on the segment $\overline{v_i v_{j-1}}$. Now, if $i > h$ then the vertex v_{j-1} can reach the outer face via the edge (v_i, v_{i-1}) which was not crossed so far. If $i = h$ then v_{j-1} is on the outer face if $P' = P_1$, otherwise it can reach the current outer face by crossing the edge (v_h, v_{h-1}) . This edge might have crossed one edge when it was initially drawn but so far no other edge.

Observe that when we draw the next maximal subpath, we place v_j at $(j, y_{\max} + 1)$, i.e., in particular in the outer face of the current drawing. The edge from v_{j-1} to v_j must be directed towards v_j since the orientation is acyclic. Thus, we can draw the edge between v_{j-1} and v_j upward with at most one crossing, causing at most a second crossing on (v_h, v_{h-1}) . ◀



■ **Figure 6** Example in- and output of our drawing algorithm (edge crossings due to Lemma 8 are highlighted in yellow; other edge crossings are highlighted in orange).

Now we describe our construction for general outerpaths; see Figure 6.

► **Theorem 9.** *Every directed acyclic outerpath admits an upward 2-planar drawing.*

Proof. Without loss of generality, we can assume that the given outerpath is maximal: if the outerpath has interior faces that are not triangles, we temporarily triangulate them using additional edges, which we direct such that they do not induce directed cycles and which we remove after drawing the maximal outerpath.

Let G' be such a graph; see Figure 6a. Let c_1, c_2, \dots, c_k be the vertices of degree at least 4 in G' (marked red in Figure 6). These vertices form a path (light red in Figure 6); let them be numbered along this path, which we call the *backbone* of G' . We assign every vertex v that does not lie on the backbone to a neighboring backbone vertex; if v is incident to an

inner edge, we assign v to the other endpoint of that edge. Otherwise v has degree 2 and is incident to a unique backbone vertex via an outer edge, and we assign v to this backbone vertex. For $i \in [k]$, backbone vertex c_i induces, together with the vertices assigned to it, a fan F_i .

We draw the backbone in an x-monotone fashion. We start by drawing F_1 with the algorithm for drawing a fan as detailed in the proof of Lemma 8; see the leftmost gray box in Figure 6b. Then, for $i \in \{2, \dots, k\}$, we set $x(c_i)$ to $x(c_{i-1})$ plus the number of inner edges incident to c_i and we set $y(c_i)$ depending (i) on the y-coordinates of the two neighbors of c_i that have already been drawn (c_{i-1} and the common neighbor u_{i-1} of c_{i-1} and c_i in F_{i-1}) and (ii) on the directions of the edges that connect these vertices to c_i ; see, for example, the placement of c_5 in Figure 6b. Then we draw F_i with respect to the position of c_i , again using the algorithm from the proof of Lemma 8 with the following modifications. In general, vertices in F_i that are adjacent to c_i via an edge directed towards c_i (resp. from c_i) are placed below (resp. above) all vertices in the drawings of F_1, \dots, F_i ; see the dark gray boxes below (resp. above) c_2, \dots, c_5 in Figure 6b. If an edge of F_i connects two neighbors of c_i one of which lies above c_i and one of which lies below c_i , then we route this edge to the left of all drawings of F_1, \dots, F_{i-1} .

An exception to this rule occurs if c_i and the common neighbor w_i of c_{i-1} and c_i in F_i must be both above or both below c_{i-1} due to the directions of the corresponding edges. Let u_{i-1} be the common neighbor of c_{i-1} and c_i in F_{i-1} . We assume, without loss of generality, that c_i is above c_{i-1} . Let P_1 and P_2 be the first and second maximal subpath from Lemma 8 applied to F_i , and let e_i be the edge connecting P_1 and P_2 . We distinguish two subcases.

If P_1 is a directed path leaving w_i , then we draw P_1 above the edge $c_{i-1}c_i$ and we draw the edge e_i straight, without going around all drawings of F_1, \dots, F_{i-1} . In this case e_i is directed from P_1 to P_2 . Hence, e_i crosses the edge $u_{i-1}c_i$ if $u_{i-1}c_i$ is directed from c_i to u_{i-1} ; see the situation for c_2 in Figure 6b. Note that e_i may receive a second crossing when we draw the remainder of F_i in the usual way.

Otherwise, that is, if P_1 contains an edge directed towards the left endpoint w_i of P_1 , let f be the first such edge. We then place the part of P_1 up to the first endpoint of f below the edge $c_{i-1}c_i$; see w_5 and f in Figure 6b. If the edge $u_{i-1}c_i$ is directed towards c_i , we draw it between w_i and the edge $c_{i-1}c_i$. Then it crosses the edge $c_{i-1}w_i$ but no other edge. We place the second endpoint of f below all vertices in $V(F_1) \cup \dots \cup V(F_{i-1})$ and continue with the remainder of F_i as usual.

In any case, if $1 < i < k$, then the last vertex u_{i-1} of F_{i-1} is connected to c_i and c_{i-1} is connected to the first vertex w_i in F_i . These two edges may cross each other; see the crossings highlighted in orange in Figure 6b. If the edge $c_{i-1}w_i$ goes, say, up but the following outer edges go down until a vertex v_k below c_i is reached, then the edge $c_{i-1}w_i$ may be crossed a second time by the edge $v_{k-1}v_k$; see the crossing labeled x on the edge c_3w_4 in Figure 6b. But due to property 4 of Lemma 8, edge $v_{k-1}v_k$ had been crossed at most once within its fan. Also $c_{i-1}w_i$ cannot have a third crossing. Thus, all edges are crossed at most twice. ◀

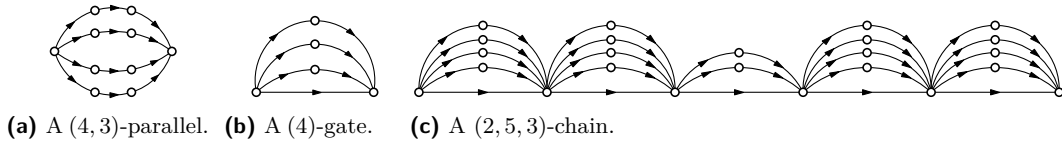
One can argue that every maximal pathwidth-2 graph can be generated from a maximal outerpath by connecting some pairs of adjacent vertices using an arbitrary number of (new) paths of length 2. In spite of the simplicity of this operation, we cannot hope to generalize the above result to pathwidth-2 graphs; see the linear lower bound on the upward local crossing number for such graphs stated in Theorem 3.

5 Complexity of Testing

Here we prove that upward 1-planarity testing is NP-complete even for structurally simple DAGs, both when a bimodal rotation system is fixed and when it is not fixed. We also show that testing upward outer-1-planarity for single-source DAGs can be solved in linear time.

5.1 Testing Upward 1-Planarity

We first define a few gadgets; all of them are planar st-graphs. For positive integers b and q , let a (b, q) -parallel be the parallel composition of b oriented paths each consisting of q edges; see Figure 7a. For a positive integer p , let a (p) -gate be the parallel composition of an oriented edge and a $(p - 1, 2)$ -parallel; see Figure 7b. For positive integers h , q , and a , let an (h, q, a) -chain consist of a series of h (q) -gates, followed by exactly one (a) -gate, followed again by h (q) -gates; see Figure 7c.



■ **Figure 7** Illustrations for the gadgets used in the construction of G_A and of G_B .

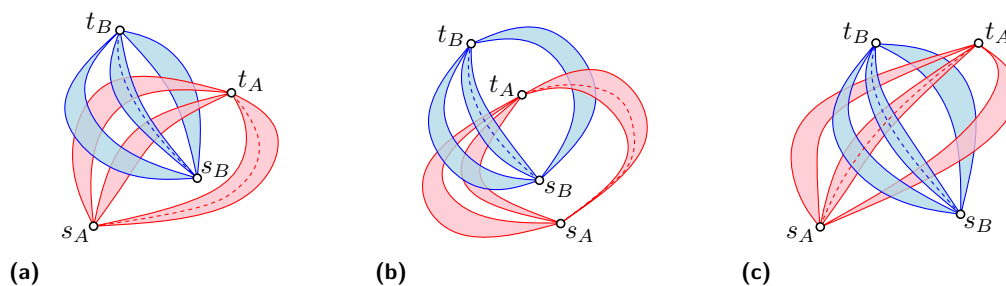
An instance of 3-PARTITION is a multiset $I = \{a_1, a_2, a_3, \dots, a_k\}$ of positive integers such that $b = k/3$ is an integer and $\sum_{i=1}^k a_i = W \cdot b$, with W integer. The 3-PARTITION problem asks if there exists a partition of the set I into b 3-element subsets such that the sum of the elements of each subset is exactly W . Since 3-PARTITION is strongly NP-hard [25], we may assume that W is bounded by a polynomial in b .

We associate with a given instance I of 3-PARTITION two planar st-graphs G_A and G_B defined as follows. Digraph G_A is the parallel composition of $(b - 1, W + 1, a_i)$ -chains, one for every $i \in \{1, \dots, k\}$. Digraph G_B is an (b, q) -parallel, with $q = W + (k - 3)(W + 1)$. Note that the underlying undirected graphs of both G_A and G_B are series-parallel.

Let G be any digraph that contains the two subgraphs G_A and G_B defined above. Let s_A and t_A (resp. s_B and t_B) be the two vertices of G that are the source and the sink of G_A (resp. G_B). Let Γ be a 1-planar drawing of G and let Γ_{AB} be the 1-planar drawing obtained by restricting Γ to the nodes and edges of G_A and G_B . We say that G_A and G_B *cross* in Γ if in Γ_{AB} every $s_A t_A$ -path (i.e., a path directed from s_A to t_A) crosses every $s_B t_B$ -path. See Figures 8a and 8b for examples of graph G_A and G_B that do not cross or cross in a drawing of Γ_{AB} , respectively.

► **Lemma 10.** *Let I be an instance of 3-PARTITION and let G_A and G_B be the two planar st-graphs associated with I . Let G be a digraph containing G_A and G_B as subgraphs such that G has an upward 1-planar drawing if and only if G_A crosses G_B . There exists an upward 1-planar drawing Γ of G if and only if I admits a solution.*

Proof sketch. We prove that if G admits an upward 1-planar drawing Γ , then Γ provides a solution of instance I of 3-PARTITION. By hypothesis, G_A and G_B cross in Γ ; see Figure 9. Observe that only one path among the b paths of G_B can traverse one (a_i) -gate G_A . Also, every path of G_B crosses all the $(W + 1, b - 1, a_i)$ -chains of G_A . In particular, every path of G_B must cross at least three (a_i) -gates since it has not enough edges to cross more than $k - 2$ $(W + 1)$ -gates. Also, if one path of G_B crossed more than three (a_i) -gates, then some other path of G_B that would cross at most two (a_i) -gates. Therefore, every path π of G_B



■ **Figure 8** Illustrations for the definition of crossing st-subgraphs. (a) and (b) Two planar st-graphs G_A and G_B that do not cross, as witnessed by the two non-crossing dashed paths. (c) Two planar st-graphs G_A and G_B that cross.

■ **Table 1** A comparison between results in the literature about the complexity of testing upward planarity and the results discussed in this paper about the complexity of testing upward 1-planarity.

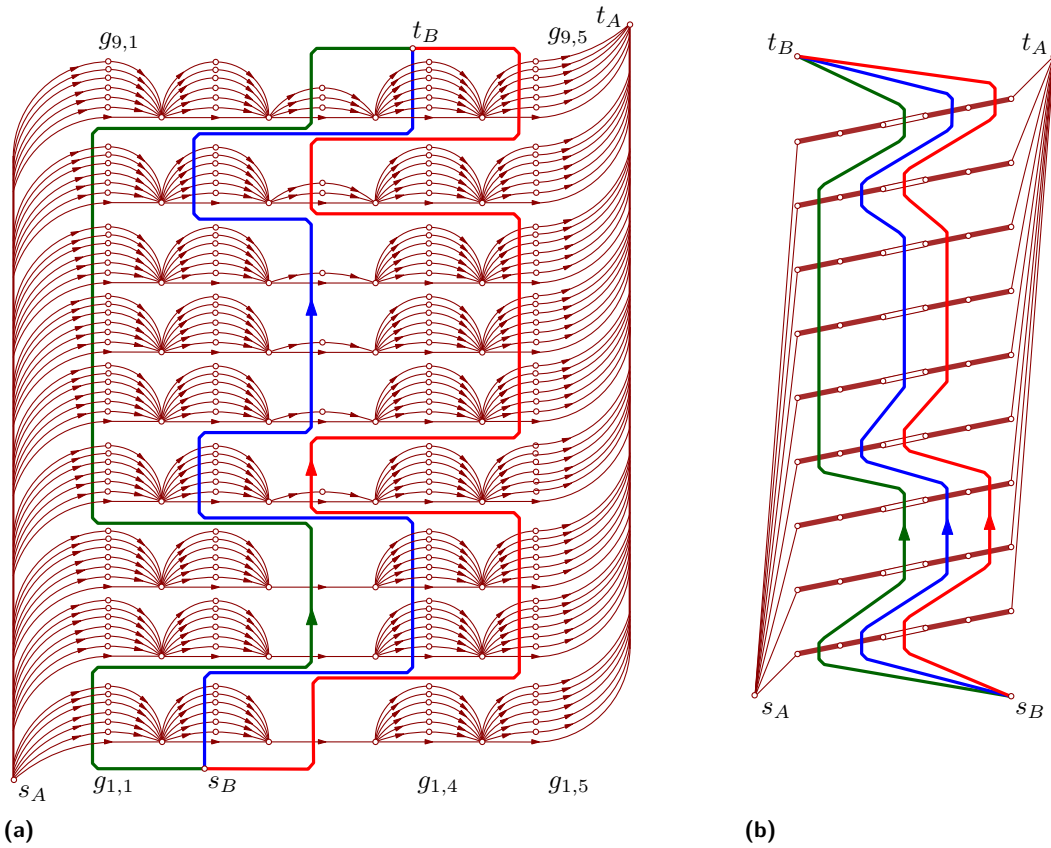
Underlying planar graph	Acyclic orientation	Upward planarity		Upward 1-planarity	
		fixed embedding	variable embedding	fixed rotation system	variable rotation system
Series-parallel	multi-source multi-sink	Polynomial [15, 18]		NP-complete Theorem 11 Case 1	NP-complete Theorem 11 Case 3
	single-source single-sink	Trivially polynomial			Trivially polynomial
General graph	multi-source multi-sink	Polynomial [9]	NP-complete [27]	NP-complete Corollary 12	NP-complete Theorem 11 Case 2
	single-source single-sink	Polynomial [14]			

must cross exactly three (a_i) -gates and $k - 3(W + 1)$ -gates in Γ . Note that the number of crossings of π with the three (a_i) -gates is exactly W . It follows that if G has an upward 1-planar drawing then the instance I of 3-PARTITION admits a solution. Conversely, if the instance I of 3-PARTITION admits a solution it is easy to construct an upward 1-planar drawing Γ_{AB} of G_A and G_B where G_A and G_B cross. ◀

► **Theorem 11.** *Testing upward 1-planarity is NP-complete even in the following restricted cases:*

1. *The bimodal rotation system is fixed, the DAG has exactly one source and exactly one sink, the underlying graph is series-parallel.*
2. *The bimodal rotation system is not fixed, the DAG has exactly one source and exactly one sink, the underlying planar graph is obtained by replacing the edges of a K_4 with series-parallel graphs.*
3. *The bimodal rotation system is not fixed, the underlying graph is series-parallel, there is one source and two sinks.*

Proof sketch. It is immediate to observe that upward 1-planarity testing is in NP, as one can guess an upward 1-planar embedding and test it in polynomial time. In order to show that the problem is NP-hard for the cases in the statement it suffices, by Lemma 10, to exhibit digraphs that contain G_A and G_B as subgraphs and that admit upward 1-planar



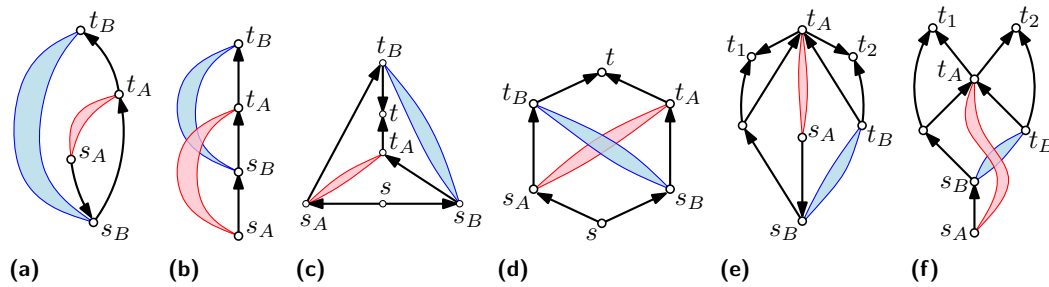
■ **Figure 9** (a) Digraph G_A (dark red) and a schematic representation of digraph G_B where each colored curve represents a directed path with $W + (k - 3)(W + 1)$ edges. The corresponding instance of 3-PARTITION is $I = \{1, 1, 1, 2, 2, 2, 2, 3, 4\}$, with $b = 3$ and $W = 6$. The 1-planar drawing corresponds to the solution $\{1, 1, 4\}$ (green path), $\{2, 2, 2\}$ (blue path), and $\{1, 2, 3\}$ (red path). The drawing in (a) is not upward but it can be made upward by stretching it vertically as shown in (b), where thick edges represent (q) -gates and the central white-filled edges represent (a) -gates.

drawings if and only if G_A and G_B cross in them. Let m_A and m_B be the number of edges of G_A and G_B , respectively. Let a *barrier* be a planar st-graph consisting of a $(d, 2)$ -parallel, where $d = m_A + m_B + 1$. Note that neither G_A nor G_B can cross a barrier in such a way that every edge is crossed at most once. The instances that we use for the cases listed in the statement are depicted in Figure 10a (Case 1), Figure 10c (Case 2) and Figure 10e (Case 3), where the thick edges represent barriers and G_A and G_B can be identified by their poles. As shown in Figure 10b, Figure 10d, and Figure 10f an upward 1-planar drawing of such graphs forces G_A and G_B to cross, hence, implies the hardness of computing such drawings. ◀

The following corollary is an immediate consequence of the argument used to prove the second case in the statement of Theorem 11.

► **Corollary 12.** *Testing upward 1-planarity is NP-complete for single source-single sink DAGs with a fixed bimodal rotation system, whose underlying planar graph is obtained by replacing the edges of a K_4 with series-parallel graphs.*

We conclude this section by remarking some differences between the complexity of upward planarity testing and upward 1-planarity testing. When the bimodal rotation system is fixed, upward planarity testing can be solved in polynomial time [9], whereas upward 1-planarity



■ **Figure 10** Some digraphs for the proof of Theorem 11. Thick black edges represent barriers.

testing is NP-hard (Theorem 11). Also, when the bimodal rotation system is not fixed and the digraph has a constant number of sources and sinks, differently from upward 1-planarity testing, upward planarity testing can again be solved in polynomial time [14]. On the other hand, any digraph whose bimodal rotation system is not fixed, whose underlying graph is series-parallel, and that has only one source and only one sink is always upward planar and thus upward 1-planar. Indeed, adding an edge between any two vertices of the undirected underlying series-parallel graph yields a planar graph (see, e.g., [17]). It follows that G can be turned into a planar st-graph by connecting its source to its sink by an edge and hence it is upward planar [26]. This discussion is summarized in Table 1.

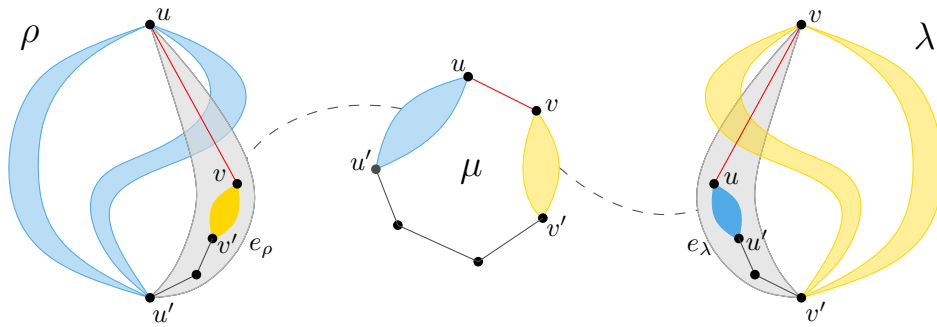
5.2 Testing Upward Outer-1-Planarity

To complement the results of Section 5.1, we consider a restricted setting that has often been studied in the “beyond planarity” literature to show the tractability of an otherwise NP-hard problem. Namely, we describe how to test whether a single-source DAG admits an *upward outer-1-planar drawing*, i.e., one that is both upward and outer-1-planar.

► **Theorem 13.** *For single-source DAGs, upward outer-1-planarity can be tested in linear time.*

This section provides the main ideas behind this result; recall that all details can be found in the full version [1]. In the following, let G be a single-source DAG. As a first step, we characterize the single-source DAGs that admit an upward outer-1-planar drawing as those that admit an outer-1-planar embedding whose planarization is acyclic. In particular, this implies that we may treat the biconnected components of G independently, and we therefore assume in the following that G is biconnected. We assume familiarity with the SPQR-tree [17]. Note that, in the version of the SPQR-tree that we use, there are no Q-nodes. Instead, skeletons contain both real and virtual edges.

Our testing algorithm builds on the results of Auer et al. [4, 5, 6] for testing outer-1-planarity. A necessary condition is that the skeleton of each R-node is a K_4 and the skeleton of each P-node contains at most four virtual edges plus, possibly, one real edge. In a nutshell, Auer et al. [5] show that there is a bijection between the outer-1-planar embeddings of a biconnected graph G and certain (non-planar) embeddings of all skeletons of the SPQR-tree \mathcal{T} of G . These non-planar embeddings need to satisfy *local conditions* which state that every virtual edge must have at least a part of it incident to the outer face, a virtual edge may only receive a crossing if it belongs to a P-node and corresponds to an S-node, and if a virtual edge receives a crossing, then the end of it that is not incident to the outer face, if any, must correspond to a real edge of the graph. As an example consider the embeddings of the skeletons of node ρ and λ in Figure 11. The shown embedding, where the segment of edge e_ρ

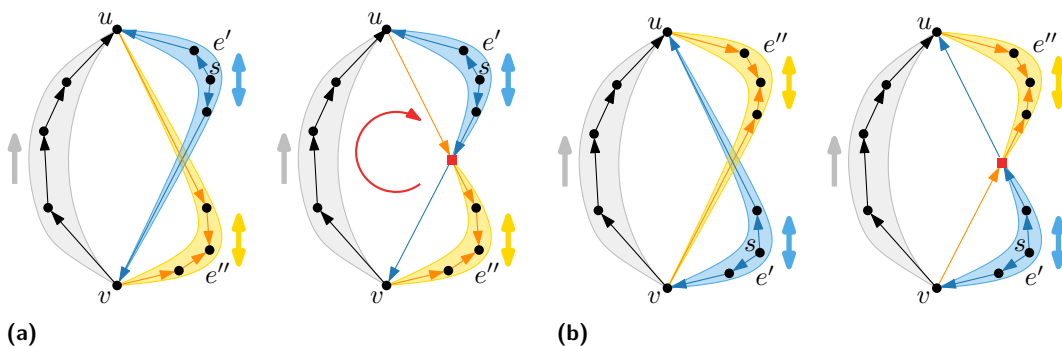


■ **Figure 11** Illustration for the necessary conditions that allow for an outer-1-planar embedding.

incident to u is not on the outer face, requires that the expansion graph of e_ρ starts with a real edge (here the edge uv) at u . We note that in this case, the edge uv is in fact a real edge in a neighboring S-node. Unlike the case of SPQR-trees and planar embeddings, the embeddings of different skeleton cannot be combined independently; instead, there is also a *global condition* that requires that no real edge receives crossings from two P-nodes. For example, the embeddings of $\text{skel}(\rho)$ and $\text{skel}(\lambda)$ shown in Figure 11 both imply a crossing on the edge uv and are therefore not compatible.

We want to restrict our attention to outer-1-planar embeddings whose planarization is acyclic. For this, we orient the edges in our skeletons, where real edges are endowed with their orientation in G and a virtual edge $\{u, v\}$ is directed as (u, v) if its expansion graph contains a directed path from u to v . Note that if the expansion graph of $\{u, v\}$ contains no directed paths between its poles, the virtual edge remains undirected. We then show that, in order to obtain an outer-1-planar embedding whose planarization is acyclic, we may only combine what we call *acyclic* embeddings of the skeletons, which do not already locally produce cycles; see Figure 12 for an example. Conversely, we prove that if we choose for each oriented skeleton an embedding that satisfies the local conditions and is acyclic, and moreover, these choices also satisfy the global condition, then they together define an outer-1-planar embedding of G whose planarization is acyclic, and hence admits an upward drawing.

The algorithm therefore works as follows. Since each skeleton admits at most 12 embeddings that satisfy the local conditions [5], we can enumerate them and test for each of them whether it is acyclic in total linear time. After this step, we have for each node μ of \mathcal{T} a set \mathcal{F}_μ



■ **Figure 12** Two embeddings of the skeleton of a P-node μ and the corresponding planarizations. The planarization in (a) contains a directed cycle, the one in (b) does not. Thick arrowed edges show the direction of the virtual edges; a double arrow indicates an undirected virtual edge.

of candidate embeddings. It remains to choose for each node μ one embedding $\mathcal{E}_\mu \in \mathcal{F}_\mu$ such that the global condition is satisfied, i.e., no two P-nodes put crossings on the same real edge. We say that such a choice of embeddings is *consistent*. To decide whether such a choice exists, we construct a conflict graph H whose vertices are the embeddings of the skeletons, each \mathcal{F}_μ forms a clique and two embeddings of different P-nodes are connected by an edge if and only if they put a crossing on the same real edge. Then a consistent choice corresponds to an independent set in H whose size is the number of nodes of \mathcal{T} . Since the size of H is linear in the size of G and we can show that the treewidth of H is at most 36, the existence of such an independent set can be tested in linear time [16]. We note that, if the test succeeds, we can also construct the upward outer-1-planar embedding of G in the same running time.

6 Conclusion

In this paper we initiated the study of upward k -planar drawings, that is, upward drawings of directed acyclic graphs such that every edge is crossed at most k times for a given constant k . We first gave upper and lower bounds for the upward local crossing number of various graph families, i.e., the minimum k such that every graph from the respective family admits an upward k -planar drawing. We strengthen these combinatorial results by proving that testing a DAG for upward k -planarity is NP-complete even for $k = 1$. On the positive side, testing upward outer-1-planarity for single source digraphs can be done in linear time. We conclude the paper by listing some open problems that may stimulate further research.

1. Is there a directed outerpath that does not admit an upward 1-planar drawing?
2. Consider the class \mathcal{O}_Δ of outerplanar graphs (or even 2-trees) of maximum degree Δ . Is there a function f such that every graph in \mathcal{O}_Δ admits an $f(\Delta)$ -planar upward drawing?
3. In light of the lower bounds in Section 3, it is natural to consider graphs with a special structure, in order to prove sublinear upper bounds on their (upward) local crossing number. For example, Wood and Telle [39, Corollary 8.3] show that every (undirected) graph of maximum degree Δ and treewidth τ admits a (straight-line) drawing in which every edge crosses $\mathcal{O}(\Delta^2\tau)$ other edges. Can the *upward* local crossing number be bounded similarly by a function in Δ and τ ?
4. Do planar graphs of maximum degree Δ have upward local crossing number $\mathcal{O}(f(\Delta)n^{1-\epsilon})$ for some function f and some constant $\epsilon > 0$?
5. Can upward outer-1-planarity be efficiently tested for multi-source and multi-sink DAGs?
6. Investigate parameterized approaches to testing upward 1-planarity.

References

- 1 Patrizio Angelini, Therese Biedl, Markus Chimani, Sabine Cornelsen, Giordano Da Lozzo, Seok-Hee Hong, Giuseppe Liotta, Maurizio Patrignani, Sergey Pupyrev, Ignaz Rutter, and Alexander Wolff. The price of upwardness. arXiv 2409.01475. URL: <http://arxiv.org/abs/2409.01475>.
- 2 Patrizio Angelini, Luca Cittadini, Giuseppe Di Battista, Walter Didimo, Fabrizio Frati, Michael Kaufmann, and Antonios Symvonis. On the perspectives opened by right angle crossing drawings. In David Eppstein and Emden R. Gansner, editors, *Proc. 17th Int. Symp. Graph Drawing (GD)*, volume 5849 of *LNCS*, pages 21–32. Springer, 2009. doi:10.1007/978-3-642-11805-0_5.
- 3 Patrizio Angelini, Luca Cittadini, Walter Didimo, Fabrizio Frati, Giuseppe Di Battista, Michael Kaufmann, and Antonios Symvonis. On the perspectives opened by right angle crossing drawings. *J. Graph Algorithms Appl.*, 15(1):53–78, 2011. doi:10.7155/JGAA.00217.

- 4 Christopher Auer, Christian Bachmaier, Franz J. Brandenburg, Andreas Gleißner, Kathrin Hanauer, Daniel Neuwirth, and Josef Reislhuber. Recognizing outer 1-planar graphs in linear time. In Stephen K. Wismath and Alexander Wolff, editors, *Proc. 21st Int. Symp. Graph Drawing (GD)*, volume 8242 of *LNCS*, pages 107–118. Springer, 2013. doi:10.1007/978-3-319-03841-4_10.
- 5 Christopher Auer, Christian Bachmaier, Franz J. Brandenburg, Andreas Gleißner, Kathrin Hanauer, Daniel Neuwirth, and Josef Reislhuber. Outer 1-planar graphs. *Algorithmica*, 74(4):1293–1320, 2016. doi:10.1007/S00453-015-0002-1.
- 6 Christopher Auer, Christian Bachmaier, Franz J. Brandenburg, Andreas Gleißner, Kathrin Hanauer, Daniel Neuwirth, and Josef Reislhuber. Correction to: Outer 1-planar graphs. *Algorithmica*, 83(11):3534–3535, 2021. doi:10.1007/S00453-021-00874-Z.
- 7 Michael A. Bekos, Giordano Da Lozzo, Fabrizio Frati, Martin Gronemann, Tamara Mchedlidze, and Chrysanthi N. Raftopoulou. Recognizing DAGs with page-number 2 is NP-complete. In Patrizio Angelini and Reinhard von Hanxleden, editors, *Proc. 30th Int. Symp. Graph Drawing & Netw. Vis. (GD)*, volume 13764 of *LNCS*, pages 361–370. Springer, 2022. doi:10.1007/978-3-031-22203-0_26.
- 8 Michael A. Bekos, Giordano Da Lozzo, Fabrizio Frati, Martin Gronemann, Tamara Mchedlidze, and Chrysanthi N. Raftopoulou. Recognizing DAGs with page-number 2 is NP-complete. *Theor. Comput. Sci.*, 946:113689, 2023. doi:10.1016/J.TCS.2023.113689.
- 9 Paola Bertolazzi, Giuseppe Di Battista, Giuseppe Liotta, and Carlo Mannino. Upward drawings of triconnected digraphs. *Algorithmica*, 12(6):476–497, 1994. doi:10.1007/BF01188716.
- 10 Sujoy Bhore, Giordano Da Lozzo, Fabrizio Montecchiani, and Martin Nöllenburg. On the upward book thickness problem: Combinatorial and complexity results. In Helen C. Purchase and Ignaz Rutter, editors, *Proc. 29th Int. Symp. Graph Drawing & Netw. Vis. (GD)*, volume 12868 of *LNCS*, pages 242–256. Springer, 2021. doi:10.1007/978-3-030-92931-2_18.
- 11 Sujoy Bhore, Giordano Da Lozzo, Fabrizio Montecchiani, and Martin Nöllenburg. On the upward book thickness problem: Combinatorial and complexity results. *Eur. J. Comb.*, 110:103662, 2023. doi:10.1016/J.EJC.2022.103662.
- 12 Carla Binucci, Giordano Da Lozzo, Emilio Di Giacomo, Walter Didimo, Tamara Mchedlidze, and Maurizio Patrignani. Upward book embeddability of st-graphs: Complexity and algorithms. *Algorithmica*, 85(12):3521–3571, 2023. doi:10.1007/S00453-023-01142-Y.
- 13 Julia Böttcher, Klaas P. Pruessmann, Anusch Taraz, and Andreas Würfl. Bandwidth, expansion, treewidth, separators and universality for bounded-degree graphs. *Europ. J. Combin.*, 31(5):1217–1227, 2010. doi:10.1016/J.EJC.2009.10.010.
- 14 Steven Chaplick, Emilio Di Giacomo, Fabrizio Frati, Robert Ganian, Chrysanthi N. Raftopoulou, and Kirill Simonov. Parameterized algorithms for upward planarity. In Xavier Goaoc and Michael Kerber, editors, *Proc. 38th Int. Symp. Comput. Geom. (SoCG)*, volume 224 of *LIPICs*, pages 26:1–26:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICs.SOCG.2022.26.
- 15 Steven Chaplick, Emilio Di Giacomo, Fabrizio Frati, Robert Ganian, Chrysanthi N. Raftopoulou, and Kirill Simonov. Testing upward planarity of partial 2-trees. In Patrizio Angelini and Reinhard von Hanxleden, editors, *Proc. 30th Int. Symp. Graph Drawing & Netw. Vis. (GD)*, volume 13764 of *LNCS*, pages 175–187, 2022. doi:10.1007/978-3-031-22203-0_13.
- 16 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshantov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
- 17 Giuseppe Di Battista and Roberto Tamassia. On-line planarity testing. *SIAM J. Comput.*, 25:956–997, 1996. doi:10.1137/S0097539794280736.
- 18 Walter Didimo, Francesco Giordano, and Giuseppe Liotta. Upward spirality and upward planarity testing. *SIAM J. Discrete Math.*, 23(4):1842–1899, 2010. doi:10.1137/070696854.
- 19 Walter Didimo, Giuseppe Liotta, and Fabrizio Montecchiani. A survey on graph drawing beyond planarity. *ACM Comput. Surv.*, 52(1):4:1–4:37, 2019. doi:10.1145/3301281.

- 20 Vida Dujmović, Ken-ichi Kawarabayashi, Bojan Mohar, and David R. Wood. Improved upper bounds on the crossing number. In Monique Teillaud, editor, *Proc. 24th ACM Symp. Comput. Geom. (SoCG)*, pages 375–384, 2008. doi:10.1145/1377676.1377739.
- 21 Fedor V. Fomin and Petr A. Golovach. Interval degree and bandwidth of a graph. *Discrete Appl. Math.*, 129(2–3):345–359, 2003. doi:10.1016/S0166-218X(02)00574-7.
- 22 Fabrizio Frati, Radoslav Fulek, and Andres J. Ruiz-Vargas. On the page number of upward planar directed acyclic graphs. In Marc J. van Kreveld and Bettina Speckmann, editors, *Proc. 19th Int. Symp. Graph Drawing (GD)*, volume 7034 of *LNCS*, pages 391–402. Springer, 2011. doi:10.1007/978-3-642-25878-7_37.
- 23 Fabrizio Frati, Radoslav Fulek, and Andres J. Ruiz-Vargas. On the page number of upward planar directed acyclic graphs. *J. Graph Algorithms Appl.*, 17(3):221–244, 2013. doi:10.7155/JGAA.00292.
- 24 Radoslav Fulek, Michael J. Pelsmajer, Marcus Schaefer, and Daniel Štefankovič. Hanani–Tutte, monotone drawings, and level-planarity. In János Pach, editor, *Thirty Essays on Geometric Graph Theory*, pages 263–287. Springer, 2013. doi:10.1007/978-1-4614-0110-0_14.
- 25 Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- 26 Ashim Garg and Roberto Tamassia. Upward planarity testing. *Order*, 12:109–133, 1995. doi:10.1007/BF01108622.
- 27 Ashim Garg and Roberto Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM J. Comput.*, 31(2):601–625, 2001. doi:10.1137/S0097539794277123.
- 28 Francesco Giordano, Giuseppe Liotta, Tamara Mchedlidze, Antonios Symvonis, and Sue Whitesides. Computing upward topological book embeddings of upward planar digraphs. *J. Discrete Algorithms*, 30:45–69, 2015. doi:10.1016/J.JDA.2014.11.006.
- 29 Lenwood S. Heath, Sriram V. Pemmaraju, and Ann N. Trenk. Stack and queue layouts of directed acyclic graphs: Part I. *SIAM J. Comput.*, 28(4):1510–1539, 1999. doi:10.1137/S0097539795280287.
- 30 Seok-Hee Hong and Takeshi Tokuyama, editors. *Beyond Planar Graphs, Communications of NII Shonan Meetings*. Springer, 2020. doi:10.1007/978-981-15-6533-5.
- 31 Paul Jungeblut, Laura Merker, and Torsten Ueckerdt. Directed acyclic outerplanar graphs have constant stack number. In *Proc. 64th IEEE Ann. Symp. Foundat. Comput. Sci. (FOCS)*, pages 1937–1952, 2023. doi:10.1109/FOCS57990.2023.00118.
- 32 Stephen G. Kobourov, Giuseppe Liotta, and Fabrizio Montecchiani. An annotated bibliography on 1-planarity. *Comput. Sci. Rev.*, 25:49–67, 2017. doi:10.1016/J.COSREV.2017.06.002.
- 33 Tamara Mchedlidze and Antonios Symvonis. Crossing-optimal acyclic Hamiltonian path completion and its application to upward topological book embeddings. In S. Das and Ryuhei Uehara, editors, *Proc. Workshop Algorithms & Comput. (WALCOM)*, volume 5431 of *LNCS*, pages 250–261. Springer, 2009. doi:10.1007/978-3-642-00202-1_22.
- 34 Martin Nöllenburg and Sergey Pupyrev. On families of planar DAGs with constant stack number. In Michael A. Bekos and Markus Chimani, editors, *Proc. Int. Symp. Graph Drawing & Netw. Vis. (GD)*, volume 14465 of *LNCS*, pages 135–151. Springer, 2023. doi:10.1007/978-3-031-49272-3_10.
- 35 Achilleas Papakostas. Upward planarity testing of outerplanar DAGs. In Roberto Tamassia and Ioanni G. Tollis, editors, *Proc. Int. Sympos. Graph Drawing (GD)*, volume 894 of *LNCS*, pages 298–306. Springer, 1994. doi:10.1007/3-540-58950-3_385.
- 36 Marcus Schaefer. The graph crossing number and its variants: A survey. *Electr. J. Combin.*, Dynamic Survey DS21, 2024. doi:10.37236/2713.
- 37 Pavel Valtr. On the pair-crossing number. In J. E. Goodman, J. Pach, and E. Welzl, editors, *Combinatorial and Computational Geometry*, volume 52 of *MSRI Publications*, pages 569–575. Cambridge University Press, 2005. URL: <https://library2.msri.org/books/Book52/files/31valtr.pdf>.

13:20 The Price of Upwardness

- 38 David R. Wood. Characterisations of intersection graphs by vertex orderings. *Australasian J. Combin.*, 34:261–268, 2006. URL: https://ajc.maths.uq.edu.au/pdf/34/ajc_v34_p261.pdf.
- 39 David R. Wood and Jan Arne Telle. Planar decompositions and the crossing number of graphs with an excluded minor. *New York J. Math.*, 13:117–146, 2007. URL: <https://nyjm.albany.edu/j/2007/13-8.html>.

Bounding the Treewidth of Outer k -Planar Graphs via Triangulations

Oksana Firman 

Universität Würzburg, Germany

Grzegorz Gutowski 

Institute of Theoretical Computer Science, Faculty of Mathematics and Computer Science,
Jagiellonian University, Kraków, Poland

Myroslav Kryven 

University of Manitoba, Canada

Yuto Okada 

Nagoya University, Japan

Alexander Wolff 

Universität Würzburg, Germany

Abstract

The *treewidth* is a structural parameter that measures the tree-likeness of a graph. Many algorithmic and combinatorial results are expressed in terms of the treewidth. In this paper, we study the treewidth of *outer k -planar* graphs, that is, graphs that admit a straight-line drawing where all the vertices lie on a circle, and every edge is crossed by at most k other edges.

Wood and Telle [New York J. Math., 2007] showed that every outer k -planar graph has treewidth at most $3k + 11$ using so-called planar decompositions, and later, Auer et al. [Algorithmica, 2016] proved that the treewidth of outer 1-planar graphs is at most 3, which is tight.

In this paper, we improve the general upper bound to $1.5k + 2$ and give a tight bound of 4 for $k = 2$. We also establish a lower bound: we show that, for every even k , there is an outer k -planar graph with treewidth $k + 2$. Our new bound immediately implies a better bound on the *cop number*, which answers an open question of Durocher et al. [GD 2023] in the affirmative.

Our treewidth bound relies on a new and simple triangulation method for outer k -planar graphs that yields few crossings with graph edges per edge of the triangulation. Our method also enables us to obtain a tight upper bound of $k + 2$ for the *separation number* of outer k -planar graphs, improving an upper bound of $2k + 3$ by Chaplick et al. [GD 2017]. We also consider *outer min- k -planar* graphs, a generalization of outer k -planar graphs, where we achieve smaller improvements.

2012 ACM Subject Classification Mathematics of computing → Graph theory

Keywords and phrases treewidth, outerplanar graphs, outer k -planar graphs, outer min- k -planar graphs, cop number, separation number

Digital Object Identifier 10.4230/LIPIcs.GD.2024.14

Related Version *Full Version*: <https://arxiv.org/abs/2408.04264>

Funding *Oksana Firman*: Supported by DFG grant Wo758/9-1.

Grzegorz Gutowski: The research cooperation was funded by the program Excellence Initiative – Research University at the Jagiellonian University in Kraków.

Yuto Okada: Supported by JST SPRING, Grant Number JPMJSP2125 and JSPS KAKENHI, Grant Number JP22H00513 (Hirotaka Ono).

Acknowledgements We thank Yota Otachi for suggestions that helped us to improve the lower bound. We also thank Hirotaka Ono for supporting our work.



© Oksana Firman, Grzegorz Gutowski, Myroslav Kryven, Yuto Okada, and Alexander Wolff;
licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 14; pp. 14:1–14:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Treewidth measures the tree-likeness of a graph via so-called tree decompositions. A *tree decomposition* of a graph G covers the vertex set of G by bags such that every edge is in some bag and the bags form a tree such that, for every vertex v of G , the bags that contain v form a subtree. The width of a tree decomposition is the maximum size of a bag minus one. The treewidth of G is the minimum width over all tree decompositions of G .

Treewidth is an important structural parameter because the running time of many graph algorithms and algorithms for drawing graphs depend on the treewidth. Recently treewidth-based techniques have been successfully applied for *convex* drawings, that is, straight-line drawings where vertices lie on a circle. For example, Bannister and Eppstein [8] ([7]) showed that if a graph admits a convex drawing with at most k crossings in total, then its treewidth is bounded by a function of k . Furthermore, they used this fact to give, for a fixed k , a linear-time algorithm (via extended monadic second-order logic and Courcelle’s theorem [13]) to decide whether a convex drawing with at most k crossings exists for a given graph. Chaplick et al. [12] ([11]) generalized this to *bundled crossings* (a crossing of two bundles of edges such that in each bundle the edges travel in parallel, which is counted as one crossing). Another prominent class of graphs with convex drawings is the class of *outer k -planar* graphs, that is, graphs that admit a convex drawing where each edge is crossed at most k times. Unlike *k -planar* graphs (without the restriction on the placement of vertices), the treewidth of outer k -planar graphs can be bounded by a function of k only (see discussion below). Similarly to Bannister and Eppstein [8], Chaplick et al. [10] used this to test in linear time, for any fixed k , whether a given graph is *full outer k -planar*, i.e., whether it admits an outer k -planar drawing where no crossing appears on the boundary of the outer face.

For disambiguation, recall that *k -outerplanar* graphs are defined as follows. A drawing of a graph is 1-outerplanar if all vertices of the graph lie on the outer face. For $k > 1$, a drawing is k -outerplanar if deleting the vertices on the outer face yields a $(k - 1)$ -outerplanar drawing. For $k \geq 1$, a graph is k -outerplanar if it admits a k -outerplanar drawing.

In this paper, we are particularly interested in the treewidth of outer k -planar graphs. Note that every graph is outer k -planar for some value of k . For a graph G , let $\text{lcr}^\circ(G)$ denote the *convex local crossing number* of G – the smallest k such that G is outer k -planar. Consult Schaefer’s survey [19] for details on this and many other types of crossing numbers. Outer k -planar graphs admit balanced separators of size $O(k)$ (more below) associated with the drawing, which makes it possible to test, for any fixed k , outer k -planarity in quasi-polynomial time [10]. This implies that the recognition problem is not NP-hard unless the Exponential Time Hypothesis fails [10].

We also consider a generalization of outer k -planar graphs, namely *outer min- k -planar graphs*, which are graphs that admit a convex drawing where, for every pair of crossing edges, at least one edge is crossed at most k times. Wood and Telle [20, Proposition 8.5] showed that any (min-) outer k -planar graph has treewidth at most $3k + 11$. More precisely, they showed that, for every outer min- k -planar graph G , one obtains a tree decomposition of width at most $3k + 11$ from a *planar decomposition* of G , which is a generalization of a tree decomposition. The proof uses a result by Bodlaender [9] on k -outerplanar graphs.

For constant k , better treewidth bounds are known. A folklore result is that outerplanar graphs ($k = 0$) have treewidth at most 2. Auer et al. [5, 6] ([4]) showed that maximal outer 1-planar graphs are chordal graphs and, therefore, have treewidth at most 3, which is tight.

Another parameter linked to the treewidth is the separation number of a graph. A pair of vertex sets (A, B) is a *separation* of a graph G if $A \cup B = V(G)$ and there is no edge between $A \setminus B$ and $B \setminus A$. A separation (A, B) is said to be *balanced* if the sizes of both $A \setminus B$ and

$B \setminus A$ are at most $2n/3$, where n is $|V(G)|$. For a (balanced) separation (A, B) , the set $A \cap B$ is called (balanced) *separator*, and $|A \cap B|$ is the *order* of (A, B) . The *separation number* of G , $\text{sn}(G)$, is the minimum integer k such that every subgraph of G has a balanced separation of order k . Note that, for any subgraph G' of G , $\text{tw}(G') \leq \text{tw}(G)$. This implies that, for every graph G , $\text{sn}(G) \leq \text{tw}(G) + 1$ [15]. On the other hand, Dvořák and Norin [15] showed that, for every graph G , $\text{tw}(G) \leq 15 \text{sn}(G)$.

Our contribution. Given an outer k -planar drawing Γ of a graph G , let the *outer cycle* of G be the cycle that connects the vertices of G in the order along the circle on which they lie in Γ (even if G does not contain all edges of this cycle). We introduce two simple methods to construct, given an outer k -planar drawing Γ of a graph G , a triangulation of the outer cycle of G with the property that each edge of the triangulation is crossed by at most k edges of G ; see Section 3. The resulting triangulations yield the following bounds; see Section 4.

- We improve the upper bound of Wood and Telle [20] regarding the treewidth of outer k -planar graphs from $3k + 11$ to $1.5k + 2$ (Theorem 13 in Section 4.1). Our proof is constructive and implies a practical algorithm; the tree decomposition that we obtain follows the weak dual of the triangulation. For outer 2-planar graphs, our methods yield an upper bound of 4 (Theorem 12), which is tight due to K_5 .
- Chaplick et al. [10] showed that, for every outer k -planar graph G , its separation number is at most $2k + 3$. We improve this upper bound to $k + 2$; see Section 4.2.
- We give new lower bounds of $k + 2$ for both the treewidth and the separation number of outer k -planar graphs; see Section 5. Note that the latter bound is tight (Theorem 17).
- Durocher et al. [14] recently proved that the cop number of general 1-planar graphs is not bounded, but the maximal 1-planar graphs have cop number at most 3. Since it is known [16] that for every graph G , $\text{cop}(G) \leq \text{tw}(G)/2 + 1$, Durocher et al. [14] observed that the treewidth bound of Wood and Telle yields, for every outer k -planar graph G , that $\text{cop}(G) \leq 1.5k + 6.5$. They asked explicitly whether the multiplicative factor of 1.5 can be improved. We answer this question in the affirmative since our new treewidth bound immediately yields the better bound $\text{cop}(G) \leq 0.75k + 1.75$.
- For outer min- k -planar graphs, our triangulation improves the bound of Wood and Telle $3k + 11$ slightly to $3k + 1$ and gives a bound $2k + 1$ on the separation number; see Section 4.

Related results. A structurally similar type of result is known for *2-layer k -planar* graphs. A 2-layer k -planar graph is a bipartite graph that admits a k -planar straight-line drawing with vertices placed on two parallel lines. The maximal 2-layer 0-planar graphs (called *caterpillars*) are the maximal pathwidth-1 graphs [17, 18]. Angelini et al. [2] ([3]) showed that 2-layer k -planar graphs have pathwidth at most $k + 1$ and gave a lower bound of $(k + 3)/2$ for every odd number k . Hence, their result can be considered as a smooth generalization of the caterpillar result. Similarly, our treewidth bound $1.5k + 2$ for outer k -planar graphs smoothly generalizes the treewidth-2 bound for maximal outerplanar graphs.

2 Preliminaries

A *convex drawing* Γ of a graph is a straight-line drawing where the vertices of the graph are placed on different points of a circle, which we call the *circle of Γ* . An *outer k -planar drawing* Γ of a graph is a convex drawing such that every edge crosses at most k other edges. A counterclockwise walk of the circle of Γ yields a cyclic order on the vertices of the graph. We say that two sets on distinct four vertices $\{v_1, w_1\}$, and $\{v_2, w_2\}$ are *intertwined* in Γ

if they are ordered $\langle v_1, v_2, w_1, w_2 \rangle$ or $\langle v_1, w_2, v_2, w_1 \rangle$ in this cyclic order. Observe that two edges $\{v_1, w_1\}$ and $\{v_2, w_2\}$ cross in Γ if and only if they span four different vertices and are intertwined. In the remainder of this section, we make some simple observations that will be helpful later; they can be skipped by more experienced readers.

► **Observation 1.** *Let G be a graph. If Γ is an outer k -planar drawing of G , then every convex drawing Γ' of G with the same cyclic vertex order as in Γ is also an outer k -planar drawing of G .*

Proof. As the cyclic orders on vertices of G defined by Γ and Γ' are the same, exactly the same pairs of edges cross in Γ and in Γ' . ◀

In the definition of convex drawings, we could have allowed placing vertices on arbitrary convex shapes (instead of a circle) and drawing edges as curves (instead of straight-line segments) – as long as every curve is drawn inside the circle and no two curves cross more than once. Using curves will be sometimes convenient in our proofs later.

► **Observation 2.** *Let G be a graph, let Γ be an outer k -planar drawing of G , and let c be a curve that joins vertices v and w of G and is contained inside the circle of Γ . If c crosses at most l edges of Γ , then the straight-line segment \overline{vw} also crosses at most l edges of Γ .*

Proof. As c must cross every edge $\{v', w'\}$ of G that is intertwined with $\{v, w\}$, we get that c must have at least one crossing with every edge crossed by the straight-line segment \overline{vw} . ◀

A graph G is *maximal outer k -planar* if G is outer k -planar and G does not contain any vertex pair $e \in V^2(G) \setminus E(G)$ such that the graph $(V(G), E(G) \cup \{e\})$ is still outer k -planar. Since removing edges increases neither treewidth nor separation number, we are interested in properties of maximal outer k -planar graphs.

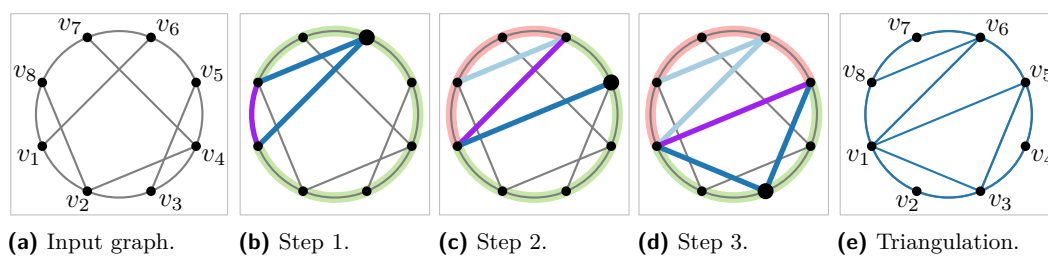
► **Observation 3.** *If G is a maximal outer k -planar graph with at least three vertices, then, in every outer k -planar drawing of G , the outer face is bounded by a simple cycle.*

Proof. Let v and w be two vertices that are consecutive in the cyclic order defined by some outer k -planar drawing Γ of G . For a contradiction, suppose that v and w are not adjacent. As v and w are consecutive on the circle of Γ , the set $\{v, w\}$ is not intertwined with any edge in Γ . Thus, Γ can be extended to an outer k -planar drawing of $G + \{v, w\}$, which contradicts the maximality of G . ◀

We remark that Observations 1–3 analogously hold for outer min- k -planar graphs.

3 Triangulations

In this section, we present two simple strategies to triangulate maximal outer k -planar graphs. These triangulations will serve as tools to construct tree decompositions and balanced separators in Section 4. We assume that some outer k -planar drawing Γ of a graph G on n vertices is given. During the triangulation procedures, we will modify the drawing, but we will never change the cyclic order of the vertices. Thus, let v_1, v_2, \dots, v_n be the vertices of G in the cyclic (counterclockwise) order defined by Γ . As we focus on maximal graphs, by Observation 3, we have that $\langle v_1, v_2, \dots, v_n, v_1 \rangle$ is a cycle in G . Our goal is to construct a triangulation \mathcal{T} of this cycle such that the edges of \mathcal{T} cross only a limited number of edges of G . We refer to the edges of \mathcal{T} as *links* in order to distinguish them from the edges of G . Every edge of the outer cycle is a link in \mathcal{T} ; we call these links *outer links*. We will select



■ **Figure 1** Steps of the splitting procedure. The active edge is purple, the (new/old) links are (dark/light) blue, the left/right side of the active edge is red/green. The split vertex is big.

a set of $n - 3$ other pairwise non-intertwined pairs of vertices as *inner* links that, together with the outer links, form a triangulated n -gon. Note that some of the inner links may coincide with edges of G . We say that a link is *pierced* by an edge of G if their endpoints are intertwined in the cyclic order. The *piercing number* of a link is the number of edges of G that pierce the link. In particular, the piercing number of outer links is 0, and if a link coincides with an edge, then its piercing number is at most k by the outer k -planarity of Γ . We define the *edge piercing number* of \mathcal{T} to be the maximum piercing number of any link of \mathcal{T} . In Section 4 we show how to use triangulations with small edge piercing number.

Our *splitting procedure* starts by declaring the link $\{v_1, v_n\}$ active and considers all other vertices to lie on the *right* side of the link. See Figure 1 for illustration. Clearly, the link $\{v_1, v_n\}$ is not pierced. In each recursive step, the input is an active link $\{v_i, v_k\}$ with $1 \leq i < k \leq n$ (promised to be pierced at most some limited number of times), along with a distinguished right side v_{i+1}, \dots, v_{k-1} , where currently no inner links have been selected apart from $\{v_i, v_k\}$. (Vertices $v_1, \dots, v_{i-1}, v_{k+1}, \dots, v_n$ form the *left* side.) The goal is to pick, among the vertices on the right side, a *split vertex* v_j such that the new links $\{v_i, v_j\}$ and $\{v_j, v_k\}$ are pierced by at most some limited number of edges. The new links are added to \mathcal{T} (completing the triangle $v_i v_j v_k$) and become active. The link $\{v_i, v_k\}$ ceases to be active. The split gives rise to two new splitting instances $\{v_i, v_j\}$ and $\{v_j, v_k\}$, which are then solved recursively.

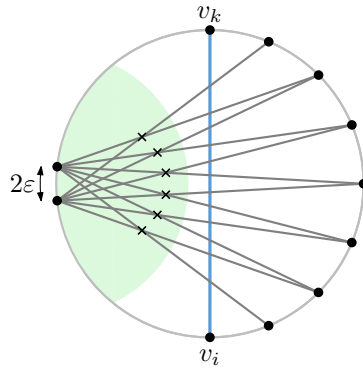
The set of edges of G that pierce the active link are the *piercing edges*. Note that every piercing edge has one left and one right endpoint. Intuitively, the edges with two left endpoints are of no concern to the splitting procedure. The next lemma allows us to push all crossings among piercing edges to the left of $\{v_i, v_k\}$.

► **Lemma 4.** *Given an outer k -planar drawing Γ of a graph G and an active link $\{v_i, v_k\}$, there exists an outer k -planar drawing Γ' with the same cyclic order as Γ , and all crossings among the edges piercing the active link are drawn to the left of the active link.*

Proof. We draw the vertices of G on the unit circle, keeping the cyclic order defined by Γ . We set $\varepsilon > 0$ to a sufficiently small value. Then we place v_i at $(0, -1)$, v_k at $(0, 1)$, all the vertices on the left side of the link within arc distance ε of $(-1, 0)$, and the vertices on the right side of the link evenly along the right semicircle, see Figure 2.

Now, Observation 1 yields that the new drawing is an outer k -planar drawing of G with the same cyclic order. As every pair of intertwined piercing edges has their left endpoints in arc distance at most ε from $(-1, 0)$ and their right endpoints in arc distance at least π/n from each other, the crossing of two such segments is to the left of the active link. ◀

Lemma 4 gives us an equivalent drawing of G in which piercing edges cross the active link in the same order from bottom to top as their right endpoints occur in the cyclic order. This allows us to draw the piercing edges without crossings on the right side of the active link.



■ **Figure 2** We can push crossings among the edges that pierce the active link $\{v_i, v_k\}$ to the left. We show only the top- and bottommost endpoints on the left. Crossings occur exclusively in the green area, which can be made arbitrarily small.

Now, we are ready to present our first and basic triangulation method. Asymptotically, it yields a worse bound than Lemma 6, but it serves as a gentle introduction into our techniques, and it will be used for small values of k and for outer min- k -planar graphs later.

► **Lemma 5.** *For $k \geq 1$, every outer k -planar drawing of a maximal outer k -planar graph admits a triangulation of the outer cycle with edge piercing number at most $2k - 1$.*

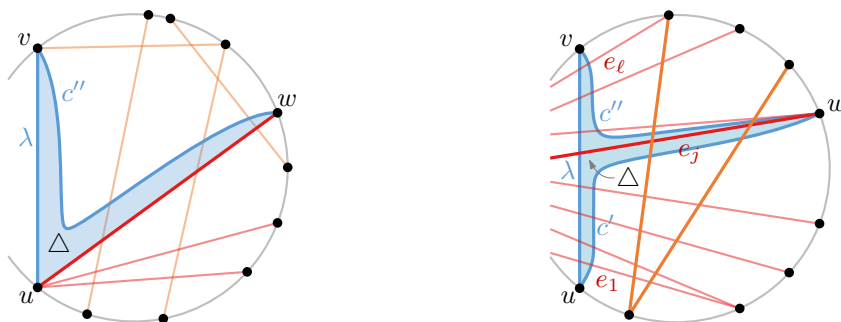
Proof. We prove the lemma by providing an appropriate splitting procedure that recursively constructs the triangulation. Let Γ be an outer k -planar drawing of the given maximal outer k -planar graph G . We prove by induction on the number of steps of the splitting procedure that every active link has piercing number at most $2k - 1$. The claim is true in the beginning, as we start with the link $\{v_1, v_n\}$, which is not pierced at all.

Now we consider a later step of the procedure. Let $\lambda = \{v_i, v_k\}$ denote the active link. For brevity, set $u = v_i$ and $v = v_k$. There are two cases.

In the first case, λ is not pierced by any edge of G . Observe that $\{v_i, v_{i+1}\}$ is an edge, so u has at least one neighbor on the right side of λ in G . We select the split vertex, w , to be the neighbor of u from the right side of λ that is closest to v in the circular order; see Figure 3a. In this case, the link $\{u, w\}$ coincides with an edge of G , and thus is pierced by at most k edges. To see that the link $\{w, v\}$ is also pierced by at most k edges, let c'' be a curve that starts in w , goes along $\{w, u\}$ towards u on the right side (in the direction of walking), then follows $\{u, v\}$ on the right side, and ends at v . This curve crosses only the edges that pierce the link $\{u, w\}$, so at most k edges pierce the link $\{w, v\}$ by Observation 2.

In the second case, λ is pierced by edges of G . Using Lemma 4, we modify the drawing Γ so that the piercing edges cross λ in the same bottom-to-top order as their right endpoints. Let e_1, e_2, \dots, e_ℓ denote the piercing edges, and let w_1, w_2, \dots, w_ℓ denote their respective, not necessarily different, right endpoints in the order from bottom to top. By induction, we have $\ell \leq 2k - 1$. Then we pick the middle edge e_j , where $j = \lceil (\ell + 1)/2 \rceil$, and set the split vertex $w = w_j$.

To bound the number of edges that pierce the link $\{u, w\}$, we draw the curve c' as in Figure 3b. We start the curve c' at u and follow λ until before it crosses e_j , then follow e_j until we arrive at w . The curve c' crosses at most $\lceil (\ell - 1)/2 \rceil$ edges that pierce λ and at most k edges that pierce e_j . Thus, there are at most $\lceil (\ell - 1)/2 \rceil + k \leq \lceil (2k - 2)/2 \rceil + k = 2k - 1$ edges that pierce the link $\{u, w\}$. We can argue symmetrically for the link $\{w, v\}$; see curve c'' in Figure 3b.



(a) Constructing the new triangle Δ if λ is not pierced by any other edge of G . (b) Constructing the new triangle Δ bounded by λ , c' , and c'' ; the orange edges that pierce both c' and c'' also cross the edge e_j .

■ **Figure 3** Constructing a triangulation \mathcal{T} with edge piercing number at most $2k - 1$.

As we have bounded the piercing number of every active link during the triangulation procedure by $2k - 1$, we get the desired bound of $2k - 1$ for the edge piercing number of the constructed triangulation. ◀

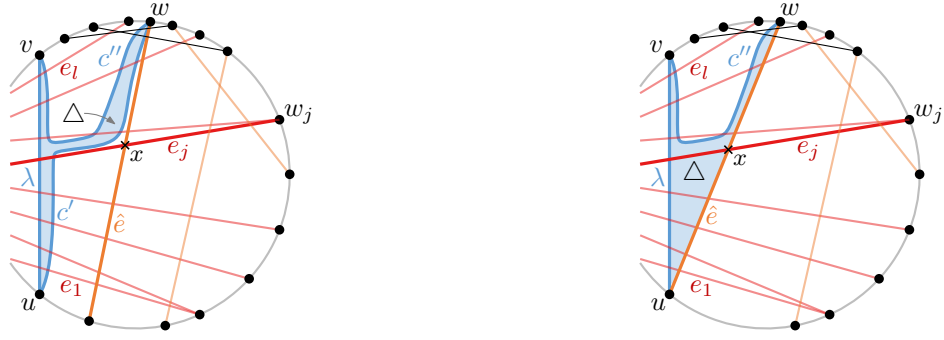
Next, we present our refined triangulation method, where we consider an additional edge to get a better bound for the edge piercing number.

► **Lemma 6.** *For every $k \geq 1$, every outer k -planar drawing of a maximal outer k -planar graph admits a triangulation of the outer cycle with edge piercing number at most k .*

Proof. The proof is similar to the proof of Lemma 5, and we use the same terminology. Let again Γ be an outer k -planar drawing of the given maximal outer k -planar graph G . We present a splitting procedure and show that every active link is pierced by at most k edges. For the base of the induction, observe that the link $\{v_1, v_n\}$ has no piercing edges. Now, let $\lambda = \{v_i, v_k\}$ be the active link and set $u = v_i$ and $v = v_k$. If λ has no piercing edges in Γ , we proceed as in the proof of Lemma 5 and pick the split vertex to be the neighbor of u from the right side of λ that is closest to v . Otherwise, λ is pierced by edges of G . Using Lemma 4, we assume that the piercing edges cross the active link in the same bottom-to-top order as their right endpoints. Now, let e_1, e_2, \dots, e_ℓ be the piercing edges, and let w_1, w_2, \dots, w_ℓ be their respective, not necessarily different, right endpoints in the order from bottom to top. We pick the middle edge e_j with $j = \lceil (\ell + 1)/2 \rceil$. If e_j has no crossings on the right side of the active link, then we pick w_j to be the split vertex. The two curves that start in w_j , follow e_j to e , and then follow λ to either u or v have at most $\lceil (\ell - 1)/2 \rceil \leq k$ crossings each, so we can proceed. Now, let x denote the first crossing on the edge e_j that occurs on the right side of λ , and let \hat{e} denote the edge that crosses e_j at x ; see Figure 4a.

We first consider the case where \hat{e} and λ are disjoint. Let a and b be the numbers of crossings on \hat{e} on the two sides of x . As $a + b + 1 \leq k$, we get that $\min\{a, b\} \leq \lfloor (k - 1)/2 \rfloor$. We select the split vertex w to be the endpoint of \hat{e} on the side that has fewer, i.e., $\min\{a, b\}$, crossings along \hat{e} . We now assume $w_j < w < v$. The case $u < w < w_j$ is symmetric.

We bound the piercing number of the link $\{v, w\}$. To this end, we start tracing a curve c'' at v and follow the link λ until just before it crosses e_j , then we follow e_j until just before it crosses \hat{e} , and then we follow \hat{e} until we arrive at w ; see Figure 4a. In the first part, the curve crosses at most $\lceil (\ell - 1)/2 \rceil \leq \lceil (k - 1)/2 \rceil$ edges. The curve does not cross any edge in the middle part, as the crossing between e_j and \hat{e} is the first crossing along e_j starting from λ . In the last part, the curve crosses at most $\min\{a, b\} \leq \lfloor (k - 1)/2 \rfloor$ edges. Thus, by Observation 2, the piercing number of the link $\{v, w\}$ is at most $\lceil (k - 1)/2 \rceil + \lfloor (k - 1)/2 \rfloor \leq k - 1$.



(a) λ and \hat{e} do not share an endpoint; the new triangle Δ is bounded by λ , c' , and c'' .

(b) λ and \hat{e} share the endpoint u ; the new triangle Δ is bounded by λ , \hat{e} , and c'' .

■ **Figure 4** Constructing a triangulation \mathcal{T} with edge piercing number k .

To bound the piercing number of the link $\{u, w\}$, we start tracing a curve c' at u , follow the link λ until just after it crosses e_j and then follow the curve c'' to w , see Figure 4a. The curve c' crosses the same set of edges as the curve c'' plus the edge e_j , so c'' has at most k crossings. Thus, the piercing number of the link $\{u, w\}$ is at most k .

Now we consider the last case, where \hat{e} and λ have a common endpoint. In this case, we choose the other endpoint of \hat{e} as w (see Figure 4b) and analyse the two subcases.

We first assume that u is the common endpoint. Then the link $\{u, w\} = \hat{e}$ is an edge of G , so it has at most k crossings. To bound the piercing number of the link $\{v, w\}$, we argue with the curve c'' that goes from v to w as defined in the previous case. Observe that the edge $\{u, w\}$ has exactly j crossings on the part from u to x (including x). Therefore, we can argue as above and conclude that the piercing number of the edge $\{v, w\}$ is at most $(\ell - j) + (k - j) = k - (2j - \ell)$. This is bounded from above by $k - 1$ since $2j - \ell = 2\lceil(\ell + 1)/2\rceil - \ell$ is 1 if ℓ is odd and 2 otherwise.

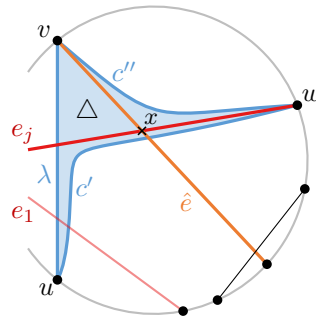
Similarly, if v is the common endpoint, then the link $\{v, w\} = \hat{e}$ is an edge of G , so it is crossed by at most k edges, and the piercing number of the link $\{u, w\}$ is at most $(j - 1) + (k - (\ell - j + 1)) = k + (2j - \ell) - 2 \leq k$. ◀

The *piercing number* of a triangle of \mathcal{T} is the sum of the piercing numbers of the links that form the sides of the triangle. We define the *triangle piercing number* of \mathcal{T} to be the maximum piercing number of any triangle of \mathcal{T} . Note that the triangle piercing number of a triangulation is at most three times its edge piercing number, but it can be smaller. If k is odd, we obtain a slightly stronger result; if k is odd, for every triangle t that we create in the proof of Lemma 6, at least one edge of t has at most $k - 1$ crossings.

► **Lemma 7.** *For every odd $k \geq 1$, every outer k -planar drawing of a maximal outer k -planar graph admits a triangulation of the outer cycle with edge piercing number at most k and triangle piercing number at most $3k - 1$.*

Proof. Following the proof of Lemma 6, in all cases, there is an edge with piercing number at most $k - 1$. If λ has no piercing edges, its piercing number is $0 \leq k - 1$. Otherwise, if \hat{e} and $\{u, v\}$ are disjoint, $\{v, w\}$ is pierced by at most $k - 1$ edges. In the last case, where \hat{e} and $\{u, v\}$ have a common endpoint, $\{v, w\}$ is pierced by at most $k - 1$ edges if ℓ is odd. If ℓ is even, it implies $\ell \neq k$ and therefore the link λ is pierced by $\ell \leq k - 1$ edges. ◀

With a slight modification of the strategy described in the proof of Lemma 6 and a careful analysis, we also obtain a better bound of the triangle piercing number for the case $k = 2$.



■ **Figure 5** λ and \hat{e} share the endpoint v ; the new triangle Δ is bounded by λ , c' , and c'' .

► **Lemma 8.** *Every outer 2-planar drawing of a maximal outer 2-planar graph admits a triangulation of the outer cycle with edge piercing number at most 2 and triangle piercing number at most 4.*

Proof. We roughly follow the strategy of the proof of Lemma 6. Recall that λ is the active link and that ℓ is the number of edges that pierce λ . If $\ell > 0$, then e_j with $j = \lceil (\ell + 1)/2 \rceil$ is the “middle” edge among those that pierce λ . If e_j has crossings on the right side of λ , then, among these, x is the crossing point of e_j closest to λ . The edge \hat{e} is the edge that crosses e_j in x . As in the proof of Lemma 6, we distinguish the following cases.

1. λ has no piercing edges.
2. e_j has no crossings to the right of λ .
3. λ and \hat{e} are disjoint (Figure 4a).
4. λ and \hat{e} meet in u (Figure 4b).
5. λ and \hat{e} meet in v .

In cases 1–4, we choose the split vertex w as in the proof of Lemma 6, which means that the new active links are pierced at most $k = 2$ times.

In case 5, we now choose w to be w_j (see Figure 5). Let c' be a curve that starts in u , follows λ until just before e_j and then follows e_j to w . Let c'' be a curve that starts in v , follows \hat{e} to x , and then follows e_j to w . In order to show that each of the two curves is crossed by at most two edges, we need to do some preparations. We say that an edge lies to the *left* of the active link if both of its endpoints lie on the left side or one endpoint lies on the left and the other is an endpoint of the active link. We further say that an edge e piercing a link λ is *anchored* with respect to λ if there exists another edge f that crosses e and either $f = \lambda$ or f lies to the left of λ . Observe that an anchored edge can have at most one crossing to the right of λ .

We claim that, during the splitting procedure described above, every edge piercing an active link is anchored (with respect to the active link). The claim is true in the beginning as $\lambda = \{v_1, v_n\}$ is not pierced. In case 1, every edge that pierces $\{u, w\}$ or $\{v, w\}$ also crosses the edge $\{u, w\}$, so it is anchored. In case 2, every edge that pierces $\{u, w\}$ or $\{v, w\}$ also pierces $\lambda = \{u, v\}$ and is anchored by induction. In case 3, observe that no edge crosses e_j between λ and x (as x is the first crossing on e_j to the right of λ), and no edge crosses \hat{e} between x and w (by the choice of w and since $\lfloor (k - 1)/2 \rfloor = 0$ for $k = 2$). Therefore, every edge that pierces $\{u, w\}$ or $\{v, w\}$ also pierces λ and is anchored by induction. In case 4, $\{u, w\}$ is an edge, so every edge piercing it is anchored. Every edge that pierces $\{v, w\}$ either crosses $\{u, w\}$ or pierces λ and is anchored by induction. Similarly, in case 5, every edge that pierces $\{u, w\}$ or $\{v, w\}$ either pierces λ or crosses e_j (which lies to the left of both new active links).

14:10 Bounding the Treewidth of Outer k -Planar Graphs via Triangulations

We now return to case 5 and bound the piercing numbers of the new active links $\{u, w\}$ and $\{v, w\}$. To do this, we count the edges that cross the curves c' and c'' , respectively. As e_j is anchored with respect to λ , we get that x is the only crossing on e_j to the right of λ . Now, it is easy to see that c' is pierced only by \hat{e} and possibly by e_1 in case $j = 2$. For c'' , we can show that there are no crossings along the curve. First, observe that any edge that crosses \hat{e} between v and x has to pierce λ , which is not possible. Second, no edge crosses e_j between x and w , as e_j is anchored and x is the only crossing on e_j to the right of λ . Combining these two observations, we see that there is no place for c'' to cross with an edge.

It remains to show that the triangle piercing number of the new triangle Δ formed by u , v , and w is at most 4. In case 1, λ has no piercing edges. In cases 2, 4, and 5, $\{v, w\}$ has no piercing edges. In case 3, we distinguish two subcases depending on the relative position of w . When $w_j < w$, then $\{v, w\}$ has no piercing edges. When $w < w_j$, then each of $\{u, w\}$ and $\{v, w\}$ is pierced at most once. ◀

Lastly, we combine the two triangulation strategies from the proofs of Lemmas 5 and 6 to obtain a triangulation method for outer min- k -planar graphs.

► **Lemma 9.** *For every $k \geq 1$, every outer min- k -planar drawing of a maximal outer min- k -planar graph admits a triangulation of the outer cycle with edge piercing number at most $2k - 1$ and triangle piercing number at most $6k - 3$.*

Proof. We proceed as in the proof of Lemma 5. We describe a splitting procedure such that every active link has piercing number at most $2k - 1$. Consult Figure 3 for a reminder of the notation and the strategy. As the given drawing is outer min- k -planar, edges may now have more than k crossings. We call such edges *heavy* and the other edges *light*. By definition, heavy edges cross only light edges.

We first explain why the previous proof breaks in both cases if the curves c' and c'' are routed along heavy edges. Recall that in the first case when λ has no piercing edges, we selected the split vertex w to be the neighbor of u from the right side of λ that is closest to v . If, however, the edge $\{u, w\}$ is heavy, then we cannot bound the number of crossings along the curve c'' . In the second case, we selected the split vertex to be the right endpoint w_j of the middle piercing edge e_j . If the edge e_j is heavy, then we cannot bound the numbers of crossings along c' and c'' .

To resolve the issue with the second case, we apply the strategy from the proof of Lemma 6. Observe that if e_j is heavy, then for the first crossing x with an edge \hat{e} on the right side of λ , the edge \hat{e} is a light. Similarly, as in the proof of Lemma 6 (see Figure 4), we select the split vertex w to be the endpoint of \hat{e} that is different from u and v and in case both endpoints are, the one on the side of x with fewer crossings. The piercing numbers of the two new links are at most $\lceil (2k - 1)/2 \rceil + k - 1 \leq 2k - 1$.

To resolve the issue with the first case, we use a similar strategy. If the edge $\{u, w\}$ is heavy, let x be the crossing on $\{u, w\}$ that is closest to u and let \hat{e} be the crossing edge. Note that \hat{e} is light. We select the split vertex w to be any right endpoint of \hat{e} . Now consider the two curves that start at w , go along \hat{e} to x , then along $\{u, w\}$ to u , and optionally to v along λ . Both curves have at most k crossings, only with edges that also cross \hat{e} . ◀

► **Remark 10.** Given the intersection graph of the edges of G , the triangulations in Lemmas 5–9 can be constructed in $O(nk)$ time. Each step of the splitting procedures presented in Lemmas 5–9 can be implemented to run in $O(k)$ time, as the edges that pierce the new active links also pierce at least one of the edges λ , e_j , and \hat{e} .

4 Applications

In this section, we show several applications of our triangulations from Section 3. Among others, we improve the bound of Wood and Telle [20] on the treewidth of outer k -planar graphs from $3k + 11$ to $1.5k + 2$ (see Section 4.1) and improve the bound of Chaplick et al. [10] on the separation number of outer k -planar graphs from $2k + 3$ to $k + 2$; see Section 4.2. In addition, we improve the bound of Wood and Telle of $3k + 11$ on the treewidth of outer min- k -planar graphs to $3k + 1$; see Theorem 14.

4.1 Treewidth

Now we show our main tool to obtain a better upper bound on the treewidth. Note that we do not refer specifically to outer k -planar graphs since our tool can be used for any drawing of a graph whose vertices lie on the outer face.

Consider a graph G with a drawing Γ whose outer face is bounded by a simple cycle C that contains all vertices of G . Let \mathcal{T} be a planar triangulation of C and f be some triangular face of \mathcal{T} . We call an edge of $E(G) \setminus E(\mathcal{T})$ *short* with respect to f if one of its endpoints is on f and it pierces a link of f . We call an edge of $E(G) \setminus E(\mathcal{T})$ *long* with respect to f if neither of its endpoints is on f and it pierces two links of f .

► **Lemma 11.** *If G is a graph with a convex drawing whose outer cycle admits a triangulation \mathcal{T} with triangle piercing number at most c , then $\text{tw}(G) \leq (c + 5)/2$.*

Proof. Let T be a tree obtained by taking the weak dual graph of \mathcal{T} and associating each node of T with a bag that contains the vertices of G that are incident to the corresponding face. We modify this tree to consider the piercing edges, ensuring that the sizes of bags do not exceed $(c + 7)/2$. (Recall that the width of a tree decomposition is the size of the largest bag minus 1.)

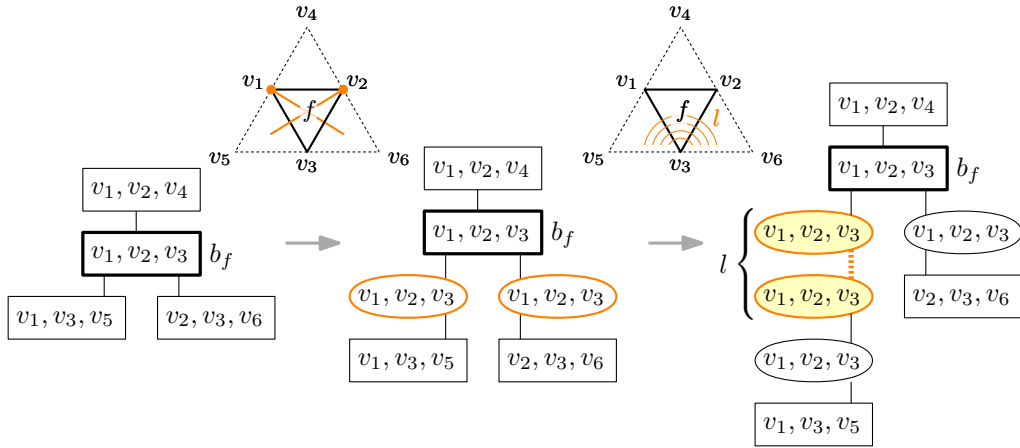
If \mathcal{T} has an unpierced inner link $\lambda = \{u, v\}$, we split \mathcal{T} along λ into two triangulations and compute tree decompositions for them. Since each of them has a bag that contains λ , we can connect them and obtain a tree decomposition of G .

Hence, from now on, we assume that every inner link of \mathcal{T} is pierced at least once. We root T at an arbitrary leaf. The basic idea is as follows. Consider an edge $\{u, v\}$ that pierces inner links of \mathcal{T} . Then we *lift* both u and v to the bag b that is the lowest common ancestor of the highest bags b_u and b_v that contain u and v , respectively. That is, we place a copy of u into each bag on the way from b_u to b and symmetrically for v . If we apply this strategy naively, then there may be a bag with $c + 3$ vertices in the worst case. In the following, we describe how to improve this bound.

Consider a face f of the triangulation \mathcal{T} . Let b_f be the bag in T that corresponds to f . Let v_1, v_2 , and v_3 be the vertices of f such that f shares the link $\{v_1, v_2\}$ with the face that corresponds to the parent of f in T (if any). An edge of $E(G) \setminus E(\mathcal{T})$ is *lineal* if one of its endpoints is in the bag b_f or above and the other endpoint is in a bag below b_f in the tree T . A long edge of $E(G) \setminus E(\mathcal{T})$ is *bent* if its endpoints are in different subtrees of T .

We perform the following two modifications of T for each face f ; see Figure 6.

- We subdivide each edge of T between b_f and its children (if any) by a new *copy bag* that is a copy of b_f . We call these bags *primary* copy bags.
- If f has two children in T , assume that piercing number of $\lambda = \{v_1, v_3\}$ is less or equal to that of $\lambda' = \{v_2, v_3\}$ and that λ is the common link of f and its left child face. If there are $l > 0$ bent edges $\{u_1, w_1\}, \dots, \{u_l, w_l\}$ w.r.t. f , then we subdivide the edge of T between b_f and its left child by additional l copies of b_f . These *secondary* copy bags are numbered $b_{f,1}, \dots, b_{f,l}$ from top to bottom; see Figure 6.



■ **Figure 6** Steps 1 and 2 of the modification. Copy bags are oval; secondary copy bags are yellow.

Next, we lift vertices in bottom-up order, e.g., in preorder. More precisely, at each *original* bag b_f (that is, not a copy bag) in our preorder traversal of T , we process each edge e of G that pierces f , according to its type as follows.

- If e is a short edge, that is, it is incident to a vertex of f , namely to v_1 or v_2 , then we lift the highest occurrence of the other endpoint u of e by one bag (as v_5 and v_7 in Figure 7a) to the primary copy bag of b_f .
- If e is a long edge that pierces the link $\{v_1, v_2\}$, then we lift the highest occurrence of the other endpoint u of e all the way to b_f ; see Figure 7b.
- If $e = \{u_i, w_i\}$ for some $i \in \{1, \dots, l\}$, then we lift both endpoints of e . First, we lift the highest occurrence of u_i from the left subtree to $b_{f,i}$. Second, we lift the highest occurrence of w_i in the right subtree of b_f to b_f . Third, we copy w_i into each bag from b_f down to $b_{f,i}$; see Figure 7c. In this way, the two endpoints of e meet in $b_{f,i}$.

The modifications described above make sure that the resulting tree T' is indeed a tree decomposition of G , i.e., for each vertex v of G , the subtree induced by the bags that contain v is connected, and for each edge e of G , there is a bag in T' that contains both endpoints of e .

In the following, we show that each bag of T' contains at most $(c+7)/2$ vertices, i.e., we have added at most $(c+1)/2$ to the initial three vertices. For each face f , the number of vertices added to the original bag b_f is the number of long edges w.r.t. f , which is at most $c/2$. It remains to bound the number of vertices that are added to each copy bag of b_f .

First, we show the bound for secondary copy bags. Recall that, for $i \in \{1, \dots, l\}$, we have added to $b_{f,i}$ from below i endpoints of the bent long edges w.r.t. f and from above $l+1-i$ endpoints of bent long edges; see Figure 7c. In addition, we have added a vertex for each lineal long edge. Therefore, the number of added vertices is at most one plus the number of long edges w.r.t. f that pierce the link $\{v_1, v_3\}$. Recall that the link $\{v_1, v_3\}$ was chosen to have at most as many piercing edges as $\{v_2, v_3\}$. Because each inner link of \mathcal{T} , including $\{v_1, v_2\}$, is pierced at least once, $\{v_1, v_3\}$ is pierced at most $(c-1)/2$ times. Thus, we have added at most $(c+1)/2$ vertices to $b_{f,i}$, and so $|b_{f,i}| \leq (c+7)/2$.

Now, we bound the number of vertices that are added to the primary (the bottommost) copy bag b'_f of b_f . Note that such a bag is a parent bag of some original bag b_g such that the faces f and g are adjacent. There are two types of vertices that are added to b'_f . The first type consists of the endpoints of the lineal long edges of b_g (at most $c/2$). The second one is just one vertex w of the face g that is not shared with f (for example, the endpoint v_5 of g

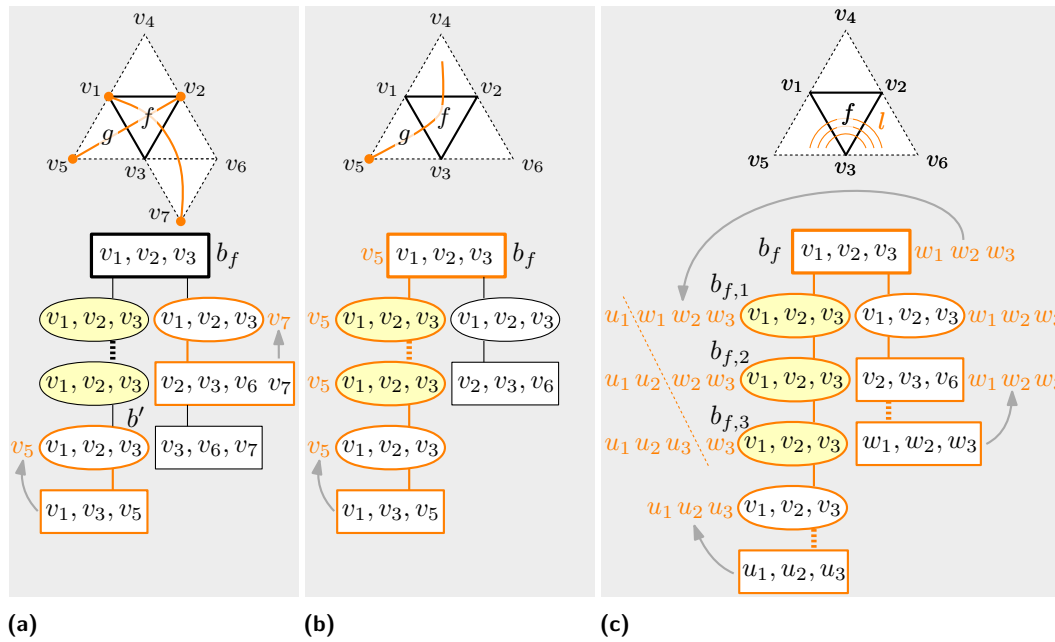


Figure 7 Lifting up endpoints of the edges that pierce links of the face f .

in Figure 7a). We have added w only if there are edges in $E(G) \setminus E(\mathcal{T})$ with lower endpoint w (lower w.r.t. T). Each such edge contributes to the piercing number of g . Hence, the number of lineal long edges w.r.t. g is at most $(c - 1)/2$. In both cases ($w \in b'$ and $w \notin b'$), we have added at most $(c + 1)/2$ vertices to b' , so $|b'| \leq (c + 7)/2$. ◀

Lemmas 8 and 11 give us an upper bound 4 for outer 2-planar graphs, extending the known tight bound $k + 2$ for $k = \{0, 1\}$ [5, 6]. Note that K_5 is outer 2-planar and $\text{tw}(K_5) = 4$.

► **Theorem 12.** *Every outer 2-planar graph has treewidth at most 4, which is tight.*

► **Theorem 13.** *Every outer k -planar graph has treewidth at most $1.5k + 2$.*

Proof. We already know better bounds for $k \leq 2$. For $k \geq 3$, by Lemmas 6, 7, and 11, we obtain a bound $1.5k + 2.5$ for even k and $1.5k + 2$ for odd k . As treewidth is an integer, we obtain $1.5k + 2$ for general k . ◀

By Lemmas 9 and 11, we also obtain the following bound for outer min- k -planar graphs, which improves the previously known bound $3k + 11$ by Wood and Telle [20]. Note that outer min-0-planar graphs are outerplanar graphs.

► **Theorem 14.** *For $k \geq 1$, every outer min- k -planar graph has treewidth at most $3k + 1$.*

4.2 Separation Number

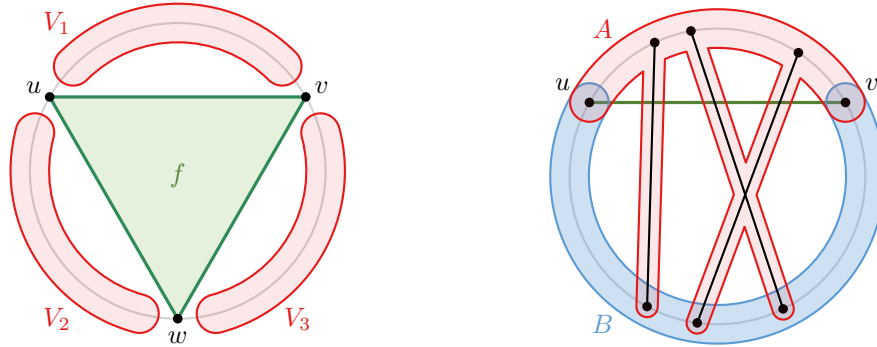
Theorems 13 and 14 immediately imply upper bounds on the separation number of outer k -planar and outer min- k -planar graphs. However, using our triangulations directly, we obtain even better bounds, namely $k + 2$ for outer k -planar graphs and $2k + 1$ for outer min- k -planar graphs. The first bound improves the bound of $2k + 3$ by Chaplick et al. [10].

14:14 Bounding the Treewidth of Outer k -Planar Graphs via Triangulations

► **Lemma 15.** *If G is a graph with a convex drawing whose outer cycle admits a triangulation \mathcal{T} with edge piercing number at most c , then G has a balanced separator of size at most $c + 2$.*

Proof. We construct a balanced separation of order at most $c + 2$ using \mathcal{T} . In short, we select a link λ of \mathcal{T} that is a balanced separator for \mathcal{T} and put its endpoints into the separator. Then, we add at most c vertices to the separator according to the edges piercing λ .

First, we find a “centroid” triangle of \mathcal{T} as follows. Let T be the tree that is the weak dual of \mathcal{T} . It is well known that every tree contains a vertex such that, after removing it, the number of vertices in each subtree is at most half of the original tree. Let $f = \{u, v, w\}$ be the triangle corresponding to such a vertex of T . We partition $V \setminus f$ into three disjoint sets V_1, V_2, V_3 ; see Figure 8a. We may assume that V_1 is the largest among the three sets.



(a) “Centroid” triangle f and vertex sets V_1, V_2, V_3 . (b) Edges piercing $\{u, v\}$ and separation (A, B) .

■ **Figure 8** A triangulation with edge piercing number c yields a balanced separator of size at most $c + 2$.

Now, we construct the desired separation with the help of the sets V_1, V_2, V_3 . As $\lambda = \{u, v\}$ is a link of \mathcal{T} , at most c edges pierce λ , connecting vertices in V_1 with vertices in $V_2 \cup \{w\} \cup V_3$. Let S be the set of endpoints of the piercing edges on the latter side. Let $A = V_1 \cup \{u, v\} \cup S$ and $B = V_2 \cup \{u, v, w\} \cup V_3$; see Figure 8b. We claim that (A, B) is a balanced separation of order at most $c + 2$.

Clearly, the order of (A, B) is $|A \cap B| = |S \cup \{u, v\}|$, which is at most $c + 2$. Hence, it suffices to show that the sizes of $A \setminus B$ and $B \setminus A$ are at most $2n/3$. To this end, we first bound the size of V_1 as follows:

$$\frac{n}{3} - 1 \leq |V_1| \leq \frac{n}{2} - 1.$$

The lower bound holds as $|V_1| + |V_2| + |V_3| = n - 3$ and $|V_1|$ is the largest. The upper bound can be shown by the fact that a triangulated outerplanar graph has exactly $F + 2$ vertices, where F is the number of triangles. By the way of choosing f , the vertex set $V_1 \cup \{u, v\}$ induces at most $n/2 - 1$ triangles. Therefore, $|V_1| = |V_1 \cup \{u, v\}| - 2 \leq (n/2 - 1 + 2) - 2 = n/2 - 1$.

Now we can confirm that the sizes of $A \setminus B$ and $B \setminus A$ are at most $2n/3$ as follows:

$$|A \setminus B| = |V_1| \leq \frac{n}{2} - 1 \leq \frac{2}{3}n \quad \text{and}$$

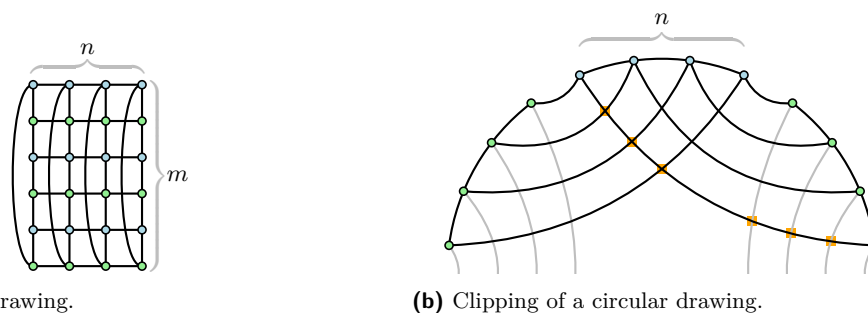
$$|B \setminus A| \leq |V_2 \cup \{w\} \cup V_3| = |V \setminus (V_1 \cup \{u, v\})| \leq n - \left(\frac{n}{3} - 1\right) - 2 = \frac{2}{3}n - 1. \quad \blacktriangleleft$$

For every k , the classes of outer k -planar graphs and outer min- k -planar graphs are closed under taking subgraphs. Hence, by Lemmas 6, 9, and 15, we obtain the following bounds.

► **Theorem 16.** *Every outer k -planar graph has separation number at most $k + 2$, and for $k \geq 1$, every outer \min - k -planar graph has separation number at most $2k + 1$.*

5 Lower Bounds

In this section, we complement the results in Section 4, by giving lower bounds for the separation number and treewidth of outer k -planar graphs. To show them, we use grid-like graphs called *stacked prisms*. The $m \times n$ stacked prism $Y_{m,n}$ is the (planar) graph obtained by connecting all pairs vertices in the topmost and bottommost row of the $m \times n$ grid that are in the same column; see Figure 9a.



(a) Grid-like drawing.

(b) Clipping of a circular drawing.

■ **Figure 9** Two drawings of the stacked prism $Y_{m,n}$.

► **Theorem 17.** *For every even number $k \geq 0$, there exists a graph G such that $\text{lcr}^\circ(G) = k$ and $\text{sn}(G) = k + 2$.*

Proof. We first observe that, if m is even, $\text{lcr}^\circ(Y_{m,n})$ is at most $2n - 2$. If m is even, we can place the m rows of $Y_{m,n}$, each of length n , one after the other, alternating in their direction, around a circle; see Figure 9b. The grid has two types of edges: the *row edges* and the *column edges*, which connect vertices in the same row or column, respectively. As Figure 9b shows, the row edges have no crossing. For each $i \in [n]$, there is a column edge e_i that spans $2i - 2$ other vertices along the perimeter of the circle. Each of these vertices is incident to exactly one column edge that crosses e_i . Hence, every column edge has at most $2n - 2$ crossings, and $\text{lcr}^\circ(Y_{m,n}) \leq 2n - 2$.

Let $n = k/2 + 1$, and let m be a sufficiently large even number. Then we claim that $Y_{m,n}$ fulfills the conditions. As we discussed $\text{lcr}^\circ(Y_{m,n}) \leq k$ holds and therefore $\text{sn}(Y_{m,n}) \leq k + 2$ follows from Theorem 16. Hence, it suffices to show $\text{sn}(Y_{m,n}) \geq k + 2 = 2n$, which also implies $\text{lcr}^\circ(Y_{m,n}) \geq 2n - 2 = k$ by Theorem 16.

Suppose that $Y_{m,n}$ has a balanced separator S of size less than $2n$. As there are n columns, there is a column that contains at most one vertex of S . This column contains a path P of length $m - 1$ that does not intersect S . Now observe that at least $m - 2n$ rows do not contain any vertex of S , and therefore, all vertices in these rows are connected to P . Hence, after removing S , the size of the connected component C that contains P is at least $n(m - 2n)$. The ratio $|V(C)|/|V(Y_{m,n})|$ is $1 - (2n/m)$, which is greater than $2/3$ if m is sufficiently large. ◀

Aidun et al. [1] showed that $\text{tw}(Y_{m,n}) = 2n$ if $m > 2n$. With the same stacked prism, we obtain the following lower bound.

► **Theorem 18.** *For every even number $k \geq 0$, there exists a graph G such that $\text{lcr}^\circ(G) = k$ and $\text{tw}(G) = k + 2$.*

6 Conclusion and Open Problems

We have introduced methods for triangulating drawings of outer k -planar graphs such that the triangulation edges cross few graph edges. These triangulations yield better bounds on treewidth and separation number of outer k -planar graphs. Our method is constructive; the corresponding treewidth decomposition and balanced separation can be computed efficiently. Via our triangulations, we improved the multiplicative constant in the upper bound on the treewidth of outer k -planar graphs from 3 to 1.5; we showed a lower bound of 1. What is the correct multiplicative constant? Finding triangulations of outer k -planar graphs with lower triangle piercing number could be a step in this direction. It would also be interesting to find other graph classes that admit triangulations with low triangle piercing number.

References

- 1 Ivan Aidun, Frances Dean, Ralph Morrison, Teresa Yu, and Julie Yuan. Treewidth and gonality of glued grid graphs. *Discrete Applied Mathematics*, 279:1–11, 2020. doi:10.1016/j.dam.2019.10.024.
- 2 Patrizio Angelini, Giordano Da Lozzo, Henry Förster, and Thomas Schneck. 2-layer k -planar graphs: Density, crossing lemma, relationships and pathwidth. *Computer Journal*, 67(3):1005–1016, 2024. doi:10.1093/comjnl/bxad038.
- 3 Patrizio Angelini, Giordano Da Lozzo, Henry Förster, and Thomas Schneck. 2-layer k -planar graphs – density, crossing lemma, relationships, and pathwidth. In David Auber and Pavel Valtr, editors, *28th International Symposium on Graph Drawing and Network Visualization (GD 2020)*, volume 12590 of *Lecture Notes in Computer Science*, pages 403–419. Springer, 2020. doi:10.1007/978-3-030-68766-3_32.
- 4 Christopher Auer, Christian Bachmaier, Franz J. Brandenburg, Andreas Gleißner, Kathrin Hanauer, Daniel Neuwirth, and Josef Reislhuber. Recognizing outer 1-planar graphs in linear time. In Stephen K. Wismath and Alexander Wolff, editors, *21st International Symposium on Graph Drawing (GD 2013)*, volume 8242 of *Lecture Notes in Computer Science*, pages 107–118. Springer, 2013. doi:10.1007/978-3-319-03841-4_10.
- 5 Christopher Auer, Christian Bachmaier, Franz J. Brandenburg, Andreas Gleißner, Kathrin Hanauer, Daniel Neuwirth, and Josef Reislhuber. Outer 1-planar graphs. *Algorithmica*, 74(4):1293–1320, 2016. doi:10.1007/s00453-015-0002-1.
- 6 Christopher Auer, Christian Bachmaier, Franz J. Brandenburg, Andreas Gleißner, Kathrin Hanauer, Daniel Neuwirth, and Josef Reislhuber. Correction to: Outer 1-planar graphs. *Algorithmica*, 83(11):3534–3535, 2021. doi:10.1007/S00453-021-00874-Z.
- 7 Michael J. Bannister and David Eppstein. Crossing minimization for 1-page and 2-page drawings of graphs with bounded treewidth. In Christian A. Duncan and Antonios Symvonis, editors, *22nd International Symposium on Graph Drawing (GD 2014)*, volume 8871 of *Lecture Notes in Computer Science*, pages 210–221. Springer, 2014. doi:10.1007/978-3-662-45803-7_18.
- 8 Michael J. Bannister and David Eppstein. Crossing minimization for 1-page and 2-page drawings of graphs with bounded treewidth. *Journal of Graph Algorithms and Applications*, 22(4):577–606, 2018. doi:10.7155/jgaa.00479.
- 9 Hans L. Bodlaender. A partial k -arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1-2):1–45, 1998. doi:10.1016/S0304-3975(97)00228-4.
- 10 Steven Chaplick, Myroslav Kryven, Giuseppe Liotta, Andre Löffler, and Alexander Wolff. Beyond outerplanarity. In Fabrizio Frati and Kwan-Liu Ma, editors, *25th International Symposium on Graph Drawing and Network Visualization (GD 2017)*, volume 10692 of *Lecture Notes in Computer Science*, pages 546–559. Springer, 2017. doi:10.1007/978-3-319-73915-1_42.

- 11 Steven Chaplick, Thomas C. van Dijk, Myroslav Kryven, Ji-won Park, Alexander Ravsky, and Alexander Wolff. Bundled crossings revisited. In Daniel Archambault and Csaba D. Tóth, editors, *27th International Symposium on Graph Drawing and Network Visualization (GD 2019)*, volume 11904 of *Lecture Notes in Computer Science*, pages 63–77. Springer, 2019. doi:10.1007/978-3-030-35802-0_5.
- 12 Steven Chaplick, Thomas C. van Dijk, Myroslav Kryven, Ji-won Park, Alexander Ravsky, and Alexander Wolff. Bundled crossings revisited. *Journal of Graph Algorithms and Applications*, 24(4):621–655, 2020. doi:10.7155/jgaa.00535.
- 13 Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.
- 14 Stephane Durocher, Shahin Kamali, Myroslav Kryven, Fengyi Liu, Amirhossein Mashghdoust, Avery Miller, Pouria Zamani Nezhad, Ikaro Penha Costa, and Timothy Zapp. Cops and robbers on 1-planar graphs. In Michael A. Bekos and Markus Chimani, editors, *31st International Symposium on Graph Drawing and Network Visualization (GD 2023)*, volume 14466 of *Lecture Notes in Computer Science*, pages 3–17. Springer, 2023. doi:10.1007/978-3-031-49275-4_1.
- 15 Zdeněk Dvořák and Sergey Norin. Treewidth of graphs with balanced separations. *Journal of Combinatorial Theory, Series B*, 137:137–144, 2019. doi:10.1016/j.jctb.2018.12.007.
- 16 Gwenaël Joret, Marcin Kamiński, and Dirk Oliver Theis. The cops and robber game on graphs with forbidden (induced) subgraphs. *Contributions to Discrete Mathematics*, 5(2):40–51, 2010. URL: <http://cdm.ucalgary.ca/cdm/index.php/cdm/article/view/154>, doi:10.11575/cdm.v5i2.62032.
- 17 Nancy G. Kinnersley and Michael A. Langston. Obstruction set isolation for the gate matrix layout problem. *Discrete Applied Mathematics*, 54(2-3):169–213, 1994. doi:10.1016/0166-218X(94)90021-3.
- 18 Andrzej Proskurowski and Jan Arne Telle. Classes of graphs with restricted interval models. *Discrete Mathematics & Theoretical Computer Science*, 3(4):167–176, 1999. doi:10.46298/dmtcs.263.
- 19 Marcus Schaefer. The graph crossing number and its variants: A survey. *Electronic Journal of Combinatorics*, DS21(version 8), 2024. doi:10.37236/2713.
- 20 David R. Wood and Jan Arne Telle. Planar decompositions and the crossing number of graphs with an excluded minor. *New York Journal of Mathematics*, 13:117–146, 2007. URL: <https://nyjm.albany.edu/j/2007/13-8.html>.

Bundling-Aware Graph Drawing

Daniel Archambault ✉ 🏠 

Newcastle University, UK

Giuseppe Liotta ✉ 🏠 

University of Perugia, Italy

Martin Nöllenburg ✉ 🏠 

TU Wien, Austria

Tommaso Piselli ✉ 

University of Perugia, Italy

Alessandra Tappini ✉ 🏠 

University of Perugia, Italy

Markus Wallinger ✉ 

TU Munich, Germany

Abstract

Edge bundling algorithms significantly improve the visualization of dense graphs by reducing the clutter of many edges visible on screen by bundling them together. As such, bundling is often viewed as a post-processing step applied to a drawing, and the vast majority of edge bundling algorithms consider a graph and its drawing as input. Another way of thinking about edge bundling is to simultaneously optimize both the drawing and the bundling. In this paper, we investigate methods to simultaneously optimize a graph drawing and its bundling. We describe an algorithmic framework which consists of three main steps, namely *Filter*, *Draw*, and *Bundle*. We then propose two alternative implementations and experimentally compare them against the state-of-the-art approach and the simple idea of drawing and subsequently bundling the graph. The experiments confirm that bundled drawings created by our framework outperform previous approaches according to standard quality metrics for edge bundling.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms; Human-centered computing → Graph drawings

Keywords and phrases Edge Bundling, Experimental Comparison, Graph Sparsification

Digital Object Identifier 10.4230/LIPIcs.GD.2024.15

Supplementary Material *Software*: <https://doi.org/10.17605/OSF.IO/G4XQW> [36]

Funding *Giuseppe Liotta*: MUR PON Proj. ARS01_00540; MUR PRIN Proj. 2022TS4Y3N; MUR PRIN Proj. 2022ME9Z78.

Martin Nöllenburg: Vienna Science and Technology Fund (WWTF) [10.47379/ICT19035].

Tommaso Piselli: MUR PON Proj. ARS01_00540; MUR PRIN Proj. 2022TS4Y3N; MUR PRIN Proj. 2022ME9Z78.

Alessandra Tappini: MUR PON Proj. ARS01_00540; MUR PRIN Proj. 2022TS4Y3N; MUR PRIN Proj. 2022ME9Z78.

Markus Wallinger: Vienna Science and Technology Fund (WWTF) [10.47379/ICT19035].

1 Introduction

The majority of bundling algorithms consider both the graph and its drawing as input [23]. With the notable exception of approaches that are inspired by confluent drawings [2, 41], the coordinates of the nodes are vital for computing a bundled drawing of high quality. However, often we would like to produce a graph layout that simultaneously optimizes the drawing and



© Daniel Archambault, Giuseppe Liotta, Martin Nöllenburg, Tommaso Piselli, Alessandra Tappini, and Markus Wallinger;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 15; pp. 15:1–15:19

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

the bundling. For some graphs, edge bundling will not be necessary: if we have a sufficiently good drawing of a graph, there is no need to bundle it. In the case of dense graphs, it will be very difficult to find a drawing of the graph that does not require some or many of the edges to be bundled to resolve its hairball-like appearance. In such cases, it may be advantageous for us to consider bundling alongside drawing.

One way to optimize the bundling alongside the drawing is to select edges that are likely to be bundled, remove them from the graph, and draw the skeleton of the edges that will be bundled against. Edge-Path bundling (EPB) algorithms [37, 38] have such a skeleton with edges bundled against paths. In this case, the smaller graph that would be computed for layout would be the edges that participate in the paths used for bundling. The confluent drawing inspired approaches of Bach et al. [2] and Zheng et al. [41] perform aggregation via a power set decomposition and create a hierarchy of coarse graphs that are used to draw and bundle the graph in a confluent way. This aggregation approach will help focus graph layouts on edges that can support bundles. However, confluent drawings have a lower degree of bundling when compared to EPB, and it may be interesting to investigate approaches with a greater degree of bundling.

In this paper, we contribute a framework which offers a greater degree of bundling than confluent-like approaches while optimizing bundling alongside drawing. Two instantiations of this framework are explored. Our **Filter-Draw-Bundle** framework selects nodes and edges that form the scaffold of the bundling, or the set of edges in the graph that will be bundled against. Edges that are more likely to be bundled are filtered from the graph. As Edge-Path bundling has a very strong connection to t -spanners [1, 38], we show how various levels of t -spanners can be used for drawing with the remaining edges bundled onto the spanner.

2 Related Work

Edge bundling algorithms have been studied for nearly 20 years now [16] with many approaches [23] created to simplify edge clutter of densely connected networks. Most edge bundling techniques [4, 9, 10, 17, 19, 20, 22, 24, 27, 29–32, 35, 39] consider a drawing and a graph as input and bundle edges together based on similar properties such as orientation, co-location in space, and other similar edge properties. Recent Edge-Path bundling techniques [37, 38] instead bundle long edges with paths. The main advantage for Edge-Path bundling is that it does not create the illusion of false connections between disconnected edges as the subtended edge always has a corresponding path. In all of the above cases, a drawing of the graph is taken as input and is used for bundling. This has been successfully applied in practice, but considering the layout in combination with bundling has not been investigated before. Furthermore, our results show that creating a layout more tailored for bundling has better performance regarding certain quality metrics.

Confluent drawings [6, 7, 14] do not consider a drawing beforehand. Rather, the graph is converted into a planar graph by contracting down bicliques, drawing the high level planar graph, and reintroducing the bicliques into the graph so that they can be bundled without ambiguity. Less restrictive versions of these drawings [2, 33, 41] do not require the strict planar condition and generally compute a decomposition of the graph into components. The decomposition of the graph is drawn and then the edges between components are bundled. While heuristics for computing confluent drawings introduce less ambiguity compared to our proposed framework, they are much more constrained in what can be bundled, leading to less overall bundling.

In the work presented here, we focus on a new way of optimizing a bundling while drawing the graph simultaneously. In our case, we filter the graph in a way that takes away edges in an Edge-Path bundling that are likely to be bundled, providing a low-stress layout of the paths to be bundled against. In our experiment, we compare against the state-of-the-art methods that produce confluent-like drawings [2, 41] as well as against the straight-forward approach of first creating a low-stress drawing of the full input graph and then applying Edge-Path bundling on it.

3 The Filter-Draw-Bundle Framework

In this section, we introduce the **Filter-Draw-Bundle** (FDB) Framework. This framework is designed to optimize three well-studied quality metrics for bundled drawings, namely the ink ratio, distortion, and ambiguity. We start by formally defining these metrics, since we shall refer to them when describing the framework.

3.1 Bundling Quality Metrics

We consider three metrics that are in line with the ones described by Wallinger et al. [37, 38], namely *ink ratio*, *distortion*, and *ambiguity*. Smaller values in all three metrics are considered better. Let $G = (V, E)$ be the input graph, let \mathcal{A}_B be a bundling algorithm and let \mathcal{L}_B be the layout obtained by applying \mathcal{A}_B to G . The metrics we consider are defined as follows.

Ink Ratio. Informally, the ink ratio is the proportion of pixels within the drawing's bounding box that are occupied by \mathcal{L}_B . It can quantify simplification through a measure of data-ink ratio [34]. More precisely, let $L_B \in \{0, \dots, 255\}^{m \times n}$ be an $m \times n$ -greyscale bitmap image of \mathcal{L}_B , and let $0 \leq \delta < 255$ be a threshold grey value below which we consider a pixel occupied; in our implementation we consider $\delta = 254$, which means all non-white pixels are occupied. We scale the drawing \mathcal{L}_B to a bitmap that has a width of 1.000 pixels when computing the ink ratio. For each pixel $L_B[i, j]$ of L_B , where $1 \leq i \leq m$ and $1 \leq j \leq n$, we set $p_B(i, j) = 1$ if $L_B[i, j] \leq \delta$, otherwise we set $p_B(i, j) = 0$. The ink ratio of L_B is then defined as:

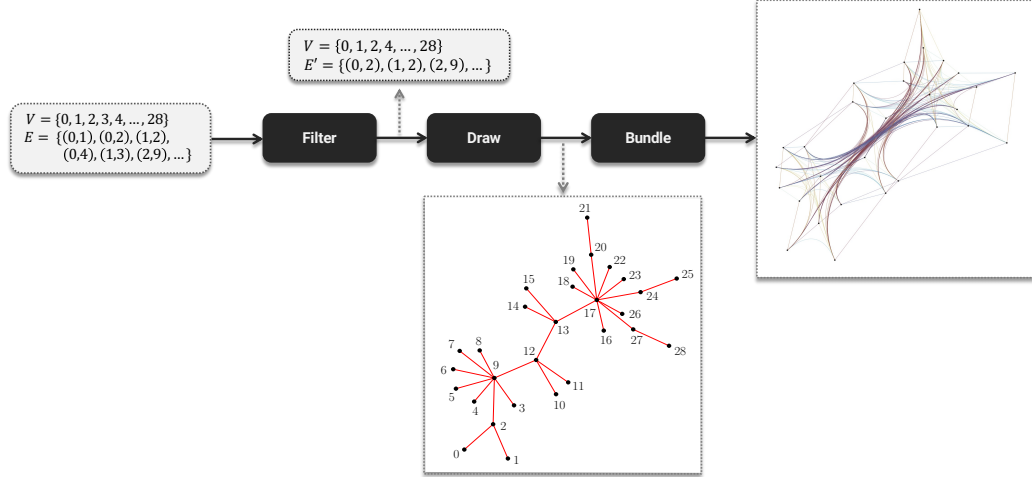
$$\text{InkRatio} = \frac{\sum_{i=1}^m \sum_{j=1}^n p_B(i, j)}{m \cdot n}. \quad (1)$$

We note that our definition of ink ratio is slightly different from Wallinger et al. [37, 38], who measured the reduction in occupied pixels compared to a straight-line input drawing. Since we do not consider any input drawing for the FDB framework, their definition is not applicable.

Distortion. The distortion of an edge is the ratio between its length in \mathcal{L}_B and the Euclidean distance of its endpoints. Human-centred studies have shown that deviations from straight line distances make paths harder to read [18]. Let $\ell_B(uv)$ be the length of an edge uv in \mathcal{L}_B and let $d(u, v)$ be the Euclidean distance between u and v . The distortion of \mathcal{L}_B is the average distortion of its edges, namely:

$$\text{Distortion} = \frac{1}{|E|} \sum_{uv \in E} \frac{\ell_B(uv)}{d(u, v)}. \quad (2)$$

Ambiguity. The ambiguity of a bundled drawing measures how many false adjacencies could be derived erroneously from the drawing \mathcal{L}_B . Avoiding such false connections would be considered more faithful [26]. More precisely, the ambiguity is defined as the ratio between



■ **Figure 1** Illustration of our FDB framework. The **Filter** step takes as input an abstract graph $G = (V, E)$ and outputs a sparsification $G' = (V, E')$ with $E' \subseteq E$; the **Draw** step computes a drawing Γ of G' ; the **Bundle** step adds to Γ the edges in E' and bundles them.

the number of perceivable false adjacencies and the total number of perceivable adjacencies. Here, an adjacency between two vertices u, v is a false adjacency if the topological distance between u and v is greater than a certain threshold. Given an edge $e = st \in E$, which is drawn as a curve in \mathcal{L}_B , the set $N_B(s, e)$ of reachable neighbors of s along e consists of all the vertices that are perceived as being connected to s . More precisely, a vertex v belongs to $N_B(s, e)$ if there is an edge $e' = uv \in E$ that is drawn in \mathcal{L}_B as a curve that intersects e by forming a flat angle which is smaller than a certain threshold θ . The set $N_B(s, e)$ may contain both true and false neighbors of s : we denote as $N_B^T(s, e)$ the set of the true neighbors and as $N_B^F(s, e) = N_B(s, e) \setminus N_B^T(s, e)$ the set of the false neighbors. We consider two vertices as true neighbors if the length of the shortest path between them in G (in hop distance) is smaller than a certain threshold δ . So if $\delta = 1$ (as in our implementation), $N_B^T(s, e)$ contains only the direct neighbors of s , otherwise it contains all vertices reachable in hop distance at most δ . The ambiguity of \mathcal{L}_B is then defined as:

$$\text{Ambiguity} = \frac{\sum_{s \in V} \sum_{e=st \in E} |N_B^F(s, e)|}{\sum_{s \in V} \sum_{e=st \in E} |N_B(s, e)|}. \quad (3)$$

3.2 Framework Description

We are now ready to describe our algorithmic framework, which we call **Filter-Draw-Bundle**, or **FDB** for short. The algorithmic framework is illustrated in Figure 1 and summarized in Algorithm 1. As the name suggests, it consists of the following three steps:

- **Filter**: Let $G = (V, E)$ be the input graph. In the **Filter** step, a subgraph G' of G is computed, with $G' = (V, E')$ such that $V' = V$ and $E' \subseteq E$. In particular, E' is supposed to be the subset of those edges of G against which the edges of $E \setminus E'$ will be bundled in the third step (in our case to optimize for Edge-Path bundling). Graph G' is called the *skeleton* of G .
- **Draw**: A drawing Γ of the skeleton G' is computed. The quality metrics of the subsequent bundled graph drawing will strongly depend on the layout of the skeleton.

- **Bundle:** Γ is enriched with those edges of G that were filtered out in the first step of FDB. In the instantiations that follow, we shall adopt the Edge-Path bundling (EPB) approach to bundle against the edges of the skeleton G' .

■ **Algorithm 1** The FDB Framework.

Input: Graph $G = (V, E)$.
Output: A bundled layout \mathcal{L}_B of G .

```

/* Filter Step */
w ← computeWeights(G)
G' = (V, E') ← sparsifyGraph(G, w) // G' is the skeleton of G
/* Draw Step */
Γ ← computeDrawing(G')
/* Bundle Step */
LB ← edgePathBundling(G, Γ)

```

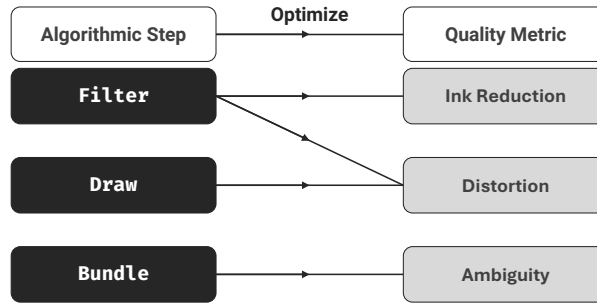
In contrast to other bundling approaches, the FDB framework does not receive the vertex coordinates of the final layout as part of the input, but it computes them during the **Draw** step. Hence, the FDB approach is not a post-processing procedure that improves the readability of a given drawing, but it takes into account the quality metrics of the target bundling during each step of its algorithmic pipeline, and in particular aims to find a drawing of a suitable skeleton graph that gives rise to a high-quality bundling.

It is also worth recalling that Bach et al. [2] and Zheng et al. [41] describe similar ideas. Namely, they first compute a hierarchical aggregation of the graph, then draw such an aggregation, and finally compute a confluent drawing. While the **Filter** step of the FDB pipeline computes a subgraph of G , the approach of [2, 41] suitably selects groups of nodes, aggregates them, and connects such groups with dummy edges. Once a layout of the hierarchical aggregation is computed, a confluent drawing of the graph is constructed by using these dummy edges as a guideline.

3.2.1 Instantiating the FDB Framework

While the flexibility of our algorithmic framework makes it possible to adopt different filtering, drawing, and bundling strategies, as a proof of concept we make specific choices for the three different steps which will be used in our experimental analysis. Namely, we describe two different instances of the FDB framework, which differ for the adopted strategy in the **Filter** step. In both cases, we consider filter and draw steps that optimize EPB. The framework can be used for other bundling algorithms, but appropriate filter and draw steps would be needed for the bundling algorithm selected. Figure 2 shows the relationship between our algorithmic choices at each step of the framework and the bundling quality metrics that we aim at optimizing.

Filter. The edges of E' for the skeleton subgraph are computed based on a weighting function. The edges that receive the highest weight will be included in the skeleton. The idea is to give a higher weight to those edges that are more “central” in the graph, based on the intuition that bundling against these edges optimizes the distortion.



■ **Figure 2** Relationships between our algorithmic choices at each step of the FDB framework and the bundling quality metrics that we aim at optimizing.

More formally, we weight the edges by two different methods. The first method uses the *edge betweenness* as a standard centrality measure for edges (see, e.g., [12]), defined as:

$$\text{EdgeBetweenness}(e) = \sum_{u,v \in V} \frac{\sigma(u,v | e)}{\sigma(u,v)}. \quad (4)$$

In the formula, $\sigma(u,v)$ denotes the number of shortest paths between any two vertices u and v of G , and $\sigma(u,v | e)$ is the number of such shortest paths passing through an edge e . The edge betweenness assigns a weight to each edge $e \in E$ based upon the computation of all-pairs shortest paths in G [3]. The weight of each edge is proportional to the number of shortest paths between all vertex pairs of G that pass through this particular edge.

A different approach, specifically tailored for the **Bundle** step, is to assign edge betweenness weights by considering the shortest paths only between those pairs of vertices that are actually adjacent in G , since these edges might later be bundled against their respective shortest path. We shall filter out those edges of G that appear in only a few such shortest paths. In the **Bundle** step, such edges will be reinserted one per time and possibly bundled against the edges of the skeleton. This leads to the following variant of the definition of edge betweenness, that we call *neighboring edge betweenness*:

$$\text{NeighboringEdgeBetweenness}(e) = \sum_{uv \in E} \frac{\sigma'(u,v | e)}{\sigma'(u,v)}, \quad (5)$$

where $\sigma'(u,v)$ is the number of shortest paths between u and v in $G_{uv} = (V, E \setminus \{uv\})$, and $\sigma'(u,v | e)$ is the number of these shortest paths passing through e . Algorithm 2 shows the pseudocode to compute the edge weights based on the neighboring edge betweenness. It is immediate to see that, since the edge betweenness of an unweighted graph can be computed in polynomial time (see, e.g., Brandes [3]), also the neighboring edge betweenness can be computed in polynomial time.

Once the edges are weighted, the skeleton G' is formed by filtering out the lighter edges from G . This can be done, for example, by computing a t -spanner [1] using the inverse weights. Recall that a t -spanner of a weighted graph G is a subgraph G' of G consisting of all the vertices of G and of a subset of its edges. The t -spanner property, for some $t > 1$, says that for each edge uv in G with weight $w(uv)$, there must be a (shortest) path π between u and v in G' whose weight $w(\pi) = \sum_{e \in \pi} w(e) \leq t \cdot w(uv)$. A typical approach to compute a t -spanner is to order the edges of the graph by increasing weight; starting from

■ **Algorithm 2** Edge weights computation with neighboring edge betweenness.

```

Input: Graph  $G = (V, E)$ .
Output: Function  $weights: E \rightarrow \mathbb{R}^+$ .
for  $uv \in E$  do
  |  $sp(uv) \leftarrow \text{shortestPaths}(u, v, G_{uv} = (V, E \setminus uv))$ 
for  $e \in E$  do
  | for  $uv \in E$  do
  | |  $sum \leftarrow 0$ 
  | | for  $p \in sp(uv)$  do
  | | | if  $p.contains(e)$  then
  | | | |  $sum \leftarrow sum + 1$ 
  | | |  $weights(e) \leftarrow weights(e) + \frac{sum}{|sp(uv)|}$ 
return  $weights$ 

```

a graph $G' = (V, \emptyset)$, the edges are processed one by one in that order, and at each step it is decided whether the edge is inserted into G' or not; refer to Algorithm 3, where we use $w(e) = 1/weights(e)$.

■ **Algorithm 3** Skeleton computation as a t -spanner [1, 38].

```

Input: Graph  $G = (V, E)$ , edge-weights  $w: E \rightarrow \mathbb{R}^+$ , value  $t > 1$ .
Output:  $t$ -spanner  $G' = (V, E' \subseteq E)$ .
 $E' \leftarrow \emptyset$ 
 $G' \leftarrow (V, E')$ 
 $sortedEdges \leftarrow \text{sortAscending}(E, w)$ 
for  $uv \in sortedEdges$  do
  |  $p \leftarrow \text{shortestPath}(G', u, v)$ 
  | if  $p$  is undefined then
  | |  $E' \leftarrow E' \cup \{uv\}$ 
  | else
  | |  $p.weight \leftarrow 0$ 
  | | for  $e \in p$  do
  | | |  $p.weight \leftarrow p.weight + w(e)$ 
  | | if  $p.weight > t * w(uv)$  then
  | | |  $E' \leftarrow E' \cup \{uv\}$ 
return  $G' = (V, E')$ 

```

According to this algorithm, the lighter edges, i.e., those with the highest (neighboring) betweenness score, are more likely to be part of G' than the heavier edges, which have low betweenness. The higher we choose the value of t the sparser the skeleton G' will be, since it is less likely during the construction algorithm that there is no path of weight at most $t \cdot w(uv)$ in G' . On the one hand, this is expected to influence the ink ratio of the bundled layout. The intuition is that the sparser the skeleton, the lower the ink ratio. Namely, when the skeleton is sparser more edges of $E \setminus E'$ are bundled against the same portion of the skeleton. On the other hand, sparser skeletons and higher values of t in the spanner construction are expected to also lead to higher distortions in the final bundled drawings. Therefore, we expect that the choice of the value of t is very influential for the final bundling quality, and it should be balanced to achieve both good ink ratio and good distortion.

Draw. The goal of this step is to compute a drawing of the (unweighted) skeleton G' such that the graph-theoretic distance between its vertices is well reflected by their Euclidean distance in the layout. The intuition behind this choice is that such a layout would, in fact, give rise to bundled drawings with small distortion because it tends to straighten those paths against which edges are bundled. To this aim, we compute the output of the **Layout** step by means of a stress majorization approach with stochastic gradient descent (SGD) [11, 40].

Bundle. This step adds to the layout of the skeleton $G' = (V, E')$ those edges that are not part of it, i.e., edges in $E \setminus E'$. Let Γ be the layout of the skeleton computed in the previous step and let $uv \in E \setminus E'$ be a *non-skeletal edge*. We incrementally add each non-skeletal edge to Γ by adapting the Edge-Path bundling approach proposed in [37, 38], as described in Algorithm 4. More precisely, for each non-skeletal edge uv we consider the path in the skeleton between u and v that minimizes the sum of the Euclidean lengths of its edges. Let π be such a path: we bundle uv to π if the distortion is not above a predefined threshold $\delta > 1$; otherwise, uv is added to Γ as a straight-line unbundled segment. In our implementation we use the threshold $\delta = t$, for the same value of t applied in the **Filter** step. The rationale behind adapting the EPB approach to our scenario is that it avoids independent edge ambiguities by design. Indeed, an edge uv can only be bundled against a path of the skeleton connecting u to v . Moreover, we ensure that edges can only be bundled against a path if their distortion is not above the selected distortion threshold δ . Finally, bundling always has the goal of saving ink, thus improving on the ink ratio score.

■ **Algorithm 4** Spanner-Edge-Path bundling algorithm [38].

Input: Graph $G = (V, E)$, skeleton $G' = (V, E')$, drawing Γ of G' , maximum distortion $\delta > 1$.

Output: Control points for the edges in $E \setminus E'$.

```

for  $uv \in E \setminus E'$  do
   $p \leftarrow \text{shortestPath}(G', \Gamma, u, v)$ 
   $p.length \leftarrow 0$ 
  for  $xy \in p$  do
     $\text{/* } d(x, y) \text{ is the Euclidean distance between } x \text{ and } y \text{ in } \Gamma \text{ */}$ 
     $p.length \leftarrow p.length + d(x, y)$ 
  if  $p.length \leq \delta * d(u, v)$  then
     $\text{/* bundle edge } uv \text{ against } p \text{ */}$ 
     $\text{controlPoints}(uv) \leftarrow p.getVertexCoordinates()$ 
  else
     $\text{/* draw edge } uv \text{ unbundled */}$ 
     $\text{controlPoints}(uv) \leftarrow (\Gamma(u), \Gamma(v))$ 
return  $\text{controlPoints}$ 

```

4 Experimental Study

In this section, we present an experimental analysis that compares the bundling performances of four algorithms. We first recall these algorithms (Section 4.1); we then discuss our experimental hypotheses (Section 4.2); next we describe the data set and give implementation details (Section 4.3); finally, we present a statistical analysis of the results (Section 4.4).

4.1 Competing Algorithms

In the experiment, we consider the following four algorithms:

- **EB-FDB**: This algorithm is the first instantiation of the FDB framework described in Section 3.2.1, where the filtering step uses edge betweenness.
- **NEB-FDB**: This algorithm is the second instantiation of the FDB framework described in Section 3.2.1, where the filtering step uses neighboring edge betweenness.
- **ConfluentDrawing**: This algorithm creates a confluent-like drawing [2, 41] using the implementation of Zheng et al. [41].
- **PP-Bundling**: This algorithm computes a drawing of the input graph using standard stress majorization [11, 40], and then it performs the Spanner Edge-Path bundling approach [38] (SEPB) on the computed drawing. It can be regarded as a representative of the common post-processing approach where the visual clutter is reduced by bundling edges on a given drawing. Hence, the graph spanner which is used for the bundling is computed on the Euclidean distances of a given drawing.

4.2 Experimental Hypotheses

We conceptually cluster our hypotheses into two groups. The first group is about the relationship between the FDB framework and the state-of-the-art bundling methods **ConfluentDrawing** and **PP-Bundling**, whereas the second group is about the direct comparison of the two instantiations **EB-FDB** and **NEB-FDB** of the framework.

The following hypotheses belong to the first group:

- **H1.1**: Algorithms **EB-FDB** and **NEB-FDB** perform better than the **PP-Bundling** algorithm in terms of ink ratio, distortion, and ambiguity.
- **H1.2**: Algorithms **EB-FDB** and **NEB-FDB** perform better than the **ConfluentDrawing** algorithm in terms of ink ratio, distortion, and ambiguity.

The following hypotheses belong to the second group:

- **H2.1**: Increasing the value of t in the t -spanner reduces the ink ratio of the bundled drawings computed with either the **EB-FDB** algorithm or the **NEB-FDB** algorithm.
- **H2.2**: For a given value of t , the **Filter** step of the **NEB-FDB** algorithm yields sparser t -spanners than those computed by the **Filter** step of the **EB-FDB** algorithm.
- **H2.3**: The **NEB-FDB** algorithm computes bundled drawings with higher distortion but lower ink ratio than those computed by the **EB-FDB** algorithm.

Rationale. Concerning **H1.1**, we recall that in the **PP-Bundling** algorithm the position of the nodes is computed by a standard stress majorization method, agnostic of the bundling step, which is in fact applied as a post-processing. On the other hand, the **Filter** step of the **EB-FDB** and the **NEB-FDB** algorithms is designed such that the edges of the path against which the bundling is executed are selected based on the graph topology and placed on a central position in the layout. This, we believe, positively impacts both ink ratio and distortion, because the long edges tend to be bundled against sets of edges that are central in the layout, avoiding long curves that pass through peripheral parts of the layout. This also potentially leads to layouts with less ambiguity compared to **PP-Bundling**: routing long edges towards the center likely reduces edge crossings and, consequently, it could reduce the number of sharp crossing angles.

Regarding **H1.2**, the **ConfluentDrawing** algorithm introduces dummy nodes and edges to the original graph. While these guides help construct a confluent drawing, we believe they might negatively impact distortion and ink ratio. Dummy elements are not part of the

actual data, and curves routed around them might become unnecessarily long compared to those computed without them. Also, by strategically placing heavier edges centrally, FDB reduces the chances of them intersecting with many other edges in peripheral areas and this could lead to less visual clutter and potentially an ambiguity reduction.

As for **H2.1**, higher values of t give rise to sparser skeletons. Therefore, by increasing t the number of edges against which the non-skeletal edges are bundled decreases, potentially leading to a smaller number of occupied pixels in the bundled layout.

Hypothesis **H2.2** originates from the observation that the **Filter** step of NEB-FDB weights the edges by only considering shortest paths between pairs of adjacent vertices, which could give rise to a smaller number of “heavily weighted” edges to be included in the skeleton than those computed by EB-FDB. We note, however, that while we expect this hypothesis to hold in typical real-world data sets, it is easy to find small counterexamples, where this is not the case. The same intuition is behind **H2.3**, as a skeleton with fewer edges gives rise to bundled drawings that use fewer pixels; however, this may have a negative impact on the distortion.

4.3 Dataset and Implementation Details

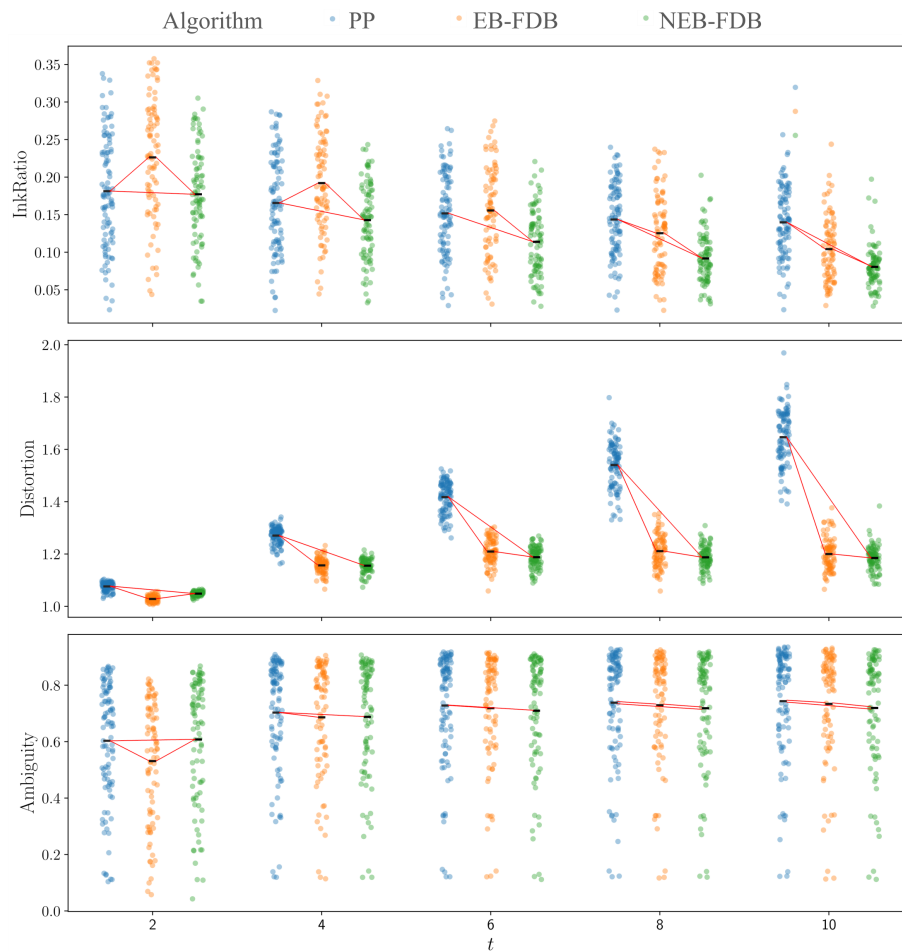
Our benchmark consists of graphs having a social network structure [21] and generated using the well-established Stochastic Block Model (SBM) [15]. The dataset is divided into four graph-size ranges: 20-50 nodes, 50-100 nodes, 100-150 nodes, and 150-200 nodes. Within each range, graphs are further categorized into five density classes based on the average number of edges per node (edge density). These classes range from sparse (2-4 edges per node) to dense (greater than 10 edges per node). We generated five graphs for each combination of graph-size range and density class, resulting in a total of 100 graphs.

The implementation of the **Filter** step of EB-FDB algorithm leverages the well-known NetworkX library [13] to compute betweenness centrality (Equation 4), which is based on the algorithm by Brandes [3]. The neighboring edge betweenness of Equation (5) is implemented in Python. The **Draw** step for both the EB-FDB algorithm and the NEB-FDB algorithm uses the Stochastic Gradient Descent method to optimize the stress of a straight line drawing of the skeleton [40]. Finally, the **Bundle** step uses the implementation of the Faster Edge-Path Bundling algorithm by Wallinger et al. [38]. The main code is in Python with integrated C++ methods. Specifically, the Stochastic Gradient Descent and the NetworkX library make use of more efficient C++ routines. Due to the different implementation languages we did not focus on performance comparisons. Still, we measured wallclock runtime and report it for EB-FDB (mean = 0.36s, max=3.30s) and NEB-FDB (mean = 23.04s, max=335.61s). We ran our experiments on a compute cluster with an AMD EPYC 7402, 2.80GHz 24-core CPU but restricted the computation to one core and 32GB of RAM, essentially simulating a powerful desktop PC.

The data set and the code used for the experimental analysis are available at the following OSF repository: <https://osf.io/g4xqw/>.

4.4 Analysis of the Results

We analyzed ink ratio, distortion and ambiguity separately. In general, a Shapiro-Wilk test revealed that no condition-metric pair was normally distributed. Hence, we applied the Wilcoxon signed-rank for comparing two conditions or the Friedmann test for three or more conditions to test for statistical significance ($\alpha = 0.01$). If statistical significance was found, we applied Bonferroni-corrected pairwise t-tests. We report effect size with the the Common Language Effect Size (CLES), i.e., the probability that the score of a random sample from one distribution is greater than the one of a random sample from some other distribution.

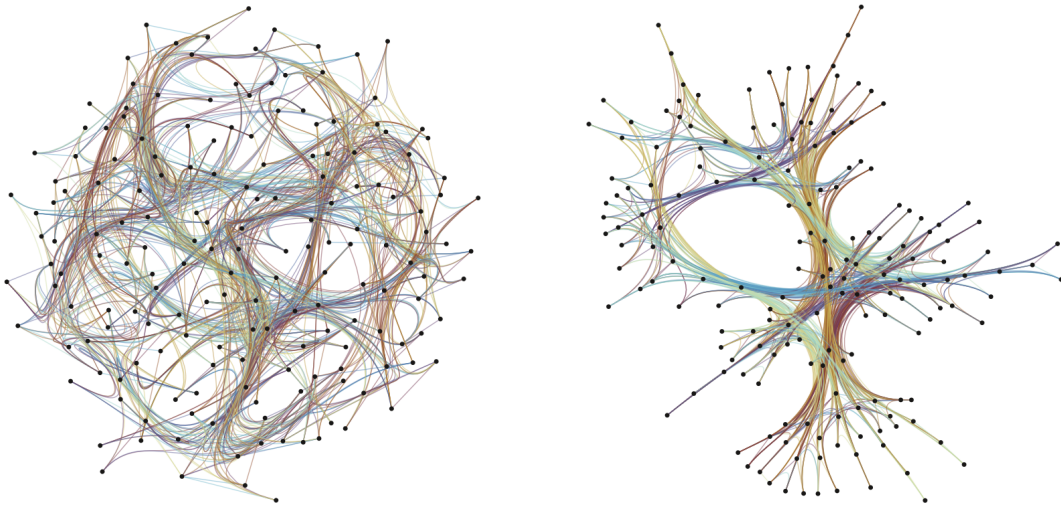


■ **Figure 3** Strip plots of the experimental data about **H1.1**. The x -axis reports different t values. The y -axis reports: (top) ink ratio; (middle) distortion, and (bottom) ambiguity. Red segments connecting the mean values indicate statistically significant differences.

4.4.1 H1.1

Recall that the hypothesis is about EB-FDB and NEB-FDB outperforming PP-Bundling in ink ratio, distortion, and ambiguity. Our results are as follows (see also Figure 3). Note that the reported effect size is stated in terms of aggregated t , while individual levels can potentially have different effect sizes.

Ink Ratio. When aggregating all instances over t we partially confirm the hypothesis. NEB-FDB (mean value = 1.12) has lower ink ratio than PP-Bundling (mean value = 1.16) (CLES = 0.67, $p < 0.01$). However, there is no statistically significant difference between EB-FDB (mean value = 1.16) and PP-Bundling. If we compare different levels of t , then there is no significant differences for $t = 6$ between EB-FDB and PP-Bundling. For all other levels of t we have statistically significant results between the algorithms. Interestingly, with increasing t EB-FDB exhibits higher ink ratio reduction than PP-Bundling.



■ **Figure 4** Instance with $|V| = 156$ and $|E| = 815$ from the benchmark dataset visualized with PP-Bundling (left) and NEB-FDB (right) with $t = 6$. The ink ratio is 0.21 for PP-Bundling and 0.13 for NEB-FDB; the mean distortion is 1.52 for PP-Bundling and 1.18 for NEB-FDB; the ambiguity is 0.91 for PP-Bundling and 0.90 for NEB-FDB.

Distortion. When aggregating t the results for distortion indicate that there is statistically significant difference between PP-Bundling (mean value = 1.39) compared to EB-FDB (mean value = 1.14) (CLES = 0.84, $p < 0.01$) and NEB-FDB (mean value = 1.13) (CLES = 0.85, $p < 0.01$). Furthermore, if we analyse individual values of t we find that there is positive correlation between t and distortion for PP-Bundling. In contrast, the mean distortion does not exhibit this behavior for EB-FDB or NEB-FDB.

Ambiguity. Again, if we aggregate instances over t the results indicate significant differences between PP-Bundling (mean value = 0.70) compared to EB-FDB (mean value = 0.63) (CLES = 0.58, $p < 0.01$) and NEB-FDB (mean value = 0.64) (CLES = 0.57, $p < 0.01$). Correlation analysis shows that for all three algorithms ambiguity correlates with increasing t .

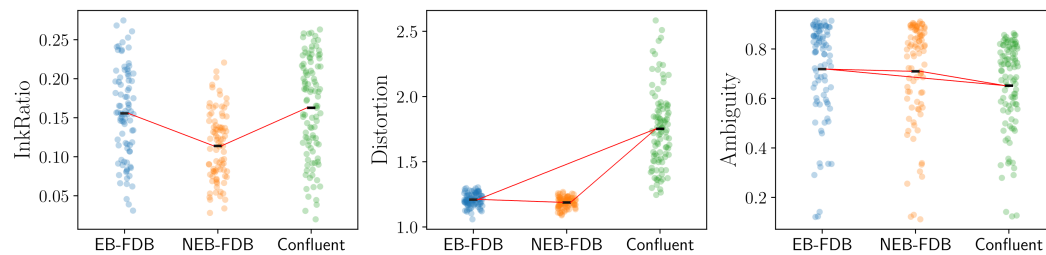
Summary. For distortion and ambiguity our results support **H1.1**. However, for ink ratio we found support only for values of $t \neq 6$, i.e., for EB-FDB the ink ratio is higher for lower values of t but decreases faster with increasing t compared to the other algorithms. Also, with increasing t the distortion for EB-FDB and NEB-FDB remains somewhat constant while PP-Bundling increases.

Figure 4 shows two drawings of the same graph. The one to the left is computed by PP-Bundling while the one to the right is computed by NEB-FDB; in both cases $t = 6$.

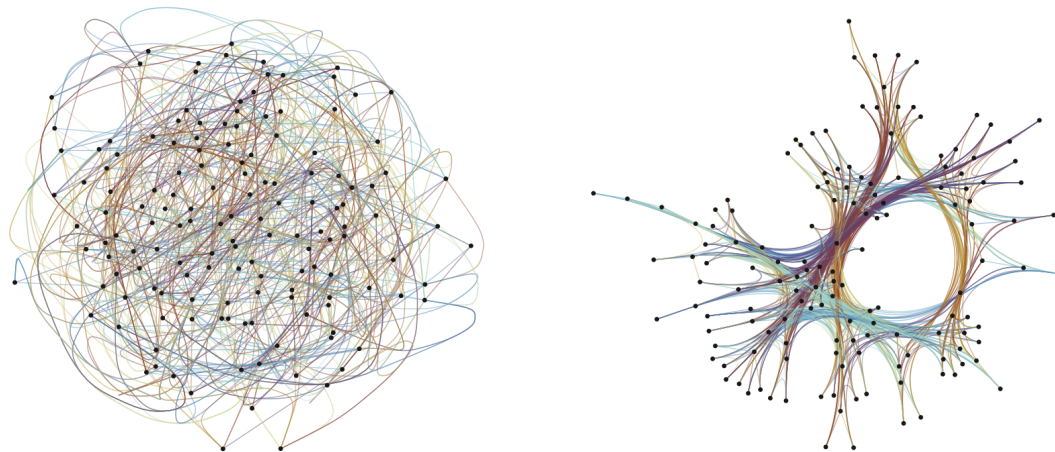
4.4.2 H1.2

Recall that the hypothesis is about EB-FDB and NEB-FDB outperforming ConfluentDrawing in ink ratio, distortion, and ambiguity. Our results are as follows (see also Figure 5).

First, we decided on a value for t to have a fair comparison between EB-FDB and NEB-FDB against ConfluentDrawing. The choice for $t = 6$ was guided by the overall balanced performance regarding all three metrics.



■ **Figure 5** Strip plots of the experimental data about **H1.2**. The x -axis reports different algorithms.



■ **Figure 6** Instance with $|V| = 147$ and $|E| = 952$ from the benchmark dataset visualized with **ConfluentDrawing** (left) and **NEB-FDB** (right) with $t = 6$. The ink ratio is 0.23 for **ConfluentDrawing** and 0.09 for **NEB-FDB**; the mean distortion is 1.74 for **ConfluentDrawing** and 1.17 for **NEB-FDB**; the ambiguity is 0.82 for **ConfluentDrawing** and 0.89 for **NEB-FDB**.

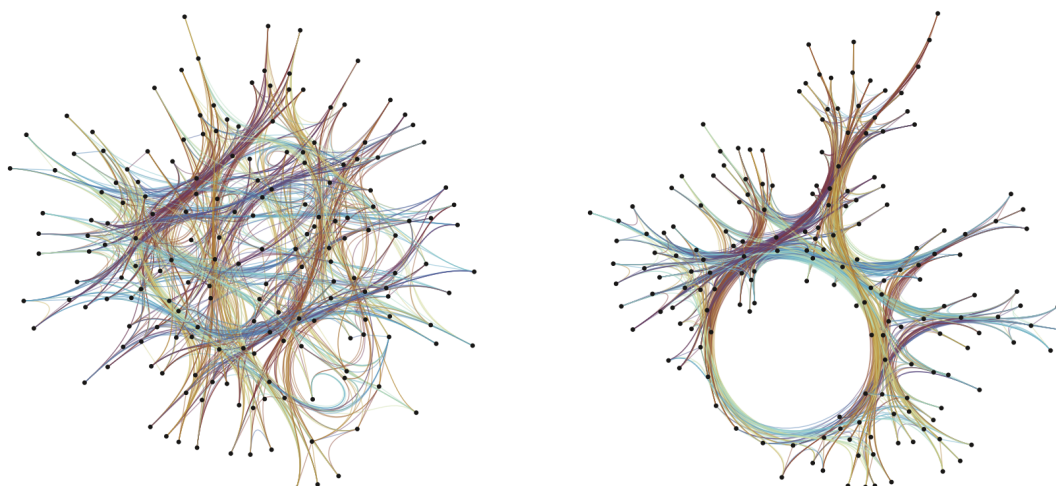
Ink Ratio. Here, the ink ratio shows a significant difference for **ConfluentDrawing** (mean value = 0.16) when compared against **NEB-FDB** (mean value = 0.12) (CLES = 0.72, $p < 0.01$) but not for **EB-FDB** (mean value = 0.15).

Distortion. The comparison of the distortion values revealed that both **EB-FDB** (mean value = 1.17) (CLES > 0.99, $p < 0.01$) and **NEB-FDB** (mean value = 1.16) (CLES > 0.99, $p < 0.01$) have drastically less distortion than **ConfluentDrawing** (mean value = 1.75).

Ambiguity. Lastly, we found that **ConfluentDrawing** (mean value = 0.65) has lower ambiguity than **EB-FDB** (mean value = 0.67) (CLES = 0.58, $p < 0.01$) and **NEB-FDB** (mean value = 0.66) (CLES = 0.57, $p < 0.01$).

Summary. For ink ratio our results partially support **H1.2** and fully support it for distortion. **H1.2** is also supported for **NEB-FDB** with respect to mean distortion for all values of t that were considered. As for ambiguity, **ConfluentDrawing** outperforms both **NEB-FDB** and **EB-FDB**.

Figure 6 shows two drawings of the same graph. The one to the left is computed by **ConfluentDrawing** while the one to the right is computed by **NEB-FDB** with $t = 6$.



■ **Figure 7** Instance with $|V| = 182$ and $|E| = 1268$ from the benchmark dataset visualized both with NEB-FDB, and with $t = 6$ (left) and $t = 10$ (right). The ink ratio is 0.16 in the drawing to the left and 0.09 in the drawing to the right.

4.4.3 H2.1

Recall that the hypothesis is that higher values of t yield lower ink ratio both with EB-FDB and with NEB-FDB.

Analyzing the ink ratio for different values of t individually for EB-FDB and NEB-FDB revealed a similar behavior. For both algorithms the ink ratio has no statistically significant differences between $t = 2$ and $t = 4$ and between $t = 8$ and $t = 10$. All other values had significant difference with effect sizes $\text{CLES} < 0.4$. This indicates that for $t \leq 4$ the spanner is dense and subsequently not many edges can be bundled. Similarly, for $t \geq 8$ the spanner is sparse and probably close to a spanning tree. Hence, the maximal amount of edges is bundled but there is no choice of path left when computing the bundling.

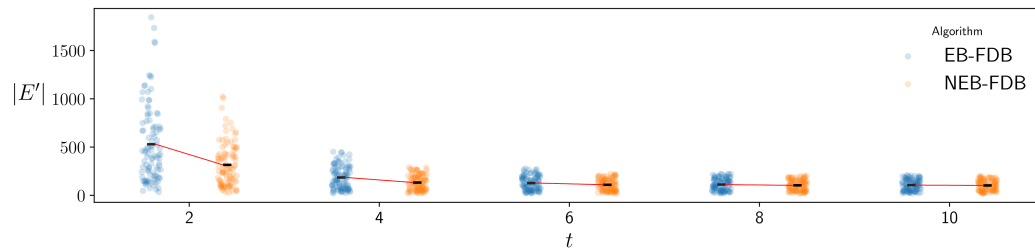
Summary. Our results support **H2.1** except when comparing $t = 2$ with $t = 4$ and when comparing $t = 8$ with $t = 10$. Figure 7 shows two drawings of the same graph computed by NEB-FDB for different t values.

4.4.4 H2.2

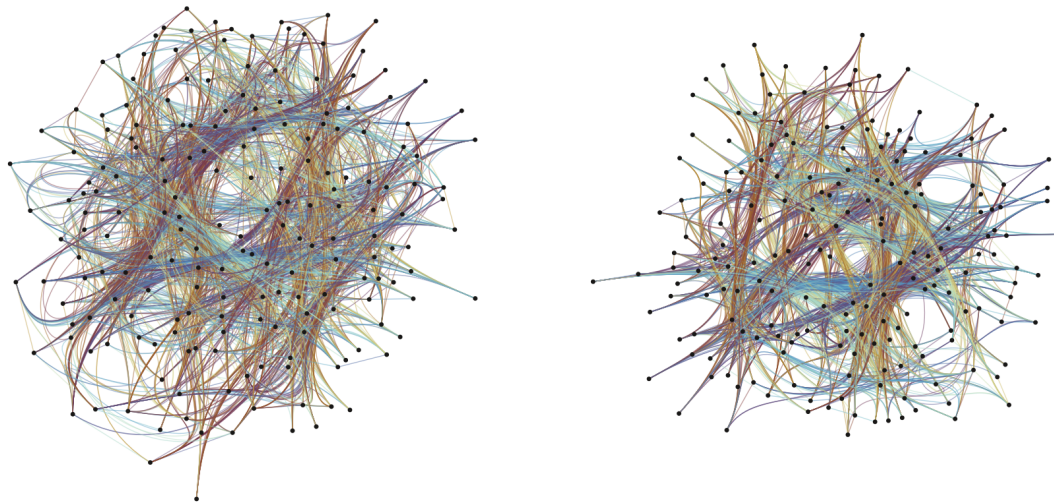
Recall that the hypothesis is about NEB-FDB using sparser t -spanners than EB-FDB. Refer to Figure 8 for our results.

Here we counted the edges in the spanner after computing it in the filter step of EB-FDB and NEB-FDB. The statistical analysis showed that instances computed with NEB-FDB have sparser spanners than EB-FDB ($\text{CLES} = 0.44$, $p < 0.01$). Furthermore, the effect size becomes stronger if we exclude higher values of t , i.e., all instances that are probably for both a spanning tree already.

Summary. Our results support **H2.2**.



■ **Figure 8** Strip plots presenting the impact of the parameter t on the number of edges in the spanners for EB-FDB and NEB-FDB.



■ **Figure 9** Instance with $|V| = 193$ and $|E| = 2614$ from the benchmark dataset visualized with EB-FDB (left) and NEB-FDB (right) with $t = 6$. The ink ratio is 0.24 in the drawing to the left and 0.21 in the drawing to the right; the mean distortion is 1.28 in the drawing to the left and 1.26 in the drawing to the right.

4.4.5 H2.3

Recall that the hypothesis is about NEB-FDB computing drawings with higher distortion but lower ink ratio than EB-FDB. Refer to Figure 3 for our results.

In direct comparison between EB-FDB and NEB-FDB we analyzed that both, ink ratio (CLES = 0.35, $p < 0.01$) and distortion (CLES = 0.48, $p < 0.01$) are higher for EB-FDB. Even though the result for distortion is statistically significant, the effect size only shows minor impact of the algorithm in the **Filter** step.

Summary. Our results partially support **H2.3**. Namely, NEB-FDB outperforms EB-FDB both in distortion and in ink ratio for the same values of t . Recall that, for a fixed value of t , the spanner computed for NEB-FDB is generally sparser than the spanner for EB-FDB. In contrast to our initial intuition, EB-FDB gives rise to higher distortion than NEB-FDB, since the curves representing the bundled edges are longer.

Figure 9 shows two drawings of the same graph. The one to the left is computed by EB-FDB while the one to the right is computed by NEB-FDB. In both cases with $t = 6$.

5 Final Remarks and Future Work

We conclude the paper addressing the lessons that we learned, pointing out to the limitations of our study, and discussing future research directions.

Lessons Learned. This study suggests that bundled drawing quality can be significantly enhanced by designing algorithms that directly optimize core quality metrics during the layout computation process, rather than relying solely on post-processing techniques. To this aim, we can use an algorithmic framework consisting of three main steps: **Filter**, **Draw**, and **Bundle**. The **Filter** step sparsifies the graph by identifying relevant edges, the **Draw** step computes a layout of the sparsified graph, and the **Bundle** step reinserts the missing edges into the layout. We showed that the computation of the sparsified graph can have significant impact on relevant bundling qualities. In particular, we weighted the edges of the graph and then applied a t -spanner. Our experiments have shown that both the edge weighting method and the choice of the parameter t in the t -spanner can influence the ink ratio, distortion, and ambiguity of the final drawing. More precisely, a value of t between 4 and 8, and in particular a value of $t = 6$, together with a weighting function based on neighboring edge betweenness, i.e., **NEB-FDB**, seems to be the most effective instantiation of the **FDB** framework in practice.

Limitations. We performed an experimental study that compares the **EB-FDB**, the **NEB-FDB**, the **PP-Algorithm**, and the **ConfluentDrawing** bundling algorithms on a data set consisting of 100 graphs having different numbers of nodes and densities. Although our results support our hypotheses in most cases, experiments involving additional datasets should be conducted. For example, one could consider a higher number of networks, having different sizes, and that are generated with algorithms different from **SBM**. Also, it would be interesting to evaluate our approach with real-world networks. At this point, the quality comparisons of the different bundled drawings are performed purely on the three quantitative measures of ink ratio, distortion, and ambiguity. Similarly, other drawing quality metrics should be tested to provide new perspectives on bundling performance.

Future Work. We studied two different algorithmic instances of the **FDB** framework, namely the **EB-FDB** algorithm and the **NEB-FDB** algorithm. We believe that the potential of the different steps of the **FDB** framework can be further investigated by considering different methods for each of the steps in the framework. It would be interesting to: (i) test different sparsification techniques such as, for example, the Simmelian backbone [28]; (ii) adopt different layout strategies for the drawing of the sparsified graph such as, for example, faithful force-directed methods [25]; (iii) test approaches that consider crossing angles in the **Bundle** step of which examples exist in the literature [5, 8]. Currently, we investigated the problem from a practical perspective, however, a complementary theoretical investigation into combining layout and bundling algorithms would be beneficial for further understanding. Also, it would be interesting to compare the drawings produced by **EB-FDB**, **NEB-FDB**, **PP-Algorithm**, and **ConfluentDrawing** in terms of aesthetic criteria such as, for example, angular resolution, crossing angle, and number of crossings. Finally, it is an open question to evaluate whether the optimized parameters are good proxies for graph readability.


References

- 1 Ingo Althöfer, Gautam Das, David P. Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discret. Comput. Geom.*, 9:81–100, 1993. doi:10.1007/BF02189308.
- 2 Benjamin Bach, Nathalie Henry Riche, Christophe Hurter, Kim Marriott, and Tim Dwyer. Towards unambiguous edge bundling: Investigating confluent drawings for network visualization. *IEEE Trans. Vis. Comput. Graph.*, 23(1):541–550, 2017. doi:10.1109/TVCG.2016.2598958.
- 3 Ulrik Brandes. A faster algorithm for betweenness centrality. *J. Math. Sociol.*, 25(2):163–177, 2001. doi:10.1080/0022250X.2001.9990249.
- 4 Weiwei Cui, Hong Zhou, Huamin Qu, Pak Chung Wong, and Xiaoming Li. Geometry-based edge clustering for graph visualization. *IEEE Trans. Vis. Comput. Graph.*, 14(6):1277–1284, 2008. doi:10.1109/TVCG.2008.135.
- 5 Emilio Di Giacomo, Walter Didimo, Luca Grilli, Giuseppe Liotta, and Salvatore Agostino Romeo. Heuristics for the maximum 2-layer RAC subgraph problem. *Comput. J.*, 58(5):1085–1098, 2015. doi:10.1093/COMJNL/BXU017.
- 6 Matthew Dickerson, David Eppstein, Michael T. Goodrich, and Jeremy Y. Meng. Confluent drawings: Visualizing non-planar diagrams in a planar way. *J. Graph Algorithms Appl.*, 9(1):31–52, 2005. doi:10.7155/jgaa.00099.
- 7 Matthew Dickerson, David Eppstein, Michael T. Goodrich, and Jeremy Yu Meng. Confluent drawings: Visualizing non-planar diagrams in a planar way. In Giuseppe Liotta, editor, *Graph Drawing (GD’03)*, volume 2912 of *LNCS*, pages 1–12. Springer, 2003. doi:10.1007/978-3-540-24595-7_1.
- 8 Walter Didimo, Giuseppe Liotta, and Salvatore Agostino Romeo. A graph drawing application to web site traffic analysis. *J. Graph Algorithms Appl.*, 15(2):229–251, 2011. doi:10.7155/JGAA.00224.
- 9 Ozan Ersoy, Christophe Hurter, Fernando Vieira Paulovich, Gabriel Cantareiro, and Alexandru C. Telea. Skeleton-based edge bundling for graph visualization. *IEEE Trans. Vis. Comput. Graph.*, 17(12):2364–2373, 2011. doi:10.1109/TVCG.2011.233.
- 10 Emden R. Gansner, Yifan Hu, Stephen C. North, and Carlos Eduardo Scheidegger. Multilevel agglomerative edge bundling for visualizing large graphs. In *Pacific Visualization Symposium (PacificVis’11)*, pages 187–194. IEEE, 2011. doi:10.1109/PACIFICVIS.2011.5742389.
- 11 Emden R. Gansner, Yehuda Koren, and Stephen C. North. Graph drawing by stress majorization. In János Pach, editor, *Graph Drawing (GD’04)*, volume 3383 of *LNCS*, pages 239–250. Springer, 2004. doi:10.1007/978-3-540-31843-9_25.
- 12 Michelle Girvan and Mark E. J. Newman. Community structure in social and biological networks. *Proc. Natl. Acad. Sci.*, 99(12):7821–7826, 2002. doi:10.1073/pnas.122653799.
- 13 Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gäel Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Python in Science Conference (SciPy’08)*, pages 11–15, 2008.
- 14 Michael Hirsch, Henk Meijer, and David Rappaport. Biclique edge cover graphs and confluent drawings. In *Graph Drawing (GD’06)*, volume 4372 of *LNCS*, pages 405–416. Springer, 2007. doi:10.1007/978-3-540-70904-6_39.
- 15 Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Soc. Netw.*, 5(2):109–137, 1983.
- 16 Danny Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Trans. Vis. Comput. Graph.*, 12(5):741–748, 2006. doi:10.1109/TVCG.2006.147.
- 17 Danny Holten and Jarke J. van Wijk. Force-directed edge bundling for graph visualization. *Comput. Graph. Forum*, 28(3):983–990, 2009. doi:10.1111/j.1467-8659.2009.01450.x.
- 18 Weidong Huang, Peter Eades, and Seok-Hee Hong. A graph reading behavior: Geodesic-path tendency. In *Pacific Visualization Symposium (PacificVis’09)*, pages 137–144. IEEE, 2009. doi:10.1109/PACIFICVIS.2009.4906848.

- 19 Christophe Hurter, Ozan Ersoy, and Alexandru Telea. Graph bundling by kernel density estimation. *Comput. Graph. Forum*, 31(3):865–874, 2012. doi:10.1111/j.1467-8659.2012.03079.x.
- 20 Antoine Lambert, Romain Bourqui, and David Auber. Winding Roads: Routing edges into bundles. *Comput. Graph. Forum*, 29(3):853–862, 2010. doi:10.1111/j.1467-8659.2009.01700.x.
- 21 Clement Lee and Darren J. Wilkinson. A review of stochastic block models and extensions for graph clustering. *Appl. Netw. Sci.*, 4(1):122, 2019. doi:10.1007/S41109-019-0232-2.
- 22 Antoine Lhuillier, Christophe Hurter, and Alexandru C. Telea. FFTEB: edge bundling of huge graphs by the fast fourier transform. In *Pacific Visualization Symposium (PacificVis'17)*, pages 190–199. IEEE, 2017. doi:10.1109/PACIFICVIS.2017.8031594.
- 23 Antoine Lhuillier, Christophe Hurter, and Alexandru C. Telea. State of the art in edge and trail bundling techniques. *Comput. Graph. Forum*, 36(3):619–645, 2017. doi:10.1111/cgf.13213.
- 24 Sheng-Jie Luo, Chun-Liang Liu, Bing-Yu Chen, and Kwan-Liu Ma. Ambiguity-free edge-bundling for interactive graph visualization. *IEEE Trans. Vis. Comput. Graph.*, 18(5):810–821, 2012. doi:10.1109/TVCG.2011.104.
- 25 Amyra Meidiana, Seok-Hee Hong, and Peter Eades. Shape-faithful graph drawings. In Patrizio Angelini and Reinhard von Hanxleden, editors, *Graph Drawing and Network Visualization (GD'22)*, volume 13764 of *LNCS*, pages 93–108. Springer, 2022. doi:10.1007/978-3-031-22203-0_8.
- 26 Quan Hoang Nguyen, Peter Eades, and Seok-Hee Hong. On the faithfulness of graph visualizations. In *Pacific Visualization Symposium (PacificVis'13)*, pages 209–216. IEEE, 2013. doi:10.1109/PacificVis.2013.6596147.
- 27 Quan Hoang Nguyen, Seok-Hee Hong, and Peter Eades. TGI-EB: A new framework for edge bundling integrating topology, geometry and importance. In *Graph Drawing (GD'11)*, volume 7034 of *LNCS*, pages 123–135. Springer, 2011. doi:10.1007/978-3-642-25878-7_13.
- 28 Bobo Nick, Conrad Lee, Pádraig Cunningham, and Ulrik Brandes. Simmelian backbones: amplifying hidden homophily in facebook networks. In Jon G. Rokne and Christos Faloutsos, editors, *Advances in Social Networks Analysis and Mining (ASONAM'13)*, pages 525–532. ACM, 2013. doi:10.1145/2492517.2492569.
- 29 Sergey Pupyrev, Lev Nachmanson, Sergey Bereg, and Alexander E. Holroyd. Edge routing with ordered bundles. In Marc J. van Kreveld and Bettina Speckmann, editors, *Graph Drawing (GD'11)*, volume 7034 of *LNCS*, pages 136–147. Springer, 2011. doi:10.1007/978-3-642-25878-7_14.
- 30 Sergey Pupyrev, Lev Nachmanson, Sergey Bereg, and Alexander E. Holroyd. Edge routing with ordered bundles. *Comput. Geom.*, 52:18–33, 2016. doi:10.1016/j.comgeo.2015.10.005.
- 31 David Selassie, Brandon Heller, and Jeffrey Heer. Divided edge bundling for directional network data. *IEEE Trans. Vis. Comput. Graph.*, 17(12):2354–2363, 2011. doi:10.1109/TVCG.2011.190.
- 32 Alexandru Telea and Ozan Ersoy. Image-based edge bundles: Simplified visualization of large graphs. *Comput. Graph. Forum*, 29(3):843–852, 2010. doi:10.1111/j.1467-8659.2009.01680.x.
- 33 Naoko Toeda, Rina Nakazawa, Takayuki Itoh, Takafumi Saito, and Daniel Archambault. Convergent drawing for mutually connected directed graphs. *J. Vis. Lang. Comput.*, 43:83–90, 2017. doi:10.1016/j.jvlc.2017.09.004.
- 34 Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 1983.
- 35 Matthew van der Zwan, Valeriu Codreanu, and Alexandru C. Telea. CUBu: universal real-time bundling for large graphs. *IEEE Trans. Vis. Comput. Graph.*, 22(12):2550–2563, 2016. doi:10.1109/TVCG.2016.2515611.
- 36 Markus Wallinger. FDB Framework. Software (visited on 2024-09-13). doi:10.17605/OSF.IO/G4XQW.

- 37 Markus Wallinger, Daniel Archambault, David Auber, Martin Nöllenburg, and Jaakko Peltonen. Edge-path bundling: A less ambiguous edge bundling approach. *IEEE Trans. Vis. Comput. Graph.*, 28(1):313–323, 2022. doi:10.1109/TVCG.2021.3114795.
- 38 Markus Wallinger, Daniel Archambault, David Auber, Martin Nöllenburg, and Jaakko Peltonen. Faster edge-path bundling through graph spanners. *Comput. Graph. Forum*, 42(6), 2023. doi:10.1111/CGF.14789.
- 39 Jieting Wu, Lina Yu, and Hongfeng Yu. Texture-based edge bundling: A web-based approach for interactively visualizing large graphs. In *Big Data (BigData'15)*, pages 2501–2508. IEEE, 2015. doi:10.1109/BigData.2015.7364046.
- 40 Jonathan X. Zheng, Samraat Pawar, and Dan F. M. Goodman. Graph drawing by stochastic gradient descent. *IEEE Trans. Vis. Comput. Graph.*, 25(9):2738–2748, 2019. doi:10.1109/TVCG.2018.2859997.
- 41 Jonathan X. Zheng, Samraat Pawar, and Dan M. Goodman. Further towards unambiguous edge bundling: Investigating power-confluent drawings for network visualization. *IEEE Trans. Vis. Comput. Graph.*, 27(03):2244–2249, 2021. doi:10.1109/TVCG.2019.2944619.

GraphTrials: Visual Proofs of Graph Properties

Henry Förster ✉ 

Department of Computer Science,
University of Tübingen, Germany

Tim Dwyer ✉ 

Faculty of Information Technology,
Monash University, Australia

Seok-Hee Hong ✉ 

School of Computer Science,
The University of Sydney, Australia

Giuseppe Liotta ✉ 

Department of Engineering,
University of Perugia, Italy

Fabrizio Montecchiani ✉ 

Department of Engineering,
University of Perugia, Italy

Falk Schreiber ✉ 

Department of Computer Science,
University of Konstanz, Germany

Felix Klesen ✉ 

Institute of Computer Science,
University of Würzburg, Germany

Peter Eades ✉ 

School of Computer Science,
The University of Sydney, Australia

Stephen G. Kobourov ✉ 

Department of Computer Science,
University of Arizona, Tucson, AZ, USA

Kazuo Misue ✉ 

Department of Computer Science,
University of Tsukuba, Japan

Alexander Pastukhov ✉ 

Department of Psychology,
University of Bamberg, Germany

Abstract

Graph and network visualization supports exploration, analysis and communication of relational data arising in many domains: from biological and social networks, to transportation and powergrid systems. With the arrival of AI-based question-answering tools, issues of trustworthiness and explainability of generated answers motivate a greater role for visualization. In the context of graphs, we see the need for visualizations that can convince a critical audience that an assertion about the graph under analysis is valid. The requirements for such representations that convey precisely one specific graph property are quite different from standard network visualization criteria which optimize general aesthetics and readability.

In this paper, we aim to provide a comprehensive introduction to visual proofs of graph properties and a foundation for further research in the area. We present a framework that defines what it means to visually prove a graph property. In the process, we introduce the notion of a visual certificate, that is, a specialized faithful graph visualization that leverages the viewer's perception, in particular, pre-attentive processing (e. g. via pop-out effects), to verify a given assertion about the represented graph. We also discuss the relationships between visual complexity, cognitive load and complexity theory, and propose a classification based on visual proof complexity. Finally, we provide examples of visual certificates for problems in different visual proof complexity classes.

2012 ACM Subject Classification Human-centered computing → Visualization theory, concepts and paradigms; Human-centered computing → Graph drawings; Human-centered computing → Information visualization

Keywords and phrases Graph Visualization, Theory of Visualization, Visual Proof

Digital Object Identifier 10.4230/LIPIcs.GD.2024.16

Related Version *Full Version*: <https://arxiv.org/abs/2409.02907>

Acknowledgements This work was initiated at Dagstuhl seminar 23051 “Perception in Network Visualization”. We thank the organizers for making this fruitful interdisciplinary exchange possible and all participants for interesting discussions and insights during the seminar week.



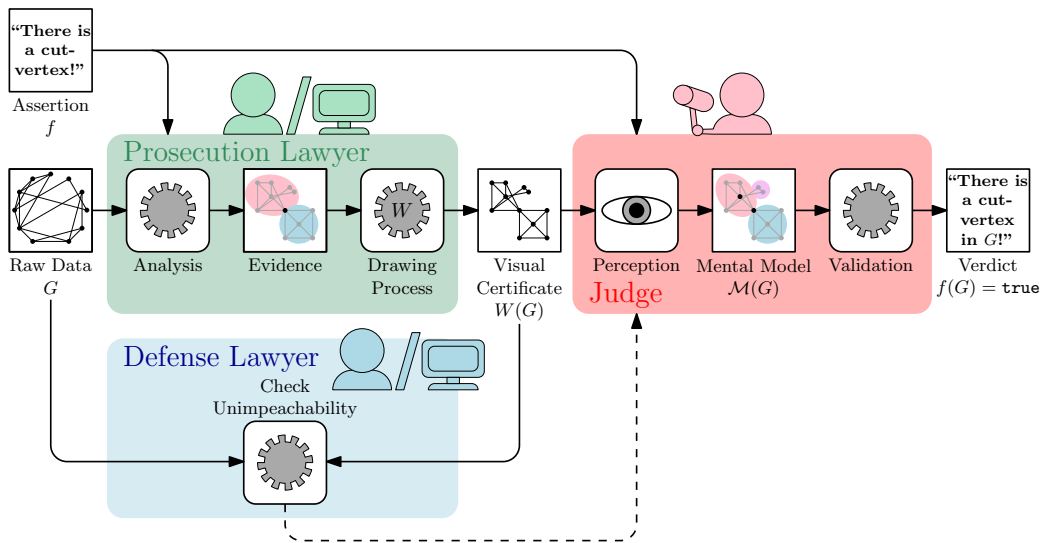
© Henry Förster, Felix Klesen, Tim Dwyer, Peter Eades, Seok-Hee Hong, Stephen G. Kobourov, Giuseppe Liotta, Kazuo Misue, Fabrizio Montecchiani, Alexander Pastukhov, and Falk Schreiber; licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 16; pp. 16:1–16:18



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** Our model GRAPHTRIALS identifies key processes for visually proving an assertion about a given graph in an adversarial setting. The *prosecution lawyer*, i. e., a software or a human (assisted by software), intends to highlight evidence for a graph being guilty of satisfying an assertion using a visual certificate drawing. To convince the *judge*, i. e., the human audience of the drawing, the visual certificate guides the judge’s perception to form a mental model which makes the assertion easy to validate. The visual certificate must be unimpeachable as a *defense lawyer* (software or human adversary) checks for reasons to doubt the certificate’s validity to influence the judge’s verdict.

1 Introduction

While state-of-the-art graph and network visualization techniques do a reasonable job of untangling graphs to convey meaning and support free-form exploration, there are certain application scenarios where these algorithms fall short. Namely, we focus on applications where it is necessary to *convince* a (possibly non-expert) audience that a particular graph has some structural property. We emphasize that this kind of application scenario differs significantly from the traditional usage of visualization to generate new knowledge. Namely, existing graph and network visualization techniques have sought mainly to represent all aspects of a graph or network structure as faithfully as possible such that a user can explore the visualization, identify structures, and gain insights about the underlying data. These traditional visualization techniques can be sufficient for journalists and other communicators to support a narrative in print or on-line media [11] by showing only selected views of graphs.

However, novel approaches are required in our setting in which a specific property of the data is to be conveyed in an adversarial setting where the validity of the evidence presented may be questioned (see also the defense lawyer role in Fig. 1 which may, e. g., represent doubts of the audience). For example, the investigative activity of the Italian Revenue Agency (IRA) exploits the visual analysis of social networks whose nodes are the actors of potential fraudulent activities and whose edges represent financial/legal transactions between the actors. The investigators of IRA who suspect a group of persons or a single individual/company of tax evasion submit a case to the Italian financial Police for possible prosecution, which also implies showing some structural properties of the network beyond reasonable doubts. See, e. g., [21–23] for references about the use of visual analytics in the context of contrasting tax evasion in Italy.¹ Below we describe introductory examples.

¹ One author has been approached by the Australian Security and Investments Commission (a governmental regulator for stock exchange) inquiring about visualizations to convince a court about illegal trades.

► **Example 1.** A network admin discovers that two critical parts of the infrastructure would not be able to communicate with each other if a particular switch fails. To increase the robustness of the network, new hardware is needed. They have to convince the manager, who has no background in network security, to fund new hardware.

► **Example 2.** In a legal court case, the prosecution discovered that money acquired in black market sales was laundered by laundromat chain as evidenced by money provably transferred via a complicated network from the dealers to the laundromats. The prosecution has to convince the judge that all suspects belong to the criminal syndicate.

► **Example 3.** A new AI based heuristic is able to efficiently decide if a given graph is Hamiltonian, i. e., to test if it contains a cycle traversing all its vertices exactly once². However, false positives must be filtered out. A human operator needs to perform this task as there is no efficient algorithm. To facilitate this, the new version of the algorithm should also create a visualization of the graph making the Hamiltonian cycle obvious to the operator.

Such scenarios have key differences to standard motivations for graph visualization. Typical graph visualization techniques (node-link layout algorithms [18, 46], matrix ordering approaches [7] and mixed approaches which either include features of different paradigms [3, 4, 29] or show different visualizations side-by-side [13, 28]) usually seek a representation showing as many graph properties as possible simultaneously (by trading off aesthetic and readability criteria [1, 8, 9, 17]). However, for the scenarios above it is better to focus on showing optimally and faithfully just one specific property, i. e., we want a *visual proof* for that property.

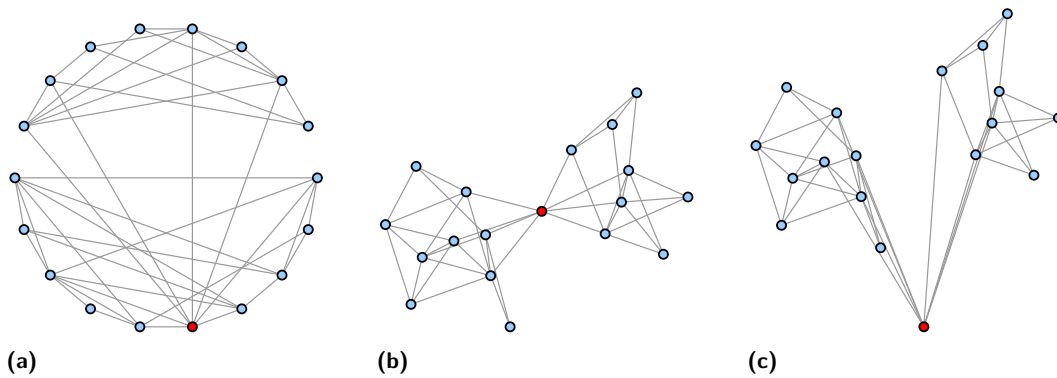
More precisely, a visual proof is a proof given by the use of a graphical or visual representation called *visual certificate*. A good visual proof should be clear and concise, conveying the main idea in an easy-to-understand way. It should be able to effectively communicate the desired message without being overly complex or cluttered. Additionally, the visual certificate should be aesthetically pleasing and easy to interpret. Somehow, it should be able to provide evidence to support the argument being made. Thus, a good visual certificate should be accurate, concise, and free of errors or mistakes.

In fact, visual proofs are already used in mathematics and other areas such as logic, graph theory, computer science, and physics [37, 57]; visual proofs are often easier to understand than algebraic proofs, as they are less abstract and easier to follow. Accessible proofs are often considered more beautiful by mathematicians; e. g., Appel and Haken employed a computer-assisted proof of the long-open four-color theorem in 1976 [5]. This new type of proof sparked philosophical debates [50] and while the theorem is broadly accepted as proved³, researchers still desire a more elegant proof [2]. Thus, we expect that visual proofs are appealing and even more convincing to experts also in fields other than mathematics.

Visual proofs can also convey properties to non-expert users or explain correctness of AI-generated solutions. As powerful chat-based interfaces are capable of generating plausible sounding – but difficult to verify – explanations of complex phenomena, we believe that there is a requirement to understand what makes a graph representation a proper visual certificate.

² Note that neural network approaches for NP-hard problems have been described, e. g., in [10]. In addition, the need for visualizations in the context of explainable deep learning has been described, e. g., in [16].

³ According to the *Oxford English Dictionary*, it is yet to be proven as a “*mathematical theorem*” [40].



■ **Figure 2** Three layouts generated with γED^4 [59]. (a) and (b) Circular and organic layouts generated with standard settings, resp. (c) Manually created layout highlighting the cut-vertex.

Contribution. We introduce a model identifying important steps and their interactions in a visual proof of a graph property. Based on this model, we formalize the concept of visual certificates and give requirements for a visualization to qualify as such. We also give examples of visual proofs for widely used graph properties and identify open research questions that should be answered to better understand visual proofs and make them algorithmically usable.

2 First Examples of Visual Proofs

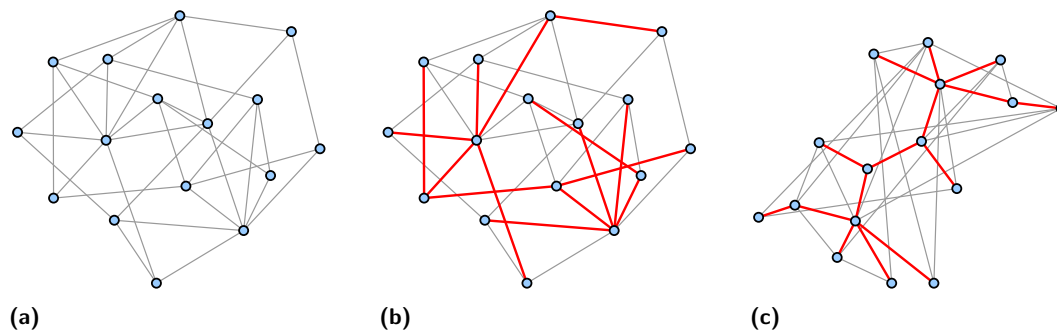
2.1 Example 1: The Graph contains a Cut-Vertex

First, we revisit Example 1. In this communication network there are two distinct parts such that all connections between them traverse a single switch. This corresponds to the graph underlying the network containing a *cut-vertex*, whose removal separates the remainder of the graph into at least two distinct components. Hence, in order to convince the manager, the network admin has to point out that the graph underlying the network can be separated by the removal of the vertex corresponding to the switch. So, they first layout the graph using a circular layout, which is a wide-spread all-purpose visualization style [44], and point the manager to the fact that the red colored vertex is a cut-vertex; see Figure 2a.

Unfortunately, the circular layout does a poor job at highlighting the cut-vertex. While it is evident to the manager that there are a top and a bottom component connected by some edges, they explain that they are not sure if all connections between both components use the suggested cut-vertex or not. Hence, the network admin prepares a second drawing using a force-directed organic layout where the cut-vertex is clearly visible; see Figure 2b. However, the engineer who designed the network becomes defensive and claims that there could be another edge hidden behind the alleged cut-vertex. This argument can be easily disproven by the network admin as they move the cut-vertex down, obtaining the drawing in Figure 2c. Presented with this new line of evidence, the engineer stops arguing and the manager agrees that the network has to be made more robust.

Discussion. This example illustrates how standard layout techniques may be unable to highlight even simple properties. In the circular layout, it is not easy to verify even when the cut-vertex is highlighted; see Figure 2a. This is due to the Gestalt principle of grouping [52,53].

⁴ Unless specified otherwise, the layouts of all visualizations in this paper have been created by the authors.



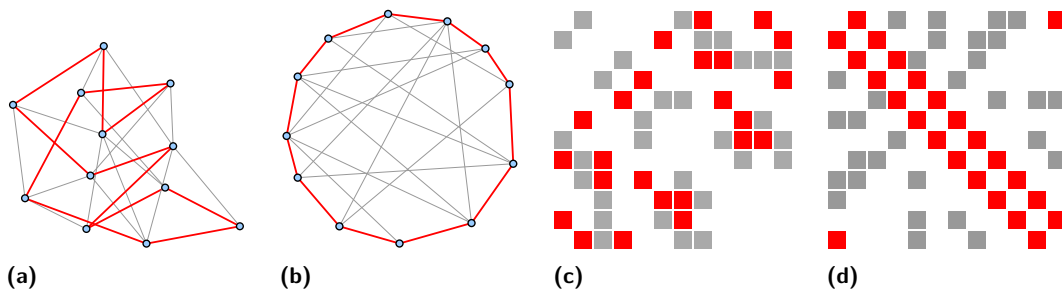
■ **Figure 3** (a)–(b) Organic layout generated with standard settings by YED [59], with a spanning tree highlighted in (b). (c) A manually created layout highlighting the spanning tree.

Here, the initial perception is guided by continuity and closure of node positions, leading to the perception of a single circular component. As a second step, an observer may see two separate components with edges biasing perception due to connectedness grouping. Thus, the observer has to analyze the entire graph, going node-by-node, to negate the automatic perceptual grouping induced by the layout to verify that there is a cut-vertex. The issue with the second illustration in Figure 2b is of different nature. Namely, the force-directed layout does a much better job at highlighting the cut-vertex. In fact, the observer discovers two dense salient features which are the two components separated by the cut-vertex and immediately notes that they are connected at a single vertex. Nevertheless, if there is an overlapping edge behind the cut-vertex, the drawing may look the same, challenging the human observer to identify that the vertex is not a cut-vertex. The drawing in Figure 2c avoids this problem by explicitly highlighting the cut-vertex via pre-attentively perceptible patterns (i. e., pop-out effects) [53]. The singular goal of highlighting the cut-vertex is achieved at the cost of traditionally accepted aesthetic metrics [42], as – compared to the circular and force-directed layouts – the general layout is unbalanced, with many crossings and poor resolution; see Table 1. Thus, visual certificates may not be useful in traditional exploratory applications, instead they focus on highlighting a specific property.

We remark that a cut-vertex proves non-2-connectivity and a similar approach can be used to visually prove that a graph is *not* k -connected: there exists a set of $k - 1$ vertices whose removal separates the graph and we can layout the graph so that all connections between two clearly separated parts run via this vertex set.

2.2 Example 2: The Graph is Connected

In Example 2, to convince the judge, the prosecution lawyer decides to visualize the network of criminals induced by the connections of provable money transfers. The prosecution lawyer draws it with a force-directed approach; see Figure 3a. While Figure 3a shows that there are many connections in the graph, it does not emphasize that there is only a single connected component. Hence, the defense lawyer argues that the component containing their client may have been drawn on top of the component with all the convicted criminals. Hence, the prosecution lawyer has to improve their visual proof. To do so, they include a highlighted *spanning tree* that shows that every vertex can be reached from every other vertex; see Figure 3b. Although the defense lawyer now has to admit that there is a smaller portion of the drawing to check, i. e., the highlighted edges, their argument stays more or less the same: that there are still crossings between edges of the spanning tree, which may be due



■ **Figure 4** Four layouts of a graph with a Hamiltonian cycle (red).

to two different highlighted components drawn on top of each other. Thus, the prosecution lawyer creates a third drawing in which the spanning tree is crossing-free; see Figure 3c. Here the spanning tree is rooted at the central vertex and vertices are drawn on concentric circles depending on distance from the root. Given this visualization, the defense rests, and the judge decides quickly that indeed all members of the network are affiliated.

Discussion. While in Example 1 we have seen that the drawing style of the entire graph can be important to visually prove a property, here we added another dimension. Namely, a subgraph is explicitly color-highlighted for pre-attentive perception. In addition, the drawing of this subgraph was very important in creating a convincing argument. In Figure 3b the drawing of the spanning tree is not very readable. Thus, even with the attention drawn to this portion of the drawing, it remains time consuming to check that a single tree connects all vertices. But when the tree is laid out in a concise and readable fashion as in Figure 3c, it is quite evident that it spans all the vertices, as the colored edges induce automatic grouping via similarity [52] and act as guidance for attention spread [30]. Similar to Example 1, while the quality of the drawing of the spanning tree is improved, the drawing of the rest of the graph does not measure well on the usual metrics; see Table 1.

2.3 Example 3: The Graph has a Hamiltonian Cycle

We may train an AI to produce a good node-link drawing for Example 3. As in Section 2.2, we observe that the quality of the drawing of the evidence relevant to the property under consideration is more important than the drawing of the full graph (Figure 4a), thus we select a circular layout with the hamiltonian cycle forming the outer face (Figure 4b). However, the human operator needs to check that all edges of the highlighted outer cycle are indeed present which can become increasingly difficult for larger graphs where resolution may become problematic. We can improve upon these issues by instead using an adjacency matrix representation. While an arbitrary permutation (see Figure 4c) does not provide any insights, an appropriate sorting of rows and columns makes the cycle composed of three components: one red diagonal and two red-cells (top-right and bottom-left); see Figure 4d.

Discussion. We observed that different visualization paradigms may perform better or worse for visually proving a property. While the node-link drawing in Figure 4b already highlights the cycle well, the adjacency matrix representation in Figure 4d composes the Hamilton cycle in three components. The perception of the red diagonal is facilitated by figure-ground separation via connectedness and similarity [52], and the two corner cells stand out due to both color difference and symmetry [58]. The particular advantage of this representation is

Table 1 Selected aesthetic metrics of the node-link drawings in this paper: stress ST [32], node resolution NR, Jaccard index JI [1], edge-length ratio EL [33, 34], crossing resolution CR [19, 20], crossing number CN, aspect ratio AR and angular resolution AN [39]. Red numbers in parentheses give the corresponding values for subgraphs highlighted in red in the corresponding figure. For ST and CN lower numbers are better, otherwise higher numbers are better.

Fig.	2			3			4			5			6			7	
	(a)	(b)	(c)	(a) - (b)	(b)	(c)	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(a)	(b)		
ST	132.1	9.2	36.7	13.6 (31.3)	32.5 (19.6)	9.3 (51.9)	45.5 (12.6)	6635	567.6 (9.1)	651.0 (1.9)	1022.7 (3.3)	58.8	94.4				
CN	81	19	26	28 (8)	63 (0)	29 (4)	45 (0)	7452	6211 (2)	9493 (0)	9992 (0)	12649	12650				
JI	.274	.332	.283	.330 (.132)	.285 (.160)	.407 (.167)	.339 (.232)	.037	.177 (.243)	.180 (.361)	.182 (.361)	.918	.920				
EL	.174	.333	.127	.360 (.404)	.149 (.382)	.401 (.649)	.185 (.542)	.090	.020 (.161)	.023 (.586)	.013 (.472)	.126	.047				
NR	.174	.104	.100	.181 (.181)	.119 (.119)	.164 (.164)	.184 (.184)	.020	.016 (.123)	.016 (.314)	.011 (.307)	.126	.047				
AR	.985	.774	.966	.884 (.884)	.796 (.796)	.871 (.871)	.987 (.987)	1	.941 (.606)	.941 (.977)	.994 (.994)	.998	.898				
CR	20.0	27.1	7.2	37.1 (37.1)	20.5 (N/A)	31.4 (48.2)	22.5 (N/A)	2.6	0.79 (73.2)	0.79 (N/A)	0.74 (N/A)	14.4	4.68				
AN	10.0	.20	.56	1.68 (20.0)	4.44 (21.4)	0.40 (15.93)	12.08 (141.0)	0.21	.017 (6.87)	.017 (112.0)	.007 (97.6)	7.20	1.23				

that the used visual cues scale nicely even for very large matrices (up to the pixel resolution of the screen) [58]. Thus, an important criterion for judging the quality of a visual proof should be the workload required by the observer to evaluate the correctness. As checking for Hamiltonicity is a difficult task with an all-purpose visualization (see also Section 3.3), both visualizations should be regarded as valid visual certificates albeit of different quality.

3 Related Theories, Frameworks and Models

3.1 Certifying Algorithms

The concept of visual certificates is related to *certifying algorithms* popularized by McConnell et al. [36], which seek to provide short and easy-to-check *certificates* for the correctness of an algorithm. Let $f : X \rightarrow Y$ be a computable, surjective function for input set X and output set Y and let W be a set of *witnesses*. Intuitively speaking, a witness describes a simple proof certifying that the output y of an algorithm for f on input x satisfies $f(x) = y$. The validity of a witness for a certain combination of inputs and outputs is assessed via the witness predicate $\mathcal{W} : X \times Y \times W \rightarrow \{\mathbf{true}, \mathbf{false}\}$ that fulfills:

1. *Witness property*: Given $(x, y, w) \in X \times Y \times W$, it holds $f(x) = y \Leftrightarrow \mathcal{W}(x, y, w) = \mathbf{true}$.
2. *Checkability*: Given $(x, y, w) \in X \times Y \times W$, it is trivial to determine $\mathcal{W}(x, y, w)$.
3. *Simplicity*: $\mathcal{W}(x, y, w) \Rightarrow f(x) = y$ has a simple proof.

An algorithm for f is now called *certifying algorithm* if for any input $x \in X$ it computes the output $y = f(x) \in Y$ and a witness $w \in W$ such that $\mathcal{W}(x, y, w) = \mathbf{true}$. It is worth noting that Properties 2 and 3 of the witness predicate are vaguely formulated. McConnell et al. [36] suggest that Property 2 can be formalized by requiring that there must be a decision algorithm for \mathcal{W} that runs in a certain time (such an algorithm is called a *checker*). On the other hand, they emphasize that Property 3 is intentionally left subjective as it relies on what is considered common knowledge. For examples of certifying algorithms, see the full version of this article.

3.2 Perception

Visual proofs are concerned with a design of visual evidence for an existence of a specific property, such as the presence of a cut-vertex, a Hamiltonian cycle, etc. In principle, a proof for such a property can be reduced to a program that returns a binary outcome, affirming or rejecting the claim. This may be sufficient for specialists who are familiar with the property itself, understand and trust the algorithm behind the code, and trust that the code is valid. However, such evidence may not be convincing to a non-specialist (a judge, a stockholder, etc.), particularly because the proof itself will be just one piece of evidence among many. Prior research shows that in such cases, presenting evidence *per se* is not enough, as information can be discounted as confusing, unimportant, or, given the wrong context, even misleading [49], as the accessibility and clarity of evidence could be as important as evidence itself [27].

Due to the diversity of graph properties there can be no general solution. Visual proof design might be guided by the principle of optimizing the *data-ink* ratio [49]. Thus, instead of optimizing overall aesthetic metrics [42], one should minimize the required number of *visual queries*, i. e., attention orientation, driving eye movements, and pattern/object recognition [54].

The human visual perception system consists of three stages: (1) rapid parallel processing involving billions of neurons, e. g., extraction of orientation, texture, color, and motion features; (2) slower processing than Stage 1, e. g., detection of 2D patterns, contours and

regions; (3) slow serial processing, involving both working and long-term memory, e. g., object identification [53]. As in Stage 1 the entire visual field is processed quickly in parallel, information that can be captured in this stage can be easily distinguished. Thus, pre-attentive (pop-out) patterns such as color, size, orientation, shapes, etc. should be utilized.

In other words, a good visual proof must ensure that a focal piece of evidence is a visual “pop-out” feature that automatically attracts viewer attention and that the visual layout is parsed and grouped into patterns that express the evidence. In case of the former, studies on visual search provide a comprehensive list of useful pop-out features such as color, size, contrast, or location [58]. Regarding the latter, one can rely on a large body of literature on principles of perceptual organization, commonly known as Gestalt principles [52]. However, yet another constraint is placed by our working memory that limits the number of nodes, edges, and components that can realistically be assessed at any single time [35].

The examples above illustrate the importance of this approach for visual proofs. For instance, consider the visual evidence for the existence of a cut-vertex in Figure 2a. While it uses color to attract the viewer’s attention to the cut-vertex and spatial arrangement to visually separate the two components, it still leads to an excessive number of visual queries, requiring multiple scans of individual vertices to ensure that they are connected only to the cut-vertex and the nodes within the component. In turn, there is a memory bottleneck that is likely to prevent a viewer from being completely certain about the validity of the proof. In contrast, in Figure 2c the graph layout groups the entire evidence in just three components and clearly shows lack of inter-component edges, so that very few visual queries are required to confirm the vertex is indeed a sole connector between the components. In short, although there cannot be a single one-size-fits-all approach for constructing visual proofs, their critical role in aiding the cognition of the viewer means they should be built based on principles of perceptual organization and around the limitations of attention and memory [54].

3.3 Computational Complexity

To evaluate the amount of the cognitive workload, we will apply concepts from complexity theory [6, 24]. It is also worth mentioning that the examples discussed so far differ in terms of their computational complexity. Namely, all cut-vertices of a graph and a spanning-tree can be found in time $O(n + m)$ based on BFS traversals where n is the number of vertices and m the number of edges while determining a Hamiltonian cycle is *NP-complete* [24]. Thus, in Example 3, we have visually proven an algorithmically difficult to solve problem.

However, there may be graph properties that cannot be visually proven. We first have to discuss how a human observer interacts with a visual certificate. In Example 1, the human observer identified two connected components and then saw that they can be separated by the removal of their shared vertex. Such a procedure could be seen as an $O(1)$ time algorithm, where the observer determined that there is only a single point where both components touch. Similarly, in Examples 2 and 3, the observer may have checked for every vertex if it was part of the highlighted structure. Even if they were to check this for every vertex one at a time, the resulting algorithm would still run in linear time. Hence, an observer is actually performing a *deterministic* validation algorithm for establishing that a certificate is correct.

Now, consider the complementary question to Example 3, i. e., we want to determine whether a graph does *not* contain a Hamiltonian cycle. This is a *CoNP-complete* problem as it is the complement to an NP-complete problem. For CoNP-complete problems it is likely that there is no certificate that can be checked in polynomial time [6], i. e., if we assume that a human observer deterministically analyzes a visualization (as could be recreated by computer vision), we have to assume that we cannot visually prove a CoNP-complete problem.

3.4 Related Visualization Models

Aside from graph visualizations, the concept of visually enhancing a proof is wide-spread. In mathematics, visual proofs for theorems have been used since ancient times [57] and there is a plethora of examples [37]. The question if such proofs can be regarded as such also has been discussed philosophically [12]. Also in computer science, visualizations are heavily used to convey knowledge, e.g., while not necessarily proving, an interactive sequential art by Bret Victor [51] beautifully explained an algorithm from a Nature paper [55].

Overall, there is a trend of increasingly sophisticated models considering an holistic integration of visualization into the sensemaking process, typically with the goal of informing the design of interactive systems for data exploration. Early models considered a linear pipeline, from data, via various transformations, to a visual display [14]. Visual analytics seeks to apply visualization to support the entire human sense-making loop [41]. More recent models aim to connect sense-making from interactive data visualization, via hypothesis formation and testing, to knowledge generation [43]. An underlying theme across most of this work is the role of computational guidance in the analytics process, and how algorithms can support the various loops in the sensemaking process [15]. By contrast, we consider a different model to conceptualize the role of algorithms, and AI, in supporting data (specifically network data) understanding. Our model for visual proofs (Fig. 1) does not seek to replace the traditional sense-making/knowledge-generation loop, but to support humans in situations where the result of a complex algorithm or property needs to be explained and justified.

There are also models related to ours from information visualisation research. Song et al. [45] considered a problem that may be seen as a complementary question to the one studied in this paper: They investigated how computer vision can understand network visualizations optimized for human users. Wickham et al. [56] proposed a two-phase procedure to convince a human observer that a data set contains statistically significant difference from randomly generated data. The human observer is first exposed to several randomly generated data sets (similar to a Rorschach test) before being exposed to a line-up consisting of the real data set and a couple randomly generated data sets. The first phase primes the human viewer for statistically insignificant variations so that, in the second phase, statistically significant differences clearly pop out from the noise. Another related model are *Gragnostics*, which are ten features suggested by Gove [25,26], that are fast to compute and provide a quantification of structural graph properties. In contrast to our model that aims to prove structural properties of graphs, Gragnostics provides the human user with a first impression of the structure of the graph at hand which may be helpful for initiating a thorough investigation. Finally, our model may also be seen as a visual communication of structural graph properties. Visual communication has been investigated in other settings for several decades, see e.g. [47,48].

4 The GraphTrials Model

We are now ready to discuss our formalization of *visual proofs*. For this, we first abstractly outline the process of visually proving properties of graphs in an adversarial setting using a model that we call GRAPHTRIALS; see also Figure 1. The model includes three distinct roles that have already appeared in our discussion of Example 2 in Section 2.2: The *prosecution lawyer* must convince the judge that a certain assertion regarding a graph is true, the *defense lawyer* may raise doubts about the validity of the prosecution lawyer’s claims, and the *judge* will determine the truth of the assertion. The roles *prosecution lawyer*, *judge* and *defense lawyer* are to be seen as abstract descriptions of the different actors in the process; e.g., in Examples 1 and 3, the prosecution lawyers were the network admin and the AI based

algorithm, respectively. The latter example further indicates that not all roles have to be assigned to a human. In fact, we only require that the judge corresponds to the human audience of the visual certificate whereas each lawyer may be either human, software or a human assisted by software. Moreover, as we have seen in Section 2.1, it can also occur that a critical audience can act as both the judge and defense lawyer roles simultaneously.

To convince the judge of a valid assertion f for the input graph G , the *prosecution lawyer* draws a visual certificate $W(G)$. To do so, they first analyze the raw data G to reveal evidence that proves the assertion f . The evidence is then embedded in $W(G)$: a visual representation of G that in some way emphasizes the evidence. Note that in the scope of our model we treat the analysis of the raw data and extraction of the evidence as a black box, i. e., we may assume that the prosecution lawyer already knows that the assertion f is true for the input graph G and may also be given the evidence as input. This allows us to *efficiently* visually prove algorithmically difficult assertions (such as the existence of a Hamiltonian cycle as in Section 2.3) and to ignore how the evidence is gathered (either algorithmically or by human interaction) in our model. The latter aspect also provides the possibility to separate the evidence gathering from the visualization process W , i. e., W could be a reusable program that embeds the evidence according to a specification⁵.

The *defense lawyer* checks the *unimpeachability* of $W(G)$ as a visual representation of G certifying $f(G)$. Thus, they may question whether the graph represented in the visualization actually corresponds to the input and they may also raise concerns if $W(G)$ is not distinguishable from a slightly different non-certificate (e. g., in Section 2.1 we encountered the case where an edge may have been hidden making it invisible to the judge's perception).

The *judge*, the human audience of the visual certificate $W(G)$, will validate the claim $f(G)$ using $W(G)$. In this step, the visual certificate $W(G)$ must guide the judge's perception so that they are able to form a mental model $\mathcal{M}(G)$ that facilitates confirmation of the validity of the assertion $f(G)$. For instance, the guidance can be formed by a suitable choice of topology which leads the judge to identify clusters of the graph as distinct salient features (as in Section 2.1) or by adding additional features such as color to draw attention to certain parts of the graph (as in Section 2.2). We discuss the judge's mental model in the full version of this article.

It is noteworthy that aside from the input graph and the verdict of the judge, the only information shared by all three roles is the *visual certificate* $W(G)$. In particular, it is the only medium that can be used by the prosecution lawyer to communicate the gathered evidence to the judge, i. e., the evidence is hidden information only accessible by the prosecution lawyer. Similarly, the judge is not communicating its mental model $\mathcal{M}(G)$ to the prosecution or defense lawyer, yet as we discussed above both roles might want to *estimate* what the mental model will look like. Furthermore, the nature of the mental model plays an important role in the validation step performed by the judge. Namely, the cognitive load put on the judge in this step depends hugely on how *complex* $\mathcal{M}(G)$ is. Finally, the defense lawyer's checking for unimpeachability is a process that is independent of the judge and prosecution lawyer and for a *faithful and readable* visual certificate we demand that there is no reason for the defense lawyer to raise doubts to the judge. As a result, there are several properties that we require from a visualization in order to call it a visual certificate and it could occur that an assertion cannot be *visually proven* for every graph for which the assertion is true (for instance we discussed issues related to scalability in Section 2.3). To this end, we also state when we want to say that a certain assertion can be visually proven for arbitrary graphs.

⁵ The examples in Section 2.3 and 5 both use visual certificates that highlight cycles.

Visual Certificates and Visual Provability

We give formal requirements inspired by the concept of certifying algorithms discussed in Section 3.1. Let $f : \mathcal{G} \rightarrow \{\mathbf{true}, \mathbf{false}\}$ be an *assertion function* for the set of graphs \mathcal{G} , i. e., for some graphs the assertion $f(G)$ is \mathbf{true} while for others it is not. For instance, if f is the existence of a cut-vertex, some graphs do contain one ($f(G) = \mathbf{true}$) while others do not ($f(G) = \mathbf{false}$). Consider a graph G with $f(G) = \mathbf{true}$ and let $W(G)$ be a visualization of G . We call $W(G)$ *visual certificate* for $f(G)$ if and only if the following hold:

1. *Unimpeachability*: We call $W(G)$ *unimpeachable*, if it satisfies the following two properties. First, $W(G)$ should provide *information faithfulness* [38], i. e., it displays the ground truth properties and structures in G . Second, $W(G)$ should provide *task readability* [38], i. e., the judge can *perceive* enough information for validating the assertion.
2. *Checkability*: Given $W(G)$, it is trivial to decide that $f(G) = \mathbf{true}$. In particular, this means that the judge’s perception leads to the formation of a *mental model* $\mathcal{M}(G)$ that makes it possible for the judge to *efficiently* validate the assertion. The number of distinct observations made by the judge in the process is called the *perceptual complexity*.
3. *Simplicity*: Given $\mathcal{M}(G)$, there is a *simple formal proof* for $f(G) = \mathbf{true}$ that relies solely on conclusions that the judge may deduce using $\mathcal{M}(G)$. In particular, this means that $W(G)$ is *perceptually distinguishable* from any possible wrong visual certificate $W'(G)$.

If a visual certificate $W(G)$ exists for each $G \in \mathcal{G}$ with $f(G) = \mathbf{true}$, we call f *visually provable*. Note that the complementary function f^c (which is \mathbf{true} if and only if $f(G) = \mathbf{false}$) needs not necessarily be visually provable. For instance, we were able to visually prove the assertion that G contains a Hamiltonian cycle in Section 2.3 but we argued that the absence of such a cycle cannot be visually proven in Section 3.3. This and requiring unimpeachability are clear differences to the concept of certifying algorithms whereas checkability and simplicity occur in both models, here considering the perceptual abilities of the judge; see also Section 3.1.

We are also interested in *how efficiently* the judge is able to validate $f(G) = \mathbf{true}$ based on $\mathcal{M}(G)$. To this end, we define the *perceptual complexity* as the time that the judge needs to check the assertion given $\mathcal{M}(G)$. The perceptual complexity may depend on the size of the graph, however, in some scenarios (e.g. Example 1) it may be independent of it. Since we assume the judge to make an objective judgment based on the evidence, we can treat the thought process as a deterministic algorithm and apply methods from complexity theory to evaluate the perceptual complexity. See the full version of this article for an application of these concepts.

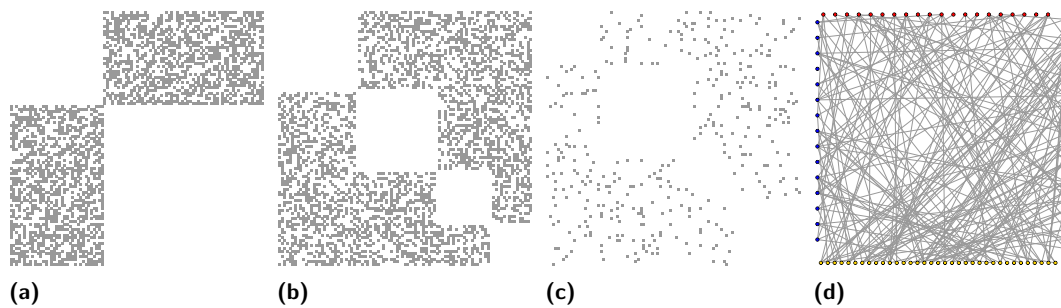
5 Visual Proofs for Graph Properties

We provide visual proofs for further widely used assertions. For a summary of our discussion, refer to Table 2. In addition, we discuss further assertions in the full version of this article.

(Non)-Bipartiteness and k -colorability. We can use a matrix representation to visually prove bipartiteness; see Figure 5a. When sorting the rows and columns according to the two independent subsets, bipartiteness can be simply checked by verifying if the two empty squares are indeed empty [58]. This approach also generalizes to k -colorability as shown in Figure 5b for 4 colors, however, for sparse graphs like in Figure 5c additional highlighting of the (supposedly) empty squares might be necessary. For small graphs, a node-link diagram might be easier to read and hence preferable, however the approach does not scale well due to resolution since the judge needs to verify there are no edges within the subsets; see Figure 5d.

■ **Table 2** Visual proofs in this paper, computational complexity of the problem and perceptual complexity of presented visual proofs (n and m denote the numbers of vertices and edges, resp.).

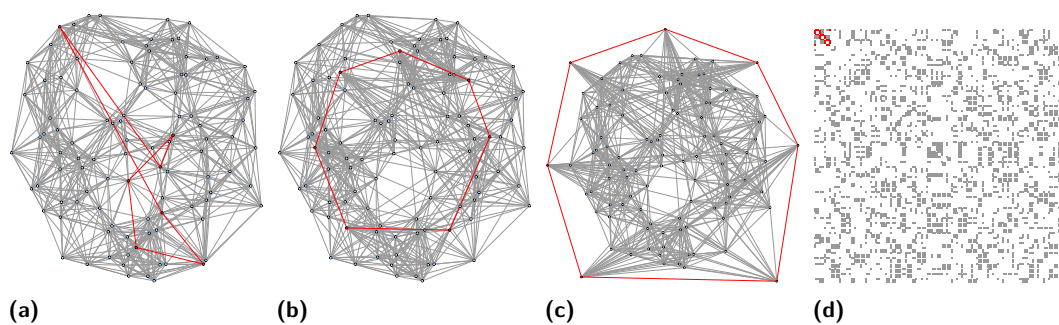
Assertion	Comp. Complexity	Percep. Complexity	Sec.
G is connected	$O(n+m)$	$O(n)$	2.2
G is <i>not</i> (2-)connected	$O(n+m)$	$O(1)$	2.1
G is <i>not</i> k -connected	$O(k^3n^2)$	$O(k)$	2.1
G is (not) complete	$O(n^2)$	$O(1)$	5
G has a Hamilt. cycle (path)	NP-complete	$O(1)$	2.3
G has a length- k cycle (path)	NP-complete	$O(k)$	2.3
G is (not) bipartite	$O(n+m)$	$O(1)$	5
G is k -colorable	NP-complete	$O(k)$	5
CoNP-complete assertions	coNP-complete	<i>Conj.:</i> No Visual Proof	3.3



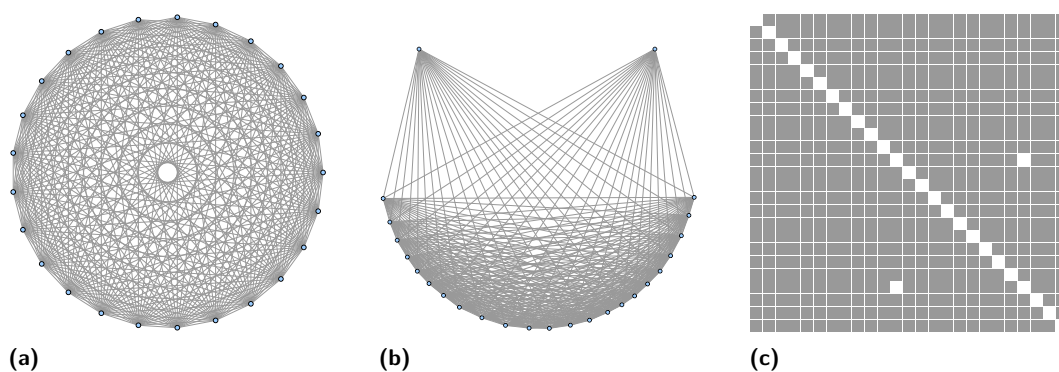
■ **Figure 5** Visualizing k -colorability. (a) A bipartite graph. (b) A dense 4-colorable graph. (c) and (d) a adjacency matrix and a node-link visualization of a sparse 4-colorable graph.

An odd-length cycle certifies that a graph is not bipartite, so non-bipartiteness can be visually proven by highlighting a shortest odd cycle in a drawing. In an arbitrary drawing, the cycle may be hard to spot, see Figure 6a. Redrawing the cycle in convex position makes it easier to read (see Figure 6b), especially if it is the convex outer cycle; see Figure 6c (this makes the rest of the graph harder to read; see Table 1). The cycle is now clearly visible and the judge just needs to assert oddness. While depending on the odd cycle length counting may be inevitable, the judge can use the symmetry of the drawing of the cycle to see that the cycle is odd (e.g., in Figure 6c, there is a single top-most but no single bottom-most vertex). For larger cycle lengths, an adjacency matrix representation may be beneficial: Sort the rows and columns along the odd cycle and mark it, then append the remaining vertices arbitrarily. Then, alter the spacing of the matrix so that even rows and columns are thicker than odd ones; see Figure 6d. The cell closing the cycle is a square if and only if the length is odd.

Completeness and Non-Completeness. Non-completeness is evidenced by a single missing edge and can be visually proven with a circular layout with the missing edge on the outer cycle. This approach does not scale well for a larger graphs; see Figure 7a. Readability and scalability can be improved by drawing focus to the missing edge, see Figure 7b. However, one can also use a matrix representation (see Figure 7c) since spotting a missing square scales well from a perception perspective [58]. This technique can also prove completeness.



■ **Figure 6** Visualizing non-bipartiteness of a graph. In (a) the odd cycle is self-intersecting, making it difficult to certify that it is in fact a cycle. Both in (b) and (c) the cycle is clearly visible where in (c) the cycle forms the outer boundary of the drawing letting it stand out even more compared to (b). Finally, in (d) the odd cycle is represented by a distinguishable pattern in the adjacency matrix.



■ **Figure 7** Visualizing non-completeness of a graph. In (a) the missing edge is very difficult to spot. In (b) and (c) on the other hand it is obvious that an edge is missing.

6 Limitations of the GraphTrials Model

Scalability. In the GRAPHTRIALS model, we must not only visualize the evidence represented in the visual certificate, but also display the remainder of the graph faithfully. This may result in higher computational complexity compared to other visualization techniques, e. g., force-directed graph layouts, whose purpose is to create an overall readable representation. Why not forgo visualization completely and use an assertion software to validate the evidence computationally? While this could drastically reduce the computation time and require fewer software components, there are in fact real-world application scenarios, e. g., in court, where it may be better to show a visual certificate accompanied by a short explanation why the certificate is indeed establishing the assertion instead of simply telling the audience that a piece of software analyzed the network and found the evidence for the assertion; see Section 1. Another benefit of visual proofs over a non-visual assertion software is that bugs in the visual proof pipeline can be spotted in the visual certificate, i. e., either the represented graph is not the input graph or the evidence is not a true evidence for the claim.

Another scalability issue is to display the entire graph faithfully. In Section 4, we assumed that the visual certificate may be represented by few components in the judge’s mental model and that the formation of that mental model can be mainly guided by usage of bottom-up and pattern recognition processes. For large input graphs, the screen resolution might not permit an information-faithful representation of the input graph so that one must resort to

techniques for displaying larger data, e. g., zooming. The introduction of such modes of user interaction may be problematic for our model as it may lead the judge to increasingly use top-down processes of perception which may influence the formation of the mental model.

Human factors. In our model, the judge is necessarily a human actor in the visual proof process. Hence, it is no surprise that human factors play an important role in the application of our model. Our model assumes that the judge is able to draw objective conclusions provided the evidence by the prosecution lawyer. This process may be hindered by insufficient background knowledge of the judge or subjective expectations towards the visualization. Moreover, the judge’s mental model cannot be directly analyzed and influenced introducing uncertainty into the model. We discuss these aspects further in the full version of this article.

7 Open Problems

(i) Are visual proofs in fact scalable? How do they extend to geospatial and dynamic graphs where the data are expected to obey spatial and/or temporal constraints? (ii) Which features contribute to perceptual complexity? (iii) Do response times depend mostly on perceptual complexity? (iv) When do human users regard a visual certificate as unimpeachable? (v) What are human limits for the perception of graph properties? For instance, the minimum perceivable slope difference is ≈ 2 degrees [31]. (vi) What is the trade-off between perceptual complexity and cognitive load?

References

- 1 Abu Reyhan Ahmed, Felice De Luca, Sabin Devkota, Stephen G. Kobourov, and Mingwei Li. Multicriteria scalable graph drawing via stochastic gradient descent, (SGD)². *IEEE Trans. Vis. Comp. Graph.*, 28(6):2388–2399, 2022. doi:10.1109/TVCG.2022.3155564.
- 2 Martin Aigner and Günter M. Ziegler. Five-coloring plane graphs. In *Proofs from THE BOOK (3. ed.)*. Springer, 2004. doi:10.1007/978-3-662-05412-3_30.
- 3 Lorenzo Angori, Walter Didimo, Fabrizio Montecchiani, Daniele Pagliuca, and Alessandra Tappini. Chordlink: A new hybrid visualization model. In Daniel Archambault and Csaba D. Tóth, editors, *Graph Drawing and Network Visualization - 27th International Symposium, GD 2019, Prague, Czech Republic, September 17-20, 2019, Proceedings*, volume 11904 of *Lecture Notes in Computer Science*, pages 276–290. Springer, 2019. doi:10.1007/978-3-030-35802-0_22.
- 4 Lorenzo Angori, Walter Didimo, Fabrizio Montecchiani, Daniele Pagliuca, and Alessandra Tappini. Hybrid graph visualizations with chordlink: Algorithms, experiments, and applications. *IEEE Trans. Vis. Comp. Graph.*, 28(2):1288–1300, 2022. doi:10.1109/TVCG.2020.3016055.
- 5 K. Appel and W. Haken. Special announcement. *Discret. Math.*, 16(2):179–180, 1976. doi:10.1016/0012-365X(76)90147-3.
- 6 Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. doi:10.1017/CB09780511804090.
- 7 Michael Behrisch, Benjamin Bach, Nathalie Henry Riche, Tobias Schreck, and Jean-Daniel Fekete. Matrix reordering methods for table and network visualization. *Comput. Graph. Forum*, 35(3):693–716, 2016. doi:10.1111/cgf.12935.
- 8 Michael A. Bekos, Henry Förster, Christian Geckeler, Lukas Holländer, Michael Kaufmann, Amadäus M. Spallek, and Jan Splett. A heuristic approach towards drawings of graphs with high crossing resolution. In Therese Biedl and Andreas Kerren, editors, *Graph Drawing and Network Visualization - 26th International Symposium, GD 2018, Barcelona, Spain, September 26-28, 2018, Proceedings*, volume 11282 of *Lecture Notes in Computer Science*, pages 271–285. Springer, 2018. doi:10.1007/978-3-030-04414-5_19.

- 9 Michael A. Bekos, Henry Förster, Christian Geckeler, Lukas Holländer, Michael Kaufmann, Amadäus M. Spallek, and Jan Splett. A heuristic approach towards drawings of graphs with high crossing resolution. *Comput. J.*, 64(1):7–26, 2021. doi:10.1093/comjnl/bzz133.
- 10 Irwan Bello, Hieu Pham, Quoc V. Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. In *Intl. Conf. Learning Represent. ICLR*. OpenReview.net, 2017. URL: <https://openreview.net/forum?id=Bk9mx1SFx>.
- 11 Liliana Bounegru, Tommaso Venturini, Jonathan Gray, and Mathieu Jacomy. Narrating networks. *Digital Journalism*, 5(6):699–730, 2017. doi:10.1080/21670811.2016.1186497.
- 12 James Robert Brown. *Philosophy of Mathematics: A Contemporary Introduction to the World of Proofs and Pictures*. Routledge, 2nd edition, 2008. doi:10.4324/9780203932964.
- 13 Michael Burch, Kiet Bennema ten Brinke, Adrien Castella, Ghassen Karray, Sebastiaan Peters, Vasil Shteriyarov, and Rinse Vlaswinkel. Guiding graph exploration by combining layouts and reorderings. In Michael Burch, Michel A. Westenberg, Quang Vinh Nguyen, and Ying Zhao, editors, *VINCI: Intl. Symp. Vis. Inform. Comm. Interact.*, pages 25:1–25:5. ACM, 2020. doi:10.1145/3430036.3430064.
- 14 Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.
- 15 Davide Ceneda, Theresia Gschwandtner, and Silvia Miksch. A review of guidance approaches in visual data analysis: A multifocal perspective. *Comput. Graph. Forum*, 38(3):861–879, 2019. doi:10.1111/CGF.13730.
- 16 Jaegul Choo and Shixia Liu. Visual analytics for explainable deep learning. *IEEE Comput. Graph. Appl.*, 38(4):84–92, 2018. doi:10.1109/MCG.2018.042731661.
- 17 Sabin Devkota, Abu Reyan Ahmed, Felice De Luca, Katherine E. Isaacs, and Stephen G. Kobourov. Stress-plus-x (SPX) graph layout. In Daniel Archambault and Csaba D. Tóth, editors, *Intl. Symp. Graph Drawing*, volume 11904 of *LNCS*, pages 291–304. Springer, 2019. doi:10.1007/978-3-030-35802-0_23.
- 18 Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, 1999.
- 19 Emilio Di Giacomo, Walter Didimo, Giuseppe Liotta, and Henk Meijer. Area, curve complexity, and crossing resolution of non-planar graph drawings. In David Eppstein and Emden R. Gansner, editors, *Graph Drawing, 17th International Symposium, GD 2009, Chicago, IL, USA, September 22-25, 2009. Revised Papers*, volume 5849 of *Lecture Notes in Computer Science*, pages 15–20. Springer, 2009. doi:10.1007/978-3-642-11805-0_4.
- 20 Emilio Di Giacomo, Walter Didimo, Giuseppe Liotta, and Henk Meijer. Area, curve complexity, and crossing resolution of non-planar graph drawings. *Theory Comput. Syst.*, 49(3):565–575, 2011. doi:10.1007/s00224-010-9275-6.
- 21 Walter Didimo, Luca Giamminonni, Giuseppe Liotta, Fabrizio Montecchiani, and Daniele Pagliuca. A visual analytics system to support tax evasion discovery. *Decis. Support Syst.*, 110:71–83, 2018. doi:10.1016/j.dss.2018.03.008.
- 22 Walter Didimo, Luca Grilli, Giuseppe Liotta, Lorenzo Menconi, Fabrizio Montecchiani, and Daniele Pagliuca. Combining network visualization and data mining for tax risk assessment. *IEEE Access*, 8:16073–16086, 2020. doi:10.1109/ACCESS.2020.2967974.
- 23 Walter Didimo, Luca Grilli, Giuseppe Liotta, Fabrizio Montecchiani, and Daniele Pagliuca. Visual querying and analysis of temporal fiscal networks. *Inf. Sci.*, 505:406–421, 2019. doi:10.1016/j.ins.2019.07.097.
- 24 M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- 25 Robert Gove. Gragnostics: Fast, interpretable features for comparing graphs. In Ebad Banissi, Anna Ursyn, Mark W. McK. Bannatyne, Nuno Datia, Rita Francese, Muhammad Sarfraz, Theodor G. Wyeld, Fatma Bouali, Gilles Venturini, Hanane Azzag, Mustapha Lebbah, Marjan Trutschl, Urska Cvek, Heimo Müller, Minoru Nakayama, Sebastian Kernbach, Loredana Caruccio, Michele Risi, Ugo Erra, Autilia Vitiello, and Veronica Rossano, editors, *Intl. Conf. Inform. Visual. IV*, pages 201–209. IEEE, 2019. doi:10.1109/IV.2019.00042.

- 26 Robert Gove. Gragnostics: Evaluating fast, interpretable structural graph features for classification and visual analytics. In Boris Kovalerchuk, Kawa Nazemi, Răzvan Andonie, Nuno Datia, and Ebad Banissi, editors, *Integrating Artificial Intelligence and Visualization for Visual Knowledge Discovery*, pages 311–336. Springer, 2022. doi:10.1007/978-3-030-93119-3_12.
- 27 Kieran Healy. *Data Visualization: A Practical Introduction*. Princeton University Press, 2018.
- 28 Nathalie Henry and Jean-Daniel Fekete. MatrixExplorer: a dual-representation system to explore social networks. *IEEE Trans. Vis. Comp. Graph.*, 12(5):677–684, 2006. doi:10.1109/TVCG.2006.160.
- 29 Nathalie Henry, Jean-Daniel Fekete, and Michael J. McGuffin. Nodetrix: a hybrid visualization of social networks. *IEEE Trans. Vis. Comp. Graph.*, 13(6):1302–1309, 2007. doi:10.1109/TVCG.2007.70582.
- 30 R. Houtkamp, H. Spekrijse, and P. R. Roelfsema. A gradual spread of attention. *Perception & Psychophysics*, 65(7):1136–1144, 2003. doi:10.3758/BF03194840.
- 31 Maurice Janssen, Marc Lauvenberg, Wesley van der Ven, Twan Bloebaum, and Herman Kingma. Perception threshold for tilt. *Otol Neurotol.*, 32(5):818–825, 2011. doi:10.1097/MAO.0b013e31821c6c7b.
- 32 Tomihisa Kamada and Satoru Kawai. An algorithm for drawing general undirected graphs. *Inf. Process. Lett.*, 31(1):7–15, 1989. doi:10.1016/0020-0190(89)90102-6.
- 33 Sylvain Lazard, William J. Lenhart, and Giuseppe Liotta. On the edge-length ratio of outerplanar graphs. In Fabrizio Frati and Kwan-Liu Ma, editors, *Graph Drawing and Network Visualization - 25th International Symposium, GD 2017, Boston, MA, USA, September 25-27, 2017, Revised Selected Papers*, volume 10692 of *Lecture Notes in Computer Science*, pages 17–23. Springer, 2017. doi:10.1007/978-3-319-73915-1_2.
- 34 Sylvain Lazard, William J. Lenhart, and Giuseppe Liotta. On the edge-length ratio of outerplanar graphs. *Theor. Comput. Sci.*, 770:88–94, 2019. doi:10.1016/j.tcs.2018.10.002.
- 35 Steven J. Luck and Andrew Richard Hollingworth. *Visual Memory*. Oxford University Press US, 2008. doi:10.1093/acprof:oso/9780195305487.001.0001.
- 36 Ross M. McConnell, Kurt Mehlhorn, Stefan Näher, and Pascal Schweitzer. Certifying algorithms. *Comput. Sci. Review*, 5(2):119–161, 2011. doi:10.1016/j.cosrev.2010.09.009.
- 37 R.B. Nelsen. *Proofs Without Words: Exercises in Visual Thinking*. Classroom resource materials. Mathematical Association of America, 1993. URL: <https://books.google.de/books?id=Kx2cjyzTIIyKc>.
- 38 Quan Hoang Nguyen, Peter Eades, and Seok-Hee Hong. On the faithfulness of graph visualizations. In Sheelagh Carpendale, Wei Chen, and Seok-Hee Hong, editors, *IEEE Pacific Vis*, pages 209–216, 2013. doi:10.1109/PACIFICVIS.2013.6596147.
- 39 Takao Nishizeki and Md. Saidur Rahman. *Planar Graph Drawing*, volume 12 of *Lect. Notes Ser. Computing*. World Scientific, 2004. doi:10.1142/5648.
- 40 Oxford English Dictionary. four-colour | four-color, adj. In *Oxford English Dictionary*. Oxford University Press, 2023. doi:10.1093/OED/3549282820.
- 41 Peter Pirolli and Stuart Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proc. Intl. Conf. Intelligence Analysis*, volume 5, pages 2–4. McLean, VA, USA, 2005.
- 42 H Purchase. Metrics for graph drawing aesthetics. *J. Vis. Lang. & Comput.*, 13(5):501–516, 2002. doi:10.1016/S1045-926X(02)90232-6.
- 43 D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. A. Keim. Knowledge generation model for visual analytics. *IEEE Trans. Vis. Comp. Graph.*, 20(12):1604–1613, 2014. doi:10.1109/TVCG.2014.2346481. doi:10.1109/TVCG.2014.2346481.
- 44 J. M. Six and I. G. Tollis. A framework for circular drawings of networks. In Jan Kratochvíl, editor, *Proc. 7th Intl. Symp. Graph Drawing*, volume 1731 of *LNCS*, pages 107–116. Springer, 1999. doi:10.1007/3-540-46648-7_11.

- 45 Sicheng Song, Chenhui Li, Dong Li, Juntong Chen, and Changbo Wang. Graphdecoder: Recovering diverse network graphs from visualization images via attention-aware learning. *IEEE Trans. Vis. Comp. Graph.*, pages 1–17, 2022. doi:10.1109/TVCG.2022.3225554.
- 46 Roberto Tamassia, editor. *Handbook on Graph Drawing and Visualization*. Chapman and Hall, 2013. URL: <https://www.crcpress.com/Handbook-of-Graph-Drawing-and-Visualization/Tamassia/9781584884125>.
- 47 Edward Rolf Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 1992.
- 48 Edward Rolf Tufte. *Visual explanations - images and quantities, evidence and narrative*. Graphics Press, 1997. URL: <https://www.worldcat.org/oclc/36234417>.
- 49 Edward Rolf Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 2001.
- 50 Thomas Tymoczko. The four-color problem and its philosophical significance. *J. Philosophy*, 76(2):57–83, 1979.
- 51 Bret Victor. Scientific communication as sequential art. <http://worrydream.com/ScientificCommunicationAsSequentialArt/>, 2011.
- 52 Johan Wagemans, James H. Elder, Michael Kubovy, Stephen E. Palmer, Mary A. Peterson, Manish Singh, and Rüdiger von der Heydt. A century of gestalt psychology in visual perception: I. perceptual grouping and figure–ground organization. *Psychol. Bulletin*, 138(6):1172–1217, 2012. doi:10.1037/a0029333.
- 53 Colin Ware. *Information Visualization – Perception for Design*. Morgan Kaufmann, 2004.
- 54 Colin Ware. *Visual Thinking For Information Design*. Morgan Kaufmann, 2021. doi:10.1016/C2016-0-01395-5.
- 55 Duncan J. Watts and Steven H. Strogatz. Collective dynamics of ‘small-world’ networks. *Nature*, 393:440–442, 1998. doi:10.1038/30918.
- 56 Hadley Wickham, Dianne Cook, Heike Hofmann, and Andreas Buja. Graphical inference for InfoVis. *IEEE Trans. Vis. Comp. Graph.*, 16(6):973–979, 2010. doi:10.1109/TVCG.2010.161.
- 57 Wikipedia. Mathematical proof. https://en.wikipedia.org/wiki/Mathematical_proof#Visual_proof, 2023.
- 58 Jeremy Wolfe. Visual search. *Cur. Biol.*, 20(8):R346–R349, 2010. doi:10.1016/j.cub.2010.02.016.
- 59 yWorks. yEd – graph editor. <https://www.yworks.com/products/yed>, 2023.

Connectivity-Faithful Graph Drawing

Amyra Meidiana ✉ 

University of Sydney, Australia

Seok-Hee Hong ✉ 

University of Sydney, Australia

Yongcheng Jing ✉ 

University of Sydney, Australia

Abstract

Connectivity is one of the important fundamental structural properties of graphs, and a graph drawing D should faithfully represent the connectivity structure of the underlying graph G . This paper investigates connectivity-faithful graph drawing leveraging the famous Nagamochi-Ibaraki (NI) algorithm, which computes a *sparsification* G_{NI} , preserving the k -connectivity of a k -connected graph G .

Specifically, we first present *CFNI*, a divide-and-conquer algorithm, which computes a sparsification G_{CFNI} , which preserves the *global* k -connectivity of a graph G and the *local* h -connectivity of the h -connected components of G . We then present *CFGD*, a connectivity-faithful graph drawing algorithm based on CFNI, which faithfully displays the global and local connectivity structure of G . Extensive experiments demonstrate that CFNI outperforms NI with 66% improvement in the connectivity-related sampling quality metrics and 73% improvement in proxy quality metrics. Consequently, CFGD outperforms a naive application of NI for graph drawing, in particular with 62% improvement in stress metrics. Moreover, CFGD runs 51% faster than drawing the whole graph G , with a similar quality.

2012 ACM Subject Classification Human-centered computing → Graph drawings

Keywords and phrases Graph connectivity, Faithful graph drawing, Graph sparsification

Digital Object Identifier 10.4230/LIPIcs.GD.2024.17

Funding This work was funded by ARC (Australian Research Council) DP (Discovery Project) grant DP190103301.

1 Introduction

Connectivity is a fundamental structural property of graphs due to its wide communication, transportation, and production applications. Consequently, tremendous progress has been made in algorithms and complexity theory related to graph connectivity [40]. For example, algorithms for various aspects of the connectivity have been presented, ranging from computing the connectivity of a graph [27], increasing the connectivity of a graph through edge augmentation [41], and decomposing a graph into connected components [18].

One notable problem is finding the minimum k -connected spanning subgraph of a k -connected graph, which is NP-complete [13]. Nevertheless, an efficient linear-time algorithm for finding a k -connected spanning subgraph of a k -connected graph with an upper bound of a linear number of edges has been presented [39]. Specifically, the NI (Nagamochi and Ibaraki) algorithm computes a k -connected spanning subgraph with $O(kn)$ edges for a k -connected graph $G = (V, E)$ in $O(m)$ time, where $n = |V|$ and $m = |E|$.

In graph drawing, the *faithfulness* is an important quality metric to measure how the drawing faithfully represents the ground truth structure of a large and complex graph. Examples include distance-faithful metrics known as stress [8], shape-based metrics [9], cluster-faithful metrics [31], symmetry-faithful metrics [32, 33], neighborhood-faithful metrics [26],



© Amyra Meidiana, Seok-Hee Hong, and Yongcheng Jing;
licensed under Creative Commons License CC-BY 4.0

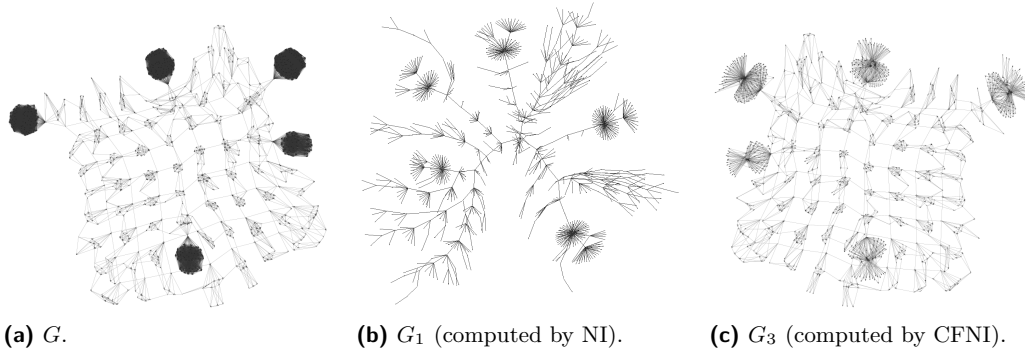
32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 17; pp. 17:1–17:17

Leibniz International Proceedings in Informatics



LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** Comparison of NI and CFNI: the sparsification computed by CFNI preserves both the global sparse connectivity and the local dense connectivity structures of the graph G , better than the sparsification computed by NI.

and change-faithful metrics [29]. In accordance, graph drawings that aim to optimize such faithfulness metrics have been investigated, such as stress minimization layouts [12, 22, 52], and the layouts to optimize shape-based metrics [30] and cluster faithfulness [3]. However, connectivity-faithful graph drawing has not yet been investigated.

In this paper, we present the first study on connectivity-faithful graph drawings by leveraging the NI algorithm, which can compute a sparse subgraph preserving the k -connectivity of a graph. Specifically, we first note that simply preserving the global k -connectivity of a graph may not be effective for connectivity-faithful graph drawing. For example, while a spanning tree preserves the global connectivity of a one-connected graph G , a drawing solely based on the spanning tree may fail to faithfully represent the local connectivity of dense subgraphs of G .

Therefore, we first present *CFNI (Connectivity-Faithful NI)*, a divide-and-conquer approach utilizing NI, which preserves both the global k -connectivity of a graph G and the local h -connectivity of each connected component of G . We then present *CFGD (Connectivity-Faithful Graph Drawing)*, which leverages CFNI to compute connectivity-faithful graph drawings. Our main contributions can be summarized as follows:

1. We present *CFNI (Connectivity-Faithful NI)*, a divide-and-conquer approach for graph sparsification utilizing NI, to compute a connectivity-faithful sparsification, preserving not only the global k -connectivity of a graph G but also the local h -connectivity of each h -connected component of G , for $h > k$. Extensive experiments demonstrate that CFNI achieves, on average, 66% better connectivity-related sampling quality metrics and 73% better proxy quality metrics [44] than NI, which outperforms the state-of-the-art *SS (Spectral Sparsification)* [49] with 52% better connectivity-related sampling quality metrics.
2. We present *CFGD (Connectivity-Faithful Graph Drawing)*, which leverages CFNI for connectivity-faithful graph drawing to faithfully represent both the global and local connectivity structures in a graph. Experiments show that CFGD obtains better quality metrics than a naive application of NI to graph drawing, particularly at up to 62% lower stress on average. Furthermore, CFGD runs faster than directly drawing the whole graph, at 51% faster, with a similar quality.

Figure 1 compares CFNI and NI for a one-connected graph G . The spanning tree G_1 in Figure 1b is computed by NI, while G_3 in Figure 1c is computed by CFNI with $h = 3$ (preserving the triconnectivity of triconnected components of G). Clearly, G_3 better preserves both the global mesh-like structure and the locally dense structures than G_1 , which loses the local connectivity.

2 Related Work

2.1 NI (Nagamochi-Ibaraki) Algorithm

Nagamochi and Ibaraki [39] presented a linear time algorithm to find a sparse k -connected spanning subgraph of a k -connected graph, based on the following main lemma:

► **Lemma 1.** *For graph $G = (V, E)$, let $F_i = (V, E_i)$ be a maximal spanning forest in $V - E_1 \cup E_2 \cup \dots \cup E_{i-1}$ for $1 \leq i \leq |E|$ where possibly $E_i = E_{i+1} = \dots = E_{|E|} = \{\}$ for some i . Each spanning subgraph $G_i = (V, E_1 \cup E_2 \cup \dots \cup E_i)$ satisfies $\lambda(x, y, G_i) = \min(\lambda(x, y, G), i)$ for all $x, y \in V$ where $\lambda(x, y, G)$ is the local connectivity between x and y in graph G .*

Based on Lemma 1, a subgraph $G_{NI} = (V, E')$ where $E' = E_1 \cup E_2 \cup \dots \cup E_k$ is k -connected if $k \leq \lambda(x, y, G)$. To compute G_{NI} , one must compute the disjoint edge subsets $E_1, E_2, \dots, E_m, m = |E|$, where each E_i is a maximal spanning forest in $G \setminus (E_1 \cup \dots \cup E_{i-1})$. G_{NI} is then constructed using the union of E_1 to E_k , i.e., $G_k = (V, E_1 \cup E_2 \cup \dots \cup E_k)$.

In other words, given a k -connected graph $G = (V, E)$, the NI algorithm computes an ordered list of disjoint edge subsets E_1, E_2, \dots, E_m , such that (V, E_1) is one-connected, $(V, E_1 \cup E_2)$ is biconnected, $(V, E_1 \cup E_2 \cup E_3)$ is triconnected, and so on, up to E_k .

The NI algorithm takes a k -connected graph $G = (V, E)$ and starts by marking all $v \in V$ and $e \in E$ as “unscanned”, and assigning a counter r to each $v \in V$, where all $r(v)$ starts at 0. The algorithm loops through every unscanned vertex, selecting a vertex with the highest r for each iteration. The algorithm then iterates through all unscanned edges $e = (x, y)$ incident on x , and adds e to the subset $E_{r(y)+1}$. If $r(x)$ is equal to $r(y)$, $r(x)$ is incremented by 1; otherwise, $r(y)$ is incremented by 1. e is then marked as “scanned”, and once all unscanned edges incident on x has been scanned, x is marked as “scanned”. The algorithm finally returns the k -connected spanning subgraph $G_{NI} = (V, E_1 \cup E_2 \cup \dots \cup E_k)$. The following theorem describes the main results:

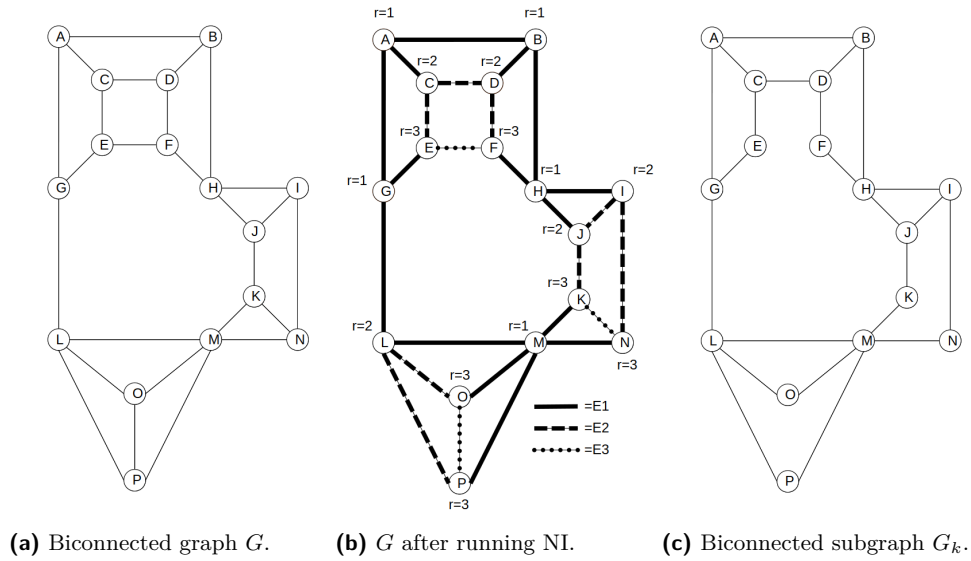
► **Theorem 2.** *Given a simple graph $G = (V, E)$, partition $E_i \subset E$ satisfying Lemma 1 can be found in $O(n + m)$ time, where $|E_i| \leq n - i$ for $i < n$ and $|E_i| = 0$ for $n \leq i \leq m$.*

The linear runtime comes from each vertex and edge being scanned once. As G is simple, $r(v)$ for $v \in V$ increases at most 1 when an incident vertex is scanned, thus $r(v) \leq n - 1$. Meanwhile, $|E_i| \leq n - 1$, as $|E_i| = n - 1$ implies that no more vertices have $r(v) < i$. Thus, the k -connected spanning subgraph $G_{NI} = (V, E_1 \cup E_2 \cup \dots \cup E_k)$ can have at most $k(n - 1)$ edges.

Figure 2 shows an example of running NI on a graph, in this case, a biconnected graph G shown in Figure 2a. Figure 2b shows the result of running NI on G , in particular showing which edges belong to each edge set E_1, E_2, E_3 as well as the r values of each vertex at the end of the NI algorithm. Figure 2c shows the biconnected spanning subgraph G_k , obtained using the union of the edge sets $E_1 \cup E_2$ from the results in Figure 2b.

2.2 Graph Sampling and Spectral Sparsification

Graph sampling has been extensively studied within graph mining, where complex analysis can be computed more efficiently on the smaller sample graph G' than on the original large and complex graph G [21, 23]. The main challenge for graph sampling is to compute G' , which is a good representative of G , preserving the structural properties of G . However, the most popular simple random sampling methods, such as Random Vertex (RV) or Random



■ **Figure 2** Example of running NI on a biconnected graph.

Edge (RE), often produce disconnected samples, failing to preserve the connectivity of G [51]. Recent random sampling methods improve the connectivity of G' and reduce the computation time of G' using the BC (Block-Cut vertex) tree decomposition [16].

Spectral sparsification [48] computes G' preserving the spectrum of G , which is closely related to important structural properties such as clustering [50] and connectivity [6]. Every n -vertex graph G has a spectral sparsification G' with $O(n \log n)$ edges, which can be computed in near-linear time [49].

More recent work on graph sampling utilizes spectral sparsification to compute G' , preserving the structural properties of G . For example, DSS (Deterministic SS) computes G' by selecting edges in decreasing order of effective resistance values [10]. Similarly, the SV (Spectral Vertex) sampling computes G' by selecting vertices in decreasing order of the sum of effective resistance values of their incident edges [20]. Both DSS and SV have been shown to perform significantly better than RE and RV , respectively, on various sampling quality metrics [10, 20].

Furthermore, spectral sparsification has been integrated with graph connectivity, such as the decomposition into biconnected (resp., triconnected) components using the BC (resp., SPQR) tree to reduce the computation time of G' and to improve the quality of G' including the connectivity, see [19, 34].

2.3 Fast Graph Drawing Algorithms using Sampling

Graph sampling methods have been successfully integrated with the most popular graph drawing methods, such as force-directed algorithms and stress minimization methods, to reduce the runtime complexity of the algorithms from quadratic time to linear time [14, 46, 52].

For example, the sparse stress-based algorithms [46, 52] sample a *pivot* set $P \subset V$ of constant size to reduce the stress computation from quadratic to linear time. Similarly, the RVS algorithm [14] uses a random vertex sampling method with a sliding window to reduce the runtime of repulsion force computation to linear time.

More recently, the fastest graph drawing algorithms using the sublinear-time force computation and stress computation have been presented [28, 36, 38]. For example, the SublinearForce framework [28] utilizes both vertex and edge sampling based on spectral sparsification to reduce the computation of both repulsion and attraction forces from linear to sublinear, while obtaining better quality than the linear-time RVS.

Sublinear-time stress computation algorithms have also been presented [38], based on the Stress Majorization and Stochastic Gradient Descent, integrating vertex sampling using spectral sparsification to reduce the stress computation from linear to sublinear time while producing drawings similar to SM and SGD.

2.4 Faithfulness Metrics and Faithful Graph Drawing

Faithfulness metrics are designed for evaluating drawings of large and complex graphs, by measuring how faithfully the ground truth structure of the graph is represented in a drawing [43]. Various faithfulness metrics have been presented based on the definition of the ground truth structure of the graph:

- *Stress* measures how proportional the geometric distances between vertices in a drawing are to the shortest path distance between the vertices in the graph [8].
- *Shape-based metrics* measure how faithfully the “shape” of the drawing, computed using the *proximity graph*, represents the ground truth structure of a graph [10, 15].
- *Proxy quality metrics* [44] measure how faithfully the drawing of a sample graph represents the ground truth structure of the original graph by computing the similarity between a graph G and the “shape” of the drawing D' of a sample graph $G' \subset G$.
- *Cluster faithfulness* [31] measures how faithfully the ground truth clustering of vertices is represented as the geometric clustering in the drawing.
- *Automorphism faithfulness* [32, 33] measures how faithfully the automorphisms of a graph are represented as symmetries in the drawing of a graph.
- *Change faithfulness* [4, 29] metrics are designed for *dynamic* graphs, measuring how proportional the change in the dynamic graph drawings is to the ground truth change of the structure of the dynamic graph.

Consequently, a number of graph drawing algorithms for optimizing faithfulness metrics have been investigated, such as stress minimization layouts [12, 22, 52], ShFR and ShSM algorithms to maximize shape faithfulness [30], and the ClusterKmeans and ClusterHAC algorithms to maximize cluster faithfulness [3].

3 CFNI: Connectivity-Faithful NI

While the NI algorithm successfully computes a spanning subgraph preserving the global k -connectivity of a graph, it may not always be sufficient to preserve the dense local connectivity structures of the graph for connectivity-faithful graph drawing. This may be an issue, especially for the graphs with a “globally sparse, locally dense” structure, such as the scale-free graphs often found in real-world social networks and biological networks [1].

To address this issue, we present CFNI, a divide-and-conquer approach leveraging NI for graph sparsification, which preserves both global and local connectivities. Given a k -connected graph G , CFNI takes as parameter a target connectivity h , and returns a subgraph preserving both the global k -connectivity of G and the local h -connectivity of each h -connected component of G .

Algorithm 1 CFNI.

```

1: Input: Graph  $G = (V, E)$ , target connectivity  $h$ 
2:  $k$ : connectivity of  $G$ 
3: if  $h > k$  then
4:   Decompose  $G$  into  $k + 1$ -connected components  $C_1, C_2, \dots, C_c$ 
5:   for each  $k + 1$ -connected component  $C_i = (V_{C_i}, E_{C_i})$  do
6:      $G_{hi} = (V_{C_i}, E'_{C_i}) = \text{CFNI}(C_i, h)$ 
7:   end for
8:    $G_{CFNI} = (V, E'_h = E'_{C_1} \cup E'_{C_2} \cup \dots \cup E'_{C_c})$ 
9:   return  $G_{CFNI}$ 
10: end if
11:  $E_1 = E_2 = \dots = E_m = \{\}$ 
12:  $V_u = V, E_u = E$  // unscanned vertices and edges
13:  $r(v) = 0$  for all  $v \in V_{C_i}$ 
14: while  $|V_u| > 0$  do
15:    $x =$  vertex in  $V_u$  with largest  $r$ 
16:   for  $\{e \in E \mid e = (x, y)\}$  do
17:      $E_{r(y)+1} = E_{r(y)+1} \cup \{e\}$ 
18:     if  $r(x) == r(y)$  then
19:        $r(x)+ = 1$ 
20:     end if
21:      $r(y)+ = 1$ ;
22:      $E_u.\text{remove}(e)$ 
23:   end for
24:    $V_u.\text{remove}(x)$ 
25: end while
26:  $G_{NI} = (V, E' = E_1 \cup E_2 \cup \dots \cup E_k)$ 
27: return  $G_{NI}$ 

```

Roughly speaking, the main idea of CFNI is to divide a k -connected graph into $k + 1$ -connected components, and then, for each $k + 1$ -connected component, recursively decompose it into $k + 2$ -connected components, and so on, until a decomposition into h -connected component is obtained. Finally, we run the NI algorithm for each connected component to preserve the local connectivity structure.

Algorithm 1 describes the steps of CFNI, which takes as input a graph $G = (V, E)$ and a target local connectivity h . h can be selected as any positive integer, not necessarily equal to the k -connectivity of G .

The algorithm first checks for the connectivity k of G . If $h > k$, the algorithm decomposes G into $k + 1$ -connected components, and recursively calls CFNI for each $k + 1$ -connected component (lines 3-7). The recursion stops when CFNI is called on a graph whose k -connectivity is no lower than h ; at this step, NI is run on G (lines 11-26).

Once the recursive calls finish for all $k + 1$ -connected components, the local h -connectivity-preserving k -connected subgraph G_{CFNI} is finally computed using the union of the edge sets of the $k + 1$ -connected subgraphs of the $k + 1$ -connected components (line 8).

The runtime complexity of CFNI depends on the runtime of the h -connected component decomposition, while running NI on each component takes linear time. For example, the decomposition of one-connected (resp., biconnected) graphs into biconnected (resp., tricon-

nected) components takes linear time [17, 18]. Running NI on each h -connected component G_{h_i} takes linear time in the number of edges in G_i , which sums up to $O(m)$ due to the number of edges in all of the connected components adding up to m .

The number of edges in G_{CFNI} is bounded by $O(hn)$. At the lowest level of recursion, CFNI decomposes a graph G into h -connected components, where NI is run on each h -connected component $G_{h_i} = (V_{h_i}, E_{h_i})$ to produce a h -connected subgraph $G'_{h_i} = (V_{h_i}, E'_{h_i})$ with $O(h|V_{h_i}|)$ edges. As G_{CFNI} is formed using the union of all G'_{h_i} , and given that the sum of all $|V_{h_i}|$ is n , the number of edges in G_{CFNI} is bounded by $O(hn)$.

4 CFGD: Connectivity-Faithful Graph Drawing

One popular method commonly used to draw big complex graphs is by utilizing *graph sparsification* [10, 19, 28, 34, 37, 38]. Namely, given a graph G , first compute a much smaller sparsified graph G' , then compute a drawing D' of G' . Finally, the sparsified edges are added back to D' , to obtain a drawing D of the whole graph G . While this approach is efficient (i.e., it has a much faster runtime than drawing the whole graph G), the effectiveness (i.e., the quality of the drawing D) depends on how well the sparsification G' preserves the structure of G .

Due to the limitation of NI in preserving the local connectivity of highly connected components of a graph G , a naive application of NI for graph drawing may not be sufficient to represent all important connectivity structures of a graph faithfully. For example, a drawing of a one-connected graph G based on the spanning tree may fail to depict cycles or misrepresent locally dense subgraphs. We, therefore, present *CFGD*, which leverages CFNI for connectivity-faithful graph drawing to overcome the weakness of NI in preserving the local connectivity of highly connected components.

Algorithm 2 CFGD.

Step 1: Compute subgraph $G_{CFNI} = (V, E'_h)$ preserving global k -connectivity and local h -connectivity of k -connected graph G using CFNI.

Step 2: Compute a drawing $D_{G_{CFNI}}$ of G_{CFNI} using a graph drawing algorithm.

Step 3: Add all edges in $E_{r_h} = E \setminus E'_h$ to $D_{G_{CFNI}}$ to obtain a drawing D of G .

We expect CFGD to be able to compute high-quality connectivity-faithful drawings due to CFNI preserving not only the global k -connectivity of a graph G but also the local h -connectivity of each h -connected component of G , while still obtaining a fast runtime due to the efficient runtime of CFNI.

5 CFNI Experiment

5.1 NI Experiment

We first evaluate the baseline performance of NI for graph sparsification by comparing NI to *SS* (*Spectral Sparsification*), which has been shown to outperform stochastic sampling methods [20, 19, 10, 34]. In summary, NI outperforms SS on several connectivity-related sampling quality metrics, most notably on the connectivity-related metrics: Closeness Centrality at 52% better, and Betweenness Centrality at 20% better. The visual comparison also demonstrates the strengths of NI in preserving the overall connectivity structures that SS often fails to preserve, for *biconnected* graphs. Thus, both the quality metrics and visual comparisons demonstrate the strengths of NI over SS for connectivity-faithful sampling. For details of the experiment, see the journal version of this paper [35].

Although NI shows a good performance on biconnected graphs, for one-connected graphs, the performance of NI deteriorates since the spanning tree misses the local connectivity structures of graphs, such as cycles and clusters. We, therefore, conduct experiments to evaluate how CFNI improves upon NI for globally sparse and locally dense one-connected graphs.

5.2 CFNI Experiment Design

We now present comparison experiments to evaluate the strengths of CFNI over NI. Specifically, we use one-connected graphs as inputs with $h = 2, 3$, since efficient linear-time algorithms are known for computing biconnected components and triconnected components [18, 17]. We denote the sparsification of a graph G computed by NI as G_1 , as $k = 1$ for the one-connected graphs. We then denote the sparsification computed by CFNI with $h = 2, 3$ as G_2 and G_3 .

We use a mix of real-world and synthetic graphs with various connectivity structures: 1) real-world benchmark *scale-free* graphs, with globally sparse, locally dense clusters and small diameters [24]; 2) *GION* graphs, biochemical networks with globally sparse, locally dense clusters and long diameters [25]; 3) *mesh* graphs, with regular grid-like structures [7]; and 4) *black-hole* graphs, synthetic graphs with globally sparse mesh- or cycle-like structures with locally dense “blobs” attached [10]. See Table 1 for details.

■ **Table 1** Data sets for the CFNI experiments.

(a) Scale-free.

G	$ V $	$ E $
soc_h	2000	16097
block_2000	2000	3992
offlights	2905	15645
tvcg	3213	10140
facebook	4039	88234
CA-GrQc	4158	13422
EVA	4475	4652
us_powergrid	4941	6594
as19990606	5188	9930
migrations	6025	9378
lastfm_asia	7624	27806

(b) Mesh.

G	$ V $	$ E $
dwt_1005	1005	4813
cage8	1015	4994
bcsstk09	1083	8677
nasa1824	1824	18692
plat1919	1919	15240
sierpinski3d	2050	6144
data	2851	15093
3elt	4720	13722

(c) GION.

G	$ V $	$ E $
2_gion	1159	6424
5_gion	1748	13957
6_gion	1785	20459
7_gion	3010	41757
8_gion	4924	52502
4_gion	5953	186279
1_gion	5452	118404
3_gion	7885	427406

(d) Black-hole.

G	$ V $	$ E $
G443	285	2009
Cycle759	377	4790
G462	733	62509
Cycle907	823	14995
Cycle896	1031	22638
G500	1080	17636
G887	4784	38135

5.3 Quality Metrics Comparison

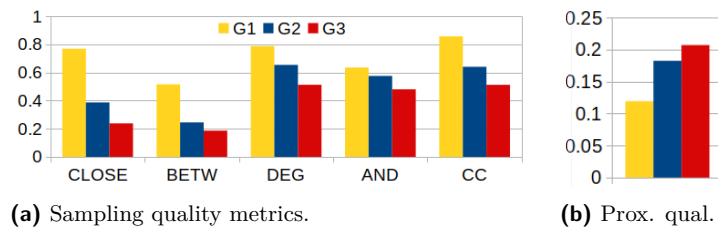
We use the well-known *sampling quality metrics* to measure how well the sparsifications preserve the following properties of the original graphs [21, 23]:

- Closeness Centrality (CLOSE) computes the “closeness” of a vertex to other vertices by summing up the length of all shortest paths between a vertex and all the other vertices [11].
- Betweenness Centrality (BETW) measures the ratio of all shortest paths between each pair of vertices that pass through a certain vertex [11].
- Degree Correlation Associativity (DEG) computes the likelihood that vertices link to other vertices of similar degrees [42].
- Average Neighbor Degree (AND) computes the average degree of a vertex’s neighbors [2].
- Clustering Coefficient (CC) measures the clustering of edges into tightly connected neighborhoods and represents the extent of clustering tendency between vertices [47].

More specifically, we measure the sampling quality metrics using the *Kolmogorov-Smirnov (KS)* goodness-of-fit-test [5], to compare the similarity of the CDF (Cumulative Distributive Function) of each graph metric of the original and sparsified graphs. The KS distance has a value between 0 and 1, where 0 means completely identical CDFs.

We compute the percentage ratio of the difference to compare the metrics computed by G_1 (i.e., computed by NI) and G_2, G_3 (i.e., computed by CFNI). For example, to compute the percentage difference of AND computed by G_1 and G_3 , we use the formula $\frac{AND(G_1) - AND(G_3)}{AND(G_1)}$.

Figure 3a shows the sampling quality metrics computed on G_1, G_2 , and G_3 , averaged over all data sets. Clearly, G_2 and G_3 achieve notably better sampling quality metrics over G_1 , and G_3 further obtains better metrics over G_2 . The largest improvements are seen on the connectivity-related metrics Closeness centrality and Betweenness centrality: averaged over both, G_2 and G_3 obtain 51% and 66% improvements, respectively, compared to G_1 . Improvements can also be seen over the other three metrics, with G_2 and G_3 obtaining 17% and 33% improvements, respectively, over G_1 .



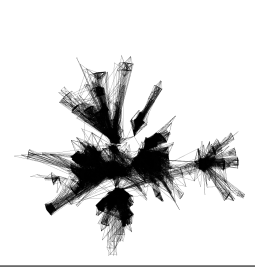
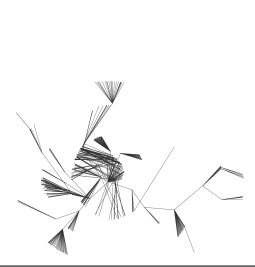
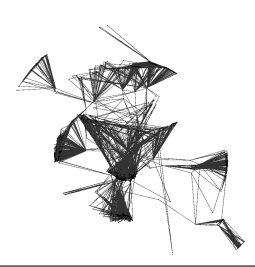
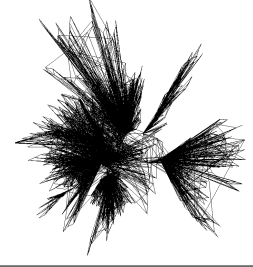
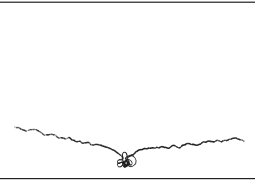
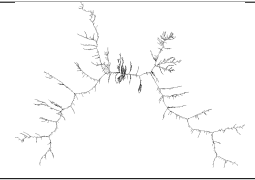
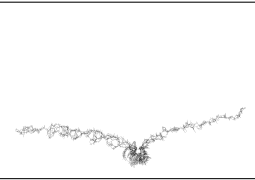
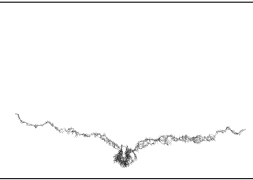
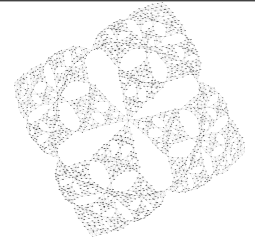

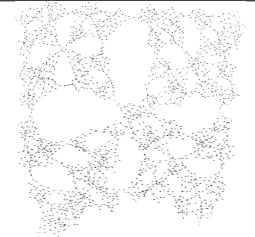
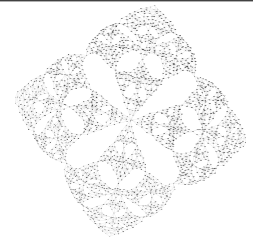
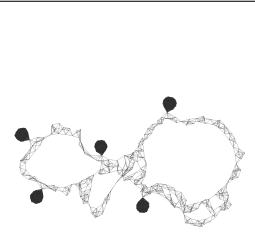

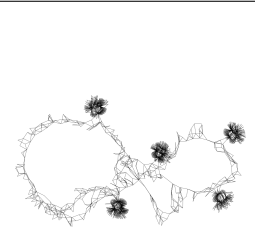
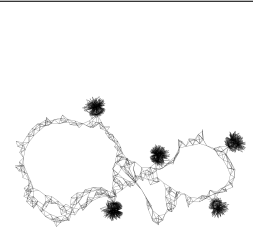
■ **Figure 3** Average sampling (lower = better) and proxy quality metrics (higher = better) for G_1, G_2 , and G_3 . On average, G_3 obtains significantly better metrics than G_1 (i.e., NI), especially on connectivity-related metrics CLOSE and BETW at 66% better on average.

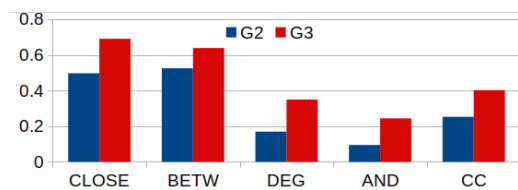
To evaluate the effectiveness of the sparsifications for the purpose of graph drawing, we compute the *proxy quality metrics* [44], for measuring how faithfully the drawing of the sparsifications represents the ground truth structure of the original graph. We use the *Backbone* layout, specifically designed to untangle “hairball” drawings of large complex graphs [45], to draw G_1, G_2 , and G_3 .

Figure 3b shows the proxy quality metrics computed on G_1, G_2 , and G_3 , averaged over all data sets. Similar to the results for sampling quality metrics, G_2 and G_3 obtain notably better proxy quality metrics than G_1 , on average 53% better by G_2 and 73% better by G_3 .

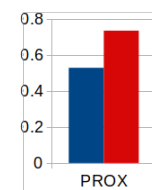
17:10 Connectivity-Faithful Graph Drawing

■ **Table 2** Visual comparison of the sparsified graphs computed by CFNI. G_2 and G_3 (by CFNI) consistently preserve the structure of the graph G better than G_1 (by NI), with G_3 significantly outperforming G_2 on some graphs, e.g., Facebook and Sierpinski3d.

G	G_1	G_2	G_3
facebook			
			
GION_1			
			
sierpinski3d			
			
Cycle896			
			



(a) Sampling metrics.



(b) Proxy metrics.

■ **Figure 4** Average improvements by G_2 and G_3 (computed by CFNI) over G_1 (computed by NI). CFNI obtains improvement over NI on all metrics, most significantly on connectivity-related sampling quality metrics CLOSE and BETW.

5.4 Visual Comparison

Table 2 shows example visual comparisons between the drawings of sparsifications computed by NI and CFNI, with the drawings of the whole graph G added as reference. Clearly, the drawings of G_2 and G_3 are very similar to those of G , i.e., they faithfully represent the original graph’s structure, while G_1 often fails to preserve the structure of G . For example, see the GION graph GION_1, where G_1 misleadingly shows four “branches” expanding from the middle cluster while G_2 and G_3 show only two, more faithful to the original G .

Moreover, sometimes only G_3 is highly similar to G , while G_2 also fails to preserve the structure of G . For example, see the mesh graph Sierpinski3d, where G_1 completely fails to maintain the mesh structure of the original graph, and while G_2 manages to maintain the structure better, it is still distorted compared to G . Meanwhile, G_3 displays almost the same structure as G .

5.5 Discussion and Summary

Extensive experiments have demonstrated the strengths of CFNI over NI, preserving both the global and local connectivity structures of graphs. Figure 4 shows the average improvements obtained by G_2 and G_3 (i.e., running CFNI with $h = 2$ and $h = 3$ respectively) over G_1 (i.e., running NI). The largest improvements are seen in Closeness centrality and Betweenness centrality, which are both distance-based centralities highly related to connectivity. On average, these improvements are 51% better for G_2 and 66% better for G_3 . Significant improvements are also seen in proxy quality metrics, at 53% better for G_2 and 73% for G_3 . In addition, G_3 further obtains an average 31% improvement over G_2 averaged between Closeness centrality and Betweenness centrality, and 13% improvement for proxy quality metrics over G_2 .

The visual comparisons in Table 2 validate the quality metrics, showing that G_3 (computed using CFNI with $h = 3$) represents both global and local connectivity structures of graphs much more faithfully than G_1 (computed by NI). In particular, for globally sparse and locally dense graphs such as the scale-free and black-hole graphs, G_3 faithfully represents both the overall global shape and the locally dense clusters better than G_1 , improving the limitations of NI. Furthermore, G_3 also outperforms G_2 in cases where G_2 still has limitations capturing the structure of G , such as seen in the Facebook graph, where the drawing of G_3 is much more similar to G compared to that of G_2 .

6 CFGD Experiment

6.1 Experiment Design

We now present experiments to evaluate the effectiveness of the CFGD approach, over a naive application of NI to graph drawing. We performed an initial experiment for the naive application of NI to graph drawing: given a k -connected graph G , we first compute the k -connected subgraph $G_{NI} = (V, E')$ using NI, then compute a drawing $D_{G_{NI}}$ of G_{NI} , and finally add back the edges in $E_r = E \setminus E'$ to produce the drawing $D_{G_{NI}+E_r}$ of G . On average, computing $D_{G_{NI}+E_r}$ is 30% faster than directly computing a drawing D of G (i.e., applying a graph drawing algorithm directly on G), with on average 11% better edge crossing and only 15% lower shape-based metrics and neighborhood preservation. However, stress is significantly higher, at 55% higher on average. For details, see the journal version of this paper [35].

We thus present experiments to evaluate how CFGD improves over a naive application of NI to graph drawing. For the CFGD experiments, we use *one-connected* graphs, with $h = 2, 3$, with the same data set as used in Section 5, and use the Backbone layout [45] for its strengths in untangling “hairballs” in drawings of large, complex graphs. We denote the drawing obtained using the sparsified graph computed by NI as D_1 , corresponding to the notation G_1 for the result of running NI on a 1-connected graph G used in Section 5. Similarly, we use D_h to simplify the notation $D_{G_{CFNI+E_{r_h}}}$ used to denote the resulting drawing from running CFGD on a graph G , i.e., we denote the drawing computed by CFGD using $h = 2$ and $h = 3$ as D_2 and D_3 respectively.

6.2 Runtime Comparison

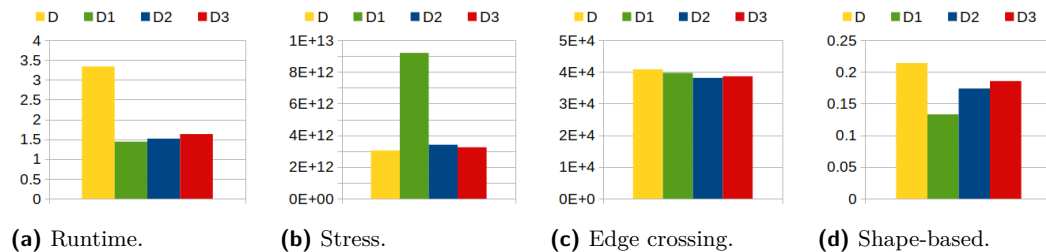
Figure 5a shows the average runtimes of computing D_1 , D_2 , D_3 compared to computing a drawing D directly from G . CFGD always runs significantly faster than drawing G directly, with over 50% runtime improvement on both D_2 and D_3 . On average, the runtime of computing D_3 is still very similar to D_1 , at only around 5% difference in runtime improvement over D , showing that CFGD still preserves much of the runtime improvement obtained by a naive application of NI for graph drawing.

6.3 Quality Metrics Comparison

To evaluate the performance of CFGD, we compare its results to those obtained from drawing a graph directly, using graph drawing quality metrics. We use a selection of commonly-used graph drawing quality metrics: stress [8], edge crossing, and shape-based metrics [10, 15]. See Section 2.4 for details on the metrics.

Stress. Figure 5b shows the average stress of D , D_1 , D_2 , and D_3 . On stress, we see the largest improvement obtained by CFGD over a naive application of NI for graph drawing: D_2 and D_3 obtain much lower stress than D_1 , at over 62% lower on average. This also brings the stress to be relatively similar to that of D , at only about 7% difference for D_3 , in contrast to D_1 obtaining over two times higher stress than D .

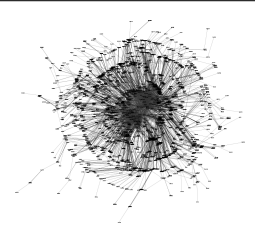
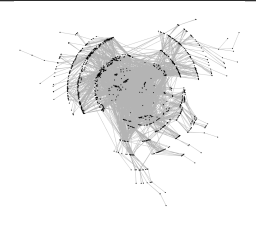
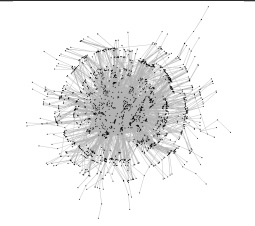
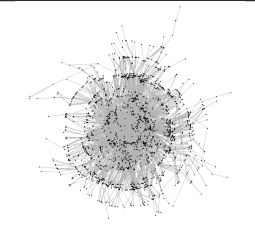
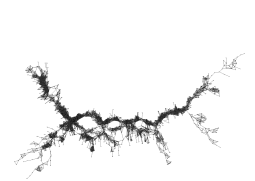
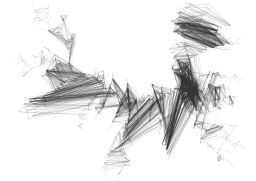
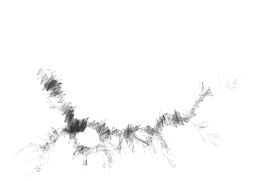
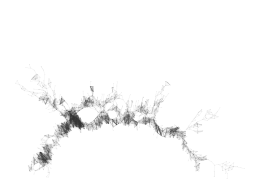
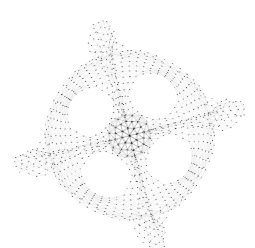
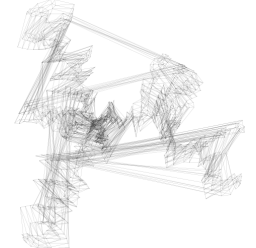
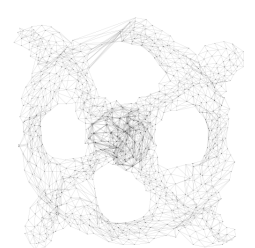
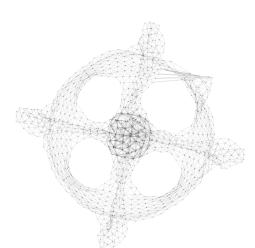
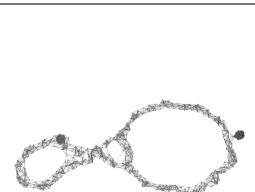
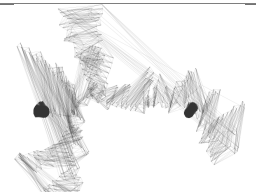
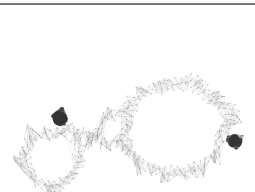
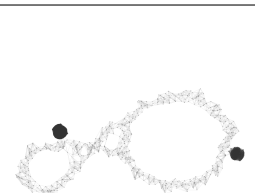
Edge crossing. Figure 5c shows the average edge crossing metrics on D , D_1 , D_2 , and D_3 . Surprisingly, even on D_1 , edge crossing is already almost the same as D , at only 3% lower on average. D_2 and D_3 also show good performance, both at around 3% lower than D_1 on average, and furthermore even slightly better than D at around 6% better on average.



■ **Figure 5** Average runtime and quality metrics (lower is better for stress and edge crossing, and higher is better for shape-based) of computing D_1 , D_2 , and D_3 compared to computing D directly on G . CFGD (D_2 and D_3) obtains significant runtime improvements over computing D directly on G , while obtaining significantly lower stress than D_1 and similar metrics to D .

Shape-based metrics. Figure 5d shows the average shape-based metrics of D , D_1 , D_2 , and D_3 . D_2 and D_3 show notable improvements over D_1 , on average 31% and 40% higher. Furthermore, this brings the shape-based metrics of the drawings computed by CFGD closer to those of the drawings computed directly from G : with D_3 , the shape-based metrics are around 13% lower than D , significantly lower than the 51% improvement in runtime.

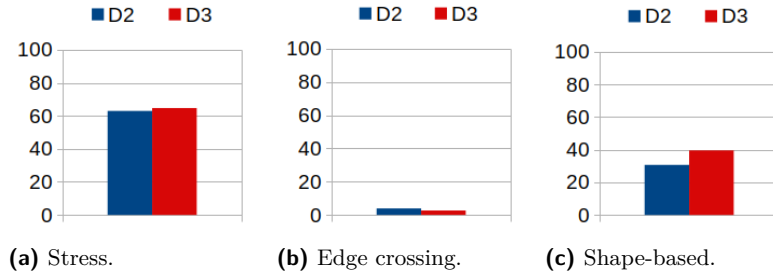
■ **Table 3** Visual comparison of the drawings computed by CFGD. D_2 and D_3 (by CFGD) clearly depict the structures of the graphs more faithfully than D_1 , with D_3 further removing some distortion issues occasionally displayed by D_2 (e.g., on `dwt_1005`).

D	D_1	D_2	D_3
migrations			
			
GION_5			
			
dwt_1005			
			
Cycle907			
			

6.4 Visual Comparison

Table 3 shows some example visual comparisons of CFGD to directly drawing graph G . It can be seen that in general, drawing D_1 , i.e., drawing the sparsification computed by NI, often fails to maintain the structure of G , as can be seen in the direct drawing D . D_2 and D_3 are often far more successful in preserving the structures of graphs, such as those seen in the scale-free graph migrations and the GION graph GION_5.

In other cases, D_3 still succeeds in preserving the structure when even D_2 fails. For example, with the mesh graph `dwt_1005`, D_2 manages to maintain the overall four-pronged shape, but the drawing is still distorted compared to D . Similarly, for the black-hole graph `Cycle907`, although D_2 preserves both the local blobs and the overall “cycle”-like global structure, the shape is somewhat distorted with “zig-zags”. Meanwhile, D_3 of both graphs mostly eliminates the distortion in the drawings compared to D_2 .



■ **Figure 6** Average improvements (in %) in quality metrics computed by D_2 and D_3 over D_1 , i.e., improvement of CFGD over a naive application of NI to graph drawing. CFGD obtains improvements on all quality metrics, with the largest improvement on stress at over 63%.

6.5 Discussion and Summary

Our experiments have demonstrated the effectiveness of CFGD. Figure 6 shows the average improvements in quality metrics obtained by D_2 and D_3 over D_1 . In particular, the largest improvement is seen on stress: D_3 obtains on average 62% lower stress than D_1 . Looking at the visual comparison, drawings D_1 often contain very long edges between vertices that neighbor each other in the original graph G but are in distant branches in the spanning tree, leading to high stress. Meanwhile, these long edges are absent in D_2 and D_3 , leading to much lower stress compared to D_1 .

Surprisingly, D_1 , D_2 , and D_3 obtain edge crossings very similar to D , even slightly better at 3%, 6%, and 6% lower on average, respectively. Most of this improvement is on scale-free and black-hole graphs, both containing graphs with globally sparse, locally dense structures. One possible explanation can be seen from the black-hole graphs, such as can be seen in graph `Cycle907` in Table 3 where the locally dense blobs are drawn with a larger area in drawings D_1, D_2, D_3 compared to D . This may have removed some of the edge crossings introduced in D due to the blob being compressed into a smaller drawing area.

7 Conclusion

We present the first study of connectivity-faithful graph drawing, by leveraging the NI algorithm to graph sparsification and drawing. Specifically, we present local connectivity-preserving divide-and-conquer approaches CFNI and CFGD, to improve on the limitations of NI by not only preserving the global k -connectivity of a k -connected graph G , but also preserving the local connectivities of h -connected components of G , where $h > k$.

We demonstrate the effectiveness of CFNI over a naive application of NI, obtaining up to 66% average better connectivity-related sampling quality metrics and 73% better proxy quality metrics over NI. We also demonstrate the effectiveness of CFGD over a naive application of NI to graph drawing, most notably with 62% lower stress; CFGD also runs 51% faster than directly drawing the whole graph with similar quality metrics.

Future work includes evaluations of CFNI and CFGD using higher local connectivity.

References

- 1 Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.
- 2 Alain Barrat, Marc Barthélemy, Romualdo Pastor-Satorras, and Alessandro Vespignani. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences*, 101(11):3747–3752, March 2004.
- 3 Shijun Cai, Seok-Hee Hong, Amyra Meidiana, Peter Eades, and Daniel Keim. Cluster-faithful graph visualization: New metrics and algorithms. In *IEEE PacificVis*, pages 192–201, 2024. doi:10.1109/PACIFICVIS60374.2024.00029.
- 4 Shijun Cai, Amyra Meidiana, and Seok-Hee Hong. Dnc: Dynamic neighborhood change faithfulness metrics. In *EuroVis*, pages 49–53, 2022. doi:10.2312/EVS.20221092.
- 5 Indra Mohan Chakravarti, Radha Govira Laha, and Jogabrata Roy. Handbook of methods of applied statistics. *Wiley Series in Probability and Mathematical Statistics*, 1967.
- 6 Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.
- 7 Timothy A Davis and Yifan Hu. The university of florida sparse matrix collection. *Transactions on Mathematical Software*, 38(1):1, 2011.
- 8 Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall PTR, 1998.
- 9 Peter Eades, Seok-Hee Hong, An Nguyen, and Karsten Klein. Shape-based quality metrics for large graph visualization. *JGAA*, 21(1):29–53, 2017. doi:10.7155/JGAA.00405.
- 10 Peter Eades, Quan Nguyen, and Seok-Hee Hong. Drawing big graphs using spectral sparsification. In *Graph Drawing*, pages 272–286, 2017.
- 11 Linton C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215–239, 1978.
- 12 Emden R Gansner, Yehuda Koren, and Stephen North. Graph drawing by stress majorization. In *Graph Drawing*, pages 239–250, 2005.
- 13 Michael R Garey and David S Johnson. *Computers and Intractability*, volume 174. freeman San Francisco, 1979.
- 14 Robert Gove. A random sampling $O(n)$ force-calculation algorithm for graph layouts. *Computer Graphics Forum*, 38(3):739–751, 2019. doi:10.1111/CGF.13724.
- 15 Seok-Hee Hong, Amyra Meidiana, James Wood, Juan Pablo Ataide, Peter Eades, and Kunsoo Park. dgg, drng, dsc: New degree-based shape-based faithfulness metrics for large and complex graph visualization. In *IEEE PacificVis*, pages 51–60, 2022. doi:10.1109/PACIFICVIS53943.2022.00014.
- 16 Seok-Hee Hong, Quan Nguyen, Amyra Meidiana, Jiayi Li, and Peter Eades. BC tree-based proxy graphs for visualization of big graphs. In *IEEE PacificVis*, pages 11–20, 2018.
- 17 John Hopcroft and Robert Tarjan. Algorithm 447: efficient algorithms for graph manipulation. *Communications of the ACM*, 16(6):372–378, 1973.
- 18 John E Hopcroft and Robert Endre Tarjan. Dividing a graph into triconnected components. *SIAM Journal on computing*, 2(3):135–158, 1973. doi:10.1137/0202012.
- 19 Jingming Hu, Tuan Tran Chu, Seok-Hee Hong, Jialu Chen, Amyra Meidiana, Marnijati Torkel, Peter Eades, and Kwan-Liu Ma. BC tree-based spectral sampling for big complex network visualization. *Appl. Netw. Sci.*, 6(1):60, 2021. doi:10.1007/S41109-021-00405-3.
- 20 Jingming Hu, Seok-Hee Hong, and Peter Eades. Spectral vertex sampling for big complex graphs. In *Complex Networks*, pages 216–227, 2019. doi:10.1007/978-3-030-36683-4_18.
- 21 Pili Hu and Wing Cheong Lau. A survey and taxonomy of graph sampling. *arXiv preprint arXiv:1308.5865*, 2013. arXiv:1308.5865.
- 22 Tomihisa Kamada, Satoru Kawai, et al. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15, 1989. doi:10.1016/0020-0190(89)90102-6.
- 23 Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *ACM SIGKDD*, pages 631–636, 2006. doi:10.1145/1150402.1150479.


- 24 Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. <http://snap.stanford.edu/data>, June 2014.
- 25 Michael R Marner, Ross T Smith, Bruce H Thomas, Karsten Klein, Peter Eades, and Seok-Hee Hong. GION: interactively untangling large graphs on wall-sized displays. In *Graph Drawing*, pages 113–124, 2014. doi:10.1007/978-3-662-45803-7_10.
- 26 Rafael Messias Martins, Rosane Minghim, Alexandru C Telea, et al. Explaining neighborhood preservation for multidimensional projections. In *CGVC*, pages 7–14, 2015.
- 27 David W Matula. Determining edge connectivity in $o(nm)$. In *SCFS*, pages 249–251, 1987. doi:10.1109/SFCS.1987.19.
- 28 Amyra Meidiana, Seok-Hee Hong, Shijun Cai, Marnijati Torkel, and Peter Eades. Sublinear-force: Fully sublinear-time force computation for large complex graph drawing. *IEEE TVCG*, 30(7):3386–3399, 2024. doi:10.1109/TVCG.2022.3233287.
- 29 Amyra Meidiana, Seok-Hee Hong, and Peter Eades. New quality metrics for dynamic graph drawing. In *Graph Drawing*, pages 450–465, 2020. doi:10.1007/978-3-030-68766-3_35.
- 30 Amyra Meidiana, Seok-Hee Hong, and Peter Eades. Shape-faithful graph drawings. In *Graph Drawing*, pages 93–108, 2022. doi:10.1007/978-3-031-22203-0_8.
- 31 Amyra Meidiana, Seok-Hee Hong, Peter Eades, and Daniel Keim. A quality metric for visualization of clusters in graphs. In *Graph Drawing*, pages 125–138, 2019. doi:10.1007/978-3-030-35802-0_10.
- 32 Amyra Meidiana, Seok-Hee Hong, Peter Eades, and Daniel Keim. Quality metrics for symmetric graph drawings. In *IEEE PacificVis*, pages 11–15, 2020. doi:10.1109/PACIFICVIS48177.2020.1022.
- 33 Amyra Meidiana, Seok-Hee Hong, Peter Eades, and Daniel A. Keim. Automorphism faithfulness metrics for symmetric graph drawings. *IEEE TVCG*, 30(7):3241–3255, 2024. doi:10.1109/TVCG.2022.3229354.
- 34 Amyra Meidiana, Seok-Hee Hong, Jiajun Huang, Peter Eades, and Kwan-Liu Ma. Topology-based spectral sparsification. In *LDAV*, pages 73–82, 2019. doi:10.1109/LDAV48142.2019.8944358.
- 35 Amyra Meidiana, Seok-Hee Hong, and Yongcheng Jing. Connectivity-Faithful Graph Sparsification for Drawing Large and Complex Graphs. *Submitted*, 2024.
- 36 Amyra Meidiana, Seok-Hee Hong, Marnijati Torkel, Shijun Cai, and Peter Eades. Sublinear Time Force Computation for Big Complex Network Visualization. *Computer Graphics Forum*, 39(3):579–591, 2020. doi:10.1111/CGF.14003.
- 37 Amyra Meidiana, Seok-Hee Hong, Marnijati Torkel, Shijun Cai, and Peter Eades. Sublinear time force computation for big complex network visualization. *Computer Graphics Forum*, 39(3):579–591, 2020. doi:10.1111/CGF.14003.
- 38 Amyra Meidiana, James Wood, and Seok-Hee Hong. Sublinear-time algorithms for stress minimization in graph drawing. In *IEEE PacificVis*, pages 166–175, 2021. doi:10.1109/PACIFICVIS52677.2021.00030.
- 39 Hiroshi Nagamochi and Toshihide Ibaraki. A linear-time algorithm for finding a sparse k -connected spanning subgraph of ak -connected graph. *Algorithmica*, 7(1):583–596, 1992. doi:10.1007/BF01758778.
- 40 Hiroshi Nagamochi and Toshihide Ibaraki. *Algorithmic Aspects of Graph Connectivity*, volume 123 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2008. doi:10.1017/CB09780511721649.
- 41 Dalit Naor, Dan Gusfield, and Charles Martel. A fast algorithm for optimally increasing the edge connectivity. *SIAM Journal on Computing*, 26(4):1139–1165, 1997. doi:10.1137/S0097539792234226.
- 42 M. E. J. Newman. Mixing patterns in networks. *Physical Review E*, 67(2), February 2003.
- 43 Quan Nguyen, Peter Eades, and Seok-Hee Hong. On the faithfulness of graph visualizations. In *IEEE PacificVis*, pages 209–216, 2013.

- 44 Quan Hoang Nguyen, Seok-Hee Hong, Peter Eades, and Amyra Meidiana. Proxy graph: Visual quality metrics of big graph sampling. *IEEE TVCG*, 23(6):1600–1611, 2017. doi:10.1109/TVCG.2017.2674999.
- 45 Arlind Nocaj, Mark Ortman, and Ulrik Brandes. Untangling hairballs - from 3 to 14 degrees of separation. In *Graph Drawing*, 2014.
- 46 Mark Ortman, Mirza Klimenta, and Ulrik Brandes. A sparse stress model. In Yifan Hu and Martin Nöllenburg, editors, *Graph Drawing*, pages 18–32, 2016. doi:10.1007/978-3-319-50106-2_2.
- 47 Jari Saramäki, Mikko Kivelä, Jukka-Pekka Onnela, Kimmo Kaski, and János Kertész. Generalizations of the clustering coefficient to weighted complex networks. *Physical Review E*, 75:027105, March 2007.
- 48 Daniel A Spielman. Spectral graph theory and its applications. In *IEEE FOCS*, pages 29–38, 2007. doi:10.1109/FOCS.2007.66.
- 49 Daniel A Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4):981–1025, 2011. doi:10.1137/08074489X.
- 50 Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2007. doi:10.1007/S11222-007-9033-Z.
- 51 Yanhong Wu, Nan Cao, Daniel Archambault, Qiaomu Shen, Huamin Qu, and Weiwei Cui. Evaluation of graph sampling: A visualization perspective. *IEEE TVCG*, 23(1):401–410, 2016. doi:10.1109/TVCG.2016.2598867.
- 52 Jonathan X Zheng, Samraat Pawar, and Dan FM Goodman. Graph drawing by stochastic gradient descent. *IEEE TVCG*, 25(9):2738–2748, 2018. doi:10.1109/TVCG.2018.2859997.

On the Uncrossed Number of Graphs

Martin Balko ✉ 

Department of Applied Mathematics, Faculty of Mathematics and Physics,
Charles University, Prague, Czech Republic

Petr Hliněný ✉ 

Faculty of Informatics, Masaryk University, Brno, Czech Republic

Tomáš Masařík ✉ 

Institute of Informatics, Faculty of Mathematics, Informatics and Mechanics,
University of Warsaw, Poland

Joachim Orthaber ✉ 

Institute of Software Technology, Graz University of Technology, Austria

Birgit Vogtenhuber ✉ 

Institute of Software Technology, Graz University of Technology, Austria

Mirko H. Wagner ✉ 

Institute of Computer Science, Osnabrück University, Germany

Abstract

Visualizing a graph G in the plane nicely, for example, without crossings, is unfortunately not always possible. To address this problem, Masařík and Hliněný [GD 2023] recently asked for each edge of G to be drawn without crossings while allowing multiple different drawings of G . More formally, a collection \mathcal{D} of drawings of G is *uncrossed* if, for each edge e of G , there is a drawing in \mathcal{D} such that e is uncrossed. The *uncrossed number* $\text{unc}(G)$ of G is then the minimum number of drawings in some uncrossed collection of G .

No exact values of the uncrossed numbers have been determined yet, not even for simple graph classes. In this paper, we provide the exact values for uncrossed numbers of complete and complete bipartite graphs, partly confirming and partly refuting a conjecture posed by Hliněný and Masařík [GD 2023]. We also present a strong general lower bound on $\text{unc}(G)$ in terms of the number of vertices and edges of G . Moreover, we prove NP-hardness of the related problem of determining the *edge crossing number* of a graph G , which is the smallest number of edges of G taken over all drawings of G that participate in a crossing. This problem was posed as open by Schaefer in his book [Crossing Numbers of Graphs 2018].

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry; Mathematics of computing \rightarrow Graph theory

Keywords and phrases Uncrossed Number, Crossing Number, Planarity, Thickness

Digital Object Identifier 10.4230/LIPIcs.GD.2024.18

Related Version *Full Version:* <https://arxiv.org/abs/2407.21206> [1]

Funding *Martin Balko:* Supported by grant no. 23-04949X of the Czech Science Foundation (GAČR) and by the Center for Foundations of Modern Computer Science (Charles Univ. project UNCE 24/SCI/008).

Tomáš Masařík: Supported by the Polish National Science Centre SONATA-17 grant number 2021/43/D/ST6/03312.

Joachim Orthaber: Supported by the Austrian Science Fund (FWF) grant W1230.

Acknowledgements We would like to thank the organizers of the Crossing Numbers Workshop 2023 where this research was initiated.



© Martin Balko, Petr Hliněný, Tomáš Masařík, Joachim Orthaber, Birgit Vogtenhuber, and Mirko H. Wagner;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 18; pp. 18:1–18:13

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

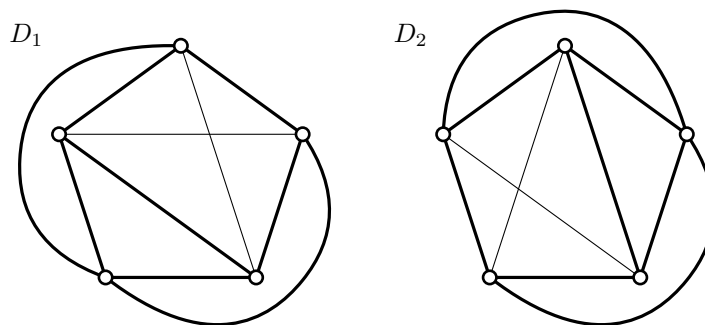
In a *drawing* of a graph G , the vertices are represented by distinct points in the plane and each edge corresponds to a simple continuous arc connecting the images of its end-vertices. As usual, we identify the vertices and their images, as well as the edges and the line segments representing them. We require that the edges pass through no vertices other than their endpoints. We assume for simplicity that any two edges have only finitely many points in common, no two edges touch at an interior point and no three edges meet at a common interior point.

A *crossing* in a drawing D of G is a common interior point of two edges of D where they properly cross. For a drawing D of a graph G , we say that an edge e of D is *uncrossed* in D if it does not share a crossing with any other edge of D .

There are two staple problems in the graph drawing field that defined the past eighty years of development in the area. The first one, dating back to World War II times [2, 21], is the problem of determining the *crossing number* $cr(G)$ of a graph G , defined as the smallest number of crossings required in any drawing of G in the plane. The crossing number problem has been intensively studied ever since, especially in the past thirty years. Computing the crossing number is NP-hard on general graphs [3], and one can find a thorough overview of the area in a recent book by Schaefer [19].

The second, only slightly newer problem, is that of determining the *thickness* $\theta(G)$ of a graph G , defined as the smallest integer k such that G can be edge-partitioned into k planar graphs. This problem was proposed by Harary [7] in 1961 and since then this concept has played an important role in graph drawing. Unlike for planarity, deciding whether a graph is *bipolar*, that is whether $\theta(G) \leq 2$, is NP-complete [14]. For an overview of the progress up to 1998, consult a survey by Mutzel, Odenthal, and Scharbrodt [16].

In this paper, we investigate a very recent notion inspired by a fusion of both concepts into one. We say that a collection $\mathcal{D}(G)$ of drawings of G is *uncrossed* if for each edge e of G there is at least one drawing in $\mathcal{D}(G)$ in which e is uncrossed; see Figure 1 for an example. Hliněný and Masařík [11], in relation to extensions of the traditional crossing number of a graph, defined the *uncrossed number* $unc(G)$ of a graph G as the smallest size of an uncrossed collection of drawings of G .



■ **Figure 1** An uncrossed collection $\mathcal{D}(K_5) = \{D_1, D_2\}$ that shows $unc(K_5) \leq 2$. The edges that are uncrossed are shown in thick lines. Since K_5 is not planar, we have $unc(K_5) = 2$.

The motivation for the uncrossed number [11] is that finding a handful of different drawings of a graph G instead of just one “flawless” drawing shall highlight different aspects of G and thus could be useful for the visualization of G , besides the theoretical interest. The requirement that each edge is uncrossed in at least one drawing is a natural way to enforce that the drawings will highlight each aspect of the graph as a whole.

Let us also formulate the decision version of the problem of determining $\text{unc}(G)$ of a given graph G .

UNCROSSEDNUMBER

Input: A graph G and a positive integer k .

Question: Are there drawings D_1, \dots, D_k of G such that, for each edge $e \in E(G)$, there exists an $i \in [k]$ such that e is uncrossed in the drawing D_i ?

Clearly, for every graph G , we have

$$\text{unc}(G) \geq \theta(G), \quad (1)$$

because the uncrossed edges in each drawing of an uncrossed collection of G induce an edge-partition of G into planar graphs. However, this new concept significantly differs from thickness (which just partitions the edges of G) in the sense that all edges of G have to be present along with the uncrossed subdrawing in each drawing of our uncrossed collection. In fact, the requirements of an uncrossed collection bring us close to the related notion of the *outerthickness* $\theta_o(G)$ of G , which is the minimum number of outerplanar graphs into which we can decompose G .

OUTERTHICKNESS

Input: A graph G and a positive integer k .

Question: Can G be decomposed into k outerplanar graphs?

As noted by Hliněný and Masařík [11], given a decomposition $\{G_1, \dots, G_k\}$ of G into outerplanar graphs, we can let D_i be an outerplanar drawing of G_i with all remaining edges of G being drawn in the outer face. This gives us

$$\text{unc}(G) \leq \theta_o(G) \quad (2)$$

for every graph G . Combining this with a result of Gonçalves [4], which implies $\theta_o(G) \leq 2\theta(G)$, we actually obtain the following chain of inequalities

$$\frac{1}{2}\theta_o(G) \leq \theta(G) \leq \text{unc}(G) \leq \theta_o(G) \leq 2\theta(G). \quad (3)$$

So far, the exact values of uncrossed numbers are not very well understood. Masařík and Hliněný [11] exactly determined $\text{unc}(G)$ of only a few sporadic examples of graphs G , such as $\text{unc}(K_7) = 3$.

Our Results

We determine the exact values of uncrossed numbers for specific and natural graph classes. First, we derive the formula for the uncrossed number of complete graphs.

► **Theorem 1.** *For every positive integer n , it holds that*

$$\text{unc}(K_n) = \begin{cases} \lceil \frac{n+1}{4} \rceil, & \text{for } n \notin \{4, 7\} \\ 3, & \text{for } n = 7 \\ 1, & \text{for } n = 4. \end{cases}$$

We also find the exact formula for the uncrossed number of complete bipartite graphs.

► **Theorem 2.** *For all positive integers m and n with $m \leq n$, it holds that*

$$\text{unc}(K_{m,n}) = \begin{cases} \left\lceil \frac{mn}{2m+n-2} \right\rceil, & \text{for } m \leq n \leq 2m-2 \\ \left\lceil \frac{mn}{2m+n-1} \right\rceil, & \text{for } n = 2m-1 \\ \left\lceil \frac{mn}{2m+n} \right\rceil, & \text{for } 6 \leq 2m \leq n \\ 1, & \text{for } m \leq 2. \end{cases}$$

Let us mention that the exact values of the thickness $\theta(K_{m,n})$ of complete bipartite graphs are not known for all values of m and n ; see [17] for further discussion.

We compare our formulas on $\text{unc}(K_n)$ and $\text{unc}(K_{m,n})$ with known formulas on the outerthickness of K_n and $K_{m,n}$. Hliněný and Masařík [11, Section 6] conjectured the uncrossed numbers and outerthickness to be the same for both complete and complete bipartite graphs except in the planar but not outerplanar cases. Guy and Nowakowski [5, 6] showed that

$$\theta_o(K_n) = \begin{cases} \left\lceil \frac{n+1}{4} \right\rceil, & \text{for } n \neq 7 \\ 3, & \text{for } n = 7 \end{cases} \quad (4)$$

and

$$\theta_o(K_{m,n}) = \left\lceil \frac{mn}{2m+n-2} \right\rceil \quad (5)$$

for all positive integers m and n with $m \leq n$. Note that it follows from Theorem 1 and Equation (4) that $\text{unc}(K_n) = \theta_o(K_n)$ for every $n \neq 4$. For $n = 4$, we have $\text{unc}(K_4) = 1$ while $\theta_o(K_4) = 2$. This confirms the conjecture of Hliněný and Masařík [11] in the case of complete graphs.

Since

$$\left\lceil \frac{mn}{2m+n} \right\rceil \leq \left\lceil \frac{mn}{2m+n-2} \right\rceil = \left\lceil \frac{mn}{2m+n} + \frac{2mn}{(2m+n-2)(2m+n)} \right\rceil \leq \left\lceil \frac{mn}{2m+n} \right\rceil + 1$$

for $n \geq 2m-1 > 1$, it follows from Theorem 2 and Equation (5) that the uncrossed number $\text{unc}(K_{m,n})$ differs from the outerthickness $\theta_o(K_{m,n})$ of $K_{m,n}$ by at most 1. In particular, we have $\text{unc}(K_{n,n}) = \theta_o(K_{n,n})$ for every positive integer n . However, Theorem 2 and (5) give, for example, $\text{unc}(K_{4,7}) = 2$ and $\theta_o(K_{4,7}) = 3$. Since $K_{4,7}$ is not planar, this refutes the conjecture of Hliněný and Masařík [11] in the case of complete bipartite graphs.

Second, we turn our attention to general graphs and their uncrossed number. We improve the trivial lower bound of $\text{unc}(G) \geq \lceil m/(3n-6) \rceil$ for any graph G with n vertices and m edges. By carefully balancing between the numbers of edges in uncrossed subdrawings of G and the numbers of edges that can be drawn within faces of uncrossed subdrawings, we derive the following estimate.

► **Theorem 3.** *Every connected graph G with $n \geq 3$ vertices and $m \geq 0$ edges satisfies*

$$\text{unc}(G) \geq \left\lceil \frac{m}{f(n,m)} \right\rceil$$

where $f(n,m) = (3n-5 + \sqrt{(3n-5)^2 - 4m})/2$.

The bound from Theorem 3 becomes interesting for $m \geq 3n - 6$. This is because we then have $f(n, m) \leq 3n - 6$ for all integers $n \geq 2$ as

$$\sqrt{(3n - 5)^2 - 4m} \leq \sqrt{9n^2 - 42n + 49} = 3n - 7$$

for any $m \geq 3n - 6 \geq 0$. It follows that the lower bound from Theorem 3 is at least as good as $\text{unc}(G) \geq \lceil m/(3n - 6) \rceil$ for any connected G with $n \geq 2$ vertices and $m \geq 3n - 6$ edges.

The lower bound from Theorem 3 gets stronger as the graph G gets denser. For example, if G contains n vertices and εn^2 edges for n sufficiently large and $\varepsilon \in (0, 1/2)$, we get

$$f(n, m) = (3n - 5 + \sqrt{(9 - 4\varepsilon)n^2 - 30n + 25})/2 \leq (3 + \sqrt{9 - 4\varepsilon})n/2.$$

Since $(3 + \sqrt{9 - 4\varepsilon})/2 < 3$ for $\varepsilon > 0$, we obtain $\text{unc}(G) \geq \lceil \frac{m}{c_\varepsilon n} \rceil$ for any $\varepsilon > 0$ and some constant $c_\varepsilon < 3$, instead of trivial $\text{unc}(G) \geq \lceil \frac{m}{3n - 6} \rceil$. We note that the best constant c_ε obtainable from Theorem 3 is $(3 + \sqrt{7})/2 \sim 2.82$ as $\varepsilon \leq 1/2$.

We also consider computational complexity aspects related to the `UNCROSSEDCROSSINGNUMBER` problem. As we will see later, a closely related problem is the one of determining the *edge crossing number* of a given graph G , which is the smallest number of edges involved in crossings taken over all drawings of G . The notion of the edge crossing number is based on results by Ringel [18], Harborth and Mengersen [8, 9], and Harborth and Thürmann [10].

`EDGECROSSINGNUMBER`

Input: A graph G and a positive integer k .

Question: Is there a drawing D of G with at most k edges involved in crossings?

The complementary problem to `EDGECROSSINGNUMBER` is the following one.

`MAXIMUMUNCROSSEDSUBGRAPH`

Input: A graph G and a positive integer k .

Question: Is there a drawing D of G with at least k edges not involved in any crossings?

In his monograph on crossing numbers, Schaefer [19] mentions that the problem of determining the computational complexity of `EDGECROSSINGNUMBER` is open. Here, we resolve this open question by showing that the problem is NP-complete.

► **Theorem 4.** *The `EDGECROSSINGNUMBER` problem is NP-complete.*

By the complementarity of the problems `MAXIMUMUNCROSSEDSUBGRAPH` and `EDGECROSSINGNUMBER`, we obtain the following result.

► **Corollary 5.** *The `MAXIMUMUNCROSSEDSUBGRAPH` problem is NP-complete.*

As a consequence of our reduction, we also obtain the following relative result.

► **Theorem 6.** *If the `OUTERTHICKNESS` problem is NP-hard, then also the `UNCROSSEDCROSSINGNUMBER` problem is NP-hard.*

However, in contrast to the complexity of the `THICKNESS` problem, which was shown to be NP-hard already in 1983 by Mansfield [14], the complexity of the `OUTERTHICKNESS` problem remains open.

2 Preliminaries

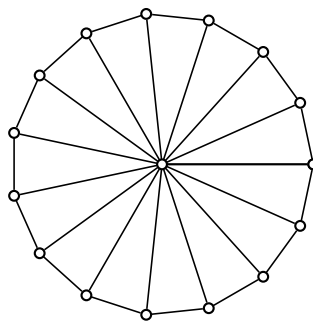
We may, without loss of generality, restrict to only simple graphs in the whole paper. This is since, in each of the formulated problems, whenever an edge e is a part of an uncrossed subdrawing (as discussed next), any other edge parallel to e can be drawn uncrossed closely along e , too.

Let D' be a subdrawing of D consisting of only uncrossed edges of D . Note that we do not require D' to contain all such edges. In this situation, we call D' an *uncrossed subdrawing* of G and we say that it *represents* the subgraph of G formed by edges that are drawn in D' . Precisely, D' is an uncrossed subdrawing of G if there exists a drawing D of a graph G such that D' is formed by a subset of the uncrossed edges of D .

► **Lemma 7.** *Let D' be an uncrossed subdrawing of a connected graph G . Then D' is a planar drawing and, for every edge $\{u, v\}$ of G , the vertices u and v are incident to a common face of D' . Moreover, there is an uncrossed subdrawing D'' of G such that D'' represents a connected supergraph of the graph represented by D' .*

Proof. The drawing D' is clearly planar, as, by the definition of D' , each edge of D' is uncrossed in a drawing D of G and thus also in D' . Moreover, it is a folklore fact that two vertices u and v in a planar drawing, here in D' , are *not* incident to a common face if and only if there exists a cycle $C \subseteq D'$ such that u and v are drawn on different sides of C . In the latter case, however, the edge $\{u, v\}$ would cross some edge of C in D , which is impossible since no edge of D' is crossed.

We prove the second part by induction on the number of connected components represented by D' . The case of one component is trivial, as $D'' = D'$. Otherwise, since G is connected, there exists an edge $e = \{u, v\}$ of G that is not drawn in D' and such that u and v belong to different components represented by D' . By the first part of the lemma, the vertices u and v are incident to the same face of D' . So, let D^+ be the planar drawing obtained from D' by adding a crossing-free arc representing the edge e . Clearly, every edge of G is still incident to a common face of D^+ , and so D^+ can be completed into a drawing of G such that D^+ stays uncrossed. The subgraph of G represented by D^+ has fewer components than we started with, and so we find the desired D'' by induction. ◀



■ **Figure 2** The wheel graph W_{15} .

For a graph G , let $h(G)$ be the maximum number of uncrossed edges in some drawing D of G . Let DW_n be a planar drawing of the wheel graph W_n on n vertices; see Figure 2. Note that DW_n is unique up to homeomorphism of the sphere and reflection as W_n is 3-connected. The following result by Ringel [18] gives a formula for $h(K_n)$ for every integer $n \geq 4$, and additionally claims that drawings of K_n with the maximum number of uncrossed edges have a unique structure.

► **Theorem 8** ([18]). *For every integer $n \geq 4$, we have $h(K_n) = 2n - 2$. Moreover, if D is a drawing of K_n with $2n - 2$ uncrossed edges, then D contains the drawing DW_n with all edges from $D \setminus DW_n$ being drawn in the outer face of DW_n .*

We also mention an analogous result for the complete bipartite graphs $K_{m,n}$, derived by Mengersen [15].

► **Theorem 9** ([15]). *For all positive integers m and n with $m \leq n$, we have*

$$h(K_{m,n}) = \begin{cases} 2m + n - 2, & \text{for } m = n \\ 2m + n - 1, & \text{for } m < n < 2m \\ 2m + n, & \text{for } 2m \leq n. \end{cases}$$

The parameter $h(G)$ can be used to estimate the uncrossed number of G . Let $\{D_1, \dots, D_k\}$ be an uncrossed collection of drawings of a graph G that has m edges. Since every drawing D_i contains at most $h(G)$ edges that are uncrossed by any other edge in D_i , we immediately obtain the following lower bound

$$\text{unc}(G) \geq \left\lceil \frac{m}{h(G)} \right\rceil. \quad (6)$$

This bound together with Theorems 8 and 9 give us quite close estimates for $\text{unc}(K_n)$ and $\text{unc}(K_{m,n})$, respectively. In particular, for $n \geq 2$ we have

$$\text{unc}(K_n) \geq \left\lceil \frac{\binom{n}{2}}{2n - 2} \right\rceil. \quad (7)$$

On the other hand, we recall the upper bound (2) on the uncrossed number of an arbitrary graph G using the notion of outerthickness of G .

3 Proof of Theorem 1

In this section, we prove Theorem 1 by providing the exact formula for the uncrossed number of complete graphs. That is, we show

$$\text{unc}(K_n) = \begin{cases} \left\lceil \frac{n+1}{4} \right\rceil, & \text{for } n \notin \{4, 7\} \\ 3, & \text{for } n = 7 \\ 1, & \text{for } n = 4 \end{cases}$$

for every positive integer n .

We start with the upper bound, which is easier to prove. For $n \notin \{4, 7\}$, the upper bound follows from (2) and (4) as we have

$$\text{unc}(K_n) \leq \theta_o(K_n) = \left\lceil \frac{n+1}{4} \right\rceil.$$

For $n = 4$, we obviously have $\text{unc}(K_4) = 1$ as K_4 is planar. Finally, $\text{unc}(K_7) = 3$ was proved by Hliněný and Masařík [11, Proposition 3.1].

It remains to prove the lower bound. Since we already know that $\text{unc}(K_7) = 3$ and $\text{unc}(K_4) = 1$ and the statement is trivial for $n \leq 3$, it suffices to consider the case $n \geq 5$ with $n \neq 7$. Let $\{D_1, \dots, D_k\}$ be an uncrossed collection of drawings of K_n and let D'_1, \dots, D'_k be corresponding uncrossed subdrawings of K_n such that $D'_1 \cup \dots \cup D'_k$ covers $E(K_n)$. By (7),

$$\text{unc}(K_n) \geq \left\lceil \frac{\binom{n}{2}}{2n - 2} \right\rceil.$$

By Theorem 8, we get that if any uncrossed subdrawing D'_i contains $2n - 2$ edges, then D'_i (as a wheel) contains a *universal vertex*, that is, a vertex that is adjacent to all remaining vertices in D'_i . If every drawing D'_i contains at most $2n - 3$ edges, then

$$\text{unc}(K_n) \geq \left\lceil \frac{\binom{n}{2}}{2n-3} \right\rceil = \left\lceil \frac{n}{4} + \frac{n}{4(2n-3)} \right\rceil = \left\lceil \frac{n+1}{4} \right\rceil \tag{8}$$

and we are done.

Thus, suppose that some drawing D'_i contains $2n - 2$ edges. Without loss of generality, we can assume $i = 1$. We then know that D'_1 contains a universal vertex v . In every drawing D'_j with $j > 1$, the edges incident to v are already counted for D'_1 , thus we can consider the drawings D'_2, \dots, D'_k to be uncrossed drawings for K_{n-1} obtained from K_n by removing v . Note that these uncrossed drawings of K_{n-1} cover $E(K_{n-1})$. Then, each D'_j contributes at most $2n - 4$ new uncrossed edges of K_{n-1} as $h(K_{n-1}) = 2(n-1) - 2 = 2n - 4$ by Theorem 8. So the number k of drawings satisfies

$$\binom{n}{2} \leq 2n - 2 + (k - 1)(2n - 4) = (2n - 4)k + 2. \tag{9}$$

However, $(2n - 4)k + 2 \leq (2n - 3)k$ when $k \geq 2$, which is satisfied for $n \geq 5$ by (9). Hence, for k' being the smallest positive integer that satisfies $\binom{n}{2} \leq k'(2n - 3)$ we obtain $k \geq k'$. Thus, we again have the inequality (8). ◀

A proof of Theorem 2 follows a similar path as that of Theorem 1, but there are several complications on the way. The upper bound requires a construction for case $n \geq 2m - 1$ besides using bounds (2) and (5). The lower bound is handled by Theorem 9 except for cases $m < n \leq 2m - 2$ that require a detailed lengthy analysis. Hence, we left the full proof for the arXiv version [1].

4 Proof of Theorem 3

Here, we show that every connected graph G with $n \geq 3$ vertices and $m \geq 0$ edges satisfies

$$\text{unc}(G) \geq \left\lceil \frac{m}{f(n, m)} \right\rceil$$

where $f(n, m) = (3n - 5 + \sqrt{(3n - 5)^2 - 4m})/2$.

Let $\mathcal{D}(G) = \{D_1, \dots, D_k\}$ be an uncrossed collection of drawings of G . For every $i \in [k]$, let D'_i be a subdrawing of D_i containing only edges of D_i that are uncrossed in D_i . By Lemma 7, each drawing D'_i is then a plane graph with the property that every edge of G that is not an edge of D'_i is contained in a single face of D'_i . Moreover, since G is connected, we can assume without loss of generality by this lemma that each D'_i represents a connected subgraph of G as to bound $\text{unc}(G)$ from below it suffices to consider drawings D'_i with as many edges as possible.

Fix an arbitrary $i \in [k]$. The number of vertices of D'_i equals n . We use m_i to denote the number of edges of D'_i and we will show that $m_i \leq f(n, m)$.

We set \mathcal{F}_i to be the set of faces of D'_i and $f_i = |\mathcal{F}_i|$. For a face F of D'_i , we use $v(F)$ for the number of vertices of D'_i that are contained in the boundary of F and we write $|F|$ for the number of times we meet an interior of an edge of D'_i when traversing F along its boundary. That is, $|F|$ is the length of the facial walk. Note that each edge can be counted once or twice in $|F|$ and so we have $v(F) \leq |F|$ as D'_i represents a connected subgraph of G . We

assume that at least one edge of F is counted once in $|F|$ and that $v(F) \geq 3$ for every face F as otherwise there is only a single face in \mathcal{F}_i and D'_i is a tree with $m_i \leq n - 1 \leq f(n, m)$ for $n \geq 3$. Also, observe that

$$\sum_{F \in \mathcal{F}_i} |F| = 2m_i \quad (10)$$

as every edge is incident to a face of D'_i from the left and from the right.

Since every edge of G that is not an edge of D'_i is contained in a single face of D'_i , we have

$$\sum_{F \in \mathcal{F}_i} \left(\binom{v(F)}{2} - v(F) \right) \geq m - m_i. \quad (11)$$

This is because vertices of each face F can span up to $\binom{v(F)}{2}$ edges of D_i and at least $v(F)$ pairs of vertices of D_i are already used for edges of D'_i as each face F contains an edge that is counted only once in $|F|$. The left hand side of (11) can be rewritten as

$$\frac{1}{2} \sum_{F \in \mathcal{F}_i} v(F)(v(F) - 3).$$

Since $v(F) \geq 3$ and $|F| \geq v(F)$ for every face F from \mathcal{F}_i , we obtain

$$\frac{1}{2} \sum_{F \in \mathcal{F}_i} |F|(|F| - 3) \geq m - m_i.$$

Since $|F| - 3 \geq 0$, the left-hand side can be bounded from above by

$$\frac{1}{2} \left(\sum_{F \in \mathcal{F}_i} |F| \right) \left(\sum_{F \in \mathcal{F}_i} (|F| - 3) \right) = m_i(2m_i - 3f_i)$$

where we used (10) twice. Altogether, we obtain $m_i(2m_i - 3f_i) \geq m - m_i$, which can be rewritten as

$$f_i \leq \frac{2m_i}{3} - \frac{m - m_i}{3m_i}.$$

Plugging this estimate into Euler's formula $n - m_i + f_i = 2$, we get

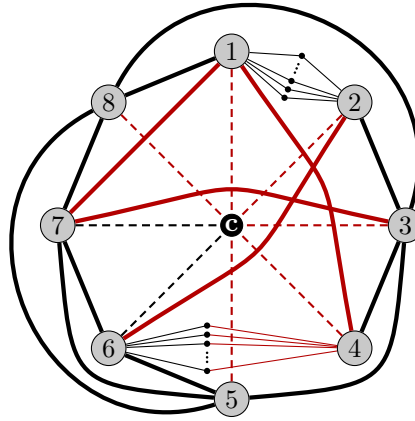
$$m_i \leq 3n - 5 - \frac{m}{m_i},$$

which after solving the corresponding quadratic inequality for m_i gives the final estimate

$$m_i \leq (3n - 5 + \sqrt{(3n - 5)^2 - 4m})/2 = f(n, m).$$

Since i was arbitrary, we see that each drawing D'_i contains at most $f(n, m)$ edges of G and therefore, we indeed have

$$k \geq \text{unc}(G) \geq \left\lceil \frac{m}{f(n, m)} \right\rceil. \quad \blacktriangleleft$$



■ **Figure 3** EDGE_CROSSING_NUMBER instance for proof of Theorem 4 in Section 5. Red edges are crossed. Thick edges represent M -bundles corresponding to edges from the MAXIMUM OUTERPLANAR SUBGRAPH instance G , see in detail the edges between 1 and 2 and between 6 and 4. The dashed edges and c form the central star.

5 Proof of Theorem 4

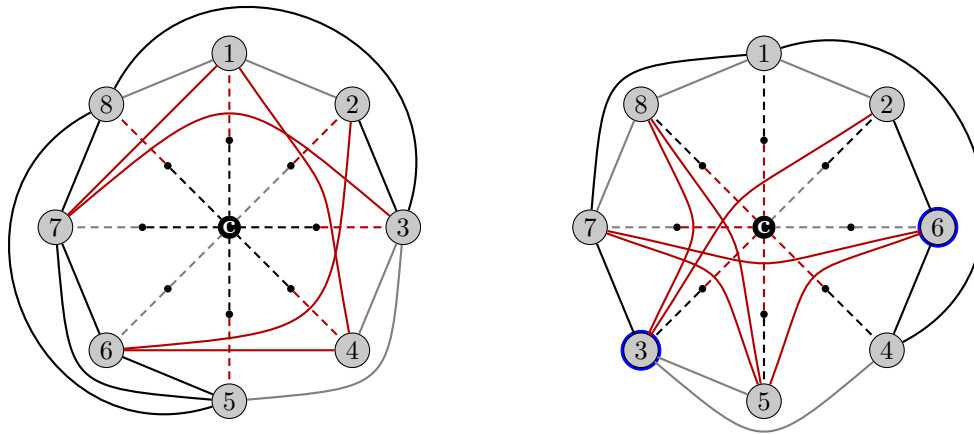
In this section, we prove that EDGE_CROSSING_NUMBER is NP-complete. Membership of this problem in the class NP is trivial. To show NP-hardness, we reduce from the following NP-complete problem [13, 20].

MAXIMUM OUTERPLANAR SUBGRAPH
Input: A graph $G = (V, E)$ and a positive integer k .
Question: Is there an outerplanar subgraph of G with at least k edges?

Assume an instance of MAXIMUM OUTERPLANAR SUBGRAPH. Let $M > |V|$, say $M = 2|V|$, and $k' = |E| - k$. We augment G into a graph G' , and show that G' can be drawn with at most $Mk' + |V|$ crossed edges, if and only if G admits an outerplanar subgraph with at least k edges. The graph G' is obtained via two augmenting steps: We add a *central star*, i.e., a vertex with an edge to each original vertex of G . Then, we replace each original edge in G by M parallel paths of length two, which we call an M -bundle. An example of this transformation can be seen in Figure 3.

Suppose there is a drawing of G' with at most $Mk' + |V|$ crossed edges. We want to modify this drawing into a drawing of G . To this end, we first remove every path belonging to an M -bundle, if either of its two edges is crossed. We also remove the central vertex and all of its incident edges. All remaining edges are uncrossed and belong to an M -bundle path. As there are at most $Mk' + |V| < M(k' + 1)$ crossed edges in the drawing, for at least $|E| - k' = k$ edges from G there is at least one path of its corresponding M -bundle that is not removed. We contract for each edge of G one edge of one of the remaining paths of its M -bundle and remove all other M -bundle paths. The vertices from G all share the face created by removing the central vertex and all vertices from M -bundles are either contracted or removed. Thus, we have an outerplanar drawing of a subgraph of G with at least k edges.

Similarly, for every outerplanar subgraph H of G with at least k edges we can construct a drawing of G' with at most $Mk' + |V|$ crossed edges. First, we draw H in an outerplanar embedding, then we draw the central star into the outer face. Next, we draw the at most



(a) First drawing.

(b) Second drawing. Vertices 3 and 6 swapped places.

■ **Figure 4** An instance of the reduction from OUTERTHICKNESS to UNCROSSEDNUMBER. The original graph G in this instance is drawn with solid edges and has outerthicknes 2 (as the two subdrawings in solid black and gray edges certify). The dashed edges and black vertices form the central star around c added to G in the reduction. In each drawing, all crossed edges are red and uncrossed edges of the particular drawing are black, and gray edges are uncrossed in both drawings.

$|E| - k = k'$ remaining edges of G in such a way that they only cross one another and the edges of the central star. Finally, we replace every edge of G with an M -bundle. The newly added vertices are positioned in such a way that at most one of the edges of each path is crossed. Therefore, there are at most Mk' crossed edges from the M -bundles and at most $|V|$ crossed edges from the central star, for a total of at most $Mk' + |V|$ crossed edges. ◀

6 Proof of Theorem 6

We show that if OUTERTHICKNESS is NP-hard, then UNCROSSEDNUMBER is NP-hard as well using a reduction from OUTERTHICKNESS to UNCROSSEDNUMBER.

The reduction employs similar arguments as used in Section 5. Let (G, k) be an instance of the problem OUTERTHICKNESS. We augment the input graph G into a graph G' by adding a vertex and connecting each vertex of G to it with a path of length two. We call the added structure the *central star*. See Figure 4 for an example of this transformation.

Let D be a drawing of G' . Consider the uncrossed subdrawing D'_G consisting of the vertices and all uncrossed edges from G . As there is a path in $D \setminus D'_G$ between each two vertices from G , we know that D'_G is outerplanar. Thus, if $\text{unc}(G') \leq k$ and D_1, \dots, D_k is an uncrossed collection of G' , then the respective subdrawings restricted to G decompose G into k outerplanar graphs.

Conversely, if G can be decomposed into $k \geq 2$ outerplanar subgraphs G_1, \dots, G_k , then we can construct an uncrossed collection D_1, \dots, D_k of G' in the following way: In every drawing D_i , we first draw G_i as an outerplanar graph and we embed the central star in the outer face. Then, we draw the remaining original edges in such a way that they only cross each other and edges from the central star. In D_1 , all crossings on the central star lie on edges incident to vertices of G , and in all other drawings, the crossings on the central star involve only edges incident to the universal vertex. This way we assure that also every edge of the central star is uncrossed in some drawing. ◀

7 Conclusions and Open Problems

We provided exact values of the uncrossed number for complete and complete bipartite graphs. The hypercube graphs form another natural graph class to consider as their outerthickness and thickness were determined exactly; see [5, 12]. However, we are not aware of any formula for the uncrossed number for the hypercube graphs.

► **Question 10.** *Determine the exact value of the uncrossed number for the hypercube graphs.*

In Theorem 3, we determined a general lower bound on $\text{unc}(G)$ in terms of the number of the edges and vertices of G by showing $\text{unc}(G) \geq \lceil \frac{m}{cn} \rceil - O(n) - O(m)$ for some constant c with $0 < c \leq 3$. In particular, we argued that the smallest obtainable constant c is approximately 2.82 for the case of dense n -vertex graphs with εn^2 edges where $\varepsilon \in (0, 1/2)$ is a fixed constant. Can one obtain a better leading constant in the general lower bound on $\text{unc}(G)$ for such dense graphs G ?

We also propose investigating other properties of the uncrossed number. We conjecture that the uncrossed number can be arbitrarily far apart from the outerthickness despite them being quite similar on the graph classes we mainly investigated in this paper. In fact, it follows from our results that the difference between the outerthickness and the uncrossed number of complete and complete bipartite graphs is never larger than one.

► **Conjecture 11.** *For every positive integer k , there is a graph G such that*

$$\theta_o(G) - \text{unc}(G) \geq k.$$


Lastly, it would be interesting to finally settle the computational complexity of the outerthickness problem. We conjecture that the OUTERTHICKNESS problem is NP-hard. Note that if true, this would also settle the computational complexity of UNCROSSEDNUMBER by Theorem 6.

References

- 1 M. Balko, P. Hliněný, T. Masařík, J. Orthaber, B. Vogtenhuber, and M.H. Wagner. On the uncrossed number of graphs. <https://arxiv.org/abs/2407.21206>, 2024. arXiv:2407.21206.
- 2 L. Beineke and R. Wilson. The early history of the brick factory problem. *Math Intelligencer*, 32:41–48, 2010. doi:10.1007/s00283-009-9120-4.
- 3 M. R. Garey and D. S. Johnson. Crossing number is NP-complete. *SIAM J. Algebr. Discrete Methods*, 4(3):312–316, September 1983. doi:10.1137/060403.
- 4 D. Gonçalves. Edge partition of planar graphs into two outerplanar graphs. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '05, pages 504–512, New York, NY, USA, 2005. Association for Computing Machinery. doi:10.1145/1060590.1060666.
- 5 R. K. Guy and R. J. Nowakowski. The outerthickness & outercoarseness of graphs. I. The complete graph & the n -cube. In *Topics in combinatorics and graph theory (Oberwolfach, 1990)*, pages 297–310. Physica, Heidelberg, 1990. doi:10.1007/978-3-642-46908-4_34.
- 6 R. K. Guy and R. J. Nowakowski. The outerthickness & outercoarseness of graphs. II. The complete bipartite graph. In *Contemporary methods in graph theory*, pages 313–322. Bibliographisches Inst., Mannheim, 1990.
- 7 F. Harary. Research problem 28. *Bull. Am. Math. Soc.*, 67(6):542, November 1961. doi:10.1090/S0002-9904-1961-10677-0.
- 8 H. Harborth and I. Mengersen. Edges without crossings in drawings of complete graphs. *J. Combinatorial Theory Ser. B*, 17:299–311, 1974. doi:10.1016/0095-8956(74)90035-5.

- 9 H. Harborth and I. Mengersen. Edges with at most one crossing in drawings of the complete graph. In *Topics in combinatorics and graph theory (Oberwolfach, 1990)*, pages 757–763. Physica, Heidelberg, 1990.
- 10 H. Harborth and C. Thürmann. Numbers of edges without crossings in rectilinear drawings of the complete graph. In *Proceedings of the Twenty-seventh Southeastern International Conference on Combinatorics, Graph Theory and Computing (Baton Rouge, LA, 1996)*, volume 119, pages 79–83, 1996.
- 11 P. Hliněný and T. Masařík. Minimizing an uncrossed collection of drawings. In *Graph Drawing and Network Visualization*, pages 110–123. Springer Nature Switzerland, 2023. doi: 10.1007/978-3-031-49272-3_8.
- 12 Michael Kleinert. Die Dicke des n -dimensionalen Würfel-Graphen. *J. Combinatorial Theory*, 3(1):10–15, 1967. doi:10.1016/S0021-9800(67)80010-3.
- 13 P. C. Liu and R. C. Geldmacher. On the deletion of nonplanar edges of a graph. In *Proceedings of the Tenth Southeastern Conference on Combinatorics, Graph Theory and Computing (Florida Atlantic Univ., Boca Raton, Fla., 1979)*, volume XXIII–XXIV of *Congress. Numer.*, pages 727–738. Utilitas Math., Winnipeg, MB, 1979.
- 14 A. Mansfield. Determining the thickness of graphs is NP-hard. *Mathematical Proceedings of the Cambridge Philosophical Society*, 93(1):9–23, 1983. doi:10.1017/S030500410006028X.
- 15 I. Mengersen. Die Maximalzahl von kreuzungsfreien Kanten in Darstellungen von vollständigen n -geteilten Graphen. *Math. Nachr.*, 85:131–139, 1978.
- 16 P. Mutzel, T. Odenthal, and M. Scharbrodt. The thickness of graphs: A survey. *Graphs Comb.*, 14(1):59–73, 1998. doi:10.1007/PL00007219.
- 17 T. Poranen and E. Mäkinen. Remarks on the thickness and outerthickness of a graph. *Comput. Math. Appl.*, 50(1-2):249–254, 2005. doi:10.1016/j.camwa.2004.10.048.
- 18 G. Ringel. Extremal problems in the theory of graphs. In *Theory of Graphs and its Applications (Proc. Sympos. Smolenice, 1963)*, pages pp 85–90. Publ. House Czech. Acad. Sci., Prague, 1964.
- 19 M. Schaefer. *Crossing Numbers of Graphs*. Discrete Mathematics and Its Applications. CRC Press, January 2018. doi:10.1201/9781315152394.
- 20 M. Yannakakis. Node-and edge-deletion NP-complete problems. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing, STOC '78*, pages 253–264, New York, NY, USA, 1978. Association for Computing Machinery.
- 21 K. Zarankiewicz. On a problem of P. Turán concerning graphs. *Fundamenta Mathematicae*, 41(1):137–145, 1955. URL: <http://eudml.org/doc/213338>.

Weakly Leveled Planarity with Bounded Span

Michael A. Bekos 
University of Ioannina, Greece

Fabrizio Frati 
Roma Tre University, Italy

Philipp Kindermann 
Trier University, Germany

Ignaz Rutter 
University of Passau, Germany

Giordano Da Lozzo 
Roma Tre University, Italy

Siddharth Gupta 
BITS Pilani, K K Birla Goa Campus, India

Giuseppe Liotta 
Perugia University, Italy

Ioannis G. Tollis 
University of Crete, Heraklion, Greece

Abstract

This paper studies planar drawings of graphs in which each vertex is represented as a point along a sequence of horizontal lines, called levels, and each edge is either a horizontal segment or a strictly y -monotone curve. A graph is s -span weakly leveled planar if it admits such a drawing where the edges have span at most s ; the span of an edge is the number of levels it touches minus one. We investigate the problem of computing s -span weakly leveled planar drawings from both the computational and the combinatorial perspectives. We prove the problem to be para-NP-hard with respect to its natural parameter s and investigate its complexity with respect to widely used structural parameters. We show the existence of a polynomial-size kernel with respect to vertex cover number and prove that the problem is FPT when parameterized by treedepth. We also present upper and lower bounds on the span for various graph classes. Notably, we show that cycle trees, a family of 2-outerplanar graphs generalizing Halin graphs, are $\Theta(\log n)$ -span weakly leveled planar and 4-span weakly leveled planar when 3-connected. As a byproduct of these combinatorial results, we obtain improved bounds on the edge-length ratio of the graph families under consideration.

2012 ACM Subject Classification Theory of computation \rightarrow Fixed parameter tractability; Theory of computation \rightarrow Computational geometry; Mathematics of computing \rightarrow Graph algorithms

Keywords and phrases Leveled planar graphs, edge span, graph drawing, edge-length ratio

Digital Object Identifier 10.4230/LIPIcs.GD.2024.19

Related Version *Full Version:* <https://arxiv.org/abs/2409.01889> [4]

Funding Research by Da Lozzo, Frati, and Liotta was supported, in part, by MUR of Italy (PRIN Project no. 2022ME9Z78 – NextGRAAL and PRIN Project no. 2022TS4Y3N – EXPAND). Research by Liotta was also supported in part by MUR PON Proj. ARS01_00540. Research by Rutter was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 541433306.

Acknowledgements This work started at the Graph and Network Visualization Workshop 2023 (GNV '23), 25–30 June, Chania.

1 Introduction

Computing crossing-free drawings of planar graphs is at the heart of Graph Drawing. Indeed, since the seminal papers by Fáry [43] and by Tutte [57] were published, a rich body of literature has been devoted to the study of crossing-free drawings of planar graphs that satisfy a variety of optimization criteria, including the area [30, 48], the angular resolution [41, 51], the face convexity [16, 17, 27], the total edge length [55], and the edge-length ratio [14, 18, 19]; see also [31, 56].



© Michael A. Bekos, Giordano Da Lozzo, Fabrizio Frati, Siddharth Gupta, Philipp Kindermann, Giuseppe Liotta, Ignaz Rutter, and Ioannis G. Tollis;
licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 19; pp. 19:1–19:19

Leibniz International Proceedings in Informatics



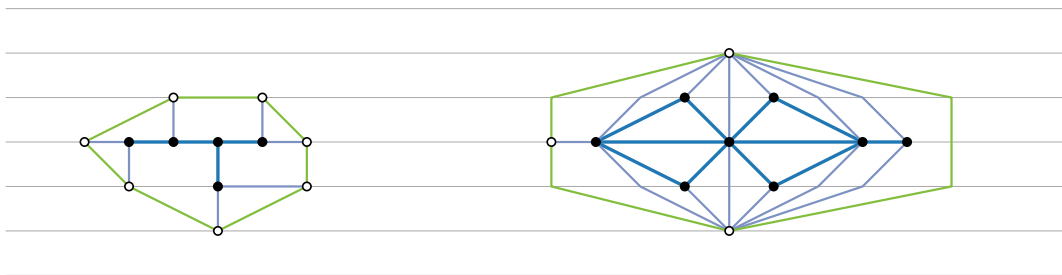
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In this paper, we focus on crossing-free drawings where the edges are represented as simple Jordan arcs and have the additional constraint of being *y-monotone*, that is, traversing each edge from one end-vertex to the other one, the *y*-coordinates never increase or never decrease. This leads to a generalization of the well-known *layered drawing* style [20, 21, 35], where vertices are assigned to horizontal lines, called *levels*, and edges only connect vertices on different levels. We also allow edges between vertices on the same level and seek for drawings of *bounded span*, i.e., in which the edges span few levels. In their seminal work [44], Heath and Rosenberg study *leveled planar drawings*, i.e., in which edges only connect vertices on consecutive levels and no two edges cross. We also mention the algorithmic framework by Sugiyama et al. [54], which yields layered drawings for the so-called hierarchical graphs. In this framework, edges that span more than one level are transformed into paths by inserting a dummy vertex for each level they cross. Hence minimizing the edge span (or equivalently, the number of dummy vertices along the edges) is a relevant optimization criterion.

Inspired by these works, we study *s-span weakly leveled planar drawings*, which are crossing-free *y*-monotone drawings in which each edge touches at most $s + 1$ levels; see Fig. 1. Note that 1-span weakly leveled planar drawings have been studied in different contexts; for example, Bannister et al. [3] prove that graphs that admit such drawings have layered pathwidth at most two¹. Felsner et al. [38, 39] show that every outerplanar graph has a 1-span weakly leveled planar drawing and use this to compute a 3D drawing of the graph in linear volume; a similar construction by Dujmović et al. [37] yields a 2-span leveled planar drawing for every outerplanar graph, which can be used to bound the queue number [44] of these graphs. In general, our work also relates to track layouts [37] and to the recently-introduced layered decompositions [36], but in contrast to these research works we insist on planarity.

Our contributions. We address the problem of computing a weakly leveled planar drawing with bounded span both from the complexity and from the combinatorial perspectives. Specifically, the *s*-SPAN (WEAKLY) LEVELED PLANARITY problem asks whether a graph admits a (weakly) leveled planar drawing where the span of every edge is at most s . The main contributions of this paper can be summarized as follows.

- In Section 3, we show that the *s*-SPAN WEAKLY LEVELED PLANARITY problem is NP-complete for any fixed $s \geq 1$ (Theorem 3). Our proof technique implies that *s*-SPAN LEVELED PLANARITY is also NP-complete. This generalizes the NP-completeness result by Heath and Rosenberg [44] which holds for $s = 1$.



■ **Figure 1** A 1-span weakly leveled planar drawing of the Frucht graph (left) and a 4-span weakly leveled planar drawing of the Goldner-Harary graph (right).

¹ Bannister et al. use the term weakly leveled planar drawing to mean 1-span weakly leveled planar drawing. We use a different terminology because we allow edges which can span more than one level.

- The para-NP-hardness of s -SPAN WEAKLY LEVELED PLANARITY parameterized by the span s motivates the study of FPT approaches with respect to structural parameters of the input graph. In Section 4, we show that the s -SPAN WEAKLY LEVELED PLANARITY problem has a kernel of polynomial size when parameterized by vertex cover number (Theorem 6) and has a (non-polynomial) kernel when parameterized by treedepth (Theorem 9). As also pointed out in [58], designing FPT algorithms parameterized by structural parameters bounded by the vertex cover number, such as the treedepth, pathwidth, and treewidth is a challenging research direction in the context of graph drawing (see, e.g., [1, 5–10, 24, 45]). Again, our algorithms can also be adapted to work for s -SPAN LEVELED PLANARITY.
- In Section 5, we give combinatorial bounds on the span of weakly leveled planar drawings of various graph classes. It is known that outerplanar graphs admit weakly leveled planar drawings with span 1 [39]. We extend the investigation by considering both graphs with outerplanarity 2 and graphs with treewidth 2. We prove that some 2-outerplanar graphs require a linear span (Theorem 10). Since Halin graphs (which have outerplanarity 2) admit weakly leveled planar drawings with span 1 [2, 33], we consider 3-connected cycle-trees [25, 29], which also have outerplanarity 2 and include Halin graphs as a subfamily. Indeed, while the Halin graphs are those graphs of polyhedra containing a face that shares an edge with every other face, the 3-connected cycle-trees are the graphs of polyhedra containing a face that shares a vertex with every other face. We show that 3-connected cycle-trees have weakly leveled planar drawings with span 4, which is necessary in the worst case (Theorem 11). For general cycle-trees, we prove $\Theta(\log n)$ span (Theorem 14); such a difference between the 3-connected and 2-connected case was somewhat surprising for us. Concerning graphs of treewidth 2, we prove an upper bound of $O(\sqrt{n})$ and a lower bound of $2^{\Omega(\sqrt{\log n})}$ on the span of their weakly leveled planar drawings (Theorem 15).

Remarks. Dujmović et al. [35] present an FPT algorithm to minimize the number of levels in a leveled planar graph drawing, where the parameter is the total number of levels. They claim that they can similarly get an FPT algorithm that minimizes the span in a leveled planar graph drawing, where the parameter is the span. Our algorithm differs from the one of Dujmović et al. [35] in three directions: (i) We optimize the span of a weakly level planar drawing, which is not necessarily optimized by minimizing the span of a leveled planar drawing; (ii) we consider structural parameters rather than a parameter of the drawing; one common point of our three algorithms is to derive a bound on the span *from* the bound on the structural parameter; and (iii) our algorithms perform conceptually simple kernelizations, while the one in [35] exploits a sophisticated dynamic programming on a path decomposition of the input graph.

Concerning the combinatorial contribution, a byproduct of our results implies new bounds on the planar edge-length ratio [46, 49, 50] of families of planar graphs. The planar edge-length ratio of a planar graph is the minimum edge-length ratio (that is, the ratio of the longest to the shortest edge) over all planar straight-line drawings of the graph. Borrazzo and Frati [19] have proven that the planar edge-length ratio of an n -vertex 2-tree is $O(n^{0.695})$. Theorem 15, together with a result relating the span of a weakly leveled planar drawing to its edge-length ratio [33, Lemma 4] lowers the upper bound of [19] to $O(\sqrt{n})$ (Corollary 16). We analogously get an upper bound of 9 on the edge-length ratio of 3-connected cycle-trees (Corollary 13).

Sketched or omitted proofs can be found in [4].

2 Preliminaries

In the paper, we only consider simple connected graphs, unless otherwise specified. We use standard terminology in the context of graph theory [34] and graph drawing [31].

A *plane graph* is a planar graph together with a *planar embedding*, which is an equivalence class of planar drawings, where two drawings are equivalent if they have the same clockwise order of the edges incident to each vertex and order of the vertices along the outer face.

A graph drawing is *y-monotone* if each edge is drawn as a strictly *y*-monotone curve and *weakly y-monotone* if each edge is drawn as a horizontal segment or as a strictly *y*-monotone curve. For a positive integer k , we denote by $[k]$ the set $\{1, \dots, k\}$. A *leveling* of a graph $G = (V, E)$ is a function $\ell: V \rightarrow [k]$. A leveling ℓ of G is *proper* if, for any edge $(u, v) \in E$, it holds $|\ell(u) - \ell(v)| = 1$, and it is *weakly proper* if $|\ell(u) - \ell(v)| \leq 1$. For each $i \in [k]$, we define $V_i = \ell^{-1}(i)$ and call it the *i -th level* of ℓ . The *height* of ℓ is $h(\ell) = \max_{v \in V} \ell(v) - \min_{v \in V} \ell(v)$. A *level graph* is a pair (G, ℓ) , where G is a graph and ℓ is a leveling of G . A *(weakly) level planar drawing* of a level graph (G, ℓ) is a planar (weakly) *y*-monotone drawing of G where each vertex is drawn with *y*-coordinate $\ell(v)$. A level graph (G, ℓ) is *(weakly) level planar* if it admits a (weakly) level planar drawing. A (weakly) leveled planar drawing of a graph G is a (weakly) level planar drawing of a level graph (G, ℓ) , for some leveling ℓ of G .

The following observation rephrases a result of Di Battista and Nardelli [32, Lemma 1] in the weakly-level planar setting.

► **Observation 1.** *Let (G, ℓ) be a level graph such that ℓ is (weakly) proper. For each $i \in [k]$, let \prec_i be a linear ordering on $\ell^{-1}(i)$. Then, there exists a (weakly) level planar drawing of (G, ℓ) that respects \prec_i (i.e., in which the left-to-right ordering of the vertices in $\ell^{-1}(i)$ is \prec_i) if and only if:*

- (i) *if $(u, v) \in E(G)$ with $\ell(u) = \ell(v) = i$, then u and v are consecutive in \prec_i ; and*
- (ii) *if (u, v) and (w, x) are two independent edges (i.e., $\{u, v\} \cap \{w, x\} = \emptyset$) with $\ell(u) = \ell(w) = i$, $\ell(v) = \ell(x) = i + 1$, and $u \prec_i w$, then $v \prec_{i+1} x$.*

The *span* of an edge (u, v) of a level graph (G, ℓ) is $\text{span}_\ell(u, v) = |\ell(u) - \ell(v)|$. The *span* of a leveling ℓ of G is $\text{span}(\ell) = \max_{(u, v) \in E} \text{span}_\ell(u, v)$. Given a graph G , we consider the problem of finding a leveling ℓ that minimizes $\text{span}(\ell)$ among all levelings where (G, ℓ) is weakly level planar. Specifically, given a positive integer s , we call *s-SPAN (WEAKLY) LEVELED PLANARITY* the problem of testing whether a graph G admits a leveling ℓ , with $\text{span}(\ell) \leq s$, such that (G, ℓ) is (weakly) level planar. The *1-SPAN LEVELED PLANARITY* problem has been studied under the name of *LEVELED PLANAR* by Heath and Rosenberg [44].

A (weakly) *y*-monotone drawing Γ of a graph defines a leveling ℓ , called the *associated leveling* of Γ , where vertices with the same *y*-coordinate are assigned to the same level and the levels are ordered by increasing *y*-coordinates of the vertices they contain. Thus, the span of an edge (u, v) in Γ is $\text{span}_\ell(u, v)$, the span of Γ is $\text{span}(\ell)$, and the height of Γ is $h(\ell)$.

The following lemma appears implicitly in the proof of Lemma 4 in [33].

► **Lemma 2.** *Any graph that admits an s -span weakly leveled planar drawing with height h has an $(2s + 1)$ -span leveled planar drawing with height $2h + 1$.*

A planar drawing of a graph is *outerplanar* if all the vertices are external, and *2-outerplanar* if removing the external vertices yields an outerplanar drawing. A graph is *outerplanar (2-outerplanar)* if it admits an outerplanar drawing (resp. 2-outerplanar drawing). A *2-outerplane* graph is a 2-outerplanar graph with an associated planar embedding which corresponds to 2-outerplanar drawings. A *cycle-tree* is a 2-outerplane graph such that removing the external

vertices yields a tree. A *Halin graph* is a 3-connected plane graph G such that removing the external edges yields a tree whose leaves are exactly the external vertices of G (and whose internal vertices have degree at least 3). Note that Halin graphs form a subfamily of the cycle-trees.

3 NP-completeness

This section is devoted to the proof of the following result.

► **Theorem 3.** *For any fixed $s \geq 1$, s -SPAN WEAKLY LEVELED PLANARITY is NP-complete.*

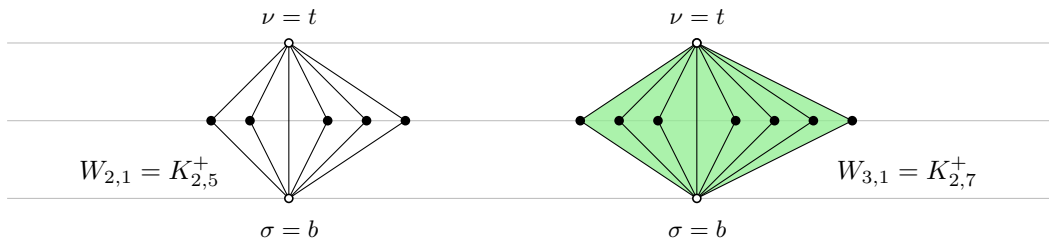
Proof sketch. The NP-membership is trivial. We prove the NP-hardness via a linear-time reduction from the 1-SPAN LEVELED PLANARITY problem, which was proved NP-complete by Heath and Rosenberg [44]. We distinguish based on whether $s = 1$ or $s > 1$.

Case $s = 1$. Starting from a (bipartite) planar graph H , we construct a graph G that is a positive instance of 1-SPAN WEAKLY LEVELED PLANARITY if and only if H is a positive instance of 1-SPAN LEVELED PLANARITY, by replacing each edge (u, v) of H with a copy $K(u, v)$ of $K_{2,4}$, where u and v are identified with the two degree-4 vertices of $K(u, v)$.

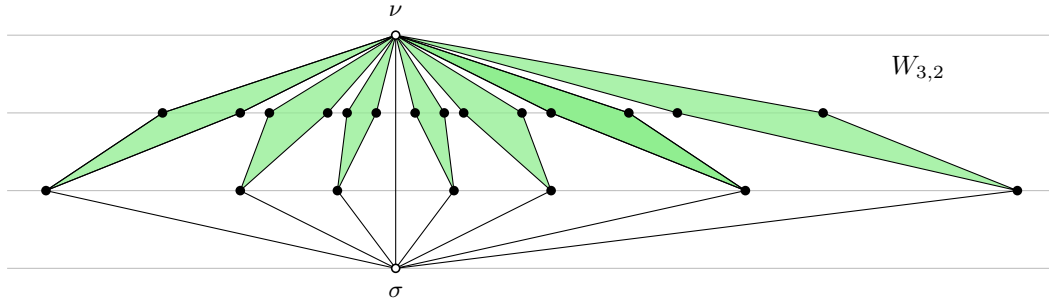
Suppose first that H admits a leveling ℓ_H in $k = h(\ell_H) + 1$ levels, with $\text{span}(\ell_H) \leq 1$, such that (H, ℓ_H) is level planar, and let Γ_H be a level planar drawing of (H, ℓ_H) . Consider the leveling ℓ_G of G on $2k$ levels computed as follows. For each vertex $w \in V(H)$, we set $\ell_G(w) = 2 \cdot \ell_H(w)$. For each vertex $w \in V(G) \setminus V(H)$ in a graph $K(u, v)$, we set $\ell_G(w) = \min\{\ell_G(u), \ell_G(v)\} + 1$. By construction, ℓ_G is proper (and thus $\text{span}(\ell_G) \leq 1$), the vertices of $V(H)$ are assigned to even levels, and the vertices in $V(G) \setminus V(H)$ are assigned to odd levels. The graph $K(u, v)$ admits a leveled planar drawing with span 1 on three levels in which u and v lie strictly above and strictly below all other vertices of $K(u, v)$, respectively. This allows us to introduce a new level between any two consecutive levels in Γ_H and replace the drawing of each edge (u, v) of H with a drawing of $K(u, v)$ as the one described above. The resulting drawing is a level planar drawing Γ_G of (G, ℓ_G) .

Suppose now that G admits a leveling ℓ_G , with $\text{span}(\ell_G) \leq 1$, such that (G, ℓ_G) is weakly level planar. We show that H admits a leveling ℓ_H , with $\text{span}(\ell_H) = 1$, such that (H, ℓ_H) is level planar. Note that any 1-span weakly leveled planar drawing of $K(u, v)$ is leveled planar and places u and v on different levels. Also, any edge of G belongs to $K(u, v)$ for some edge $(u, v) \in E(H)$. Thus, any 1-span weakly leveled planar drawing of G is leveled planar. Moreover, ℓ_G is proper. Let Γ_G be a level planar drawing of (G, ℓ_G) . To construct a level planar drawing Γ_H of (H, ℓ_H) , we simply set the ordering of the vertices on level i in Γ_H to be the ordering of these vertices on level $2i$ in Γ_G . We claim that such orderings satisfy Conditions (i) and (ii) of Observation 1, which proves that Γ_H is level planar.

Case $s > 1$. In our proof, we exploit special graphs $W_{i,h}$, with $1 \leq h < i$, having two designated vertices ν and σ , called *poles*; specifically, ν is the *north pole* and σ is the *south pole* of $W_{i,h}$; refer to Fig. 2. In the following, we denote by $K_{2,\alpha}^+$ the graph obtained from the complete bipartite graph $K_{2,\alpha}$ by adding an edge between the two vertices t and b of the size-2 bipartition class of the vertex set of $K_{2,\alpha}$. For any $i \geq 2$, the graphs $W_{i,h}$ are defined as follows. If $h = 1$, the graph $W_{i,1}$ coincides with $K_{2,2i+1}^+$; see Fig. 2a. If $h > 1$, the graph $W_{i,h}$ is obtained from $K_{2,2i+1}^+$ by removing each edge (t, x) , with $x \neq b$, and by identifying t and b with the north and south pole of a copy of $W_{i,h-1}$, respectively; see Fig. 2b.



(a) The graphs $W_{2,1}$ (left) and $W_{3,1}$ (right).



(b) The graph $W_{3,2}$. The seven green shaded regions are copies of $W_{3,1}$.

■ **Figure 2** Illustration for the construction of graphs $W_{i,h}$. Pole vertices are white filled.

The reduction for $s > 1$ is similar to the one for $s = 1$, but the role of $K_{2,4}$ is now played by $W_{s,s-1}$. For an edge $(u, v) \in E(H)$, we denote by $W_s(u, v)$ the copy of $W_{s,s-1}$ used to replace (u, v) . The correctness of the reduction is based on the following claims.

▷ **Claim 4.** For any $i \geq 2$ and $h < i$, the graph $W_{i,h}$ admits a levelled planar drawing with span $h + 1$ in which the north pole of $W_{i,h}$ lies strictly above all the other vertices of $W_{i,h}$ and the south pole of $W_{i,h}$ lies strictly below all the other vertices of $W_{i,h}$.

▷ **Claim 5.** For any $i \geq 2$ and $h < i$, in any weakly levelled planar drawing of $W_{i,i-1}$ with span at most i , the edge connecting the poles of $W_{i,i-1}$ has span i .

We conclude the proof by observing that the construction of G can be done in polynomial time, for any fixed value of s ; in particular, the number of vertices of G is bounded by the number of vertices of H times a computable function only depending on s . ◀

The proof of Theorem 3 also shows that, for any fixed $s \geq 1$, deciding whether a graph admits a (non-weakly) levelled planar drawing with span at most s is NP-complete, which generalizes the NP-completeness result by Heath and Rosenberg [44], which is limited to $s = 1$.

4 Parameterized Complexity

Motivated by the NP-hardness of the s -SPAN WEAKLY LEVELLED PLANARITY problem (Theorem 3), we consider the parameterized complexity of the problem. Recall that a problem \mathcal{P} whose input is an n -vertex graph G is *fixed-parameter tractable* (for short, *FPT*) with respect to some parameter k if it can be solved via an algorithm with running time $O(f(k) \cdot p(n))$, where f is a computable function and p is a polynomial function. A *kernelization* for \mathcal{P} is an algorithm that constructs in polynomial time (in n) an instance (G', k') , called *kernel*, such that: (i) the *size* of the kernel, i.e., the number of vertices in G' , is some computable function of k ; (ii) (G', k') and (G, k) are equivalent instances; and (iii) k' is some computable function of k . If \mathcal{P} admits a kernel w.r.t. some parameter k , then it is FPT w.r.t. k .

► **Theorem 6.** *Let (G, s) be an instance of s -SPAN WEAKLY LEVELED PLANARITY with a vertex cover C of size k . There exists a kernelization that applied to (G, s) constructs a kernel of size $O(k^2)$. Hence, the problem is FPT with respect to the size of a vertex cover.*

Proof sketch. First, we give a kernel (with respect to a parameterization by k and s) of size $O(k \cdot s)$. Second, we show that any planar graph G with vertex cover number k admits a weakly leveled planar drawing with span at most $3k$, which allows us to assume $s \leq 3k$.

For the kernel with respect to $k + s$, we follow a classical reduction approach. By planarity, the number of vertices of $G - C$ with three or more neighbors in C can be bounded by $2k$ (e.g., using [40, Lemma 13.3]), and the number of pairs from C with a degree-2 neighbor in $G - C$ is at most $3k$. For each vertex $c \in C$ with more than three degree-1 neighbors in $G - C$, we only keep three of such neighbors. Then in any drawing of the reduced instance, a neighbor v_c of c is not on the same level as c , and thus we can reinsert the removed vertices next to v_c . Also, for each pair of vertices $\{c, d\} \in C$ that are common neighbors of more than $4s + 5$ degree-2 vertices in $G - C$, we only keep $4s + 5$ of such degree-2 vertices. Then in any drawing of the reduced instance with span at most s , a neighbor v_{cd} of c and d lies strictly between the levels of c and d , and thus we can reinsert the removed vertices next to v_{cd} . As these reductions can be performed in polynomial time, this yields a kernel of size $O(k \cdot s)$.

To bound the span, we consider a more strict trimming operation that removes all degree-1 vertices of $G - C$ and replaces all degree-2 vertices of $G - C$ with the same two neighbors $u, v \in C$ by a single edge (u, v) . As above, the size of this trimmed graph is $O(k)$. It therefore admits a planar leveled drawing of height (and thus also span) $O(k)$, into which the removed vertices can be inserted without asymptotically increasing the height. ◀

Let (G, s) be an instance of s -SPAN WEAKLY LEVELED PLANARITY and b be a parameter. A set M of vertices of G is a *modulator to components of size b* (*b -modulator* for short) if every connected component of $G - M$ has size at most b . Note that a 1-modulator is a vertex cover. We show that testing whether a graph with a b -modulator of size k admits a weakly leveled planar drawing with span s is FPT w.r.t. $b + k$. The neighbors in M of a component C of $G - M$ are its *attachments*, and are denoted by $\text{att}(C)$. We also denote by $\text{bridge}(C)$ the graph consisting of C , $\text{att}(C)$, and the edges between C and its attachments.

We generalize the technique for vertex cover and give a kernel with respect to $b + k + s$ and further show that any planar graph admits a leveled planar drawing with height (and hence span) bounded by $(5b + 1)bk$. Hence, s can be bounded by a function of $b + k$, which yields the result. Differently from vertex cover, the components of $G - M$ are not single vertices but have up to b vertices. Nevertheless, we use planarity to bound the number of components with three or more attachments by $2k$ and the number of pairs $\{u, v\} \in M$ that are the attachments of components of $G - M$ by $3k$. The most challenging part is again dealing with the components of $G - M$ with one or two attachments, as their number might not be bounded by any function of $b + k$. Here, the key insight is that, since these components have size at most b , there is only a bounded number of “types” of these components. More precisely, two components C_1, C_2 of $G - M$ that have the same attachments are called *equivalent* if there is an isomorphism between $\text{bridge}(C_1)$ and $\text{bridge}(C_2)$ that leaves the attachments fixed. We use the fact that there is a computable (in fact exponential, see e.g. [26]) function $f: \mathbb{N} \rightarrow \mathbb{N}$ such that the number of equivalence classes is bounded by $f(b)$.

We can show that if an equivalence class of components with one attachment (with two attachments) is sufficiently large, that is, it is at least as large as a suitable function of s and b , then in any leveled planar drawing with bounded span, one such component must be drawn

19:8 Weakly Leveled Planarity with Bounded Span

on levels that are all strictly above or all strictly below its attachment (resp. on levels that are strictly between its two attachments). Then arbitrarily many equivalent components can be inserted into a drawing without increasing its span. This justifies the following reduction rules.

► **Rule 1.** For every vertex $v \in M$, let \mathcal{C}_v be a set containing all and only the equivalent components C of $G - M$ such that $\text{att}(C) = \{v\}$. If $|\mathcal{C}_v| > (4s + 4)b$, then remove all but $(4s + 4)b$ of these components from G .

► **Rule 2.** For every pair of vertices $\{u, v\} \in M$, let \mathcal{C}_{uv} be a set containing all and only the equivalent components C of $G - M$ with $\text{att}(C) = \{u, v\}$. If $|\mathcal{C}_{uv}| > (8s + 8)(b + 1)$, then remove all but $(8s + 8)(b + 1)$ such components from G .

We can now sketch a proof of the following theorem.

► **Theorem 7.** Let (G, s) be an instance of s -SPAN WEAKLY LEVELED PLANARITY with a b -modulator of size k . There exists a kernelization that applied to (G, s) constructs a kernel of size $O(f(b) \cdot k^2 \cdot b^4)$. Hence, the problem is FPT with respect to $k + b$.

Proof sketch. As mentioned above, the number of components of $G - M$ with three or more attachments is at most $2k$. Rules 1 and 2 bound the number of equivalent components with one and two attachments. The fact that the number of equivalence classes is bounded by $f(b)$ then yields a kernel of size $O(f(b) \cdot k \cdot b^2 \cdot s)$. Note that testing whether two components are equivalent can be reduced to an ordinary planar graph isomorphism problem [47], by connecting each attachment to a sufficiently long path, which forces the isomorphism to leave the attachments fixed. Hence, the described reduction can be performed in polynomial time.

To drop the dependence on s of the kernel size, we prove that every planar graph admits a leveled planar drawing whose height (and hence span) is at most $(5b + 1)bk$. To this end, we use a trim operation that removes all components of $G - M$ with a single attachment and replaces multiple components of $G - M$ with attachments $\{u, v\}$ by a single edge (u, v) . The size of the trimmed graph is at most $(5b + 1)k$, therefore there exists a leveled planar drawing of the trimmed graph whose span is bounded by the same function. Then the removed components, which have size at most b , can be reinserted by introducing b new levels below each level, i.e., the span increases by a linear factor in b . This yields the desired leveled planar drawing with span at most $(5b + 1)bk$ and thus a kernel of the claimed size. ◀

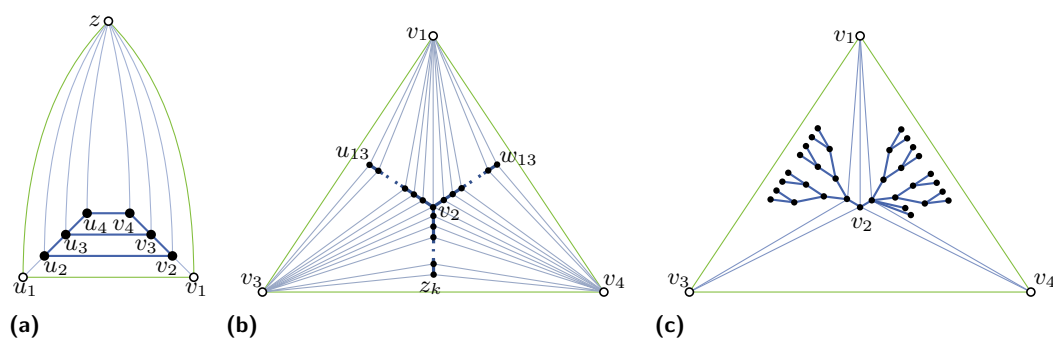
We now move to treedepth. A treedepth decomposition of a graph $G = (V, E)$ is a tree T on vertex set V with the property that every edge of G connects a pair of vertices that have an ancestor-descendant relationship in T . The *treedepth* of G is the minimum depth (i.e., maximum number of vertices in any root-to-leaf path) of a treedepth decomposition T of G .

Let td be the treedepth of G and T be a treedepth decomposition of G with depth td . Let r be the root of T . For a vertex $u \in V$, we denote by T_u the subtree of T rooted at u , by V_u the vertex set of T_u , by $d(u)$ the depth of u (where $d(u) = 1$ if u is a leaf and $d(u) = td$ if $u = r$), and by $R(u)$ the set of vertices on the path from u to r (end-vertices included).

As for Theorems 6 and 7, we can show that the instance (G, s) is positive if s is sufficiently large, namely larger than $((5td)^{td} + 1)^{td}$. Hence, we can assume that s is bounded w.r.t. td .

► **Theorem 8.** Every planar graph with treedepth td has a leveled planar drawing of height at most $((5td)^{td} + 1)^{td}$.

Proof sketch. We apply a trimming operation similar to the one described in Theorem 7. The effect of this operation, when applied to a vertex v , is to bound the number of children of v in T by $5td$. Assume that every vertex in $V_v \setminus \{v\}$ has at most $5td$ children. Then $R(v)$



■ **Figure 3** (a) A n -vertex 2-outerplanar graph requiring $\Omega(n)$ span in every weakly leveled planar drawing. (b) An n -vertex 3-connected cycle-tree requiring span 4 in every weakly leveled planar drawing. (c) An n -vertex cycle-tree requiring $\Omega(\log n)$ span in every weakly leveled planar drawing. The graph resulting from the removal of the vertices incident to the outer face is drawn bold.

is a vertex set of size at most td and, due to the degree bound, any connected component of $G - R(v)$ whose vertices are in V_v has size bounded by $(5td)^{td}$. Similarly as for modulators, we can remove such components with a single attachment and replace multiple components with the same two attachments by an edge. This bounds the degree of v in T to $5td$ ($2td$ for components with three or more attachments and $3td$ for components with two attachments).

We apply this trimming operation in batches. The first batch processes all leaves (which has no effect); each next batch consists of all the vertices whose children are already processed. After at most td batches the whole tree is processed (and in fact reduced to a single vertex by processing the root). We then undo these steps while maintaining a leveled planar drawing. The key point here is that, due to the degree bound, each of the removed components contains at most $(5td)^{td}$ vertices, and we can simultaneously reinsert all removed components at the cost of inserting $(5td)^{td}$ levels below each existing level in the current drawing. Therefore, the number of levels multiplies by $(5td)^{td} + 1$ per batch. ◀

We thus obtain the following.

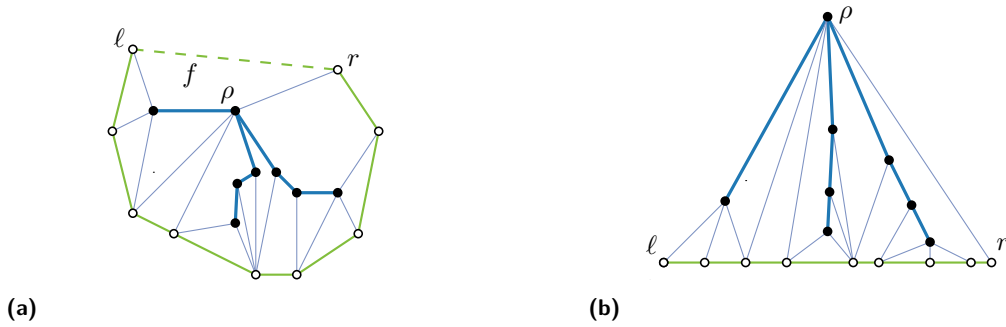
► **Theorem 9.** *Let (G, s) be an instance of s -SPAN WEAKLY LEVELED PLANARITY with treedepth td . There exists a kernelization that applied to (G, s) constructs a kernel whose size is a computable function of td . Hence, the problem is FPT with respect to the treedepth.*

Proof sketch. We perform a kernelization with respect to $td + s$. We then apply Theorem 8 to bound the span in terms of treedepth. We use a strategy similar to the one in Theorem 8 for processing the vertices in td batches. However, instead of the trimming operation used there, we use Rules 1 and 2 to bound, for each vertex v , the number of components of $G - R(v)$ whose vertices are in V_v by a function $g(td, s)$. Eventually, we obtain an equivalent instance in which the vertices have their degree bounded by $g(td, s)$ in T . This is the desired kernel. ◀

5 Upper and Lower Bounds

In this section, we establish upper and lower bounds on the span of weakly leveled planar drawings of certain graph classes.

► **Theorem 10.** *There exists an n -vertex 2-outerplanar graph such that every weakly leveled planar drawing of it has span in $\Omega(n)$.*



■ **Figure 4** (a) An almost-3-connected path-tree G , drawn with solid edges. The path-vertices are white and the tree-vertices are black. (b) The path-tree G with root ρ .

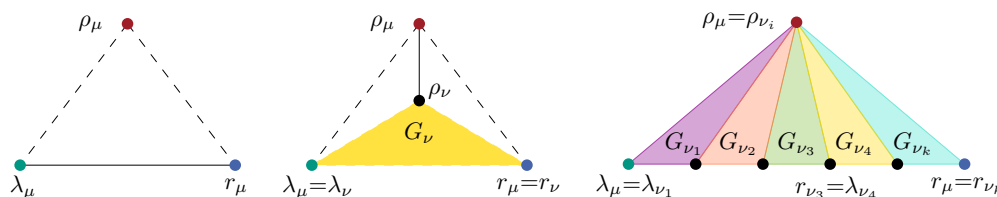
Proof sketch. The lower bound is provided by the graph G_k of Fig. 3a, which is composed of $k := \lfloor (n-1)/2 \rfloor$ “1-fused stacked cycles” and introduced by Biedl [11]. It is easy to observe that each cycle has an edge that spans two more levels than any edge of a cycle stacked inside it, from which the linear lower bound follows. ◀

There is however a well-studied graph family, the Halin graphs, which have outerplanarity 2 and admit 1-span weakly levelled planar drawings [2, 33]. This motivates the study of cycle-trees [29], a superclass of Halin graphs still having outerplanarity 2. We first consider 3-connected cycle-trees showing a constant span and then extend the study to general cycle-trees. The approach for 3-connected cycle-trees relies on removing an edge from the external face so to obtain a graph for which we construct a suitable decomposition tree. We conclude by discussing the span of weakly levelled planar drawings of planar graphs with treewidth 2.

Path-Trees. A *path-tree* is a plane graph G that can be augmented to a cycle-tree G' by adding the edge $e = (\ell, r)$ in its outer face; see Fig. 4a. W.l.o.g., let ℓ occur right before r in clockwise order along the outer face of G' ; then ℓ and r are the *leftmost* and *rightmost path-vertex* of G , respectively. The external (internal) vertices of G' are *path-vertices* (*tree-vertices*). The tree-vertices induce a tree in G . We can select any tree-vertex ρ incident to the unique internal face of G' incident to e as the *root* of G . Then G is *almost-3-connected* if it becomes 3-connected by adding the edges (ρ, ℓ) , (ρ, r) , and (ℓ, r) , if they are not already part of G . If G is almost-3-connected, the path-vertices induce a path in G .

SPQ-decomposition of path-trees. Let G be an almost-3-connected path-tree with root ρ , leftmost path-vertex λ , and rightmost path-vertex r . We define the *SPQ-decomposition* of G , introduced in [29], which constructs a tree \mathcal{T} , called the *SPQ-tree* of G . The nodes of \mathcal{T} are of three types: *S*-, *P*-, and *Q*-nodes. Each node μ of \mathcal{T} corresponds to a subgraph G_μ of G , called the *pertinent graph* of μ , which is an almost-3-connected rooted path-tree. We denote by ρ_μ the root of G_μ (a tree-vertex), by λ_μ the leftmost path-vertex of G_μ , and by r_μ the rightmost path-vertex of G_μ . To handle the base case, we consider as a path-tree also a graph whose path is the single edge (λ, r) and whose tree consists of a single vertex ρ , possibly adjacent to only one of λ and r . Also, we consider as almost-3-connected a path-tree such that adding (ρ, r) , (ρ, λ) , and (λ, r) , if missing, yields a 3-cycle.

We now describe the decomposition.



■ **Figure 5** Path-trees corresponding to a Q-node (left), an S-node (middle), and a P-node (right). Dashed edges may or may not belong to G_μ . Shaded triangles represent smaller path-trees G_{ν_i} .

- **Q-NODE:** the pertinent graph G_μ of a *Q-node* μ is an almost-3-connected rooted path-tree which consists of ρ_μ , λ_μ , and r_μ . The edge (λ_μ, r_μ) belongs to G_μ , while the edges (ρ_μ, λ_μ) and (ρ_μ, r_μ) may not exist; see Fig. 5(left).
- **S-NODE:** the pertinent graph G_μ of an *S-node* μ is an almost-3-connected rooted path-tree which consists of ρ_μ and of an almost-3-connected path-tree G_ν , where ρ_μ is adjacent to ρ_ν and, possibly, to λ_ν and r_ν . We have that μ has a unique child in \mathcal{T} , namely a node ν whose pertinent graph is G_ν . Further, we have $\lambda_\mu = \lambda_\nu$ and $r_\mu = r_\nu$; see Fig. 5(middle).
- **P-NODE:** the pertinent graph G_μ of a *P-node* μ is an almost-3-connected rooted path-tree which consists of almost-3-connected rooted path-trees $G_{\nu_1}, \dots, G_{\nu_k}$, with $k > 1$. This composition is defined as follows. First, we have $\rho_\mu = \rho_{\nu_1} = \dots = \rho_{\nu_k}$. Second, we have $\lambda_{\nu_i} = r_{\nu_{i-1}}$, for $i = 2, \dots, k$. Third, μ has children ν_1, \dots, ν_k (in this left-to-right order) in \mathcal{T} , where G_{ν_i} is the pertinent graph of ν_i , for $i = 1, \dots, k$. Finally, we have $\lambda_\mu = \lambda_{\nu_1}$ and $r_\mu = r_{\nu_k}$; see Fig. 5(right).

In the following, all considered SPQ-trees are *canonical*, that is, the child of every P-node is an S- or Q-node. For a given path-tree, a canonical SPQ-tree always exists [23].

3-connected Cycle-Trees. Let G be a plane graph with three consecutive vertices u, v, w encountered in this order when walking in clockwise direction along the boundary of the outer face of G . A leveling of G is *single-sink with respect to (u, v, w)* if all vertices of G have a neighbor on a higher level, except for exactly one of $\{u, v, w\}$. A single-sink leveling ℓ with respect to (u, v, w) is *flat* if $\ell(u) < \ell(v) < \ell(w)$ or $\ell(w) < \ell(v) < \ell(u)$; ℓ is a *roof* if $\ell(v) > \ell(u)$ and $\ell(v) > \ell(w)$. Note that a single-sink leveling is necessarily either roof or flat.

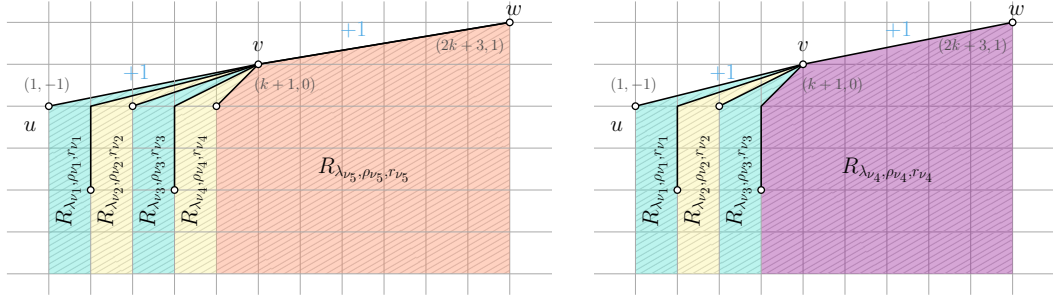
Given a single-sink leveling ℓ of G with respect to (u, v, w) , a *good* weakly leveled planar drawing Γ of (G, ℓ) is one with the following properties:

1. Γ respects the planar embedding of G ;
2. it holds that $x(u) < x(w)$ in Γ ; and
3. all vertices of $V(G) \setminus \{u, v, w\}$ are contained in the interior of the bounded region R_{uvw} defined by the path (u, v, w) , by the vertical rays starting at u and w , and by the horizontal line $y := \min_{z \in V(G)} \ell(z)$.

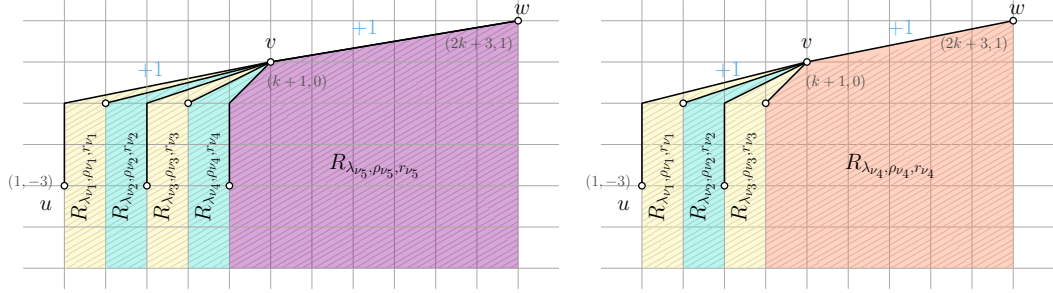
Let a and b be two non-zero integers. A good weakly leveled planar drawing Γ of (G, ℓ) is an *(a, b) -flat drawing* if ℓ is flat, $a = \ell(v) - \ell(u)$, and $b = \ell(w) - \ell(v)$; it is an *(a, b) -roof drawing* if ℓ is roof, $a = \ell(v) - \ell(u)$, and $b = \ell(w) - \ell(v)$. Note that, by definition, in an (a, b) -flat drawing we have that a and b are either both positive or both negative, while in an (a, b) -roof drawing a is positive and b is negative.

► **Theorem 11.** *Every 3-connected cycle-tree admits a 4-span weakly leveled planar drawing. Also, for all $n \geq 43$, there exists an n -vertex 3-connected cycle-tree G such that every weakly leveled planar drawing of G has span greater than or equal to 4.*

19:12 Weakly Levelled Planarity with Bounded Span

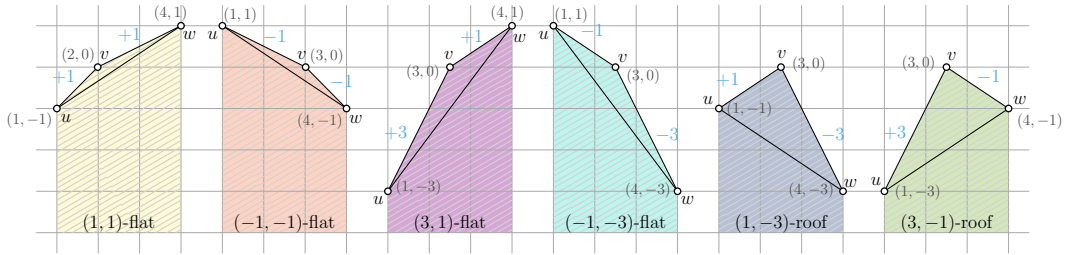


(a) A (1,1)-flat weakly leveled planar drawing when μ has an odd (left) or even (right) number of children.



(b) A (3,1)-flat weakly leveled planar drawing when μ has an odd (left) or even (right) number of children.

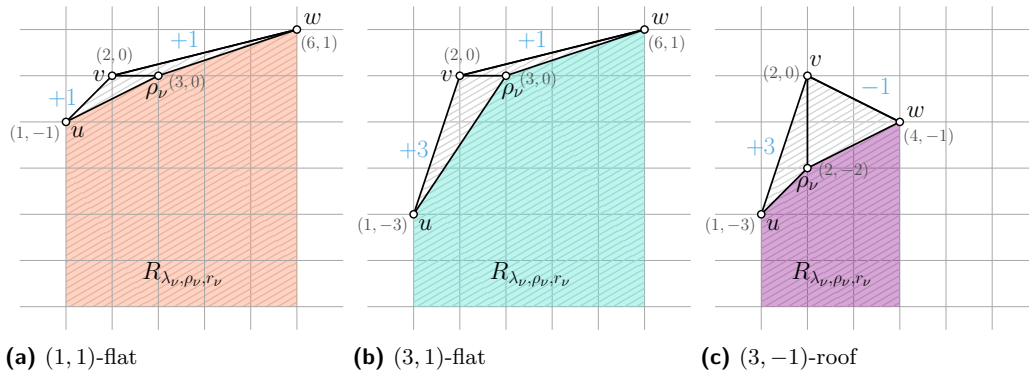
■ **Figure 6** Illustrations for the proof of Theorem 11, when μ is a P-node.



■ **Figure 7** Illustrations for the proof of Theorem 11, when μ is a Q-node and the height of \mathcal{T} is 0.

Proof sketch. We first prove the statement for almost 3-connected path trees. Let G be such a graph and \mathcal{T} be its SPQ-tree with root μ . Let $u = \lambda_\mu$, $v = \rho_\mu$, and $w = r_\mu$. Since removing edges does not increase the span of a weakly leveled planar drawing, we can assume that the edges (u, v) and (v, w) belong to G and that G is internally triangulated. That is, we prove the statement when G is a maximal almost-3-connected path-tree. The proof is based on recursively constructing a drawing of $G = G_\mu$, where the recursion is on the SPQ-tree \mathcal{T} of G , according to the following case distinction (for details, see [4]).

If μ is a **P-node**, then G_μ has flat levelings ℓ_μ^i for $i = 1, \dots, 4$, with $\text{span}(\ell_\mu^i) \leq 4$, such that (G_μ, ℓ_μ^i) admits a p_i -flat weakly leveled planar drawing with $p_1 = (-1, -1)$, $p_2 = (1, 1)$, $p_3 = (-1, -3)$, and $p_4 = (3, 1)$, see Fig. 6. Let k be the number of children of μ in \mathcal{T} . Each flat leveling ℓ_μ^i is obtained by combining roof levelings for the pertinent graphs of the leftmost (or the rightmost) $k - 1$ children of μ with a flat leveling of the pertinent graph of the rightmost (resp. leftmost) child of μ . In particular, the $k - 1$ children of μ for which flat levelings are used alternate, in left-to-right order, between (1, -3)-roof drawings and (3, -1)-roof drawings. If μ is a **Q-node**, then G_μ has flat levelings ℓ_μ^i for $i = 1, \dots, 4$, with $\text{span}(\ell_\mu^i) \leq 4$, such



■ **Figure 8** Illustrations for the proof of Theorem 11, when μ is an S-node.

that (G_μ, ℓ_μ^i) admits a q_i -flat weakly leveled planar drawing with $q_1 = (1, 1)$, $q_2 = (-1, -1)$, $q_3 = (3, 1)$, and $q_4 = (-1, -3)$. Also, G_μ has roof levelings ℓ_μ^j for $j = 5, 6$, with $\text{span}(\ell_\mu^j) \leq 4$, such that (G_μ, ℓ_μ^j) admits a q_j -roof weakly leveled planar drawing with $q_5 = (1, -3)$ and $q_6 = (3, -1)$, see Fig. 7. Finally, if μ is an **S-node**, then G_μ has the same type of levelings and weakly leveled planar drawings as in the case in which it is a Q-node, see Fig. 8. Each of such levelings is obtained from a flat leveling of the pertinent graph of the unique child of μ .

For triconnected cycle-trees, we remove an edge e on the outer face, after an augmentation we obtain a $(1, 1)$ -flat weakly leveled planar drawing, and insert back e with span 2.

The proof of the theorem is completed by observing that some 3-connected cycle-trees, like the one in Fig. 3b, require span at least 4. ◀

The approach in the proof of Theorem 11 can be implemented in quadratic time. To get linear time, we can maintain only the order of the vertices on their levels and calculate the exact coordinates at the end of the algorithm.

Similar to [37, Lemma 14], one can prove that s -span weakly leveled planar graphs have queue number at most $s + 1$; see [4] for details. Thus, we have the following.

► **Corollary 12.** *The queue number of 3-connected cycle-trees is at most 5.*

The *edge-length ratio* of a straight-line graph drawing is the maximum ratio between the Euclidean lengths of e_1 and e_2 , over all edge pairs (e_1, e_2) . The *planar edge-length ratio* of a planar graph G is the infimum edge-length ratio of Γ , over all planar straight-line drawings Γ of G . Constant upper bounds on the planar edge-length ratio are known for outerplanar graphs [50] and for Halin graphs [33]. We exploit the property that graphs that admit s -span weakly leveled planar drawings have planar edge-length ratio at most $2s + 1$ [33, Lemma 4] to obtain a constant upper bound on the edge-length ratio of 3-connected cycle trees.

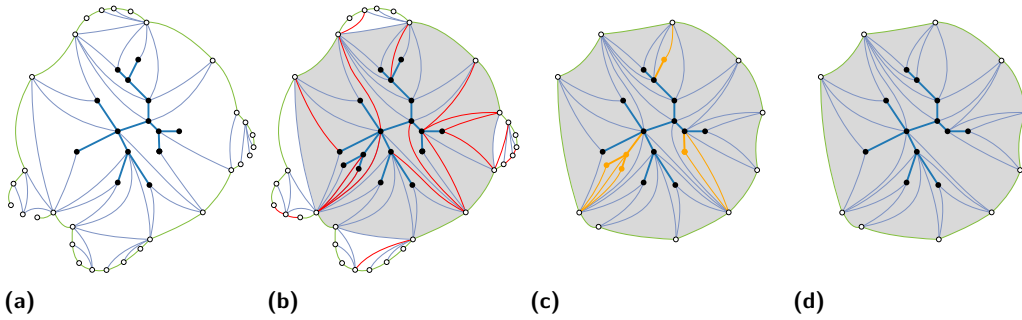
► **Corollary 13.** *The planar edge-length ratio of 3-connected cycle-trees is at most 9.*

General Cycle-Trees. We now discuss general cycle-trees, for which we can prove a $\Theta(\log n)$ bound on the span of their weakly leveled planar drawings.

► **Theorem 14.** *Every n -vertex cycle-tree has an s -span weakly leveled planar drawing such that $s \in O(\log n)$. Also, there exists an n -vertex cycle-tree such that every weakly leveled planar drawing of G has span in $\Omega(\log n)$.*

Proof sketch. For the lower bound, we observe that some cycle-trees require span $\Omega(\log n)$. Indeed, in any planar drawing of the graph in Fig. 3c, a cycle with 3 vertices contains a complete binary tree with $\Omega(n)$ vertices in its interior. Then the lower bound on the span follows from the fact that any weakly levelled planar drawing Γ of a complete binary tree with $\Omega(n)$ vertices has height $\Omega(\log n)$ (because it has $\Omega(\log n)$ pathwidth [15, 52] and the height of Γ is lower-bounded by a linear lower function of the pathwidth of the tree [35]).

For the upper bound, let G be a connected n -vertex cycle-tree. Let \mathcal{E} be a plane embedding of G in which the outer face is delimited by a walk W , so that removing the vertices of W from G one gets a tree T ; see Fig. 9a. We add the maximum number of edges connecting vertices of W with vertices of W and of T , while preserving planarity, simplicity, and the property that every vertex of W is incident to the outer face; see Fig. 9b.



■ **Figure 9** (a) An embedding \mathcal{E} of a cycle-tree G , where the tree T is represented by bold lines. (b) The augmentation of G ; the added edges are red. The face f of \mathcal{E}_{G_W} is gray. (c) Removing the components of G outside C . (d) Removing the components of G inside C .

We now remove some parts of the graph, so that it turns into a 3-connected cycle-tree H . Let G_W be the subgraph of G induced by the vertices of W and let \mathcal{E}_{G_W} be the restriction of \mathcal{E} to G_W . There is a unique face f of \mathcal{E}_{G_W} that contains T in its interior; let C be the cycle delimiting f . We remove from G the vertices of G_W not in C . The removed vertices induce connected subgraphs of G , called *components of G outside C* ; see Fig. 9c. Also, we remove from G all the vertices of T that have at most one neighbor in C . This results in the removal of subtrees of T , which we call *components of G inside C* ; see Fig. 9d.

We next apply Theorem 11 to construct a weakly levelled planar drawing Λ of H with $O(1)$ span and insert $O(\log n)$ levels between any two consecutive levels of Λ . We use such levels to re-introduce the components of G inside and outside C , thus obtaining a weakly levelled planar drawing of G with $O(\log n)$ span. The components of G inside C are trees that can be drawn inside the internal faces of H with $O(\log n)$ height, while ensuring the required vertex visibilities, via an algorithm similar to well-known tree drawing algorithms [22, 28, 53]. The components of G outside C are outerplanar graphs that can be drawn in the outer face of H with $O(\log n)$ height via a suitable combination of results by Biedl [11, 12]. ◀

Planar Graphs with Treewidth 2. In this section, we show that sub-linear span can be achieved for planar graphs with treewidth 2. Note that this is not possible for planar graphs of larger treewidth, as the graph in Fig. 3a has treewidth three and requires span $\Omega(n)$.

► **Theorem 15.** *Every n -vertex planar graph with treewidth 2 has an s -span weakly levelled planar drawing such that $s \in O(\sqrt{n})$. Also, there exists an n -vertex planar graph with treewidth 2 such that every weakly levelled planar drawing of G has span in $2^{\Omega(\sqrt{\log n})}$.*

Proof sketch. Biedl [11] proved that every n -vertex planar graph G with treewidth 2 admits a planar y -monotone grid drawing Γ with $O(\sqrt{n})$ height, that is, the drawing touches $O(\sqrt{n})$ horizontal grid lines. Interpreting the placement of the vertices along these lines as a leveling shows that G admits a leveled planar drawing Γ with height, and hence span, $O(\sqrt{n})$.

The lower bound uses a construction by Frati [42]. Note that $2^{\Omega(\sqrt{\log n})}$ is larger than any poly-logarithmic function of n , but smaller than any polynomial function of n . ◀

Since graphs that admit s -span weakly leveled planar drawings have planar edge-length ratio at most $2s + 1$ [33, Lemma 4], we obtain the following result a corollary of Theorem 15, improving upon a previous $O(n^{0.695})$ bound by Borrizzo and Frati [19]

► **Corollary 16.** *Treewidth-2 graphs with n vertices have planar edge-length ratio $O(\sqrt{n})$.*

6 Open Problems

We studied s -span weakly leveled planar drawings from an algorithmic and a combinatorial perspective. We conclude by listing natural open problems arising from our research:

- Does s -SPAN WEAKLY LEVELED PLANARITY have a kernel of polynomial size when parameterized by the treedepth? Is the problem FPT with respect to the treewidth?
- Theorem 15 shows a gap between the lower and upper bounds in the span for the family of 2-trees. It would be interesting to reduce and possibly close this gap.
- It would also be interesting to close the gap between the lower bound of $\Omega(\log n)$ [13, 14] and the upper bound of $O(\sqrt{n})$ of Corollary 16 on the edge-length ratio of 2-trees.

References

- 1 Martin Balko, Steven Chaplick, Robert Gania, Siddharth Gupta, Michael Hoffmann, Pavel Valtr, and Alexander Wolff. Bounding and computing obstacle numbers of graphs. In Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman, editors, *30th Annual European Symposium on Algorithms, ESA 2022, September 5-9, 2022, Berlin/Potsdam, Germany*, volume 244 of *LIPICs*, pages 11:1–11:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICs.ESA.2022.11.
- 2 Michael J. Bannister, William E. Devanny, Vida Dujmović, David Eppstein, and David R. Wood. Track layouts, layered path decompositions, and leveled planarity. *Algorithmica*, 81(4):1561–1583, 2019. doi:10.1007/s00453-018-0487-5.
- 3 Oliver Bastert and Christian Matuszewski. Layered drawings of digraphs. In Michael Kaufmann and Dorothea Wagner, editors, *Drawing Graphs, Methods and Models*, volume 2025 of *Lecture Notes in Computer Science*, pages 87–120. Springer, 1999. doi:10.1007/3-540-44969-8_5.
- 4 Michael Bekos, Giordano Da Lozzo, Fabrizio Frati, Siddharth Gupta, Philipp Kindermann, Giuseppe Liotta, Ignaz Rutter, and Ioannis G. Tollis. Weakly leveled planarity with bounded span, 2024. arXiv:2409.01889.
- 5 Sujoy Bhore, Giordano Da Lozzo, Fabrizio Montecchiani, and Martin Nöllenburg. On the upward book thickness problem: Combinatorial and complexity results. In Helen C. Purchase and Ignaz Rutter, editors, *Graph Drawing and Network Visualization - 29th International Symposium, GD 2021, Tübingen, Germany, September 14-17, 2021, Revised Selected Papers*, volume 12868 of *Lecture Notes in Computer Science*, pages 242–256. Springer, 2021. doi:10.1007/978-3-030-92931-2_18.
- 6 Sujoy Bhore, Giordano Da Lozzo, Fabrizio Montecchiani, and Martin Nöllenburg. On the upward book thickness problem: Combinatorial and complexity results. *Eur. J. Comb.*, 110:103662, 2023. doi:10.1016/J.EJC.2022.103662.

- 7 Sujoy Bhore, Robert Ganian, Fabrizio Montecchiani, and Martin Nöllenburg. Parameterized algorithms for book embedding problems. In Daniel Archambault and Csaba D. Tóth, editors, *Graph Drawing and Network Visualization - 27th International Symposium, GD 2019, Prague, Czech Republic, September 17-20, 2019, Proceedings*, volume 11904 of *Lecture Notes in Computer Science*, pages 365–378. Springer, 2019. doi:10.1007/978-3-030-35802-0_28.
- 8 Sujoy Bhore, Robert Ganian, Fabrizio Montecchiani, and Martin Nöllenburg. Parameterized algorithms for book embedding problems. *J. Graph Algorithms Appl.*, 24(4):603–620, 2020. doi:10.7155/JGAA.00526.
- 9 Sujoy Bhore, Robert Ganian, Fabrizio Montecchiani, and Martin Nöllenburg. Parameterized algorithms for queue layouts. In David Auber and Pavel Valtr, editors, *Graph Drawing and Network Visualization - 28th International Symposium, GD 2020, Vancouver, BC, Canada, September 16-18, 2020, Revised Selected Papers*, volume 12590 of *Lecture Notes in Computer Science*, pages 40–54. Springer, 2020. doi:10.1007/978-3-030-68766-3_4.
- 10 Sujoy Bhore, Robert Ganian, Fabrizio Montecchiani, and Martin Nöllenburg. Parameterized algorithms for queue layouts. *J. Graph Algorithms Appl.*, 26(3):335–352, 2022. doi:10.7155/JGAA.00597.
- 11 Therese Biedl. Small drawings of outerplanar graphs, series-parallel graphs, and other planar graphs. *Discret. Comput. Geom.*, 45(1):141–160, 2011. doi:10.1007/s00454-010-9310-z.
- 12 Therese Biedl. Height-preserving transformations of planar graph drawings. In Christian A. Duncan and Antonios Symvonis, editors, *22nd International Symposium on Graph Drawing, GD 2014, September 24-26, 2014, Würzburg, Germany*, volume 8871 of *Lecture Notes in Computer Science*, pages 380–391. Springer, 2014. doi:10.1007/978-3-662-45803-7_32.
- 13 Václav Blazej, Jirí Fiala, and Giuseppe Liotta. On the edge-length ratio of 2-trees. In David Auber and Pavel Valtr, editors, *Graph Drawing and Network Visualization - 28th International Symposium, GD 2020, Vancouver, BC, Canada, September 16-18, 2020, Revised Selected Papers*, volume 12590 of *Lecture Notes in Computer Science*, pages 85–98. Springer, 2020. doi:10.1007/978-3-030-68766-3_7.
- 14 Václav Blazej, Jirí Fiala, and Giuseppe Liotta. On edge-length ratios of partial 2-trees. *Int. J. Comput. Geom. Appl.*, 31(2-3):141–162, 2021. doi:10.1142/S0218195921500072.
- 15 Hans L. Bodlaender. A partial k -arboretum of graphs with bounded treewidth. *Theor. Comput. Sci.*, 209(1-2):1–45, 1998. doi:10.1016/S0304-3975(97)00228-4.
- 16 Nicolas Bonichon, Stefan Felsner, and Mohamed Mosbah. Convex drawings of 3-connected plane graphs. In János Pach, editor, *Graph Drawing, 12th International Symposium, GD 2004, New York, NY, USA, September 29 - October 2, 2004, Revised Selected Papers*, volume 3383 of *Lecture Notes in Computer Science*, pages 60–70. Springer, 2004. doi:10.1007/978-3-540-31843-9_8.
- 17 Nicolas Bonichon, Stefan Felsner, and Mohamed Mosbah. Convex drawings of 3-connected plane graphs. *Algorithmica*, 47(4):399–420, 2007. doi:10.1007/s00453-006-0177-6.
- 18 Manuel Borrizzo and Fabrizio Frati. On the edge-length ratio of planar graphs. In Daniel Archambault and Csaba D. Tóth, editors, *Graph Drawing and Network Visualization - 27th International Symposium, GD 2019, Prague, Czech Republic, September 17-20, 2019, Proceedings*, volume 11904 of *Lecture Notes in Computer Science*, pages 165–178. Springer, 2019. doi:10.1007/978-3-030-35802-0_13.
- 19 Manuel Borrizzo and Fabrizio Frati. On the planar edge-length ratio of planar graphs. *J. Comput. Geom.*, 11(1):137–155, 2020. doi:10.20382/jocg.v11i1a6.
- 20 Guido Brückner, Nadine Davina Krisam, and Tamara Mchedlidze. Level-planar drawings with few slopes. In Daniel Archambault and Csaba D. Tóth, editors, *Graph Drawing and Network Visualization - 27th International Symposium, GD 2019, Prague, Czech Republic, September 17-20, 2019, Proceedings*, volume 11904 of *Lecture Notes in Computer Science*, pages 559–572. Springer, 2019. doi:10.1007/978-3-030-35802-0_42.
- 21 Guido Brückner, Nadine Davina Krisam, and Tamara Mchedlidze. Level-planar drawings with few slopes. *Algorithmica*, 84(1):176–196, 2022. doi:10.1007/S00453-021-00884-X.

- 22 Timothy M. Chan. Tree drawings revisited. *Discret. Comput. Geom.*, 63(4):799–820, 2020. doi:10.1007/S00454-019-00106-W.
- 23 Steven Chaplick, Giordano Da Lozzo, Emilio Di Giacomo, Giuseppe Liotta, and Fabrizio Montecchiani. Planar drawings with few slopes of halin graphs and nested pseudotrees. *Algorithmica*, 86(8):2413–2447, 2024. doi:10.1007/S00453-024-01230-7.
- 24 Steven Chaplick, Emilio Di Giacomo, Fabrizio Frati, Robert Ganian, Chrysanthi N. Raftopoulou, and Kirill Simonov. Parameterized algorithms for upward planarity. In Xavier Goaoc and Michael Kerber, editors, *38th International Symposium on Computational Geometry, SoCG 2022, June 7-10, 2022, Berlin, Germany*, volume 224 of *LIPICs*, pages 26:1–26:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICs.SOCG.2022.26.
- 25 Steven Chaplick, Giordano Da Lozzo, Emilio Di Giacomo, Giuseppe Liotta, and Fabrizio Montecchiani. Planar drawings with few slopes of Halin graphs and nested pseudotrees. In Anna Lubiw and Mohammad R. Salavatipour, editors, *17th Algorithms and Data Structures Symposium, WADS 2021, August 9-11, 2021, Halifax, Nova Scotia, Canada*, volume 12808 of *Lecture Notes in Computer Science*, pages 271–285. Springer, 2021. doi:10.1007/978-3-030-83508-8_20.
- 26 Guillaume Chapuy, Éric Fusy, Omer Giménez, Bojan Mohar, and Marc Noy. Asymptotic enumeration and limit laws for graphs of fixed genus. *J. Comb. Theory, Ser. A*, 118(3):748–777, 2011. doi:10.1016/J.JCTA.2010.11.014.
- 27 Marek Chrobak and Goos Kant. Convex grid drawings of 3-connected planar graphs. *Int. J. Comput. Geom. Appl.*, 7(3):211–223, 1997. doi:10.1142/S0218195997000144.
- 28 Pierluigi Crescenzi, Giuseppe Di Battista, and Adolfo Piperno. A note on optimal area algorithms for upward drawings of binary trees. *Comput. Geom.*, 2:187–200, 1992. doi:10.1016/0925-7721(92)90021-J.
- 29 Giordano Da Lozzo, William E. Devanny, David Eppstein, and Timothy Johnson. Square-contact representations of partial 2-trees and triconnected simply-nested graphs. In Yoshio Okamoto and Takeshi Tokuyama, editors, *28th International Symposium on Algorithms and Computation, ISAAC 2017, December 9-12, 2017, Phuket, Thailand*, volume 92 of *LIPICs*, pages 24:1–24:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPICs.ISAAC.2017.24.
- 30 Hubert de Fraysseix, János Pach, and Richard Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990. doi:10.1007/BF02122694.
- 31 Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, 1999.
- 32 Giuseppe Di Battista and Enrico Nardelli. Hierarchies and planarity theory. *IEEE Trans. Syst. Man Cybern.*, 18(6):1035–1046, 1988. doi:10.1109/21.23105.
- 33 Emilio Di Giacomo, Walter Didimo, Giuseppe Liotta, Henk Meijer, Fabrizio Montecchiani, and Stephen Wismath. New bounds on the local and global edge-length ratio of planar graphs, 2023. *Proceedings of the 40th European Workshop on Computational Geometry, EuroCG 2024*. doi:10.48550/arXiv.2311.14634.
- 34 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.
- 35 Vida Dujmović, Michael R. Fellows, Matthew Kitching, Giuseppe Liotta, Catherine McCartin, Naomi Nishimura, Prabhakar Ragde, Frances A. Rosamond, Sue Whitesides, and David R. Wood. On the parameterized complexity of layered graph drawing. *Algorithmica*, 52(2):267–292, 2008. doi:10.1007/s00453-007-9151-1.
- 36 Vida Dujmović, Gwenaél Joret, Piotr Micek, Pat Morin, Torsten Ueckerdt, and David R. Wood. Planar graphs have bounded queue-number. *J. ACM*, 67(4):22:1–22:38, 2020. doi:10.1145/3385731.
- 37 Vida Dujmović, Attila Pór, and David R. Wood. Track layouts of graphs. *Discret. Math. Theor. Comput. Sci.*, 6(2):497–522, 2004. doi:10.46298/DMTCS.315.

- 38 Stefan Felsner, Giuseppe Liotta, and Stephen K. Wismath. Straight-line drawings on restricted integer grids in two and three dimensions. In Petra Mutzel, Michael Jünger, and Sebastian Leipert, editors, *Graph Drawing, 9th International Symposium, GD 2001 Vienna, Austria, September 23-26, 2001, Revised Papers*, volume 2265 of *Lecture Notes in Computer Science*, pages 328–342. Springer, 2001. doi:10.1007/3-540-45848-4_26.
- 39 Stefan Felsner, Giuseppe Liotta, and Stephen K. Wismath. Straight-line drawings on restricted integer grids in two and three dimensions. *J. Graph Algorithms Appl.*, 7(4):363–398, 2003. doi:10.7155/jgaa.00075.
- 40 Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization: Theory of Parameterized Preprocessing*. Cambridge University Press, 2019. doi:10.1017/9781107415157.
- 41 Michael Formann, Torben Hagerup, James Haralambides, Michael Kaufmann, Frank Thomson Leighton, Antonios Symvonis, Emo Welzl, and Gerhard J. Woeginger. Drawing graphs in the plane with high resolution. *SIAM J. Comput.*, 22(5):1035–1052, 1993. doi:10.1137/0222063.
- 42 Fabrizio Frati. Lower bounds on the area requirements of series-parallel graphs. *Discret. Math. Theor. Comput. Sci.*, 12(5):139–174, 2010. doi:10.46298/dmtcs.500.
- 43 István Fáry. On straight lines representation of planar graphs. *Acta Sci. Math. (Szeged)*, 11:229–233, 1948.
- 44 Lenwood S. Heath and Arnold L. Rosenberg. Laying out graphs using queues. *SIAM J. Comput.*, 21(5):927–958, 1992. doi:10.1137/0221055.
- 45 Petr Hliněný and Abhisekh Sankaran. Exact crossing number parameterized by vertex cover. In Daniel Archambault and Csaba D. Tóth, editors, *Graph Drawing and Network Visualization - 27th International Symposium, GD 2019, September 17-20, 2019, Prague, Czech Republic*, volume 11904 of *Lecture Notes in Computer Science*, pages 307–319. Springer, 2019. doi:10.1007/978-3-030-35802-0_24.
- 46 Michael Hoffmann, Marc J. van Kreveld, Vincent Kusters, and Günter Rote. Quality ratios of measures for graph drawing styles. In *26th Canadian Conference on Computational Geometry, CCCG 2014, August 11-13, 2014, Halifax, Nova Scotia, Canada*. Carleton University, Ottawa, Canada, 2014. URL: <http://www.cccg.ca/proceedings/2014/papers/paper05.pdf>.
- 47 John E. Hopcroft and J. K. Wong. Linear time algorithm for isomorphism of planar graphs (preliminary report). In Robert L. Constable, Robert W. Ritchie, Jack W. Carlyle, and Michael A. Harrison, editors, *6th Annual ACM Symposium on Theory of Computing, STOC 1974, April 30 - May 2, 1974, Seattle, Washington, USA*, pages 172–184. ACM, 1974. doi:10.1145/800119.803896.
- 48 Goos Kant. Drawing planar graphs using the canonical ordering. *Algorithmica*, 16(1):4–32, 1996. doi:10.1007/BF02086606.
- 49 Sylvain Lazard, William J. Lenhart, and Giuseppe Liotta. On the edge-length ratio of outerplanar graphs. In Fabrizio Frati and Kwan-Liu Ma, editors, *Graph Drawing and Network Visualization - 25th International Symposium, GD 2017, Boston, MA, USA, September 25-27, 2017, Revised Selected Papers*, volume 10692 of *Lecture Notes in Computer Science*, pages 17–23. Springer, 2017. doi:10.1007/978-3-319-73915-1_2.
- 50 Sylvain Lazard, William J. Lenhart, and Giuseppe Liotta. On the edge-length ratio of outerplanar graphs. *Theor. Comput. Sci.*, 770:88–94, 2019. doi:10.1016/J.TCS.2018.10.002.
- 51 Seth M. Malitz and Achilleas Papakostas. On the angular resolution of planar graphs. *SIAM J. Discret. Math.*, 7(2):172–183, 1994. doi:10.1137/S0895480193242931.
- 52 Petra Scheffler. *Die Baumweite von Graphen als ein Maß für die Kompliziertheit algorithmischer Probleme*. PhD thesis, Akademie der Wissenschaften der DDR, Berlin, 1989.
- 53 Yossi Shiloach. *Linear and Planar Arrangements of Graphs*. PhD thesis, Weizmann Institute of Science, 1976.

- 54 Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiro Toda. Methods for visual understanding of hierarchical system structures. *IEEE Trans. Syst. Man Cybern.*, 11(2):109–125, 1981. doi:10.1109/TSMC.1981.4308636.
- 55 Roberto Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.*, 16(3):421–444, 1987. doi:10.1137/0216030.
- 56 Roberto Tamassia, editor. *Handbook on Graph Drawing and Visualization*. Chapman and Hall/CRC, 2013. URL: <https://cs.brown.edu/people/rtamassi/gdhandbook>.
- 57 William Thomas Tutte. How to draw a graph. *Pro. London Math. Society*, 3(1):743–767, 1963.
- 58 Meirav Zehavi. Parameterized analysis and crossing minimization problems. *Comput. Sci. Rev.*, 45:100490, 2022. doi:10.1016/J.COSREV.2022.100490.

Quantum Algorithms for One-Sided Crossing Minimization

Susanna Caroppo ✉ 

Roma Tre University, Rome, Italy

Giordano Da Lozzo ✉ 

Roma Tre University, Rome, Italy

Giuseppe Di Battista ✉ 

Roma Tre University, Rome, Italy

Abstract

We present singly-exponential quantum algorithms for the ONE-SIDED CROSSING MINIMIZATION (OSCM) problem. We show that OSCM can be viewed as a set problem amenable for exact algorithms with a quantum speedup with respect to their classical counterparts. First, we exploit the quantum dynamic programming framework of Ambainis et al. [*Quantum Speedups for Exponential-Time Dynamic Programming Algorithms*. SODA 2019] to devise a QRAM-based algorithm that solves OSCM in $\mathcal{O}^*(1.728^n)$ time and space. Second, we use quantum divide and conquer to obtain an algorithm that solves OSCM without using QRAM in $\mathcal{O}^*(2^n)$ time and polynomial space.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms; Mathematics of computing → Graph theory; Theory of computation → Quantum complexity theory

Keywords and phrases One-sided crossing minimization, quantum graph drawing, quantum dynamic programming, quantum divide and conquer, exact exponential algorithms

Digital Object Identifier 10.4230/LIPIcs.GD.2024.20

Related Version *Full Version*: <https://arxiv.org/abs/2409.01942> [3]

Funding This research was supported, in part, by MUR of Italy (PRIN Project no. 2022ME9Z78 – NextGRAAL and PRIN Project no. 2022TS4Y3N – EXPAND).

Acknowledgements We acknowledge the CINECA award under the ISCRA initiative, for the availability of high-performance computing resources and support.

1 Introduction and Preliminaries

We study, from the quantum perspective, the ONE-SIDE CROSSING MINIMIZATION (OSCM) problem, one of the most studied problems in Graph Drawing, which is defined below.

2-Level Drawings. In a 2-level drawing of a bipartite graph the vertices of the two sets of the bipartition are placed on two horizontal lines and the edges are drawn as straight-line segments. The number of crossings of the drawing is determined by the order of the vertices on the two lines. Formally, let $G = (U, V, E)$ be a bipartite graph, where U and V are the two parts of the vertex set of G and $E \subseteq U \times V$ is the edge set of G . In the following, we write n , n_U , and n_V for $|U \cup V|$, $|U|$, and $|V|$, respectively; also, for every integer h , we use the notation $[h]$ to refer to the set $\{1, \dots, h\}$. A *2-level drawing* of G is a pair (π_U, π_V) , where $\pi_U : U \leftrightarrow \{1, \dots, |U|\}$ is a linear ordering of U and $\pi_V : V \leftrightarrow \{1, \dots, |V|\}$ is a linear ordering of V . We denote the vertices of U by u_i ($i \in [n_U]$), and the vertices of V by v_j ($j \in [n_V]$). Two edges (u_1, v_1) and (u_2, v_2) in E *cross* in (π_U, π_V) if: (i) $u_1 \neq u_2$ and $v_1 \neq v_2$ and (ii) either $\pi_U(u_1) < \pi_U(u_2)$ and $\pi_V(v_2) < \pi_V(v_1)$, or $\pi_U(u_2) < \pi_U(u_1)$ and $\pi_V(v_1) < \pi_V(v_2)$. The number of crossings of a 2-level drawing (π_U, π_V) is the number $cr(G, \pi_U, \pi_V)$ of distinct (unordered) pairs of edges that cross. Problem OSCM is defined as follows:



© Susanna Caroppo, Giordano Da Lozzo, and Giuseppe Di Battista;
licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 20; pp. 20:1–20:9

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

ONE-SIDED CROSSING MINIMIZATION (OSCM)

Input: A bipartite graph $G = (U, V, E)$ and a linear ordering $\pi_U : U \leftrightarrow [n_U]$.
Output: A linear ordering $\pi_V : V \leftrightarrow [n_V]$ such that $cr(G, \pi_U, \pi_V)$ is minimum.

State of the art. The importance of the OSCM problem, which is NP-complete [10] even for sparse graphs [19], in Graph Drawing was first put in evidence by Sugiyama in [22].

Exact solutions of OSCM have been searched with branch-and-cut techniques, see e.g. [16, 20, 24], and with FPT algorithms. The parameterized version of the problem, with respect to its natural parameter $k = \min_{\pi_V} cr(G, \pi_U, \pi_V)$, has been widely investigated. Dujmovic et al. [7, 8] were the first to show that OSCM can be solved in $f(k)n^{\mathcal{O}(1)}$ time, with $f \in \mathcal{O}(\psi^k)$, where $\psi \approx 1.6182$ is the golden ratio. Subsequently, Dujmovic and Whitesides [5, 6] improved the running time to $\mathcal{O}(1.4656^k + kn^2)$. Fernau et al. [11], exploiting a reduction to weighted FAST and the algorithm by Alon et al. [1], gave a subexponential parameterized algorithm with running time $2^{\mathcal{O}(\sqrt{k} \log k)} + n^{\mathcal{O}(1)}$. The reduction also gives a PTAS using [17]. Kobayashi and Tamaki [18] gave the current best FPT result with running time $\mathcal{O}(k2^{\sqrt{2k}} + n)$.

Quantum Graph Drawing has recently gained popularity. Caroppo et al. [4] applied Grover's search [14] to several Graph Drawing problems obtaining a quadratic speedup over classical exhaustive search. Fukuzawa et al. [12] studied how to apply quantum techniques for solving systems of linear equations [15] to Tutte's algorithm for drawing planar 3-connected graphs [23]. Recently, in a paper that pioneered *Quantum Dynamic Programming*, several vertex ordering problems related to Graph Drawing have been tackled by Ambainis et al. [2].

Our Results. First, we exploit the quantum dynamic programming framework of Ambainis et al. to devise an algorithm that solves OSCM in $\mathcal{O}^*(1.728^n)$ time and space. We compare the performance of our algorithm against the algorithm proposed in [18], based on the value of k . We have that the quantum algorithm performs asymptotically better than the FPT algorithm, when $k \in \Omega(n^2)$. Second, we use quantum divide and conquer to obtain an algorithm that solves OSCM using $\mathcal{O}^*(2^n)$ time and polynomial space. Both our algorithms improve the corresponding classical bounds in either time or space or both.

In our first result, we adopt the QRAM (quantum random access memory) model of computation [13], which allows (i) accessing quantum memory in superposition and (ii) invoking any T -time classical algorithm that uses a (classic) random access memory as a subroutine spending time $\mathcal{O}(T)$. In the second result we do not use the QRAM model of computation since we do not need to explicitly store the results obtained in partial computations.

Notation. For ease of notation, given positive integers a and b , we denote $\lceil \frac{a}{b} \rceil$ as $\frac{a}{b}$ and $\lceil \log a \rceil$ as $\log a$. If $f(n) = \mathcal{O}(n^c)$ for some constant c , we will write $f(n) = \text{poly}(n)$. In case $f(n) = d^n \text{poly}(n)$ for some constant d , we use the notation $f(n) = \mathcal{O}^*(d^n)$ (see, e.g., [25]).

Quantum Tools. We will use quantum search primitives, such as the one of Theorem 1, and exploit the fact that they can perform condition checking on data stored in QRAM.

► **Theorem 1** (Quantum Minimum Finding, QMF [9]). *Let $f : D \rightarrow C$ be a polynomial-time computable function, whose domain D has size N and whose codomain C is a totally ordered set (such as \mathbb{N}) and let \mathcal{F} be a procedure that computes f . There exists a bounded-error quantum algorithm that finds $x \in D$ such that $f(x)$ is minimized using $\mathcal{O}(\sqrt{N})$ applications of \mathcal{F} .*

Because of space limitations, some proofs are sketched or omitted. They can be found in the full version of the paper [3].

■ **Algorithm 1** Procedure `QuantumDP` is the algorithm of Lemma 2. Procedure `OPT` is a recursive procedure invoked by `QuantumDP`. Procedure `QMF` performs quantum minimum finding.

```

1: procedure QUANTUMDP( $X$ )
2:   Input: Set  $X$  of size  $n$ ; Output: the value  $OPT_{\mathcal{P}}(X)$ .
3:   for all sets  $W \subset X$  such that  $|W| \leq (1 - \alpha)n/4$  do      ▷ in order of increasing size
4:     Compute  $OPT_{\mathcal{P}}(W)$  classically via dynamic programming    ▷ use Equation (1)
5:     with  $k = |W| - 1$ 
6:     Store  $OPT_{\mathcal{P}}(W)$  in QRAM
7:   end for
8:   return  $OPT(X)$ 
9: end procedure
10: procedure OPT( $S$ )
11:  Input: Subset  $S \subseteq X$ ; Output: the value  $OPT_{\mathcal{P}}(S)$ .
12:  if  $|S| \leq (1 - \alpha)n/4$  then
13:    return value  $OPT_{\mathcal{P}}(S)$  stored in QRAM
14:  else
15:    return the result of QMF over all  $S \subset X$  to find
        
$$\min_{W \subset S, |W| = \lfloor \frac{|S|}{2} \rfloor} \{OPT(W) + OPT(S \setminus W) + f_{\mathcal{P}}(W, S \setminus W)\}$$

16:  end if
17: end procedure

```

2 Quantum Dynamic Programming for One-Sided Crossing Minimization

In this section, we first describe the quantum dynamic programming framework of Ambainis et al. [2], which is applicable to numerous optimization problems involving sets. Then, we show that OSCM is a set problem over V that falls within this framework. We use this fact to derive a quantum algorithm (Theorem 4) exhibiting a speedup over the corresponding classical singly-exponential algorithm (given in [3]) in both time and space complexity.

Quantum dynamic programming for set problems. The framework by Ambainis et al. is defined by the following lemma derivable from [2].

► **Lemma 2.** *Let \mathcal{P} be an optimization problem (say a minimization problem) over a set X . Let $|X| = n$ and let $OPT_{\mathcal{P}}(X)$ be the optimal value for \mathcal{P} over X . Suppose that there exists a polynomial-time computable function $f_{\mathcal{P}} : 2^X \times 2^X \rightarrow \mathbb{R}$ such that, for any $S \subseteq X$, it holds that for any $k \in [|S| - 1]$:*

$$OPT_{\mathcal{P}}(S) = \min_{W \subset S, |W|=k} \{OPT_{\mathcal{P}}(W) + OPT_{\mathcal{P}}(S \setminus W) + f_{\mathcal{P}}(W, S \setminus W)\} \quad (1)$$

Then, $OPT_{\mathcal{P}}(X)$ can be computed by a quantum algorithm that uses QRAM in $\mathcal{O}^(1.728^n)$ time and space.*

Proof sketch. The algorithm for the proof of the lemma is presented as Algorithm 1. The main idea of the algorithm is to precompute solutions for smaller subsets using classical dynamic programming and then recombine the results of the precomputation step to obtain the optimal solution for the whole set (recursively) applying QMF (see Theorem 1). ◀

Quantum dynamic programming for OSCM. In the following, let (G, π_U) be an instance of OSCM. For any $S \subseteq E$, consider the subgraph $H = (U, V, S)$ of G . For ease of notation, we denote $cr(H, \pi_U, \pi_V)$ simply as $cr_S(\pi_U, \pi_V)$. Also, let π_V be a linear ordering of the vertices in V and $V_1, V_2 \subseteq V$ be two subsets of the vertices of V such that $V_1 \cap V_2 = \emptyset$. We say that V_1 *precedes* V_2 in π_V , denoted as $V_1 \prec_{\pi_V} V_2$, if for any $v_1 \in V_1$ and $v_2 \in V_2$, it holds that $\pi_V(v_1) < \pi_V(v_2)$. Also, for a any $W \subseteq V$, we denote by $E(W)$ the subset of E defined as follows $E(W) := \{(u_a, v_b) : (u_a, v_b) \in E \wedge v_b \in W\}$. We will exploit the following.

► **Lemma 3.** *Let $G = (U, V, E)$ be a bipartite graph and let $\pi_U : U \leftrightarrow [n_U]$ be a linear ordering of the vertices of U . Also, let $V_1, V_2 \subseteq V$ be two subsets of the vertices of V such that $V_1 \cap V_2 = \emptyset$. Then, there exists a constant $\gamma(\pi_U, V_1, V_2)$ such that, for every linear ordering $\pi_V : V \leftrightarrow [n_V]$ with $V_1 \prec_{\pi_V} V_2$ we have that:*

$$\gamma(\pi_U, V_1, V_2) = cr_{E(V_1) \cup E(V_2)}(\pi_U, \pi_V) - cr_{E(V_1)}(\pi_U, \pi_V) - cr_{E(V_2)}(\pi_U, \pi_V) \quad (2)$$

Observe that, given an ordering π_V of V such that V_1 precedes V_2 in π_V , the value $\gamma(\pi_U, V_1, V_2)$ represents the number of crossings in a 2-level drawing (π_U, π_V) of G determined by pairs of edges, one belonging to $E(V_1)$ and the other belonging to $E(V_2)$.

We are now ready to derive our dynamic programming quantum algorithm for OSCM. To this aim, we argue next that the framework of Lemma 2 can be applied to the optimization problem associated with OSCM (i.e., computing the minimum number of crossings over all 2-level drawings (π_U, π_V) of G with π_U fixed), which we call MINOSCM. First, we have that MINOSCM is a set problem over V , whose optimal solution respects a recurrence of the same form as Equation (1). In fact, for a subset S of V , let $OPT(S)$ denote the minimum number of crossings in a 2-level drawing (π_U, π_S) of the graph $G_S = (U, S, E(S))$, where $\pi_S : S \leftrightarrow [|S|]$ is a linear ordering of the vertices of S . Then, by Lemma 3, we can compute $OPT(S)$ by means of the following recurrence for any $k \in [|S| - 1]$:

$$OPT(S) = \min_{W \subseteq S, |W|=k} \{OPT(W) + OPT(S \setminus W) + \gamma(\pi_U, W, S \setminus W)\}$$

Clearly, $OPT(V)$ corresponds to the optimal solution for (G, π_U) . Also, function γ plays the role of function $f_{\mathcal{P}}$ of Lemma 2. Second, we have that γ can be computed in $poly(n)$ time.

In [3], we show that Algorithm 1 applied to MINOSCM can also be adapted to compute an ordering π_V of V that yields a drawing with the minimum number of crossings.

Altogether, we have finally proved the following.

► **Theorem 4.** *There is a bounded-error quantum algorithm that solves OSCM in $O^*(1.728^{n_V})$ time and space.*

Comparing Theorem 4 against the current best FPT result [18] solving OSCM in $\mathcal{O}(k2^{\sqrt{2k}} + n)$ time, where k bounds the number of allowed crossings, we have the following.

► **Corollary 5.** *The algorithm of Theorem 4 is asymptotically more time-efficient than the FPT algorithm parameterized by the number k of crossings in [18] when $k \in \Omega(n^2)$.*

3 Quantum Divide and Conquer for One-Sided Crossing Minimization

Shimizu and Mori [21] used divided and conquer to obtain quantum exponential-time polynomial-space algorithms for coloring problems that do not rely on the use of QRAM. In this section, we first generalize their ideas to obtain a framework designed to speedup, without using QRAM, some classical exponential-time polynomial-space divide and conquer

■ **Algorithm 2** The quantum algorithm of Lemma 6.

```

1: procedure QUANTUMDC( $X$ ):
2:   Input: Set  $X$  of size  $n$ ; Output: the value  $OPT_{\mathcal{P}}(X)$ .
3:   if  $|S| \leq c_{\mathcal{P}}$  then
4:     return  $f_{\mathcal{P}}(S, \emptyset)$ 
5:   end if
6:   return the result of QMF over all  $W \subset S$  with  $|W| = \lfloor \frac{|S|}{2} \rfloor$  to find

```

$$\min_{W \subset S, |W| = \lfloor \frac{|S|}{2} \rfloor} \{ \text{QuantumDC}(W) + \text{QuantumDC}(S \setminus W) + f_{\mathcal{P}}(W, S \setminus W) \}$$

```

7: end procedure

```

algorithms for set problems. Then, we show that OSCM is a set problem over V that falls within this framework. We use this fact to derive a quantum algorithm (Theorem 7) that improves the time bounds of the corresponding classical singly-exponential algorithm (given in the full version of the paper [3]), while maintaining polynomial space complexity.

Quantum divide and conquer for set problems. In the remainder, we provide a general quantum framework, defined by the following lemma.

► **Lemma 6.** *Let \mathcal{P} be an optimization problem (say a minimization problem) over a set X . Let $|X| = n$ and let $OPT_{\mathcal{P}}(X)$ be the optimal value for \mathcal{P} over X . Suppose that there exists a polynomial-time computable function $f_{\mathcal{P}} : 2^X \times 2^X \rightarrow \mathbb{R}$ and a constant $c_{\mathcal{P}}$ such that, for any $S \subseteq X$, it holds that: (i) If $|S| \leq c_{\mathcal{P}}$, then $OPT_{\mathcal{P}}(S) = f_{\mathcal{P}}(S, \emptyset)$; (ii) If $|S| > c_{\mathcal{P}}$, then*

$$OPT_{\mathcal{P}}(S) = \min_{W \subset S, |W| = \lfloor \frac{|S|}{2} \rfloor} \{ OPT_{\mathcal{P}}(W) + OPT_{\mathcal{P}}(S \setminus W) + f_{\mathcal{P}}(W, S \setminus W) \} \quad (3)$$

We have that, $OPT_{\mathcal{P}}(X)$ can be computed by a quantum algorithm without using QRAM in $\mathcal{O}^*(2^n)$ time and polynomial space.

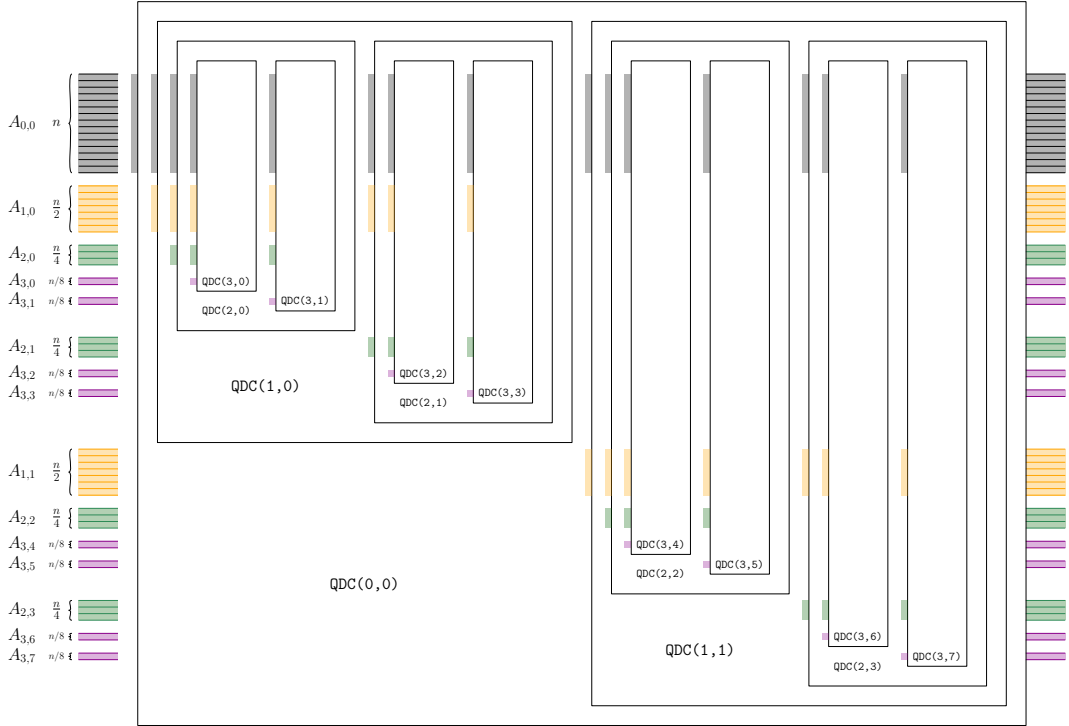
Proof. The algorithm for the proof of the lemma is presented as Algorithm 2 and is based on the recurrence in Equation (3). It works recursively as follows. If the input set X is sufficiently small, i.e., $|X| \leq c_{\mathcal{P}}$, then the optimal value for X is computed directly as $f_{\mathcal{P}}(X, \emptyset)$. Otherwise, it uses QMF to find the optimal pair $(S, X \setminus S)$ that achieves $OPT_{\mathcal{P}}(X)$ according to Equation (3), where $OPT_{\mathcal{P}}(S)$ and $OPT_{\mathcal{P}}(X \setminus S)$ have been recursively computed.

The running time $Q(k)$ of Algorithm 2 when $|X| = k$ obeys the following recurrence:

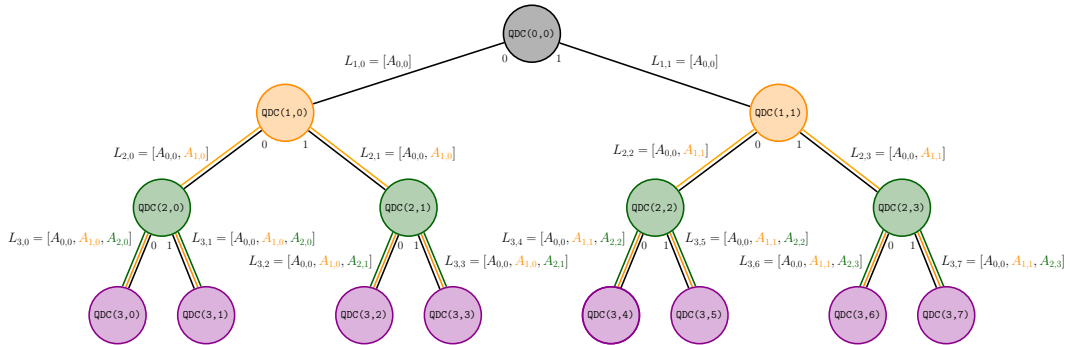
$$Q(k) \leq \sqrt{\mathcal{O}\left(\binom{k}{\lfloor k/2 \rfloor}\right)} \left(Q(\lfloor k/2 \rfloor) + Q(k/2) + \text{poly}(k) \right)$$

Hence, $Q(k) \leq 2^k \text{poly}(k)$, and the total running time of Algorithm 2 is bounded by $\mathcal{O}^*(2^n)$.

The space complexity of Algorithm 2 (procedure QUANTUMDC) can be proved polynomial as follows; see Figure 1. The execution of QUANTUMDC determines a rooted binary tree \mathcal{T} whose nodes are associated with its recursive calls; see Figure 2. Each call corresponds to a circuit in Figure 1. We denote by QDC(i, j) the circuit, at the i^{th} -level of the recursion tree \mathcal{T} , with $i = 0, \dots, \log n - 1$, associated with the j^{th} -call, with $j \in 0, \dots, 2^i - 1$. The input to each of such circuits consists of a set of registers defined as follows. For $i = 0, 1, \dots, \log n - 1$ and $j = 0, 1, \dots, 2^i - 1$, there exists a register $A_{i,j}$ with $\frac{n}{2^i}$ qubits. It stores a superposition



■ **Figure 1** Schematic representation of the circuit realizing Algorithm 2 for a set X with $n = 16$. The qubits in $L_{i,j}$ in input to the circuit $\text{QDC}(i, j)$ are incident to its left boundary.



■ **Figure 2** The tree \mathcal{T} whose nodes are associated with the recursive calls of Algorithm 2.

corresponding to a subset $S_{i,j}$ of X (to be defined later) of size $\frac{n}{2^i}$, which represents all possible ways of splitting the subset into two equal-sized subsets. Specifically, a status 0 for $A_{i,j}[k]$ corresponds to assigning the k^{th} -element of the subset associated with $A_{i,j}$ to one side of the split, while a status 1 of $A_{i,j}[k]$ corresponds to assigning the k^{th} -element of such a subset to the other side of the split. In Figure 2, we associate the split defined by the status-0 qubits (by the status-1 qubits) with the left (right) child of a node. Moreover, in Figure 2, each edge of \mathcal{T} is labeled with the registers representing the corresponding split.

The input of $\text{QDC}(i, j)$ is a set $L_{i,j}$ of $i + 1$ registers of size $n, \frac{n}{2}, \frac{n}{4}, \dots, \frac{n}{2^i}$, respectively; see Figure 1. The registers in input to $\text{QDC}(i, j)$ can be recursively defined as follows. The register $A_{i-1, \lfloor j/2 \rfloor}$ belongs to $L_{i,j}$ and it is the smallest register in this set. Also, if $A_{c,d}$ with $c \geq 1$ belongs to $L_{i,j}$, then $A_{c-1, \lfloor d/2 \rfloor}$ also belong to $L_{i,j}$. In particular, observe that $L_{i,j}$

always contains $A_{0,0}$. The circuit $\text{QDC}(i, j)$ solves problem \mathcal{P} on a subset $S_{i,j}$ of X of size $\frac{n}{2^i}$, which is defined by the states of the registers in $L_{i,j}$. In particular, the set $S_{i,j}$ can be determined by following the path of \mathcal{T} connecting $\text{QDC}(i, j)$ to the root, and observing that the parity of j determines whether a node in the path is the left or right child of its parent.

We can finally bound the space complexity of Algorithm 2, in terms of both bits and qubits. Since our algorithm does not rely on external classic memory, we only need to bound the latter. Note that the number of circuits $\text{QDC}(i, j)$ (which are in a bijection with the nodes of \mathcal{T}) is linear in n and that the number of qubits in $L_{i,j}$, which form the input of $\text{QDC}(i, j)$, is at most $\sum_{i=0}^{\log n} \frac{n}{2^i} = 2n$. Hence, the space complexity of Algorithm 2 is polynomial. ◀

Quantum divide and conquer for OSCM. We now describe a quantum divide and conquer algorithm for OSCM. We start by showing that the framework of Lemma 6 can be applied to MINOSCM (see Section 2). This can be done in a similar fashion as for the Lemma 2. In particular, the fact that the MINOSCM problem is a set problem over V immediately follows from the observation that Equation (3) is the restriction of Equation (1) to the case in which $k = |W| = \frac{|S|}{2}$. Moreover, recall that γ can be computed in $\text{poly}(n)$ time. The execution of Algorithm 2 produces as output a superposition of the registers $A_{i,j}$ such that the state with the highest probability of being returned, if measured, corresponds to an ordering π_V of V that yields a drawing with the minimum number of crossings. In [3], we show how to obtain π_V from such a state. Altogether we have proved the following.

► **Theorem 7.** *There is a bounded-error quantum algorithm that solves OSCM in $\mathcal{O}^*(2^{n_V})$ time and polynomial space.*

4 Conclusions

We presented singly-exponential quantum algorithms for OSCM, exploiting both quantum dynamic programming and quantum divide and conquer. We believe that this research will spark further interest in the design of exact quantum algorithms for hard graph drawing problems. In [3], we highlight two meaningful applications of our results.

References

- 1 Noga Alon, Daniel Lokshtanov, and Saket Saurabh. Fast FAST. In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris E. Nikolettseas, and Wolfgang Thomas, editors, *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part I*, volume 5555 of *Lecture Notes in Computer Science*, pages 49–58. Springer, 2009. doi:10.1007/978-3-642-02927-1_6.
- 2 Andris Ambainis, Kaspars Balodis, Janis Iraids, Martins Kokainis, Krisjanis Prusis, and Jevgenijs Vihrovs. Quantum speedups for exponential-time dynamic programming algorithms. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1783–1793. SIAM, 2019. doi:10.1137/1.9781611975482.107.
- 3 Susanna Caroppo, Giordano Da Lozzo, and Giuseppe Di Battista. Quantum algorithms for one-sided crossing minimization, 2024. arXiv:2409.01942.
- 4 Susanna Caroppo, Giordano Da Lozzo, and Giuseppe Di Battista. Quantum graph drawing. In Ryuhei Uehara, Katsuhisa Yamanaka, and Hsu-Chun Yen, editors, *WALCOM: Algorithms and Computation - 18th International Conference and Workshops on Algorithms and Computation, WALCOM 2024, Kanazawa, Japan, March 18-20, 2024, Proceedings*, volume 14549 of *Lecture Notes in Computer Science*, pages 32–46. Springer, 2024. doi:10.1007/978-981-97-0566-5_4.

- 5 Vida Dujmovic, Henning Fernau, and Michael Kaufmann. Fixed parameter algorithms for one-sided crossing minimization revisited. In Giuseppe Liotta, editor, *Graph Drawing, 11th International Symposium, GD 2003, Perugia, Italy, September 21-24, 2003, Revised Papers*, volume 2912 of *Lecture Notes in Computer Science*, pages 332–344. Springer, 2003. doi:10.1007/978-3-540-24595-7_31.
- 6 Vida Dujmovic, Henning Fernau, and Michael Kaufmann. Fixed parameter algorithms for one-sided crossing minimization revisited. *J. Discrete Algorithms*, 6(2):313–323, 2008. doi:10.1016/J.JDA.2006.12.008.
- 7 Vida Dujmovic and Sue Whitesides. An efficient fixed parameter tractable algorithm for 1-sided crossing minimization. In Stephen G. Kobourov and Michael T. Goodrich, editors, *Graph Drawing, 10th International Symposium, GD 2002, Irvine, CA, USA, August 26-28, 2002, Revised Papers*, volume 2528 of *Lecture Notes in Computer Science*, pages 118–129. Springer, 2002. doi:10.1007/3-540-36151-0_12.
- 8 Vida Dujmovic and Sue Whitesides. An efficient fixed parameter tractable algorithm for 1-sided crossing minimization. *Algorithmica*, 40(1):15–31, 2004. doi:10.1007/S00453-004-1093-2.
- 9 Christoph Dürr and Peter Høyer. A quantum algorithm for finding the minimum. *CoRR*, quant-ph/9607014, 1996. arXiv:quant-ph/9607014.
- 10 Peter Eades and Nicholas C. Wormald. Edge crossings in drawings of bipartite graphs. *Algorithmica*, 11(4):379–403, 1994. doi:10.1007/BF01187020.
- 11 Henning Fernau, Fedor V. Fomin, Daniel Lokshtanov, Matthias Mnich, Geevarghese Philip, and Saket Saurabh. Ranking and drawing in subexponential time. In Costas S. Iliopoulos and William F. Smyth, editors, *Combinatorial Algorithms - 21st International Workshop, IWOCA 2010, London, UK, July 26-28, 2010, Revised Selected Papers*, volume 6460 of *Lecture Notes in Computer Science*, pages 337–348. Springer, 2010. doi:10.1007/978-3-642-19222-7_34.
- 12 Shion Fukuzawa, Michael T. Goodrich, and Sandy Irani. Quantum Tutte embeddings. *CoRR*, abs/2307.08851, 2023. doi:10.48550/arXiv.2307.08851.
- 13 Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Phys. Rev. Lett.*, 100:160501, April 2008. doi:10.1103/PhysRevLett.100.160501.
- 14 Lov K. Grover. A fast quantum mechanical algorithm for database search. In Gary L. Miller, editor, *STOC 1996*, pages 212–219. ACM, 1996. doi:10.1145/237814.237866.
- 15 Aram W. Harrow. Quantum algorithms for systems of linear equations. In *Encyclopedia of Algorithms*, pages 1680–1683. Springer New York, 2016. doi:10.1007/978-1-4939-2864-4_771.
- 16 Michael Jünger and Petra Mutzel. Exact and heuristic algorithms for 2-layer straightline crossing minimization. In Franz-Josef Brandenburg, editor, *Graph Drawing, Symposium on Graph Drawing, GD '95, Passau, Germany, September 20-22, 1995, Proceedings*, volume 1027 of *Lecture Notes in Computer Science*, pages 337–348. Springer, 1995. doi:10.1007/BFB0021817.
- 17 Claire Kenyon-Mathieu and Warren Schudy. How to rank with few errors. In David S. Johnson and Uriel Feige, editors, *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 95–103. ACM, 2007. doi:10.1145/1250790.1250806.
- 18 Yasuaki Kobayashi and Hisao Tamaki. A fast and simple subexponential fixed parameter algorithm for one-sided crossing minimization. *Algorithmica*, 72(3):778–790, 2015. doi:10.1007/S00453-014-9872-X.
- 19 Xavier Muñoz, Walter Unger, and Imrich Vrto. One sided crossing minimization is NP-hard for sparse graphs. In Petra Mutzel, Michael Jünger, and Sebastian Leipert, editors, *Graph Drawing, 9th International Symposium, GD 2001 Vienna, Austria, September 23-26, 2001, Revised Papers*, volume 2265 of *Lecture Notes in Computer Science*, pages 115–123. Springer, 2001. doi:10.1007/3-540-45848-4_10.
- 20 Petra Mutzel and René Weiskircher. Two-layer planarization in graph drawing. In Kyung-Yong Chwa and Oscar H. Ibarra, editors, *Algorithms and Computation, 9th International Symposium, ISAAC '98, Taejeon, Korea, December 14-16, 1998, Proceedings*, volume 1533 of *Lecture Notes in Computer Science*, pages 69–78. Springer, 1998. doi:10.1007/3-540-49381-6_9.

- 21 Kazuya Shimizu and Ryuhei Mori. Exponential-time quantum algorithms for graph coloring problems. *Algorithmica*, 84(12):3603–3621, 2022. doi:10.1007/S00453-022-00976-2.
- 22 Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiro Toda. Methods for visual understanding of hierarchical system structures. *IEEE Trans. Syst. Man Cybern.*, 11(2):109–125, 1981. doi:10.1109/TSMC.1981.4308636.
- 23 William Thomas Tutte. How to draw a graph. *Proceedings of the London Mathematical Society*, 3(1):743–767, 1963.
- 24 Vicente Valls, Rafael Martí, and Pilar Lino. A branch and bound algorithm for minimizing the number of crossing arcs in bipartite graphs. *European journal of operational research*, 90(2):303–319, 1996. doi:10.1016/0377-2217(95)00356-8.
- 25 Gerhard J. Woeginger. Open problems around exact algorithms. *Discret. Appl. Math.*, 156(3):397–405, 2008. doi:10.1016/J.DAM.2007.03.023.

The Perception of Stress in Graph Drawings

Gavin J. Mooney   

Monash University, Melbourne, Australia

Helen C. Purchase   

Monash University, Melbourne, Australia

Michael Wybrow   

Monash University, Melbourne, Australia

Stephen G. Kobourov   

Technical University of Munich, Campus Heilbronn, Germany

Jacob Miller   

Technical University of Munich, Campus Heilbronn, Germany

Abstract

Most of the common graph layout principles (a.k.a. “aesthetics”) on which many graph drawing algorithms are based are easy to define and to perceive. For example, the number of pairs of edges that cross each other, how symmetric a drawing looks, the aspect ratio of the bounding box, or the angular resolution at the nodes. The extent to which a graph drawing conforms to these principles can be determined by looking at how it is drawn – that is, by looking at the marks on the page – without consideration for the underlying structure of the graph. A key layout principle is that of optimising “stress”, the basis for many algorithms such as the popular Kamada & Kawai algorithm and several force-directed algorithms. The stress of a graph drawing is, loosely speaking, the extent to which the geometric distance between each pair of nodes is proportional to the shortest path between them – over the whole graph drawing. The definition of stress therefore relies on the underlying structure of the graph (the “paths”) in a way that other layout principles do not, making stress difficult to describe to novices unfamiliar with graph drawing principles, and, we believe, difficult to perceive. We conducted an experiment to see whether people (novices as well as experts) can see stress in graph drawings, and found that it is possible to train novices to “see” stress – even if their perception strategies are not based on the definitional concepts.

2012 ACM Subject Classification Human-centered computing → Empirical studies in visualization

Keywords and phrases Stress, Graph Drawing, Visual Perception

Digital Object Identifier 10.4230/LIPIcs.GD.2024.21

Related Version *Full Version*: <http://arxiv.org/abs/2409.04493>

1 Introduction

Algorithms for depicting graphs are based on conformance to one or more “layout principles” (or “aesthetics”). These principles are chosen so that the resulting drawing is considered “good”, in terms of being easier to read or understand when compared to a random layout that does not take any useful principles into account. Some empirical work has been undertaken to determine whether these principles really do enhance human understanding, with support found for reducing the number of edge crossings [19, 20], depicting symmetry [19], and having wide edge crossing angles [13].

Many of these principles can easily be related to the process of visual perception. For example, the principle that edges adjacent at a node should subtend as wide an angle as possible relates to the limitations of visual acuity, as does the need to keep adequate space between nodes and edges; the principle of depicting symmetric subgraphs in symmetric



© Gavin J. Mooney, Helen C. Purchase, Michael Wybrow, Stephen G. Kobourov, and Jacob Miller; licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 21; pp. 21:1–21:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

21:2 The Perception of Stress in Graph Drawings

form refers to the Gestalt law of symmetry; bent edges are contrary to the Gestalt law of continuity; an aspect ratio that is extremely different from the Golden Ratio can result in a sense of visual imbalance.

Optimising “stress” in a graph drawing (e.g. by [11, 14, 30]) is a common layout principle that is much-cited by graph drawing researchers. It relates to the extent to which the geometric distance between each pair of nodes in the graph drawing is proportional to the shortest graph-theoretic path between them – over the whole graph drawing. The definition of stress therefore relies on calculations associated with the underlying structure of the graph (the “paths”) in a way that other layout principles do not.

So, while most layout principles can clearly relate solely to visual perception, “stress” cannot – it requires analysis of the graph structure as well. This does not mean that there has been no empirical work that refers to stress, simply that it has not done so with explicit use of the word “stress”. Marner et al. conducted an experiment where participants were asked to untangle large graphs on a wall-display, with the support of a novel algorithmic technique for moving several nodes at once [17]. They found that the drawings created by users did not have lower stress or fewer edge crossings when compared with the graphs drawn using the Fruchterman & Reingold algorithm [9], and they question the importance of these two metrics for creating graphs suitable for human understanding. Their instructions to their participants make no explicit reference to the property of stress.

Chimani et al. claim that “people prefer less stress and fewer crossings” [5]. Their two-alternative-forced-choice experiment asked participants to choose their drawing of preference, with the stimuli carefully chosen with variations in both stress and edge crossings. While it was shown that participants tended to prefer the drawings with lower stress (57%) and fewer crossings (65%), the former is an implicit variable (unseen and unexplained to the participants) and the latter is explicit – that is, it is obviously seen.

Stress-preference experimental papers like these make use of the computational stress measure in their data analysis, but do not educate the participants about the concept of stress – it is treated as a “hidden” feature, that cannot be “seen” by participants. This paper represents the first attempt to determine whether people can “see” stress in a graph drawing when the concept is explained to them. This is important for graph perception experiments that attempt to determine the most effective (or most preferred, or most efficient) layout algorithms for human understanding. Stress is a key feature of many common algorithms – if we cannot be sure that participants understand what it is, how can we assess the extent to which they value it in graph drawings?

Anecdotally, graph drawing researchers often claim that they can “see stress” – i.e., that it is as immediately perceptible as other layout principles. While this may be the case for those fluent in graph drawing principles forms (who have an internally and possibly intuitive understanding of the stress principle), it is unclear that “stress” can be explained sufficiently well to non-experts that they too can perceive stress in a graph drawing.

In this paper, we explore whether it is possible to “see” a feature of a graph drawing that is implicitly defined by both graph structure as well as visual form. We conduct a series of human experiments, asking participants to distinguish between pairs of graph drawings, identifying which has lower stress. While our results necessarily depend on the quality of the explanation of stress given to novices, we find that even experts find the task challenging.

2 Background

2.1 Applications of Stress in Graph Drawing

Kamada and Kawai [14] first introduced the idea of using stress in graph drawing in the late 80s, with the motivation that a good drawing should accurately represent something about the graph structure. In this case, that drawn distances be proportional to graph-theoretic distances. The Kamada-Kawai layout algorithm has seen several improvements such as stress majorization [10, 11] and stochastic gradient descent [30]. These most recent versions are among the most popular layout algorithms today, as can be seen by their implementation in libraries like NetworkX (`kamada_kawai_layout`), GraphViz (Neato), yEd (Organic), and others. Stress-based optimisation of multiple graph layout aesthetics is considered by Ahmed et al. [2] and Devkota et al. [6].

Kruiger et al. [15] and Zhu et al. [32] generalize the classical dimensionality reduction algorithm t-SNE [26] and include stress as one of the evaluation criteria. Other papers that optimise stress-related functions and evaluate stress include those by Zhong et al. [31] and Xue et al. [29].

Regardless of whether an algorithm explicitly optimises stress, it is very commonly used as a metric to evaluate the quality of graph drawings and layout algorithms. A recent survey of graph layout algorithms [7] shows that it is the third most common metric employed in the GD community, behind running time and number of edge crossings. Stress is used in the evaluation of graph layout algorithms in papers by Hong et al. [12] and Marmer et al. [17], as well in dynamic graph layout methods, as in Simonetto et al. [22] and Arleo et al. [3]. Brandes and Pich [4] evaluate several graph layout algorithms based on how well they optimise stress and Welch and Kobourov show that stress can be used as an alternative measure of symmetry in graph layouts [28].

2.2 Stress Definitions

The definition of stress is much older than the Kamada-Kawai algorithm, with roots in statistical analysis. Stress as it is known in the graph drawing community began with Torgerson, who proposed a technique now known as metric Multi-Dimensional Scaling (MDS) [25]. Torgerson aims to provide a low dimensional representation of a set of objects on which distances are given. Importantly, little restriction is given on where these distances come from, e.g. they may be from a traditional metric space \mathbb{R}^d , responses to a Likert scale, or indeed, graph-theoretic distances. However the distances are obtained, they are collected into a matrix $D \in \mathbb{R}^{n \times n}$, with the cell $d_{i,j}$ containing the distance between object i and object j . MDS aims to find a matrix $X \in \mathbb{R}^{n \times 2}$ where X_i represents a low dimensional coordinate of object i such that distances between rows in X are exactly the distances in D . This is not possible in general, so the deviation of these distances are measured as a function:

$$\sum_{i < j} \frac{(\|X_i - X_j\| - d_{i,j})^2}{d_{i,j}^2} \quad (1)$$

Note that this function is differentiable with respect to X , so gradient-based optimisation schemes can be employed to find a local minimum of the equation.

Sometime later, Kruskal defined what is now known as non-metric MDS [16], with the motivation that preservation of exact distances is often very difficult and too restrictive. In the non-metric variant, it is instead important to maintain the *rank* or ordering of distances

i.e. from each object, the object that is first, second, and third closest in the original space should still be first, second, and third closest in the projection respectively. This non-metric stress is defined with a deceptively similar function:

$$\sqrt{\frac{\sum_{i<j} (\|X_i - X_j\| - \hat{d}_{i,j})^2}{\sum_{i<j} \|X_i - X_j\|^2}} \quad (2)$$

However, the matrix \hat{D} is not the original distance matrix from the input. Instead, a Shepard diagram [21] is formed by forming a set of coordinates for each pair of objects i, j : $c_{i,j} = (\|X_i - X_j\|, d_{i,j})$. A Shepard diagram created from an ideal drawing would have all points lying exactly on a straight line, since this would correspond to all input distances being exactly all output distances. To measure the deviation from this line, Kruskal performs a monotonic regression with $\binom{n}{2}$ points to best fit the diagram. The matrix \hat{D} is defined such that $\hat{d}_{i,j}$ is the distance in the x -coordinate to the fitted line from $c_{i,j}$.

In terms of information, the metric stress of equation 1 is more restrictive than the non-metric stress of equation 2. However, both are measuring how well distances are maintained in the output. Non-metric stress has the additional advantage that the resulting number will be in the range 0-1 and is hence more suitable as an evaluation metric. Scale can impact normalised stress values and several scale-invariant stress measures are detailed in recent papers by Ahmed et al. [1], Smelser et al. [23], and Wang et al. [27].

3 Methodology

3.1 Experimental Methodology

Our overriding Research Question is “Can people see stress in a graph drawing?” As a simple yes/no question, this is tricky to address directly: it does not make sense to simply ask a participant to look at a graph drawing and to state whether it “is” or “is not” stressed. We therefore address this question by investigating whether people can see the difference in graph drawings with different stress values; thus, the research question becomes “Can people see differences in stress between two drawings of the same graph?”

Our methodology was mostly exploratory and incremental. We first addressed the research question by explaining the concept of stress to novices, giving them some training on the task (with feedback), and collecting performance data (accuracy, time, confidence) on 45 trials – over three different graph sizes. Each trial consisted of a pair of drawings of the same graph, with the participants being asked to indicate which has lower stress.

The results of this initial experiment were very encouraging, and so our follow-up study attempted to see how the extent of training affects performance; in this case, the participants received the same explanation of stress, but did not receive feedback on the training questions. We also asked some of our graph drawing expert collaborators to do the experiment as well; they did not have any training session.

Section 3.4 explains the experimental procedure in more detail.

3.2 Measuring Stress

To test the perception of stress, we need a set of graph drawings which have different values of stress. To create these, we must define a measurement of stress which is independent of graph size, structure, and drawing space. While the general definition (i.e. the metric definition of equation 1 and its commonly used variants), can be used to calculate the stress of a given drawing, it does not meet this criteria. For example, using this definition, Welch

and Kobourov show that the stress of a given drawing varies with respect to the scale of the drawing space [28]. This means graph drawings with different geometric scales cannot be compared.

To create the stimuli, we use the definition of stress provided by Kruskal [16] (equation 2). While intended for measuring the quality of multi-dimensional scaling techniques, the method is applicable to measuring stress in graph drawings and has numerous benefits. Firstly, the measure is normalised, and hence applicable to graphs of different size and structure. This definition also has the benefit of being independent of the rotation and geometric scale of the drawing space. To integrate this measure with other graph drawing metrics, such as those described by Mooney et al. [18], we subtract the Kruskal stress value from 1. This means that, as a metric, a value of 1 represents the extreme that is intuitively assumed to be good – in this case, zero stress. For the rest of this section, we refer to this definition as the “Kruskal Stress Metric” (KSM).

Hopefully, Section 2.2 leaves one convinced that Kruskal’s definition is an appropriate measure of stress. To further justify this choice, we used the large dataset described by Mooney et al. [18], and calculated the Kruskal stress of nearly half a million graph drawings, then compared this to a normalised stress function based on the Kamada and Kawai definition [14]. We found the correlation between the two definitions to be 0.871. This strong correlation highlights the similarity in the respective definitions.

3.3 The stimuli

We ran three concurrent experiments, for graphs with node count of 10, 25 and 50. For each graph size, we created five random graphs using the Erdős–Rényi model [8], with the constraint that the graph is connected and the number of edges is less than two times the number of nodes. This constraint ensures that drawings do not end up as “hairballs”. Preliminary testing when creating the stimuli showed that denser graphs (and hence denser drawings) tend to have higher stress than sparser graphs, and so, inclusion of them would limit the range of possible stress values.

We create graph drawings with KSM values in the range of 0.4 to 0.8, with intervals of 0.05 (nine drawings in a set). For each graph size, we generate three sets of drawings using a basic hill climbing algorithm. The evaluation function for this algorithm is the absolute difference between the KSM of the current drawing and the target KSM value. Nodes are initially positioned randomly within a unit grid. Each iteration updates the drawing by selecting a random node, and moving this node to a new random position within the initial bounding box of the drawing space. This new random position is also bounded by a circle of decreasing radius (as the number of iterations increases) centred on the chosen node’s current position. If the new drawing has KSM closer to the target value, it is taken to be the current drawing, otherwise it is discarded. The algorithm terminates when the KSM of the drawing is within 0.01 units of the target value. Example stimuli are shown in Table 1.

3.4 Experimental Procedure

The methodology for each novice experiment is the same, but with different stimuli (corresponding to the three different graph sizes). The participants are first shown an overview of the aims of the experiment and asked for their consent to participate. Next, participants are shown a page which outlines the key concepts of graph drawings and stress (details available in the online version of the paper). We use layman terms such as “network” instead of “graph”, “object” instead of “node”, and “connection” instead of “edge”. This page explains the following:

21:6 The Perception of Stress in Graph Drawings

■ **Table 1** Example drawings of different KSM values. Each row consists of different drawings of the same graph. Higher KSM values indicate lower stress.

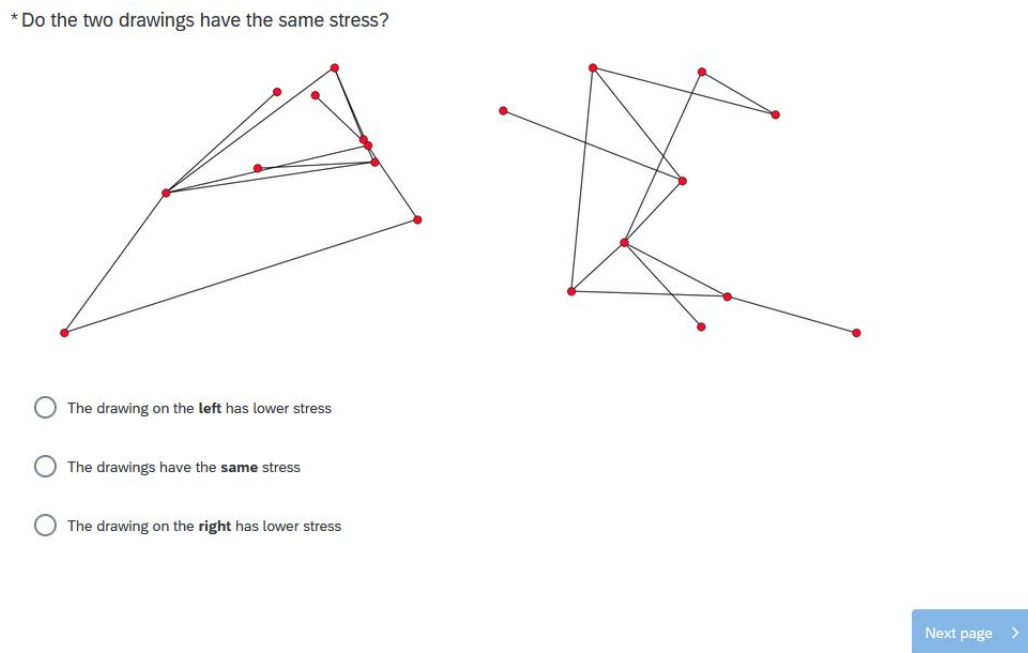
		KSM		
		0.4	0.6	0.8
Size	10			
	25			
	50			

- What a graph is, including definitions for nodes and edges.
- What a graph drawing is, and that one graph can have many different drawings.
- That different graph drawings have different visual properties, such as symmetry.
- What is meant by a “path” and “shortest path” between two nodes.
- What stress means in relation to graph drawing, with very basic examples of low and high stress drawings.
- Four pairs of drawings with low and high stress.
- That the participants should not spend a long time trying to calculate the stress of a drawing, but should rely on their immediate perception.

Participants are then shown nine predetermined pairs of drawings and asked to choose which has lower stress (or that they have the same stress). The nine drawings are of graphs that are the same size as the given experiment with differences of KSM ranging from 0.4 to 0, decrementing by 0.05 (in that order). In the first set of experiments there are 25 participants for each graph size (75 total). These 75 participants are given feedback immediately after each pair to inform them if they were correct or, if not, what the correct response was. Where feedback is given, over 50% of the responses to these training questions must be correct to continue to the main experiment. In the second set of experiments there are 10 participants for each graph size (30 total). These 30 participants are given no feedback on the nine training questions.

In the main part of the experiment, participants are shown 45 pairs of drawings (trials), with KSM values in the range 0.4–0.8, and asked to choose the drawing with lower stress (or that they have the same stress). Each pair consists of two drawings of the same graph. There

are five graphs and nine unique KSM differences (0, 0.05, 0.1...0.4). The exact drawings for each KSM difference is random – e.g., for graph one, and KSM difference of 0.3, one participant may be shown drawings with KSM of 0.4 and 0.7, whilst another may be shown drawings with KSM of 0.5 and 0.8. The order in which pairs of drawings are shown is randomised for each participant, as well as which two drawings are chosen (from a pool of three) and the order (left/right) that they appear on the page. This randomness is incorporated to lower the chance of a learning effect. The participants are not given any feedback about their choice. After each choice, the participant is asked if they were confident about their choice. An example trial is shown in Figure 1.



■ **Figure 1** Example trial for the $n=10$ experiment. Here the drawing on the left has a KSM value of 0.45, while the right drawing has a KSM value of 0.7 (but this information is, of course, not shown to the participants).

After all 45 pairs have been shown, the participant is asked some follow up questions about the experiment and asked for some demographic information. See Section 3.4.1 for more details.

The expert experiment follows the same methodology as the novice experiments, however the experts are not shown the nine training pairs and complete the 45 trials for each graph size (135 total). The experts are shown the 135 trials in blocks of 45, corresponding to each graph size (10, then 25, then 50), with an opportunity for a break between blocks.

3.4.1 Data Collection

In each experiment we collect the following data:

- The number of incorrect training responses (out of nine).
- Which pairs of drawings were shown to participants and their KSM values, as well as the order they were shown in. Within each pair we also keep track of which drawing was displayed on the left and which was displayed on the right.

21:8 The Perception of Stress in Graph Drawings

- The response from the participant about which drawing had lower stress (The drawing on the left has lower stress/The drawings have the same stress/The drawing on the right has lower stress).
- How confident the user was about each choice (Confident/Not confident).
- The time taken to submit an answer for each pair.

After all pairs of drawings are shown we also collect data for some additional questions and demographic information:

- The overall strategy used to determine which drawing had lower stress.
- The participant's overall confidence in their responses (Very confident/Somewhat confident/Not very confident/Not confident at all).
- How difficult the participant found the experiment (Very difficult/Difficult/Easy/Very Easy).
- How familiar the participant is with network diagrams (Very familiar/Somewhat familiar/Not very familiar/They are new to me).
- The participants' age and gender.

3.4.2 Experimental Conduct

The online survey platform “Qualtrics” was used to set up and run the experiments, and collect the data.

Participants were recruited from the online platform “Prolific” and paid at a rate of £9 GBP per hour. The pool of participants was limited to users over the age of 18 and residing in either the United Kingdom or Australia. Participants who were included in one study were excluded from participating in any of the others to reduce the chance of a learning effect.

For each graph size, we collected results from 25 participants who received feedback on the training, and 10 participants who did not. For the participants who received feedback on the training, those who answered more than 50% incorrectly were excluded from the main experiment. For the experiments on graphs of size 10 and 25, only one participant did not pass the training. For the experiment on graphs of size 50, three participants did not pass the training. For each experiment, we recruited more participants until we had 25 who passed the training and completed the whole experiment. Each experiment was first piloted by two participants to ensure the experiments ran smoothly and that the data was being collected correctly. No changes were made after the successful pilots.

The expert participants were invited via email, of whom ten participated in the experiment. Two of these ten responses were incomplete, leaving us with eight complete expert responses.

This research was approved by the Monash University Human Research Ethics Committee with ID 42695.

4 Results

4.1 Trained Novices (TN)

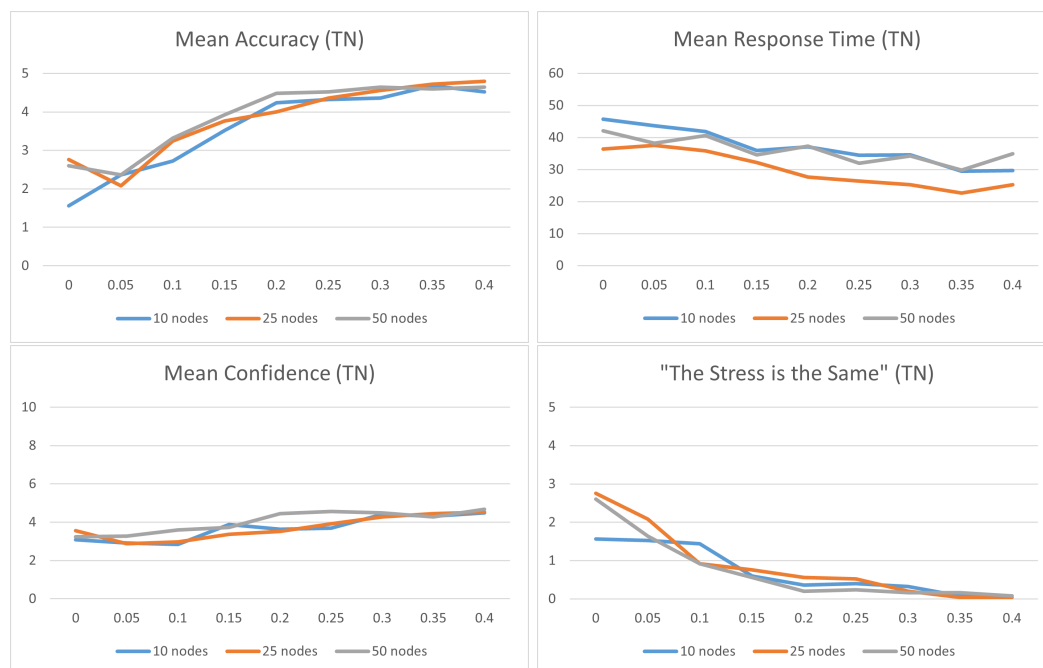
We address our research question (“Can people see the differences in stress between two drawings of the same graph?”) by analysing our data along two dimensions, giving two sub-questions:

- RQa: “How much does the quantifiable difference in stress (the “delta”) between two drawings of the same graph affect the perception of stress?”
- RQb: “How much does the size of the graph affect the perception of stress?”

In all cases, our dependent variables for measuring perception are accuracy, response time, and confidence. Of the 75 participants, there were 38 women, 36 men and 1 gender-diverse. The median age range was 36-45, with 16 participants between 18-25, and one over 76. Four participants said they were “very familiar” with graph drawing; 55 claimed novice status (at most “not very familiar”).

4.1.1 Delta trends

We consider each graph size in turn, plot the average accuracy, response time, and confidence over all 25 participants, and consider trends with respect to delta (Figure 2). There were five trials per delta value, so accuracy is between 0 and 5; high confidence is recorded as 2 for each trial (so confidence per delta is between 0 and 10); time is measured in seconds. There are eight response time data points for which the response time was over 200 seconds (six for $n=50$, two for $n=25$). All other 3,367 response time data points were less than 140 seconds. These eight data points (0.24% of the total data points) were replaced with the mean of the relevant participants’ other responses. Participants were given the option to indicate that the stress in the two drawings was the same. Only five of the forty-five trials were pairs with zero delta. Use of this option for other deltas indicates difficulty in distinguishing stress.



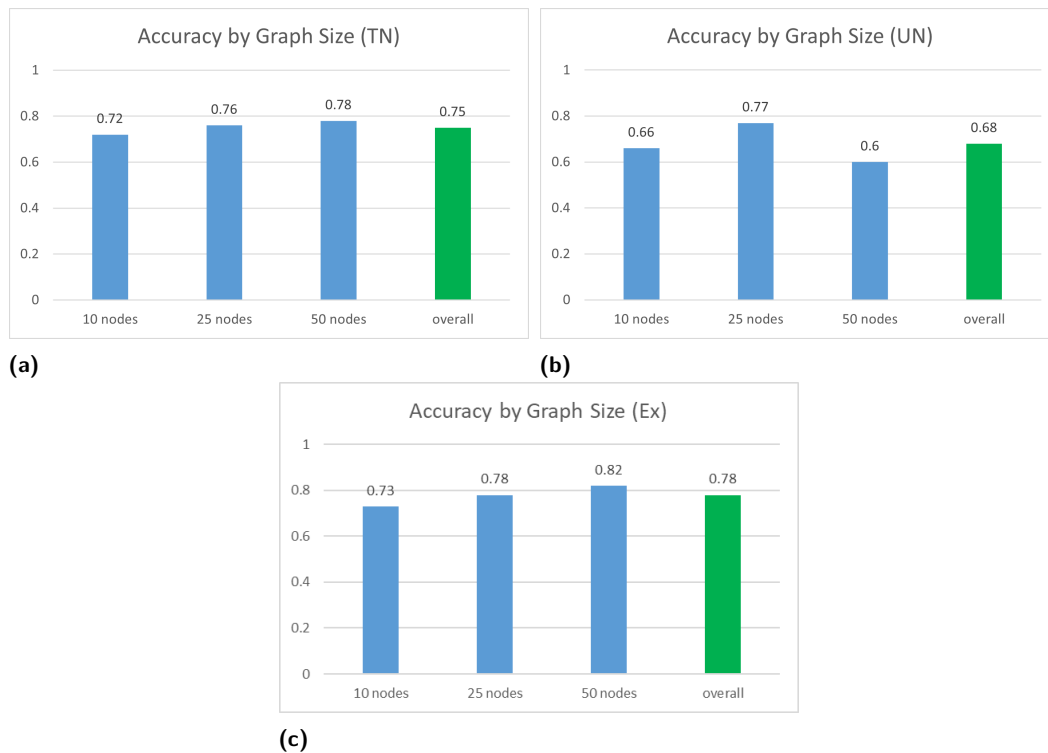
■ **Figure 2** Delta trends for Trained Novices.

4.1.2 Overall accuracy

In this case, we ignore the delta values, and focus on whether the size of the graph makes a difference to the overall mean accuracy of stress perception (Figure 3a). An independent measures t-test between the data for 10 nodes and that for 50 nodes reveals a p-value of 0.0495 – a barely significant difference.

34 of the 75 participants (45%) said they found the task difficult; only three said that it was “very easy”.

21:10 The Perception of Stress in Graph Drawings



■ **Figure 3** Overall accuracy over all trials for each participant type, with respect to graph size.

4.1.3 Discussion: Performance

With an overall accuracy rate of 75.2% (considerably higher than chance: 33% for three-way multiple choice responses), our data shows that novices who have had the notion of stress explained to them, and have had the opportunity to be trained with feedback, can indeed “see” stress. Contrary to our expectations, the size of the graph does not make the task more difficult, and we see expected trends in accuracy, response time and confidence with respect to difficulty (as measured by the delta). While there is some uncertainty with the lower deltas of 0, 0.05 and 0.1, by the time the delta reaches 0.15, participants are getting 80% of the tasks correct, rising as high as 96% accuracy for 25 nodes with $\text{delta}=0.4$. Looking at the average number of times that participants judged the stress to be the same, we can see that any ambiguity caused by deltas of 0.05 and 0.01 diminishes when the delta increases to 0.15. We were surprised by these quantitative results. We had assumed that the complexity of the notion of stress would be difficult for novices to grasp easily, and that it would be more difficult to determine in larger graphs.

4.1.4 Discussion: Strategies and Perception

We had asked the participants to explain their strategy in determining the stress differences. Many participants (24) said that they looked at the distances between the nodes, or the distribution of the nodes (5). Some were more specific, noting that they looked at the length of the edges between the nodes (11). Apart from some references to “messiness/busy-ness” (10), other features highlighted were symmetry (2), edge crossings (5) or edge closeness (4), angles (5) and “clusters” (5). Only one participant made a value judgement on the form of

the drawings: “I choose the one that made me feel less stressed i.e the one that was easiest to understand”. Other notable text summaries include: “The tightness of the angles, the grouping and how close it was, how “spiky” or chaotic the diagrams looked”, “the ones that looked more erratic and all over the place felt more stressful” and “if the lines looked messy, it had more stress.”

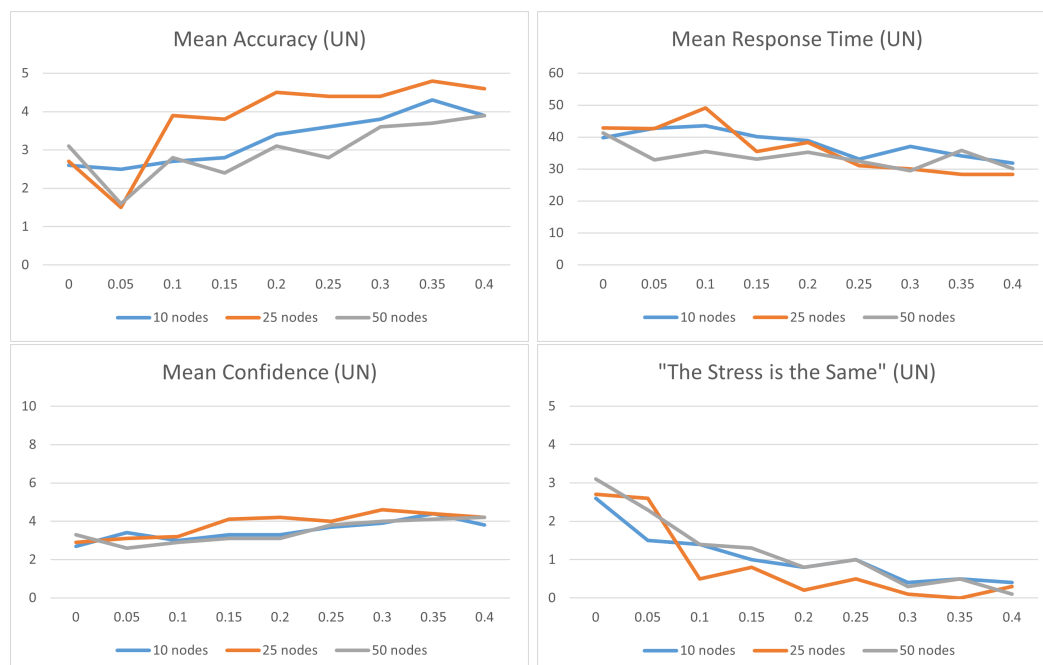
Our trained novices were therefore able to perceive “stress” in a graph drawing, using the perception of edge length, node distribution and visual edge density (i.e. the closeness and compactness of edges, including edge crossings) as proxy measures.

4.2 Untrained Novices (UN)

Our positive results for trained novices clearly depended on the nature and extent of the training offered to them: not only were they given a detailed written explanation of the stress measures (with examples), they had nine opportunities to attempt the task, with feedback given as to the correct answer for each. We conducted the experiment again, for 10 participants for each size graph, and omitting the feedback.

Of the 30 participants, there were 16 women, 13 men and 1 unknown. The median age range was 26-35, with 7 participants between 18-25, and four between 56-65. No-one claimed to be “very familiar” with graph drawing; 24 claimed novice status (at most “not very familiar”).

Figure 4 shows the data for the untrained participants; Figure 3b shows the overall accuracy. An independent measures t-test between the data for 25 nodes and that for 50 nodes reveals a p-value of 0.0561 – an insignificant difference. 23 of the 30 participants said that they found the task difficult.



■ **Figure 4** Delta trends for Untrained Novices.

4.2.1 Discussion

The overall trend pattern for the untrained novices is comparable to that of the trained novices: any ambiguity when delta is 0.05 or 0.1 is resolved by the time it increases to 0.15. It appears that the medium-sized graph ($n=25$) was easier; it is not clear why this might be the case.

The strategies used by these participants were similar to those used by the trained participants: distance between nodes and node distribution (14), “chaotic” (5), and edge lengths (3). There were three references to “open space” (not mentioned by the trained participants): “Anything that looked congested or compact I found stressful to look at where as when it was more spaced with no sudden lines pointing out it looked much more relaxed.” Two participants referred to analogies: “it reminded me of chemical and biological bonds, those with random arrangements where the links/bonds were more likely to break, be fragile, have weaker chain strengths. Those with consistent links/bonds where the lengths and widths are the same were stronger and less likely to separate”; “some looked like animals, which was less stressful to me. if there was a rogue line jutting out of somewhere I found this stressful. when they had more space I found it less stressful.”

The overall accuracy of the untrained novices was 67.6% – still greater than chance (33%) but less than the accuracy for the trained novices (75.2%). An independent-samples t-test between the overall mean accuracy for trained and untrained novices gives a p-value of 0.009; thus, the additional training made a significant difference to the performance of the novice.

4.3 Experts

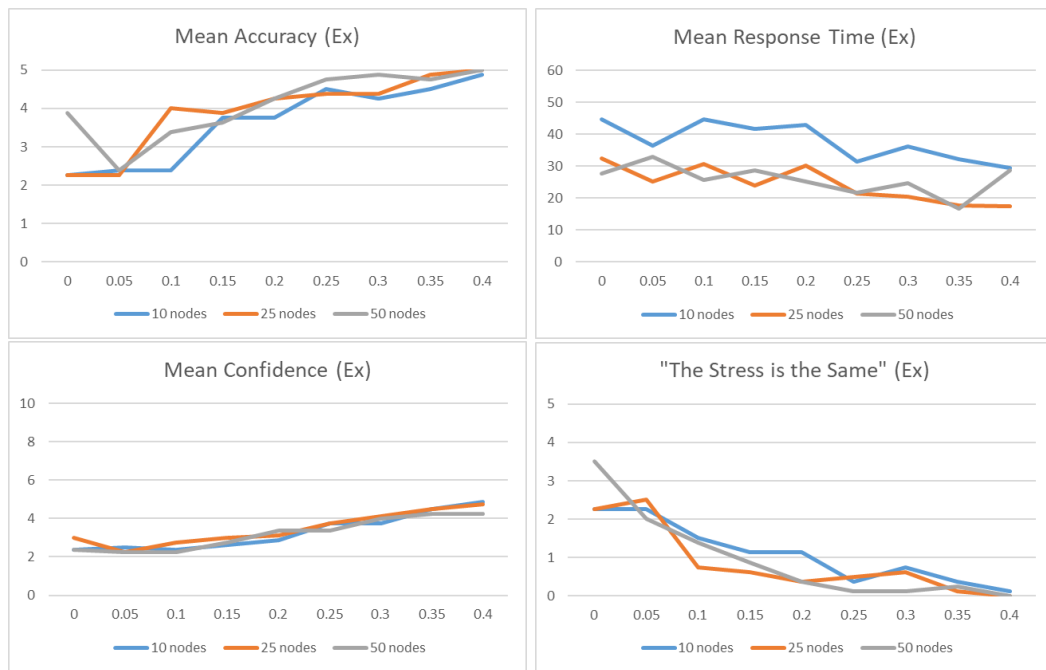
Having gathered data on how well novices can perceive stress after being presented with a rudimentary explanation of the concept, we were interested in how graph drawing experts (who already know the concept well) would fare in the same experiment. Eight graph drawing experts – known colleagues of the authors (7M, 1F, median age 36-45) – completed all the trials for all three graph sizes. The experts were shown the same written explanation of “stress”, but were not given any training.

Figure 5 shows the data for the expert participants; Figure 3c shows the overall accuracy. A repeated measures t-test between the data for 10 nodes and that for 50 nodes reveals a p-value of 0.089 – an insignificant difference.

4.3.1 Discussion

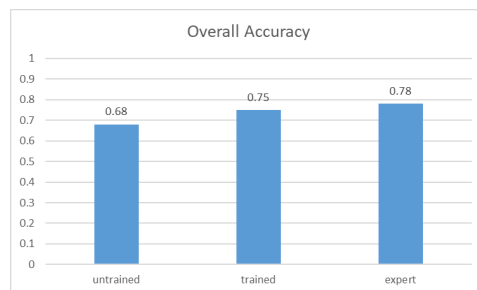
The trend patterns for the experts are remarkably similar to the novices. As with both trained and untrained novices, confusions between the pairs with deltas of 0.1 and less are resolved for those with values of 0.15 and above. The longer response time for the graphs with 10 nodes can be explained by the fact that these were the trials that all experts completed first, before they then looked at the larger graphs (suggesting a “warming-up” period). The experts’ confidence is remarkably low, closely matching that of the novices.

The strategies used by the experts included line length (2), distances between nodes (2), and edge crossings (2). There were, however, some less specific considerations and some clear value judgements: “drawings with Low stress just look “right”... high stress drawings look more random showing features that a human graph drawer would attempt to correct in a second iteration”; “general feeling how much the nodes are settled into place”; “how well the overall structure is highlighted”; “how well untangled they are.” It appears that experts do not (like novice) need to rely so much on visual proxies, but their experience gives them an



■ **Figure 5** Delta trends for Expert participants.

intuitive “feel” for stress. This “feel” is, however, not fail-safe, with the experts having only an overall accuracy of 77.5%. Figure 6 shows the relative accuracy of the three categories of participant.



■ **Figure 6** Overall accuracy over all trials for each participant group.

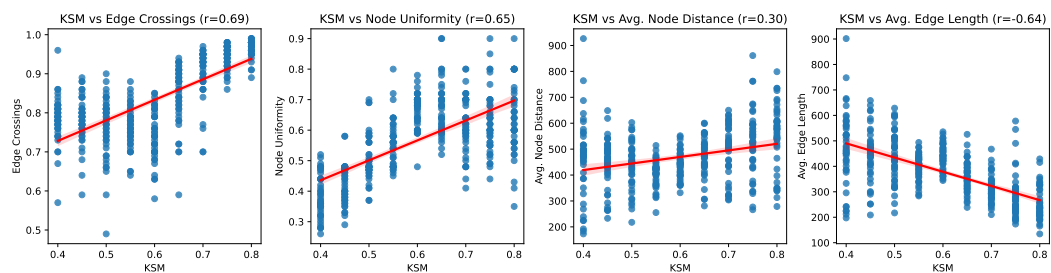
An independent-samples t-test between the overall mean accuracy for untrained and trained novices gives a p-value of 0.009; thus, the additional training made a significant difference to the performance of the novices.

An independent-samples t-test between the overall mean accuracy for trained and expert gives a p-value of 0.361; thus, there is no significant difference in accuracy between trained novices and experts. There is also no significant difference between mean response time for trained novices (7.89s) and experts (5.86s); p=0.065.

We speculate that trained novices thought more carefully about their choices, using visual proxies they had identified during the training sessions, while the experts made rapid “gut-feel” decisions. Despite this, all participants reported low confidence in their responses – even the experts.

4.4 Correlations Between Stress and Other Metrics

We calculated the correlations between KSM and four other visual properties present in the “strategy” responses: edge crossings, distribution of nodes in the drawing space, average distance between nodes, and average edge length (Figure 7). For the distribution of nodes and edge crossings, we use the Node Uniformity and Edge Crossing metrics defined by Mooney et al. [18] (details available in the online version of the paper). We compute these metrics on our set of 405 stimuli drawings.



■ **Figure 7** Correlations between KSM and other graph drawing metrics.

We find that KSM is positively correlated with the average distance between nodes (0.30), the distribution of nodes (0.65), and the number of edge crossings (0.69). KSM is negatively correlated with average edge length (-0.64). Recalling that KSM values closer to 1.0 correspond to lower stress, we can make the following observations:

- Drawings with fewer edge crossings have lower stress.
- Drawings in which the nodes are more evenly distributed around the drawing space have lower stress.
- Drawings with nodes which are farther apart on average have lower stress.
- Drawings which have shorter edge lengths on average have lower stress.

These observations suggest that some of the participants’ “strategies” are effective visual proxies for stress, though do not fully capture it.

5 Conclusion

These experiments represent the first investigation of whether people can “see” stress, an invisible property of graph drawings that is defined by both the geometric form of the graph drawing as well the underlying structure of the graph. We find that it is indeed possible to describe the notion of stress to people unfamiliar with graphs and graph drawing in manner that allows them to perceive it. Unprompted, they are able to devise their own visible proxies for the invisible stress feature – that is, they easily identify appropriate visual features of drawings which can be used to assess stress: in particular the geometric length of edges, distances between nodes and node distribution, and “compactness”/ “clustering”/ “density”. We can’t be sure that the participants utilised the exact geometric definition of stress but their performance suggests that they were able to perceive its visual features. A few participants referred to the personal stress that they felt while looking at the drawings, thus transferring a mathematical concept into an emotion.

Surprisingly, experts do not perform much better than trained novices. While there is some evidence of them also using visual proxies, they also appear to rely on high-level concepts like “untangling”, “general feeling” and “settled nodes” – more than is evident in the strategies described by the novices.

Our findings do not differ with respect to graph size. We were surprised at this, but this could be explained by the fact that the participants focused on these overall visual proxies rather than the edges themselves. Our results therefore appear to be generalisable to graph drawings of different sizes.

This work is important for any human experimental work on the perception and use of graph drawings where stress is deemed an important feature. Not only have we shown that people can understand the concept of stress sufficiently well to be able to perceive it, we have also identified those visual features that they consider to be most related to stress: edge length, node distribution and concentrated areas of edges and edge crossings. Future experimental work can now build on these results to investigate, for example, nuances of different implementations of stress, trade-offs between stress and other layout principles, the smallest delta for stress perception, and the form of user-generated drawings where low stress is an explicitly stated goal.

5.1 Future Work

All empirical studies are limited by necessary practical parameters. In this case, the stimuli are small and sparse graphs. A follow up experiment on larger and denser graphs would inform on how the perception of stress varies as graph sizes increase. While our results show only a small difference in terms of perception between graphs of size 10 and 50, this may not be the case in general as the size of the graph increases. However, as the size and density of the graph increases, producing drawings with low stress becomes increasingly challenging.

Our results suggest that stress deltas of 0.1 and below are confusing, but that stress differences of 0.15 and above are more discernible. These observations are also limited by the range of KSM values used in the experiment. It may be the case that the discernible difference in KSM is non-uniform; i.e., the difference between drawings with KSM of 0 and 0.15 may be more or less discernible than that of 0.85 and 1.0. A more comprehensive “Just Noticeable Difference” methodology (such as that used by Soni et al. to measure the perception of graph density [24]) could validate this, and perhaps result in a more specific threshold. This methodology would allow us to see what the smallest perceptible difference in stress is, and see if this follows Weber’s law. This requires careful creation of a broader set of stimuli as well as a larger number of participants.

Investigation into additional “non-visual” metrics, which are not associated with the principles of visual perception, such as Neighbourhood Preservation, would provide insights into their perceptibility. This may also inform on the usefulness of such metrics as criteria for optimisation and/or evaluation of graph drawings.

References

- 1 Reyan Ahmed, Cesim Erten, Stephen Kobourov, Jonah Lotz, Jacob Miller, and Hamlet Taraz. Size should not matter: Scale-invariant stress metrics. *arXiv preprint arXiv:2408.04688*, 2024. doi:10.48550/arXiv.2408.04688.
- 2 Reyan Ahmed, Felice De Luca, Sabin Devkota, Stephen Kobourov, and Mingwei Li. Graph drawing via gradient descent, $(GD)^2$. In *Graph Drawing and Network Visualization*, volume 12590 of *Lecture Notes in Computer Science*, pages 3–17. Springer International Publishing, 2020. doi:10.1007/978-3-030-68766-3_1.
- 3 A. Arleo, S. Miksch, and D. Archambault. Event-based dynamic graph drawing without the agonizing pain. *Computer Graphics Forum*, 41(6):226–244, 2022. doi:10.1111/cgf.14615.
- 4 Ulrik Brandes and Christian Pich. An experimental study on distance-based graph drawing. In Ioannis G. Tollis and Maurizio Patrignani, editors, *Graph Drawing*, volume 5417 of *Lecture Notes in Computer Science*, pages 218–229, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. doi:10.1007/978-3-642-00219-9_21.

21:16 The Perception of Stress in Graph Drawings

- 5 Markus Chimani, Patrick Eades, Peter Eades, Seok-Hee Hong, Weidong Huang, Karsten Klein, Michael Marner, Ross T Smith, and Bruce H Thomas. People prefer less stress and fewer crossings. In *Proceedings of International Symposium on Graph Drawing (GD)*, pages 523–524. Springer, 2014.
- 6 Sabin Devkota, Reyhan Ahmed, Felice De Luca, Katherine E. Isaacs, and Stephen Kobourov. Stress-plus-x (SPX) graph layout. In *Graph Drawing and Network Visualization*, volume 11904 of *Lecture Notes in Computer Science*, pages 291–304. Springer International Publishing, 2019. doi:10.1007/978-3-030-35802-0_23.
- 7 Sara Di Bartolomeo, Tarik Crnovrsanin, David Saffo, Eduardo Puerta, Connor Wilson, and Cody Dunne. Evaluating graph layout algorithms: A systematic review of methods and best practices. In *Computer Graphics Forum*, page e15073. Wiley Online Library, 2024.
- 8 P Erdős and A Rényi. On random graphs I. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.
- 9 Thomas M. J. Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, November 1991. doi:10.1002/spe.4380211102.
- 10 E. R. Gansner, Yifan Hu, and S. North. A maxent-stress model for graph layout. *IEEE Transactions on Visualization and Computer Graphics*, 19(6):927–940, June 2013. doi:10.1109/tvcg.2012.299.
- 11 Emden R. Gansner, Yehuda Koren, and Stephen North. Graph drawing by stress majorization. In *Graph Drawing*, volume 3383 of *Lecture Notes in Computer Science*, pages 239–250. Springer Berlin Heidelberg, 2005. doi:10.1007/978-3-540-31843-9_25.
- 12 Seok-Hee Hong, Peter Eades, Marnijati Torkel, Ziyang Wang, David Chae, Sungpack Hong, Daniel Langerenken, and Hassan Chafi. Multi-level graph drawing using infomap clustering. In *Graph Drawing and Network Visualization*, volume 11904 of *Lecture Notes in Computer Science*, pages 139–146. Springer International Publishing, 2019. doi:10.1007/978-3-030-35802-0_11.
- 13 Weidong Huang, Peter Eades, Seok-Hee Hong, and Chun-Cheng Lin. Improving multiple aesthetics produces better graph drawings. *Journal of Visual Languages & Computing*, 24(4):262–272, August 2013. doi:10.1016/j.jv1c.2011.12.002.
- 14 Tomihisa Kamada and Satoru Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15, 1989. doi:10.1016/0020-0190(89)90102-6.
- 15 J. F. Kruiger, P. E. Rauber, R. M. Martins, A. Kerren, S. Kobourov, and A. C. Telea. Graph layouts by t-sne. *Computer Graphics Forum*, 36(3):283–294, 2017. doi:10.1111/cgf.13187.
- 16 Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.
- 17 Michael R. Marner, Ross T. Smith, Bruce H. Thomas, Karsten Klein, Peter Eades, and Seok-Hee Hong. GION: Interactively untangling large graphs on wall-sized displays. In *Graph Drawing*, volume 8871 of *Lecture Notes in Computer Science*, pages 113–124. Springer International Publishing, 2014. doi:10.1007/978-3-662-45803-7_10.
- 18 Gavin J. Mooney, Helen C. Purchase, Michael Wybrow, and Stephen G. Kobourov. The multi-dimensional landscape of graph drawing metrics. In *17th IEEE Pacific Visualization Symposium (PACIFICVIS)*, pages 122–131. IEEE, IEEE, 2024. doi:10.1109/PACIFICVIS60374.2024.00022.
- 19 Helen Purchase. Which aesthetic has the greatest effect on human understanding? In Giuseppe DiBattista, editor, *Graph Drawing*, volume 1353 of *Lecture Notes in Computer Science*, pages 248–261. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997. doi:10.1007/3-540-63938-1_67.
- 20 Helen C. Purchase, C. Pilcher, and B. Plimmer. Graph Drawing Aesthetics—Created by Users, Not Algorithms. *IEEE Transactions on Visualization and Computer Graphics*, 18(1):81–92, January 2012. doi:10.1109/TVCG.2010.269.

- 21 Roger N Shepard. The analysis of proximities: multidimensional scaling with an unknown distance function. i. *Psychometrika*, 27(2):125–140, 1962.
- 22 Paolo Simonetto, Daniel Archambault, and Stephen Kobourov. Drawing dynamic graphs without timeslices. In *Graph Drawing and Network Visualization*, volume 10692 of *Lecture Notes in Computer Science*, pages 394–409. Springer International Publishing, 2018. doi:10.1007/978-3-319-73915-1_31.
- 23 Kiran Smelser, Jacob Miller, and Stephen Kobourov. "normalized stress" is not normalized: How to interpret stress correctly. *arXiv preprint arXiv:2408.07724*, 2024.
- 24 Utkarsh Soni, Yafeng Lu, Brett Hansen, Helen C. Purchase, Stephen Kobourov, and Ross Maciejewski. The perception of graph properties in graph layouts. *Computer Graphics Forum*, 37(3):169–181, 2018. doi:10.1111/cgf.13410.
- 25 Warren S Torgerson. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17(4):401–419, 1952. doi:10.1007/BF02288916.
- 26 Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008.
- 27 Xiaoqi Wang, Kevin Yen, Yifan Hu, and Han-Wei Shen. SmartGD: A GAN-Based graph drawing framework for diverse aesthetic goals. *IEEE Transactions on Visualization and Computer Graphics*, 2023.
- 28 E. Welch and S. Kobourov. Measuring Symmetry in Drawings of Graphs. *Computer Graphics Forum*, 36(3):341–351, June 2017. doi:10.1111/cgf.13192.
- 29 Mingliang Xue, Zhi Wang, Fahai Zhong, Yong Wang, Mingliang Xu, Oliver Deussen, and Yunhai Wang. Taurus: Towards a unified force representation and universal solver for graph layout. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):886–895, 2023. doi:10.1109/TVCG.2022.3209371.
- 30 Jonathan X. Zheng, Samraat Pawar, and Dan F. M. Goodman. Graph drawing by stochastic gradient descent. *IEEE Transactions on Visualization and Computer Graphics*, 25(9):2738–2748, 2019. doi:10.1109/TVCG.2018.2859997.
- 31 Fahai Zhong, Mingliang Xue, Jian Zhang, Fan Zhang, Rui Ban, Oliver Deussen, and Yunhai Wang. Force-directed graph layouts revisited: A new force based on the t-distribution. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–14, 2023. doi:10.1109/TVCG.2023.3238821.
- 32 Minfeng Zhu, Wei Chen, Yuanzhe Hu, Yuxuan Hou, Liangjun Liu, and Kaiyuan Zhang. Drgraph: An efficient graph layout algorithm for large-scale graphs by dimensionality reduction. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1666–1676, 2021. doi:10.1109/TVCG.2020.3030447.

Boundary Labeling in a Circular Orbit

Annika Bonerath ✉ 

University of Bonn, Germany

Martin Nöllenburg ✉ 

Algorithms and Complexity Group, TU Wien, Austria

Soeren Terziadis ✉ 

TU Eindhoven, The Netherlands

Markus Wallinger ✉ 

Chair for Efficient Algorithms, TU Munich, Germany

Jules Wulms ✉ 

TU Eindhoven, The Netherlands

Abstract

Boundary labeling is a well-known method for displaying short textual labels for a set of point features in a figure alongside the boundary of that figure. Labels and their corresponding points are connected via crossing-free leaders. We propose *orbital boundary labeling* as a new variant of the problem, in which (i) the figure is enclosed by a circular contour and (ii) the labels are placed as disjoint circular arcs in an annulus-shaped orbit around the contour. The algorithmic objective is to compute an orbital boundary labeling with the minimum total leader length. We identify several parameters that define the corresponding problem space: two leader types (straight or orbital-radial), label size and order, presence of candidate label positions, and constraints on where a leader attaches to its label. Our results provide polynomial-time algorithms for many variants and NP-hardness for others, using a variety of geometric and combinatorial insights.

2012 ACM Subject Classification Human-centered computing → Visualization; Theory of computation → Computational geometry

Keywords and phrases External labeling, Orthoradial drawing, NP-hardness, Polynomial algorithms

Digital Object Identifier 10.4230/LIPIcs.GD.2024.22

Related Version A preliminary abstract of this paper has been presented at EuroCG 2024.

Full Version: <https://arxiv.org/abs/2403.19052> [8]

Funding *Annika Bonerath*: German Research Foundation under Germany’s Excellence Strategy, EXC-2070 – 390732324 – PhenoRob.

Martin Nöllenburg: Vienna Science and Technology Fund (WWTF) [10.47379/ICT19035].

Soeren Terziadis: Vienna Science and Technology Fund (WWTF) [10.47379/ICT19035] and the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No 101034253.

Markus Wallinger: Vienna Science and Technology Fund (WWTF) [10.47379/ICT19035].

1 Introduction

Labeling spatial data on a map is a well-studied topic in computational geometry [1, 10, 17]. Commonly the feature points are annotated with labels that display the names or additional descriptions, ensuring non-overlapping labels to guarantee full readability. The labels are placed either next to the feature points [16] (*internal* label positions), or remotely along the contour of a bounding shape such that the feature points are connected to their labels by crossing-free leaders (*external* labeling models) [5]. Often, for high feature-point densities, the external labeling model is more advantageous since the background map is not obscured by



© Annika Bonerath, Martin Nöllenburg, Soeren Terziadis, Markus Wallinger, and Jules Wulms; licensed under Creative Commons License CC-BY 4.0

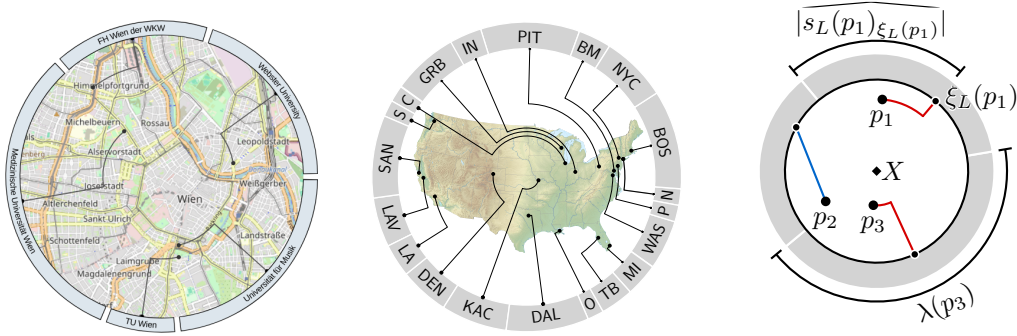
32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 22; pp. 22:1–22:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** An orbital labeling highlighting points of interest in a map section (left) and an orbital labeling of NFL-Championship winning teams, where the label sizes are scaled with the number of won titles (center). Our notation (right) with an SL-leader in blue and OR-leaders in red.

annotations. A special case of external labeling is boundary labeling [3,4] where the labels are attached to the (mostly rectangular) boundary of the map. The interest in visualizing data on round displays, e.g., on smartwatch faces (see Figure 1) or on round displays in cockpits, is growing, as discussed by Islam et al. [14] in their recent survey; but, from a visualization perspective, it is still an under-explored topic compared to traditional rectangular displays. The design space description of Islam et al. [14] includes geospatial data representations as well as placement of text labels and icons on a round display. With this in mind, we initiate the investigation of boundary labeling for maps with circular boundaries, surrounded by a peripheral fixed-width ring, in which the labels are placed. We call these labels *orbital*. We assume that the lengths of the orbital labels are normalized, s.t., they sum up to at most the perimeter of the boundary of the map. Orbital labels can also be used for other purposes, e.g., to display donut charts representing statistical data values of different feature points within the map, such that the label sizes are proportional to the data values (Figure 1). Previous research on circular map displays considered either multirow circular labels where the sum of label lengths does not equate to the map’s boundary length [12], radial labels [2,9], or horizontal labels [9,13,15]. The latter two settings are relevant on rectangular displays but not suitable for circular displays with a narrow annulus reserved for labels. Furthermore, these settings differ in their geometric properties and hence their labeling algorithms do not immediately generalize to orbital labels.

Formally, we assume that we are given a disk D in the plane \mathbb{R}^2 centered at a point X . The disk contains n points $P = \{p_1, \dots, p_n\}$. We call the set P of points *features* and we refer to the boundary of the disk as the *boundary* B . The feature closest to X is denoted by p_{\min} . We may assume that D has a radius of 1. Throughout this paper, unless otherwise specified, the *angle between* two points p, q refers to the smallest angle with the center X of D , i.e., $\min(\angle pXq, \angle qXp)$. Every feature $p \in P$ has an associated *label* representing additional information to be placed along a circular arc on the boundary starting at a point $b_1 \in B$ and ending at a point $b_2 \in B$. The circular arc along B is denoted as $\widehat{b_1 b_2}$. Usually, the start and endpoint of the label are not fixed in the input, however, the length of the arc is part of the input. We represent the associated label simply as a number $\lambda(p)$, which indicates the length of the associated label. We assume that $\sum_{i=1}^n \lambda(p_i)$ is equal to the circumference of D , i.e., all labels can be placed in a non-overlapping way without gaps between the arcs on B .

In a *labeling* L , every feature $p \in P$ is assigned a label with starting point $s_L(p) \in B$ and an endpoint $e_L(p) \in B$, s.t., $|\widehat{s_L(p) e_L(p)}| = \lambda(p)$. We require that all labels in L are pairwise non-overlapping. Additionally, every feature p is connected to its label via a curve called a *leader*. We denote the length of a leader ψ using $|\psi|$. We call the point on a

label arc where the leader connects to B the port $\xi_L(p)$. We represent a port by the *port ratio* $\rho(p) = \frac{|s_L(p)\xi_L(p)|}{\lambda(p)}$ which is the ratio of the arc from the starting point $s_L(p)$ to the port $\xi_L(p)$ and the arc from the start-point to the end-point. $\varepsilon \leq |s_L(p)\xi_L(p)|$ and $\varepsilon \leq |\xi_L(p)e_L(p)|$, i.e., there is always at least a small distance between the ends and the port of the label. A distance of ε between two points on B should be interpreted as a distance along B . Now, we define the generic orbital labeling problem.

► **Problem 1** (ORBITAL BOUNDARY LABELING). *Given a disk D , containing n feature points P with their labels $\lambda(p)$ for $p \in P$, compute a labeling L , in which all leaders are pairwise interior-disjoint and where the sum of leader lengths is minimal.*¹

A labeling in which the sum over all leader lengths is minimum is also called a *leader length minimal labeling*. We consider two leader types in this paper. A *straight-line leader* or *SL-leader* is simply a straight-line segment starting at p and ending at $\xi_L(p)$. Its length is the Euclidean distance between p and $\xi_L(p)$. An *orbital-radial leader* or *OR-leader* consists of two parts: a (possibly empty) orbital circular arc with center point X starting at the feature p and ending at a bend point q , and a radial segment that connects q to $\xi_L(p)$; see Figure 1. We call the line through X and $\xi_L(p)$ the *supporting line* of the radial part. Note that for any pair of feature and port, there are exactly two possible OR-leaders. We call an OR-leader leaving its feature in clockwise direction a *clockwise leader* and analogously define *counter-clockwise leaders*. We will also refer to the OR-leader whose orbital part spans an angle larger than π as the feature's *long* and to the other one as its *short* leader. For a feature p_i let r_i be the radius of the circle concentric with D containing p_i . The length of the OR-leader can be expressed as a function $g : P \times [0, 2\pi] \rightarrow \mathbb{R}$ with $g(p_i, \theta) = r_i\theta + (1 - r_i)$, where $0 \leq \theta < 2\pi$ describes the angle spanned by the orbital part of a leader connected to p_i .

► **Observation 2.** *For a fixed feature p , the function $g(p, \theta)$ is continuous and linear in θ .*

Based on this problem description, we delineate the space of the possible problem variants and a suitable naming scheme for such variants in the following section.

2 Problem Space

In this paper, we refer to variants of ORBITAL BOUNDARY LABELING based on the six-dimensional *T-COSA-ORBITAL BOUNDARY LABELING* scheme introduced in this section. The first dimension, denoted by T , determines the leader type, whereas the other dimensions, denoted by variables *COSA*, characterize properties of the labels (A encodes two dimensions). We use OR and SL as substitutes for T in the *T-COSA* scheme. Without the T - prefix, we refer to both OR-*COSA* and SL-*COSA* (leader types are not mixed in a labeling). We mostly focus on OR-leaders, while still discussing which of the results extend to SL-leaders.

For the five dimensions regarding the labels, we use each letter of *COSA* to describe the variants for the respective dimension.

- **[C] Candidate port positions on the boundary.** If we are given a set C of candidate positions on B , we require in any valid labeling L that the set Ξ_L of all ports in L is a subset of C , we say *the port candidates are locked* (and use the symbol C^{\blacksquare}) otherwise they are *free* (C^{\blacklozenge}). For variants with C^{\blacksquare} we assume that sufficiently many, but no more than linearly many candidates are specified ($n \leq |C|$). Otherwise, C^{\blacklozenge} is the more reasonable choice.

¹ This definition does allow a leader to contain another feature, i.e., the endpoint of another leader.

■ **Table 1** An overview of the problem space and our results. Only **locked** port ratios ($A^{\hat{a}}$) are shown. The abbreviation “w. NP-h” denotes weakly NP-hard. Red cells represent polytime, due to the reduction to the algorithm of Benkert et al. [7]. Light red cells are covered by the reduction but are superseded by faster dedicated approaches. Blue cells with a question mark are conjectures. Cells indicate the section containing the relevant result; SL-leader results are in the full version [8].

				$C^{\hat{a}}$		$C^{\hat{r}}$	
				$A_{\equiv}^{\hat{a}}$	$A_{\neq}^{\hat{a}}$	$A_{\equiv}^{\hat{r}}$	$A_{\neq}^{\hat{r}}$
OR	$O^{\hat{a}}$	S_{\equiv}	$O(C n^2)$ [S. 4.1]	$O(C n^2)$ [S. 4.1]	$O(n^2)$ [S. 4.3]	$O(n^2)$ [S. 4.3]	
		S_{\neq}	$O(C n^2)$ [S. 4.1]	$O(C n^2)$ [S. 4.1]	$O(n^2)$ [S. 4.3]	$O(n^2)$ [S. 4.3]	
	$O^{\hat{r}}$	S_{\equiv}	$O(C ^2n)$ [S. 4.2]	$O(C n^3)$ [S. 3]	$O(n^5)$ [S. 3]	$O(n^5)$ [S. 3]	
		S_{\neq}	$O(C ^4)$ [S. 3]	$O(C ^4)$ [S. 3]	w. NP-h [S. 5]	w. NP-h [S. 5]	
SL	$O^{\hat{a}}$	S_{\equiv}	$O(C n^2)$ [S. B.1]	$O(C n^2)$ [S. B.1]	$O(n^2)$? [S. B.3]	$O(n^2)$? [S. B.3]	
		S_{\neq}	$O(C n^2)$ [S. B.1]	$O(C n^2)$ [S. B.1]	$O(n^2)$? [S. B.3]	$O(n^2)$? [S. B.3]	
	$O^{\hat{r}}$	S_{\equiv}	$O(C ^2n)$ [S. B.2]				
		S_{\neq}			w. NP-h [S. B.4]	w. NP-h [S. B.4]	

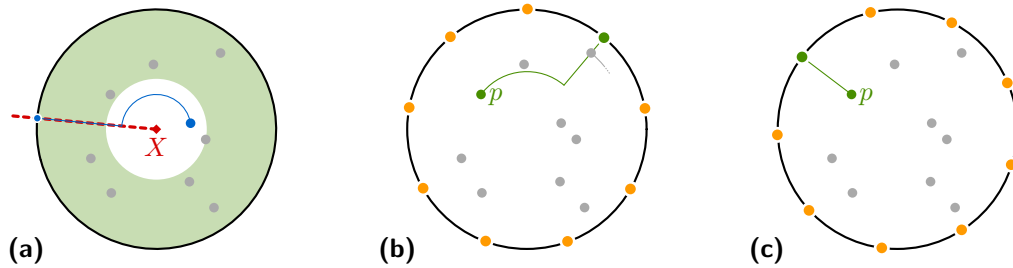
- **[O] Order.** Consider the cyclic order of labels around B . If a certain label order is pre-specified, we say *the label order is locked* ($O^{\hat{a}}$); otherwise, for the unconstrained setting, we say *the label order is free* ($O^{\hat{r}}$).
- **[S] Size of labels.** We distinguish the setting where $\lambda(p) = \frac{2\pi}{n}$ for all $p \in P$, in which case we say that *the label size is uniform* (S_{\equiv}), otherwise *the label size is non-uniform* (S_{\neq}).
- **[A] Port position on labels.** We differentiate between *uniform port ratios*, where $\rho(p) = \rho(q)$ for all $p, q \in P$ (A_{\equiv}), and *non-uniform port ratios* (A_{\neq}). Additionally the port ratios can be *fixed* as part of the input ($A^{\hat{a}}$) or can be *free* to be chosen ($A^{\hat{r}}$).

To specify a T -COSA variant we substitute C , O , S and A with $C^{\hat{r}}/C^{\hat{a}}$, $O^{\hat{r}}/O^{\hat{a}}$, S_{\equiv}/S_{\neq} and $A_{\equiv}^{\hat{r}}/A_{\neq}^{\hat{r}}/A_{\equiv}^{\hat{a}}/A_{\neq}^{\hat{a}}$, respectively. Whenever a statement applies to all variants along a COSA dimension, we drop the sub- or superscript. For example, $C^{\hat{r}}O^{\hat{a}}SA_{\equiv}^{\hat{a}}$ refers to the variants where the leader style is either OR or SL, the port candidates are free ($C^{\hat{r}}$), the order is locked ($O^{\hat{a}}$), the label sizes could be fixed to be uniform or they could be non-uniform (S) and all port ratios are fixed ($A^{\hat{a}}$) to the same (A_{\equiv}) given value.

Contributions. The remainder of the paper is structured as follows (see also Table 1). Our focus is OR -COSA $^{\hat{a}}$ and we show how results for these variants extend to SL -COSA $^{\hat{a}}$ and eventually to all COSA variants. Section 3 presents a reduction from OR -COSA to a problem called BOUNDARY LABELING [7], for which polynomial time algorithms are known. This approach is applicable to a number of OR -COSA variants. In Section 4 we introduce dedicated approaches for some variants, which improve the runtime of the reduction approach. In Section 5 we prove that OR - $C^{\hat{r}}O^{\hat{r}}S_{\neq}A^{\hat{a}}$ variants are weakly NP-hard. Section 6 outlines the extensions to straight-line leaders and free port ratios (details in Appendices B and C of the full version [8], respectively), while Section 7 provides some concluding remarks.

3 Reduction to Boundary Labeling

We begin by investigating the relation between ORBITAL BOUNDARY LABELING and an already established related problem called BOUNDARY LABELING. In an instance of BOUNDARY LABELING, we are given a set of points which are entirely left of a vertical line ℓ . Additionally, we have a set of n disjoint rectangular labels, which are placed to the right of ℓ . The goal is



■ **Figure 2** All leaders of gray points in (a) are entirely contained within the green annulus. The radial part of the blue leader spans the entire intersection between its supporting line and the green annulus. In the labeling shown in (b), a clockwise rotation of the ports is not possible without introducing a crossing between the green leader and the one in gray (indicated with a dashed line). In (c) any rotation clockwise or counterclockwise increases the length of the green leader. Both (b) and (c) show the implied ports in yellow.

to find a set of parallel-orthogonal (po) leaders, which consist of a line segment parallel to ℓ starting at a feature point and a second line segment orthogonal to ℓ connecting to a port on a label. If the label sizes are interchangeable, i.e., every point can be connected to any label, there is an $O(n^3)$ dynamic program [6, 7], which minimizes an arbitrary badness function for a leader (under the assumption that the function can be computed in linear time). The dynamic program, which first splits the instance into horizontal strips, defined by points and the boundaries of the rectangles, then assigns a specific horizontal strip to the feature farthest from ℓ and then recurses on the two sub-instances defined by this assignment. One instance contains all features and rectangles above the assigned horizontal strip, the second everything below.

To create an instance of BOUNDARY LABELING from an instance of ORBITAL BOUNDARY LABELING, we consider D as an annulus (by introducing a small circle at the center, which does not contain any features), find a line segment s orthogonal to B from the boundary of the small circle to a point on B and transform the annulus into a rectangle, turning B into a straight line. The basis for this reduction is the following observation, which – intuitively speaking – prove that there exists a radial line, which we can use to cut and unroll our instance to remove the cyclic nature of ORBITAL BOUNDARY LABELING.

► **Observation 3.** *In a crossing-free leader-length minimal labeling for problems in OR-COSA, the supporting line of the radial part for the leader of p_{min} intersect no other leader.*

Observation 3 is also illustrated in Figure 2a. Based on this observation we know that in any crossing free labeling, there is always a radial line, which does not intersect a leader other than the one of p_{min} . Moreover it is sufficient to check all possibilities for the port of p_{min} to obtain such a line. Next, we need to show that there is only a polynomial number of possibilities for the port of p_{min} . The exact number depends on the variant we are considering. If we are considering a labeling for a problem in the set OR- C^{\square} OSA, we are given the set C of possible ports, leading to the following observation.

► **Observation 4.** *For problems in C^{\square} OSA there are only $|C|$ possibilities for $\xi_L(p_{min})$.*

It is less obvious if and how we can discretize these options for problems without a fixed candidate set for the ports. If we are considering a problem in the set C° OSA, we can reduce the number of relevant options using the following lemma.

► **Lemma 5.** *For problems in $\text{OR-C}^{\circ}\text{O}^{\square}\text{SA}$ there are only n^2 possibilities for $\xi_L(p_{\min})$.*

Proof. Let S be the set of the n intersection points between B and any ray starting at X through a feature. We prove that there is a leader-length minimal labeling, in which at least one port is in S . Assume that L is a leader-length minimal labeling, where no port is in S . Consider a small rotation of all ports clockwise. If such a rotation is not possible without introducing a crossing, the radial segment of a clockwise leader already contains another feature, and therefore its port was in S (see Figure 2b). Otherwise, a small enough rotation neither changes the order of labels nor does it introduce any intersections and therefore results in a new valid labeling. If this rotation decreases the total leader length, then L was not optimal, which contradicts our assumption. If the total leader length stays the same, we can continue the rotation until either the orbital segment of a counter-clockwise leader reaches length 0 or the radial segment of a clockwise leader hits another feature. In both cases, its port is in S . If the leader length increases, we instead rotate counter-clockwise. Again, if the total leader length decreases or stays the same, the arguments above apply. Assume therefore that the total leader length again increases. Since by Observation 2 the change in leader length is linear in the angle by which we rotate, there must be a single leader that increases its length in both rotation directions, which implies that its orbital part has length 0 in L (see Figure 2c).

Therefore in any optimal labeling, at least one port is in S . Since the order is fixed, choosing a specific label to have its port at an element in S fixes all other ports and specifically the location of the port of p_{\min} . Therefore every element of S induces n possible choices for this port which results in n^2 choices in total. ◀

In general, the previous method does not extend to the problems in $\text{OR-C}^{\circ}\text{O}^{\circ}\text{SA}$, since choosing one element in S still leaves $(n-1)!$ possible orders of the remaining labels. However, in the special case where both the label sizes and the port ratios are uniform, we know that all ports in any valid labeling are distributed equally along B , which implies that again every element in S induces exactly n possible labels (even if we do not know which label is connected to which port). Therefore, we state the following observation.

► **Observation 6.** *For problems in $\text{OR-C}^{\circ}\text{O}^{\circ}\text{S}_{\equiv}\text{A}_{\equiv}$ there are only n^2 possibilities for $\xi_L(p_{\min})$.*

We can now proceed to describe how we create an instance of BOUNDARY LABELING based on a given instance of ORBITAL BOUNDARY LABELING.

► **Lemma 7.** *Given a port ξ for p_{\min} , we can reduce any problem in $\text{OR-CO}^{\square}\text{SA}^{\square}$ to BOUNDARY LABELING with po -leaders [7].*

Proof. We first map all features to points in the plane (also shown in Figure 3). Let X be the center of D . For any feature $p_i \in P$, let α_i be the angle between ξ and p_i (recall that this is defined as the smaller angle formed between these two points at X) and let $r_i = |Xp_i|$. Using polar coordinates we then create a point $q_i = (r_i, \alpha_i)$ for every point $p_i \in P$. We now place a vertical line ℓ at $x = 1$. Recall that the radius of D is 1 and therefore all points q_1, \dots, q_n are left of ℓ . Since the port ratio is locked as part of the input, the fixed position of ξ also fixes the exact position of the label of p_{\min} . If the problem is in the set $\text{OR-CO}^{\square}\text{SA}^{\square}$, the order of labels is fixed, and therefore ξ also fixes the position of all other labels. For every point p_i we place a rectangle of height $\lambda(p_i) - 2\varepsilon$ in the order of the labels (recall that any port has to have a minimal distance of ε to the boundary of the label), s.t., the lowest point of the lowest rectangle is at height $|\widehat{s_L(p_{\min})\xi}| + \varepsilon$.



(a) Instance of ORBITAL BOUNDARY LABELING.

(b) Instance of BOUNDARY LABELING.

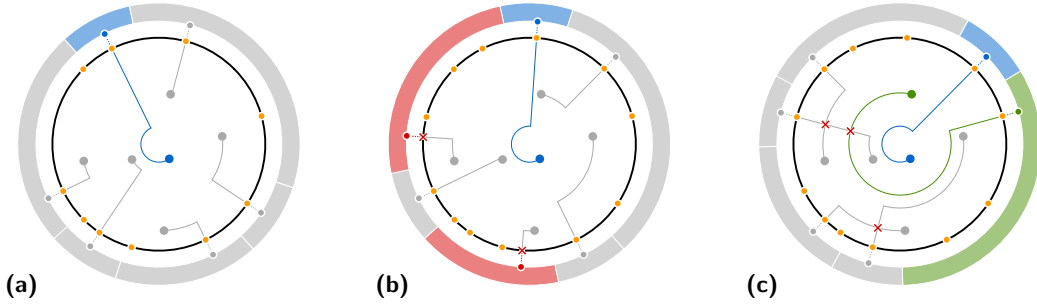
■ **Figure 3** An instance (a) of ORBITAL BOUNDARY LABELING together with a fixed port for p_{\min} . We use an appropriate mapping to construct an instance of BOUNDARY LABELING, whose solution (b) corresponds to the solution of ORBITAL BOUNDARY LABELING. This maps the angle σ between two features in (a) to their horizontal distance between the mapped features in (b) and the distance ρ of a feature to B in (a) to the distance of the mapped feature to ℓ in (b). Note that only the remaining non-fixed labels are part of the new instance.

Now, the length of the radial segment of an orbital-radial leader connecting p_i to a point $b \in B$ in our problem is equal to the length of the orthogonal linepart of the po -leader connecting q_i to a point v on ℓ . The relation between the length of the parallel part of the po -leader and the length of the orbital segment of the OR-leader is more complicated. The length of the parallel part of the po -leader is simply the difference in y -coordinate between q_i and v . Note that we mapped the clockwise angle of a point q_i relative to ξ to the y -coordinate of q_i . However, the length of the orbital part of the OR-leader is dependent on the distance of p_i to X , i.e., two OR-leaders whose orbital part span the same angle can have different lengths. Specifically, the length of an orbital part in an OR-leader of a feature p_i is exactly $r_i \cdot \alpha_i$. Therefore, we define our badness function simply as $\text{bad}(q_i, v) = 1 - r_i + r_i(|\alpha_i - y(q_i)|)$, where $y(q)$ is the y -coordinate of q_i . Note that the restriction of port placement on ℓ to a specific range (e.g., a point in a candidate set C or a point within a label corresponding to a fixed port ratio) can be encoded in this badness function too, by setting the value of $\text{bad}(q_i, v) = \infty$ if v lies outside of the permitted range. Finally, also note that we can check in $O(n)$ time if the split into sub-instances induced by the combination of q and v respects the fixed order of the labels and, if it does not, also set $\text{bad}(q_i, v) = \infty$. This completes the reduction. ◀

If we do not have a fixed order of labels, we have to pay attention to the order in which the rectangles are placed. However, if the labels have uniform size, any order will result in the same set of rectangles resulting in the following observation.

► **Observation 8.** *Given a port ξ for p_{\min} , we can reduce any problem in $\text{OR-CO}^{\circ}S_{=A}^{\square}$ to po -BOUNDARY LABELING [7].*

Lastly, if the label sizes are not uniform, and if a candidate set C is fixed, we can adapt the dynamic program slightly to leverage this fact, by placing smaller rectangles at every possible candidate position and avoid overlap by restricting the sub-instances to appropriate ranges. The proof of the following lemma can be found in Appendix A of the full version [8].



■ **Figure 4** Three rotations for case $\text{OR-C}^{\circ} \text{O}^{\circ} \text{SA}^{\circ}$. Candidates are shown in yellow. The blue feature is the first that is placed and we iteratively test every port candidate. Due to the fixed order, the other leaders are directly obtained. In a valid labeling **(a)** all ports (obtained due to a fixed port ratio) coincide with a candidate and no two leaders cross. A labeling is invalid if ports do not coincide with a candidate – highlighted red in **(b)** – or the obtained leaders contain crossings between themselves, which is shown with the red crosses in **(c)**. In **(c)** the green leader changed from a clockwise to a counter-clockwise leader to avoid crossing the blue leader.

► **Lemma 9.** *Given a port ξ for p_{\min} , any problem in $\text{OR-C}^{\circ} \text{O}^{\circ} \text{S}_{\equiv} \text{A}^{\circ}$ can be solved in $O(|C|^3)$ time by adapting the dynamic program of Benkert et al. [7].*

With all the pieces assembled, we obtain the runtime of solving a number of problem variants via this reduction. We present the relevant runtimes in the following theorem.

► **Theorem 10.** *For any problem P in $\text{OR-CO}^{\circ} \text{SA}^{\circ}$, $\text{OR-C}^{\circ} \text{OSA}^{\circ}$ and $\text{OR-C}^{\circ} \text{O}^{\circ} \text{SA}^{\circ}$ let A_P be the size of the set of possible ports for p_{\min} , s.t., the set can be computed in $O(A_P)$ time and let the time required to solve the created instance of BOUNDARY LABELING be $O(B_P)$. Then P can be solved in time $O(A_P \cdot B_P)$.*

Proof. We compute the set of possible ports for p_{\min} as stated in Observation 4, Lemma 5 or Observation 6. For any element of these sets, we create an instance of BOUNDARY LABELING and solve it as described in Lemma 7, Observation 8 and Lemma 9. We obtain a po -labeling minimizing the badness function, which has a one-to-one correspondence to a leader length minimal OR-labeling of our original instance. ◀

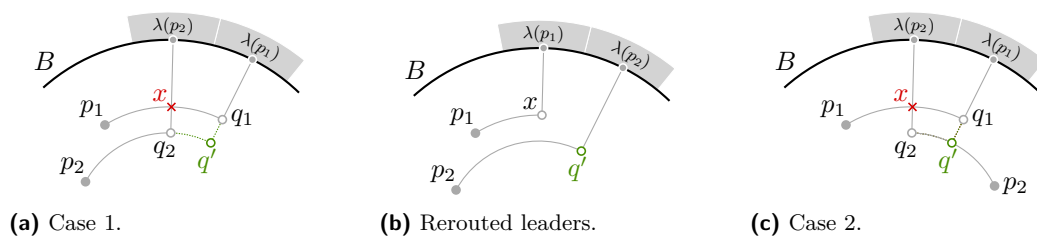
We remark that Benkert et al. [7] point out that their algorithm can be changed to handle non-uniform labels leading to a pseudopolynomial time algorithm. By Theorem 10 this would result in pseudopolynomial time algorithms for the problems in $\text{OR-C}^{\circ} \text{O}^{\circ} \text{S}_{\equiv} \text{A}^{\circ}$, however, these are superseded by a dedicated approach in Section 4, so we omit further details here.

4 Improvement via dedicated approaches

Having established the reduction as a general approach baseline, we will now present a variety of bespoke approaches, which improve the runtime implied by Theorem 10.

4.1 Locked Port Candidates and Locked Order

We begin by investigating the problem set $\text{OR-C}^{\circ} \text{O}^{\circ} \text{SA}^{\circ}$. Recall that we are given a set C of candidate positions for the ports. The placement of the label of p_{\min} determines the position of all other labels. The following lemma (which is applicable to a larger set of problems) shows that the same is true for all OR-leaders.



■ **Figure 5** Given a free label order O° we can reroute the leaders to arrive at a crossing-free solution with a shorter total leader length.

► **Lemma 11.** For $\text{OR-CO}^\circ\text{SA}^\circ$, the choice of a port for p_{\min} uniquely determines the OR-leader of any other feature p' , which does not cross the OR-leader of p_{\min} .

Proof. Note that the orbital parts of the two possible OR-leaders connecting p' to any point on B form a circle concentric with B . Therefore one of the two orbital parts crosses the radial part of the leader of p_{\min} (we also refer again to Figure 2a). The lemma follows. ◀

By Lemma 11 it is sufficient to place $\lambda(p_{\min})$ with $\xi_L(p)$ coinciding with one candidate ($O(|C|)$ possibilities). Then we check in $O(n)$ time if the ports of the remaining labels in the correct order also coincide with candidates, in $O(n^2)$ time that no two leaders cross, and finally we compute in $O(n)$ time the total leader length, leading to a total runtime of $O(|C|n^2)$ (see Figure 4) and to the following theorem.

► **Theorem 12.** The problems $\text{OR-C}^\circ\text{O}^\circ\text{SA}^\circ$ can be solved in $O(|C|n^2)$ time.

4.2 Locked Port Candidates, Free Order and Uniform Port Distribution

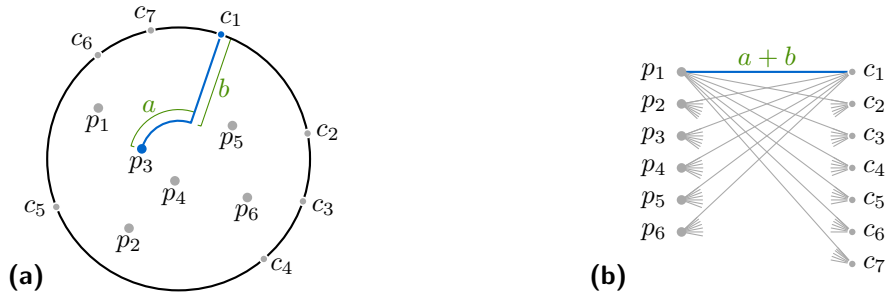
If the order can freely be chosen, then a uniform port ratio together with uniform labels guarantee that any two labels can be exchanged without creating overlap between two labels. We can thus utilize a matching algorithm to solve the problems $\text{OR-C}^\circ\text{O}^\circ\text{S}_{\equiv}\text{A}_{\equiv}$. To obtain a crossing-free labeling we first prove the following lemma showing that a solution minimizing total leader length naturally does not contain any crossings.

► **Lemma 13.** Given an instance of a problem variant in $\text{OR-CO}^\circ\text{S}_{\equiv}\text{A}_{\equiv}$ every leader-length minimal labeling L is crossing-free.

Proof. Assume a leader-length minimal labeling L contains two crossing leaders γ_1 and γ_2 connecting p_1 to its port $\xi_L(p_1) = \xi_1$ and p_2 to its port $\xi_L(p_2) = \xi_2$, respectively. Both leaders begin with an orbital segment $\widehat{p_1q_1}$ (or $\widehat{q_1p_1}$) and $\widehat{p_2q_2}$ (or $\widehat{q_2p_2}$), respectively, connecting to their bend points q_1 and q_2 , followed by their radial straight-line segment $q_1\xi_1$ and $q_2\xi_2$. Clearly, the crossing x occurs between the radial segment of the point closer to the center of D and the orbital segment of the point closer to B . We assume, w.l.o.g., that $x = \widehat{p_1q_1} \cap q_2\xi_2$, and that p_1 is on the counter-clockwise end of $\widehat{p_1q_1}$. Let q' be the intersection of the supporting line of $q_1\xi_1$ and the circle containing $\widehat{q_2p_2}$. There are two cases, as shown in Figure 5.

In the first case (Figure 5a), we can replace γ_1 with a curve consisting of $\widehat{p_1x}$ and $x\xi_2$ and γ_2 with curve consisting of $\widehat{p_2q'}$ and $q'\xi_1$ (Figure 5b). Since $|q_2x| = |q'q_1|$ and $|xq_1| > |q_2q'|$, the total leader length has decreased. Note that the rerouting might have introduced new crossings, but since this method reduces the total leader length, we can iteratively apply this procedure and will never obtain an already seen labeling. Since there is a finite number of

22:10 Boundary Labeling in a Circular Orbit



■ **Figure 6** Case $C^{\square}O^{\circ}S_{\equiv}A_{\equiv}$. Each feature p_1, \dots, p_6 and port candidate c_1, \dots, c_7 (a) introduces a vertex in the weighted complete bipartite graph (b). An edge in the bipartite graph corresponds to a leader and is weighted with the leader's length.

possible solutions, we have to arrive at a solution, which does not contain crossings anymore (otherwise we could apply the procedure infinitely many times contradicting the finite number of possible solutions). We arrive at a labeling, that has a smaller total leader length, which is a contradiction to L being optimal. While the second case (Figure 5c) looks different geometrically, we can resolve the crossing identically to the first case to again reduce the sum of leader lengths. In the special case where both features have the same distance to X , it is again obvious that by simply switching the ports the newly obtained leaders are a subset of the old leaders, removing the overlap and reducing the total leader length. In all three cases, we arrived at a labeling that is better than L which is a contradiction, concluding the proof. ◀

Next, we show that we can always use the shorter of the two OR-leaders.

► **Lemma 14.** *Given an instance of a problem variant in $OR-CO^{\circ}S_{\equiv}A_{\equiv}$ any leader-length minimal labeling L uses only the shorter of the two possible OR-leaders for any point.*

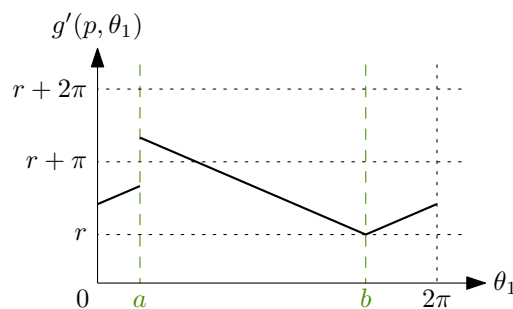
Proof. Assume L contains a long OR-leader for a point p . Replacing the leader with the short leader between p and its port yields a labeling L' with a shorter total leader length. If L' is crossing-free this is a contradiction to the optimality of L , otherwise we can iteratively apply the uncrossing procedure of the proof of Lemma 13, which, by Lemma 13, results in a crossing free labeling with the same or smaller total leader length compared to L' again contradicting the optimality of L . ◀

With Lemmas 13 and 14 we know that every combination of feature and port defines a unique leader and the shortest possible set of such leaders is crossing free. This leads naturally to a formulation of the problem as finding a minimum-weight matching between features and ports. However, we still have to guarantee that the ports we select in such a matching are equally distributed around B ; a requirement stemming from uniform label-sizes and fixed uniform port ratios. To this end, we state the following observation.

► **Observation 15.** *Given a set C of candidate ports, we can partition C into at most $k = \lfloor \frac{|C|}{n} \rfloor$ subsets C_1, \dots, C_k of size n , s.t., all candidates in one set C_i are equally distributed around B . This can be done in $O(|C|^2)$ time.*

Now we can state the central result of this subsection.

► **Theorem 16.** *The problems in $OR-C^{\square}O^{\circ}S_{\equiv}A_{\equiv}$ can be solved in $O(|C|^2n)$ time.*



■ **Figure 7** An example of the function $g'(p, \theta_1)$ for a point p with a distance of $1 - r$ to X . the points a and b are defined as in the proof of Lemma 17.

Proof. We begin by creating the k subsets (Observation 15). For each subset C_i we let G_i be a weighted complete bipartite graph between the set of features P and C_i using the length of the (short) leader between a feature $p \in P$ and a potential port $c \in C$ as the weight of the edge (p, c) ; see Figure 6. For OR-leaders it is by Lemma 14 sufficient to use the length of the shorter of the two leaders. Now a minimum weight bipartite matching in G corresponds to a leader-length minimal labeling. We know by Lemma 13, that such a labeling is crossing-free and an optimal solution (for C_i). Such a matching in a bipartite graph with $|V|$ vertices and $|E|$ edges can be computed in $O(|V|^2 \log |V| + |V||E|)$ time [11]. In our case $|V| = 2n$ and $|E| = n^2$. Therefore the runtime for one subset is $O(n^3)$ and since we run this algorithm for at most $\frac{|C|}{n}$ subsets, we arrive at a final runtime of $O(|C|^2 + |C|n^2)$. ◀

It is important to note that the total runtime of all iterations of the matching algorithm is strictly better than the runtime of $O(|C|n^3)$ yielded by Theorem 10, due to the preprocessing runtime of $O(|C|^2)$. Theorem 16 only guarantees an improvement for $|C| \in o(n^3)$.

4.3 Free Candidates and Locked Order

Now we turn to the problem set $\text{OR-}C^{\circ}O^{\circ}SA^{\circ}$. Intuitively, these are problem variants, in which we can rotate the ports continuously around B to obtain other labelings as long as we do not change the label order or introduce any crossings. To solve these problems, we formulate a univariate piece-wise linear function of bounded complexity, whose global minimum corresponds to an optimal labeling.

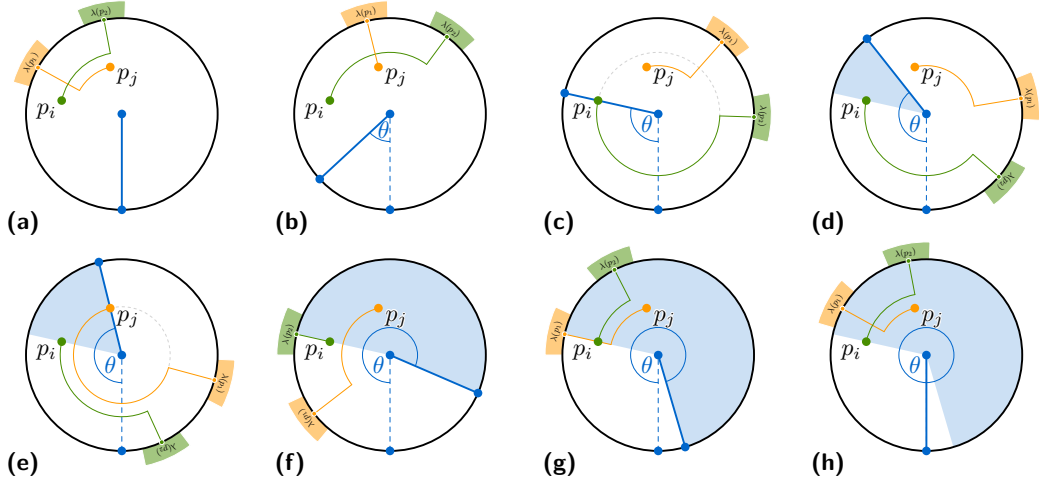
Let $\omega(i)$ be the index of the label of p_i in the fixed order (assuming $\omega(1) = 1$). For any feature p_i let θ_i^L be the angle between p_i and $\xi_L(p_i)$ in a specific labeling L . Recall that by Lemma 11 the choice of the port for the innermost feature in L also fixes all other ports and leaders. We will therefore replace θ_i^L with $\Theta_i(\theta_1^L)$, i.e., a function which simply returns the value of θ_i^L implied by the value of θ_1^L . Since the order of labels is fixed, it is not guaranteed that every feature is connected to its port using the short OR-leader. To do so we define a Boolean variable $cw(i, \theta_1^L)$. Let s_i be the intersection of B and a ray starting at X through p_i . Then $cw(i, \theta_1^L)$ is true if (in the labeling L implied by the value θ_1) starting at s_i and traversing B clockwise we encounter $\xi_L(p_i)$ before $\xi_L(p_1)$ and false otherwise.

With this we define the function $g' : P \times [0, 2\pi] \rightarrow \mathbb{R}$ as

$$g'(p_i, \theta_1) = \begin{cases} g(p_i, \Theta_i(\theta_1)) & \text{if } cw(i, \theta_1) \text{ or } i = 1 \\ g(p_i, 2\pi - \Theta_i(\theta_1)) & \text{otherwise} \end{cases} \quad (1)$$

We now provide a lemma bounding the complexity of these functions.

22:12 Boundary Labeling in a Circular Orbit



■ **Figure 8** Illustration of the interval (blue area) of θ in which the two leaders of the points p_i and p_j do not cross, while increasing θ from 0 (a) to 2π (h). The heavy blue line is the supporting line for the radial part of the leader of p_{\min} . The endpoints are the value of θ for which the leader of p_i changes from clockwise to counter-clockwise to avoid crossing the leader of the innermost point (c) and the value for which the radial part of the leader of p_j crosses p_i (g).

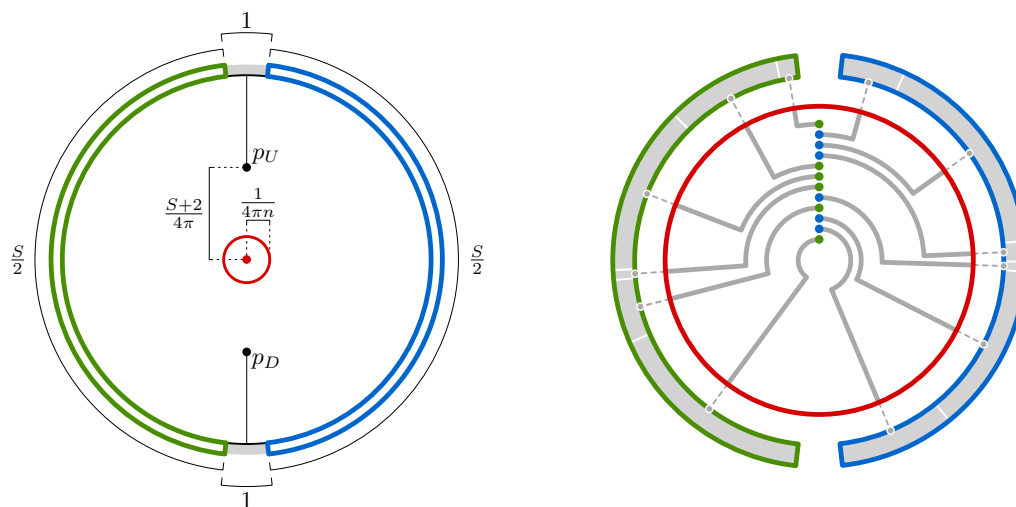
► **Lemma 17.** Any function $g'(p_i, \theta_1)$ consists of at most three continuous linear parts in the interval $0 \leq \theta_1 < 2\pi$.

Proof. Assume $cw(i, 0)$ is true, i.e., we encounter $\xi_L(p_i)$ before $\xi_L(p_1)$ when traversing B clockwise starting at s_i . While increasing θ_1 the value of $cw(i, \theta_1)$ changes to false exactly when $\xi_L(p_1) = s_i$ and back to true exactly when $\xi_{L'}(p_i) = s_i$, where L and L' are labelings implied by some values $\theta_1 = a$ and $\theta_1 = b$, respectively, s.t., $a < b$ (see also Figure 7). Since for $\theta_1 \in [0, a) \cup [b, 2\pi)$ we have $g'(p_i, \theta_1) = g(p_i, \Theta_i(\theta_1))$ and for $\theta_1 \in [a, b)$ we have $g'(p_i, \theta_1) = g(p_i, 2\pi - \Theta_i(\theta_1))$ and by Observation 2 the function g is continuous and linear in these intervals. The case for $cw(i, 0)$ being false is symmetrical and the lemma follows. ◀

We can now create a function $h(\theta_1) = \sum_{i=1}^n g'(p_i, \theta_1)$, which exactly captures the total leader length of a labeling implied by θ_1 , which adheres to the order ω . However, it is important to observe that, while such a labeling does not contain any crossing between the leader of p_1 and any other leader, crossings between any other pair of leaders are still possible. To avoid these, we restrict θ_1 to $O(n^2)$ intervals which capture exactly the values, in which the labeling implied by θ_1 does not contain any crossing. We define a new Boolean variable $cr(i, j, \theta_1)$, which is true if the two OR-leaders of p_i and p_j that are implied by θ_1 , cross. Let p_i be the feature closer to the center of D than p_j . Similar to the proof of Lemma 17, we can find two values θ_a, θ_b , s.t., $cr(i, j, x) = cr(i, j, x')$ for any value $x, x' \in [\theta_a, \theta_b)$ and $cr(i, j, x) = cr(i, j, x')$ for any value $x, x' \in [0, \theta_a) \cup [\theta_b, 2\pi)$.

► **Lemma 18.** Let \mathcal{I} be the set of intervals, s.t., θ_1 implies a labeling in which all leaders are crossing-free if and only if $\theta_1 \in I$ for some $I \in \mathcal{I}$. Then $|\mathcal{I}| \leq n^2$.

Proof. For any pair of features, there is exactly one interval in which their leaders do not cross and one interval in which they do (considering the intervals modulo 2π). An example is shown in Figure 8. We prove the lemma by induction. The base case is a single feature, which is always crossing free, therefore $|\mathcal{I}| = 1^2$. Assume that there are at most n^2 such intervals for n features. We add the $(n + 1)$ -th feature p . For any of the existing n features,



(a) Placement of two blocker features p_U and p_D .

(b) Detail of the placement of p_1, \dots, p_n .

■ **Figure 9** Visualization of the reduction. The two points p_U and p_D are placed close to the boundary (a) and all points p_1, \dots, p_n are placed in the very small red circle. A zoomed-in picture of the red circle is shown in (b).

p defines a single interval in which their leaders do not cross. Consider the possibilities for adding one of these intervals I' to \mathcal{I} . Let $I \in \mathcal{I}$. If $I \cap I' = \emptyset$, we remove I , if $I \cap I' = I$, we retain I and if $I \cap I' = I'$, we remove I and add I' . Otherwise, I contains one or two endpoints of I' but not the entire interval. If I contains only one endpoint of I' , we remove I and add $I \cap I'$. Since I' is an interval, this can happen with at most two other intervals. Both times we remove one interval and add a new one, thereby not changing the size of \mathcal{I} . Finally, if I contains both endpoints of I' but not the entire interval – intuitively the intervals wrap around B in two different directions – then we remove I and add the two continuous parts of $I \cap I'$ increasing $|\mathcal{I}|$ by one. In this case all other intervals of \mathcal{I} are either entirely contained in I or disjoint from I and therefore $|\mathcal{I}|$ does not increase any further. We iteratively add all n new intervals and increase $|\mathcal{I}|$ by a total of at most n . Therefore $|\mathcal{I}| \leq n^2 + n < n^2 + 2n + 1 = (n + 1)^2$. ◀

With this, we have everything in place to prove the following theorem.

► **Theorem 19.** *The problem variants in $\text{OR-}C^{\circ}O^{\circ}SA$ can be solved in $O(n^2)$ time.*

Proof. We set up the function $g'(p_i, \theta_1)$ as in Equation 1 and let $h(\theta_1) = \sum_{i=1}^n g'(p_i, \theta_1)$ as above. Then we restrict θ_1 to the set \mathcal{I} of n^2 intervals (Lemma 18) in which the implied labelings do not contain any crossings. Since by Lemma 17 there are at most $O(1)$ continuous linear pieces for any function g' , we conclude that h consists of $O(n)$ continuous linear pieces and thus also has at most $O(n)$ local minima. To find the global minimum of h it is sufficient to check all $O(n)$ local minima as well as the $O(n^2)$ endpoints of the intervals in \mathcal{I} . ◀

5 Free Candidates, Free Order and Non-uniform labels are NP-hard

The results presented so far cover most of the top half of Table 1. It remains to address the problems in $\text{OR-}C^{\circ}O^{\circ}S_{\equiv}A^{\circ}$. Due to the flexibility of non-uniform labels, as well as the free order and lack of a discrete candidate set, these problems turn out to be weakly NP-hard.

► **Theorem 20.** *Given an instance of OR- $C^{\circ}O^{\circ}S_{\equiv}A^{\square}$ together with $k \in \mathbb{R}$ it is (weakly) NP-hard to decide whether there exists a labeling L with a total leader length of less than k .*

Proof. For the purpose of this proof, we will relax the requirement that D has a radius of 1. The final construction can be scaled down to meet that requirement. The reduction (visualized in Figure 9) is from the weakly NP-hard problem PARTITION, where we are given a set \mathcal{X} of n integers with $S = \sum_{x \in \mathcal{X}} x$ and need to decide if \mathcal{X} can be partitioned into two sets \mathcal{X}_1 and \mathcal{X}_2 , s.t., $\sum_{x \in \mathcal{X}_1} x = \sum_{x \in \mathcal{X}_2} x = S/2$. For the reduction we place for every $x_i \in \mathcal{X}$ a feature $p_i = (0, \frac{i}{4\pi n^2})$. Additionally we place two features $p_U = (0, r)$ and $p_D = (0, -r)$, where $r > \frac{S+2}{4\pi}$. We define $\lambda(p_i) = x_i$ for all $1 \leq i \leq n$ and $\lambda(p_U) = \lambda(p_D) = 1$. Note that $\sum_{i=1}^n \lambda(p_i) + \lambda(p_U) + \lambda(p_D) = S + 2$ and the radius of the enclosing disk is therefore $\frac{S+2}{2\pi}$.

Any feature p_i , s.t., $1 \leq i \leq n$ is contained in a disk of radius $\frac{1}{4\pi n}$ and circumference $\frac{1}{2n}$ centered at the origin. Let $o(i)$ and $r(i)$ be the orbital and radial part of $\gamma_L^{p_i}$, respectively. Note that the sum over all $r(i)$ is equal in all labelings. Let this sum be equal to L_{radial} . Further note that for any p_i , $o(i) < \frac{1}{2n}$. Therefore the sum $\sum_{i=1}^n o(i) < \frac{1}{2}$.

For the problem variants OR- $C^{\circ}O^{\circ}S_{\equiv}A_{\equiv}$, in any labeling L the port ratios $\rho(p_U)$, and $\rho(p_D)$ are necessarily equal. For the variant OR- $C^{\circ}O^{\circ}S_{\equiv}A^{\square}$ port ratios are described as part of the input and we define them, s.t., $\rho(p_U) = \rho(p_D)$. Finally we set $k = 1/2 + L_{\text{radial}}$.

If there exists a partition of \mathcal{X} into two sets $\mathcal{X}_1, \mathcal{X}_2$, s.t., $\sum_{x \in \mathcal{X}_1} x = \sum_{x \in \mathcal{X}_2} x$, then we can make three observations. First, there exists a labeling L in which the length of the orbital part of $\gamma_L^{p_U}$ and $\gamma_L^{p_D}$ is equal to 0 and therefore $\gamma_L^{p_U}$ and $\gamma_L^{p_D}$ are straight lines. Second, in L both spaces between the labels of p_U and p_D are equally spaced, i.e., $|e_L(p_U)s_L(p_D)| = |e_L(p_D)s_L(p_U)|$, since $\rho(p_U) = \rho(p_D)$. Third, in a labeling L' , in which the length of the orbital part of $\gamma_L^{p_U}$ or $\gamma_L^{p_D}$ is not equal to 0, the sum of the length of the orbital parts of $\gamma_{L'}^{p_U}$ and $\gamma_{L'}^{p_D}$ (and therefore the sum over the lengths of all orbital parts of leaders in L') is at least $\frac{2\pi}{S+2} \cdot \frac{S+2}{4\pi} = \frac{1}{2}$. This is because the difference between $\overline{e_{L'}(p_U)s_{L'}(p_D)}$ and $\overline{e_{L'}(p_D)s_{L'}(p_U)}$ is at least 1 (since the label sizes are integers). Therefore the sum over all leader lengths in L is less than $1/2 + L_{\text{radial}}$, while in L' it is at least $1/2 + L_{\text{radial}}$ and L' can never be optimal.

Assume now that \mathcal{X} can be partitioned into two subsets $\mathcal{X}_1, \mathcal{X}_2$, s.t., $\sum_{x \in \mathcal{X}_1} x = \sum_{x \in \mathcal{X}_2} x$. Then the labels can be equally partitioned and the leaders of p_U and p_D can be straight lines. Therefore the total sum of leader lengths is less than k . Conversely, assume no such partition exists. Then the leaders of p_U and p_D must together contain orbital segments of length at least $1/2$ and the total sum of leader lengths is at least k , concluding the proof. ◀

6 Extensions to SL-Leaders and Free Port Ratios

So far we considered problems in OR- $COSA^{\square}$, i.e., settings which use OR-leaders and have a fixed port ratio. Some results can be translated (with small to medium effort adaptations) to the settings using SL-leaders and/or free port ratios. In this section, we will give a high-level overview of which results can be adapted and how. The set of results for free port ratios is shown in Table 2; results for SL-leaders are included as the bottom part of both Table 1 and 2.

Some results (e.g., Theorem 12) are independent of the leader length (beyond computation of leader length). Using a matching algorithm as stated in Theorem 16 also extends to the SL variant (although it requires proving that here also leader-length minimal labelings are crossing-free). Extending Lemma 19 is harder, since the number of possible local minima of a function describing the sum of SL leader lengths is not obvious. We include a conjecture for the settings SL- $C^{\circ}O^{\circ}SA^{\square}$ one of which extends to a setting with free port ratio due to

■ **Table 2** A tabular overview of the problem space and our results. Only **free** port ratios are shown. The abbreviation “w. NP-h” stands for weakly NP-hard. Blue cells are conjectures. These results can be found in the full version [8]; cells indicate the section containing the relevant result.

		C°		C^{\bullet}		
		A_{\equiv}°	A_{\equiv}^{\bullet}	A_{\equiv}°	A_{\equiv}^{\bullet}	
OR	O°	S_{\equiv}	$O(C n^2)$ [8, C.1]	$O(C ^2n^3)$ [8, C.2]	$O(n^2)$ [8, C.1]	$O(n^6)$ [8, C.2]
		S_{\equiv}		$O(C ^5n)$ [8, C.2]		$O(n^7)$ [8, C.2]
	O^{\bullet}	S_{\equiv}	$O(C ^2n)$ [8, C.1]	$O(C ^2n^3)$ [8, C.2]	$O(n^5)$ [8, C.1]	$O(n^6)$ [8, C.2]
		S_{\equiv}			w. NP-h [8, C.3]	
SL	O°	S_{\equiv}	$O(C n^2)$ [8, C.1]		$O(n^2)$? [8, C.1]	
		S_{\equiv}				
	O^{\bullet}	S_{\equiv}	$O(C ^2n)$ [8, C.1]			
		S_{\equiv}			w. NP-h [8, C.3]	

the equivalence of labelings. The construction of our NP-hardness reduction can be reused for the SL-leader case, however, it requires new arguments that it still works as intended. Details can be found in Section B of the full version [8].

While OR-leaders are related to *po*-leaders, no such immediate relation exists for SL-leaders, one reason being that OR-leaders monotonically increase their distance to X , which is not true for SL-leaders. Therefore the reduction to BOUNDARY LABELING explained in Section 3 does not extend to SL-leaders. It also does not work as is for free port ratios, since we relied on the fact that a position of the port of p_{\min} fixes the position of its label, which is not true for free port ratios. For some instances, we can leverage candidate positions or a fixed order to prove again that only a certain set of linear or quadratic size needs to be considered for such a label. Therefore we obtain similar runtimes with an additional linear or quadratic factor. However this reduction only works for non-uniform port ratios, since otherwise the subproblems in the dynamic program of Benkert et al. [7] are not independent. Detailed results for free port ratios can be found in Section C of the full version [8].

7 Conclusion

We have introduced orbital labeling as a variant of boundary labeling for circular boundaries, in which labels are placed as circular arcs in an annulus along the boundary. We provided a broad overview of problem variants, based on five different parameters of this labeling problem. We showed that from an algorithmic point of view, the different parameter combinations lead to distinctively different computational problems. In general, it appears that (unsurprisingly) the non-uniform label setting is computationally harder than the uniform setting. Similarly computing the layout for a fixed order is easier than for a free order of labels. The fixed candidate setting discretizes the problem and allows for an exhaustive search through all possible solutions, in contrast to the free candidate setting. In opposition to leader length, non-uniform port ratios seem to make the problem more approachable if the port ratios are also free, since free but uniform port ratios introduce a property, which has to be fulfilled globally, while free non-uniform port ratios can be fixed locally. Concerning leader types, the linear behaviour of the length of an OR-leader relative to the angle of their port with the x -axis allows for some approaches, which are not (or at least not immediately) applicable to SL-leaders.

It would be interesting to further investigate orbital labeling through a visualization lens as well: the variants have distinct visual styles and portray varying levels of visual complexity and uniformity. We are conducting user experiments to determine whether certain variants are superior to others.

References

- 1 Pankaj K. Agarwal, Marc van Kreveld, and Subhash Suri. Label placement by maximum independent set in rectangles. *Computational Geometry: Theory and Applications*, 11(3-4):209–218, 1998. doi:10.1016/S0925-7721(98)00028-5.
- 2 Sarah E. Battersby, John E. Stewart, Ana Lopez-De Fede, Kevin C. Remington, and Kathy Mayfield-Smith. Ring maps for spatial visualization of multivariate epidemiological data. *Journal of Maps*, 7(1):564–572, 2011. doi:10.4113/jom.2011.1182.
- 3 Michael A. Bekos, Michael Kaufmann, Antonios Symvonis, and Alexander Wolff. Boundary labeling: Models and efficient algorithms for rectangular maps. In János Pach, editor, *Graph Drawing (GD'04)*, volume 3383 of *LNCS*, pages 49–59. Springer, 2004. doi:10.1007/978-3-540-31843-9_7.
- 4 Michael A Bekos, Michael Kaufmann, Antonios Symvonis, and Alexander Wolff. Boundary labeling: Models and efficient algorithms for rectangular maps. *Computational Geometry*, 36(3):215–236, 2007. doi:10.1016/J.COMGEO.2006.05.003.
- 5 Michael A. Bekos, Benjamin Niedermann, and Martin Nöllenburg. *External Labeling: Fundamental Concepts and Algorithmic Techniques*. Synthesis Lectures on Visualization. Morgan & Claypool Publishers, 2021. doi:10.2200/S01115ED1V01Y202107VIS014.
- 6 Marc Benkert, Herman Haverkort, Moritz Kroll, and Martin Nöllenburg. Algorithms for multi-criteria one-sided boundary labeling. In Seok-Hee Hong and Takao Nishizeki, editors, *Graph Drawing (GD'07)*, volume 4875 of *LNCS*, pages 243–254. Springer, 2008. doi:10.1007/978-3-540-77537-9_25.
- 7 Marc Benkert, Herman J Haverkort, Moritz Kroll, and Martin Nöllenburg. Algorithms for multi-criteria boundary labeling. *Journal of Graph Algorithms and Applications*, 13(3):289–317, 2009. doi:10.7155/jgaa.00189.
- 8 Annika Bonerath, Martin Nöllenburg, Soeren Terziadis, Markus Wallinger, and Jules Wulms. On orbital labeling with circular contours, 2024. arXiv:2403.19052, doi:10.48550/arXiv.2403.19052.
- 9 Martin Fink, Jan-Henrik Haunert, André Schulz, Joachim Spoerhase, and Alexander Wolff. Algorithms for labeling focus regions. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2583–2592, 2012. doi:10.1109/TVCG.2012.193.
- 10 Michael Formann and Frank Wagner. A packing problem with applications to lettering of maps. In *Symposium on Computational Geometry (SoCG'91)*, pages 281–288. ACM, 1991. doi:10.1145/109648.109680.
- 11 Michael L. Fredman and Robert Endre Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM*, 34(3):596–615, 1987. doi:10.1145/28869.28874.
- 12 Andreas Gemsa, Jan-Henrik Haunert, and Martin Nöllenburg. Multirow boundary-labeling algorithms for panorama images. *ACM Transactions on Spatial Algorithms and Systems*, 1(1):1–30, 2015. doi:10.1145/2794299.
- 13 Jan-Henrik Haunert and Tobias Hermes. Labeling circular focus regions based on a tractable case of maximum weight independent set of rectangles. In Falko Schmid, Christian Kray, and Holger Fritze, editors, *SIGSPATIAL International Workshop on Interacting with Maps (MapInteract@GIS'14)*, pages 15–21, 2014. doi:10.1145/2677068.2677069.
- 14 Alaul Islam, Tingying He, Anastasia Bezerianos, Bongshin Lee, Tanja Blascheck, and Petra Isenberg. Visualizing information on smartwatch faces: A review and design space. *CoRR*, abs/2310.16185, 2023. doi:10.48550/arXiv.2310.16185.

- 15 Benjamin Niedermann and Jan-Henrik Haunert. Focus+context map labeling with optimized clutter reduction. *International Journal of Cartography*, 5(2-3):158–177, 2019. doi:10.1080/23729333.2019.1613072.
- 16 Marc van Kreveld, Tycho Strijk, and Alexander Wolff. Point labeling with sliding labels. *Computational Geometry: Theory and Applications*, 13(1):21–47, 1999. doi:10.1016/S0925-7721(99)00005-X.
- 17 Alexander Wolff. Graph drawing and cartography. In Roberto Tamassia, editor, *Handbook on Graph Drawing and Visualization*, pages 697–736. Chapman and Hall/CRC, 2013.

Intersection Graphs with and Without Product Structure

Laura Merker ✉ 

Karlsruhe Institute of Technology (KIT), Germany

Lena Scherzer ✉

Karlsruhe Institute of Technology (KIT), Germany

Samuel Schneider ✉ 

Karlsruhe Institute of Technology (KIT), Germany

Torsten Ueckerdt ✉ 

Karlsruhe Institute of Technology (KIT), Germany

Abstract

A graph class \mathcal{G} admits *product structure* if there exists a constant k such that every $G \in \mathcal{G}$ is a subgraph of $H \boxtimes P$ for a path P and some graph H of treewidth k . Famously, the class of planar graphs, as well as many beyond-planar graph classes are known to admit product structure. However, we have only few tools to prove the absence of product structure, and hence know of only a few interesting examples of classes. Motivated by the transition between product structure and no product structure, we investigate subclasses of intersection graphs in the plane (e.g., disk intersection graphs) and present necessary and sufficient conditions for these to admit product structure.

Specifically, for a set $S \subset \mathbb{R}^2$ (e.g., a disk) and a real number $\alpha \in [0, 1]$, we consider *intersection graphs of α -free homothetic copies of S* . That is, each vertex v is a homothetic copy of S of which at least an α -portion is not covered by other vertices, and there is an edge between u and v if and only if $u \cap v \neq \emptyset$. For $\alpha = 1$ we have contact graphs, which are in most cases planar, and hence admit product structure. For $\alpha = 0$ we have (among others) all complete graphs, and hence no product structure. In general, there is a threshold value $\alpha^*(S) \in [0, 1]$ such that α -free homothetic copies of S admit product structure for all $\alpha > \alpha^*(S)$ and do not admit product structure for all $\alpha < \alpha^*(S)$.

We show for a large family of sets S , including all triangles and all trapezoids, that it holds $\alpha^*(S) = 1$, i.e., we have no product structure, except for the contact graphs (when $\alpha = 1$). For other sets S , including regular n -gons for infinitely many values of n , we show that $0 < \alpha^*(S) < 1$ by proving upper and lower bounds.

2012 ACM Subject Classification Mathematics of computing \rightarrow Graphs and surfaces; Mathematics of computing \rightarrow Graph theory

Keywords and phrases Product structure, intersection graphs, linear local treewidth

Digital Object Identifier 10.4230/LIPIcs.GD.2024.23

Related Version *Full Version:* [arXiv:2409.01732](https://arxiv.org/abs/2409.01732) [23]

Funding *Lena Scherzer:* supported by the Deutsche Forschungsgemeinschaft – 520723789.

Samuel Schneider: supported by the Deutsche Forschungsgemeinschaft – 520723789.

Torsten Ueckerdt: supported by the Deutsche Forschungsgemeinschaft – 520723789.

1 Introduction

We are interested in properties P of graph classes, for which, if a graph class \mathcal{G} admits P , this certifies that \mathcal{G} is well-behaving in some sense. For example, having property P for \mathcal{G} can provide a common structure of graphs $G \in \mathcal{G}$, which can be exploited to prove statements for all graphs in \mathcal{G} , or to derive efficient algorithms for graphs in \mathcal{G} . A particularly nice property is that of having *bounded treewidth*, i.e., that there exists a constant t such that



© Laura Merker, Lena Scherzer, Samuel Schneider, and Torsten Ueckerdt;
licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 23; pp. 23:1–23:19

Leibniz International Proceedings in Informatics

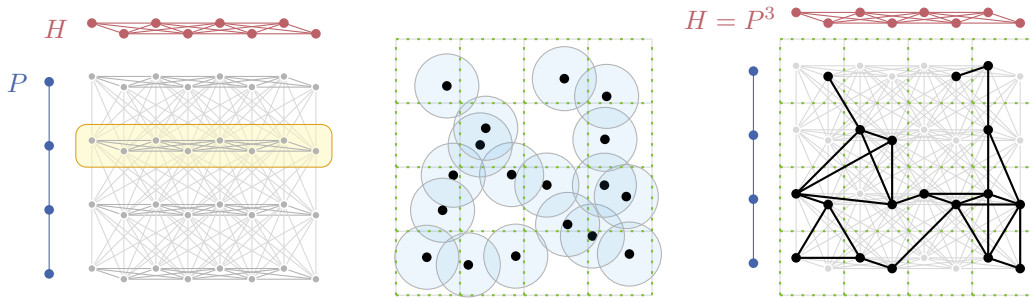


LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

every graph $G \in \mathcal{G}$ is a subgraph of H for some t -tree¹ H . Using the simple structure of t -trees, one can for example show that graphs in \mathcal{G} have small balanced separators, or that MAXINDEPENDENTSET can be solved in linear time. However, the $n \times n$ -grid graph has treewidth n and hence, already the class of planar graphs does not have bounded treewidth.

In 2019, Dujmović et al. [12] introduced with **product structure** a novel concept that generalizes the property of having bounded treewidth. We say that a graph class \mathcal{G} admits *product structure* if there exists a constant t such that every graph $G \in \mathcal{G}$ is a subgraph of $H \boxtimes P$ for a path P and some t -tree H . For example, the $n \times n$ -grid graph is a subgraph of $P_n \boxtimes P_n$, i.e., contained in the strong product of a path and a graph of treewidth 1. In fact, the class of all planar graphs admits product structure with constant $t = 6$ [25].

This constant t is also called the *row treewidth* (denoted rtw) and specifically, for a graph G we write $\text{rtw}(G) \leq t$ if $G \subseteq P \boxtimes H$ for a path P and some t -tree H . The vertices of $P \boxtimes H$ are partitioned into “rows” (one row for each vertex of P), with each row inducing a copy of H . In particular, from any vertex-ordering σ of H , we obtain a natural drawing that reflects the structure of the graph by putting the vertices of the i -th row on y -coordinate (roughly) i and x -coordinate according to σ . See Figure 1 for some illustrating examples.



■ **Figure 1** The strong product $P \boxtimes H$ with a row highlighted (left), a unit disk intersection representation of a graph G (center), and its product structure representation $G \subseteq P \boxtimes H$ (right).

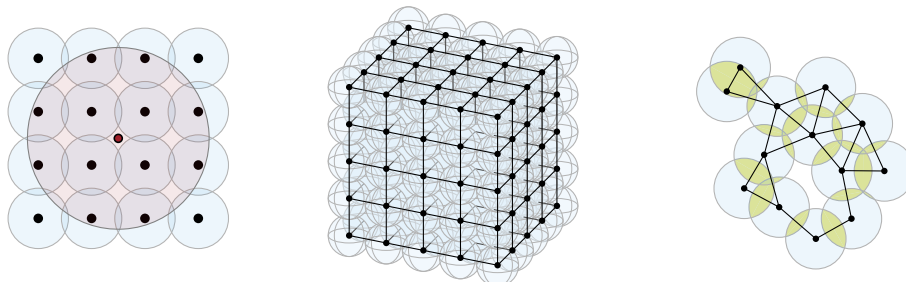
Interestingly, one can reverse this procedure in case G is an intersection graph of unit disks in \mathbb{R}^2 as follows. Roughly speaking, we superimpose a square grid on the intersection representation and let w be the largest number of disk centers in any grid cell. Then we find G as a subgraph of $P \boxtimes H$, where $H = P^{2w-1}$ is the $(2w - 1)$ -th power of a path, i.e., a t -tree for $t = 2w - 1$. This way, one can conclude for every w that $K_{w/4}$ -free unit disk intersection graphs admit product structure. We refer to [17] for a complete, formal proof.

Linear local treewidth. On the other side, we can show that a graph class \mathcal{G} has *no* product structure by showing that \mathcal{G} fails to have another (easier to check) property that in turn would be necessary for product structure. To this end, note that in the product $P \boxtimes H$ each edge either runs within a row or between two consecutive rows. It follows that for every k the k -th closed neighborhood $N^k[v] = \{u \in V \mid d(u, v) \leq k\}$ of a vertex v in $P \boxtimes H$ is completely contained in at most $2k + 1$ rows, each having treewidth $t = \text{tw}(H)$. Therefore, the treewidth of $N^k[v]$ is at most $(2k + 1)t$, i.e., grows only linearly in k for constant row treewidth t .

► **Definition 1.** A graph class \mathcal{G} has linear local treewidth if for all graphs $G \in \mathcal{G}$ and all vertices $v \in V(G)$ the treewidth of the k -th closed neighborhood of v is in $\mathcal{O}(k)$.

¹ Definitions of t -trees, treewidth, and strong products will be given in Section 2.

Linear local treewidth is a necessary (though not sufficient) condition for a class to admit product structure [12, 15, Lemma 6]. For example, in K_4 -free disk intersection graphs we can have an $n \times n$ -grid in the neighborhood of a single vertex (Figure 2-left), and hence K_4 -free disk intersection graphs do not admit product structure. Similarly, the $n \times n \times n$ -grid graph has treewidth $\Omega(n^2)$, but lies completely in $N^{3n}[v]$ of any of its vertices v (Figure 2-center). Hence, K_3 -free intersection graphs of unit balls in \mathbb{R}^3 do not admit product structure.



■ **Figure 2** K_4 -free disk intersection graphs (left) and K_3 -free unit ball intersection graphs (center) with no product structure, and a $\frac{1}{2}$ -free disk intersection graph with the free area in blue (right).

Intersection graphs of α -free homothetic shapes. We consider the question when a class of intersection graphs has product structure, and when not. Crucially, we want the vertices to be represented by homothetic shapes in \mathbb{R}^2 of different sizes. By the discussion above, we must bound the clique size, as well as the size of grids in the neighborhood of any vertex. We do this by requiring that for every vertex, its corresponding set has an α -fraction of its area disjoint from all other shapes (Figure 2-right)². For a shape $S \subseteq \mathbb{R}^2$, let us denote its area by $\|S\|$.

► **Definition 2.** Let $S \subseteq \mathbb{R}^2$ be a set and $\alpha \in [0, 1]$ be a real number. A collection $\mathcal{C} = \{S_v\}_{v \in V}$ of homothetic (obtained from S by positive scaling and translation) copies of S is α -free if

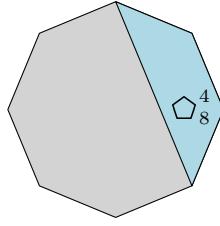
$$\text{for every } v \in V \text{ we have } \|S_v - \bigcup_{u \in V-v} S_u\| \geq \alpha \cdot \|S_v\|.$$

We denote by $\mathcal{G}(S, \alpha)$ the class of all intersection graphs of α -free homothetic copies of S .

Whether $\mathcal{G}(S, \alpha)$ has product structure or not clearly depends on S and α . In general, there is a threshold value $\alpha^*(S) \in [0, 1]$ such that $\mathcal{G}(S, \alpha)$ has product structure for $\alpha > \alpha^*(S)$, and no product structure for $\alpha < \alpha^*(S)$. For an integer $n \geq 3$, we denote by $\diamond_n \subseteq \mathbb{R}^2$ a fixed regular n -gon with area $\|\diamond_n\| = 1$. Our main interest is to determine $\alpha^*(\diamond_n)$.

When two homothetic copies S, S' of \diamond_n intersect, a number $m \leq n$ of corners of one, say S , is contained in the other, S' . Given that exactly m corners of S are covered by S' , the smallest area of S is covered when $S \cap S'$ is the convex hull of m consecutive corners of S . Let us call such a polygon an n -gon segment with m corners and denote it by \diamond_n^m . See Figure 3 for an example. Since $\|\diamond_n\| = 1$, the size $\|\diamond_n^m\|$ of \diamond_n^m is the fraction of \diamond_n covered by \diamond_n^m . In particular, we have $\|\diamond_n^m\| \in [0, 1]$. In fact, for most of the results in this paper we take $\alpha = \|\diamond_n^m\|$ for an appropriate choice of m (possibly depending on n).

² This restriction is better suited to investigate the threshold between product structure and no product structure than the (possibly more common) ply, i.e., the number of shapes that may meet at the same point. In fact, Figure 2 left already shows that graphs with ply 3 do not admit product structure.



■ **Figure 3** The 8-gon segment with 4 corners \square_{8}^4 .

Our results. We investigate whether $\mathcal{G}(S, \alpha)$ has product structure or not. As soon as $\alpha > 0$, there are indeed constants $w = w(S, \alpha)$ and $k = k(S, \alpha)$, such that all graphs in $\mathcal{G}(S, \alpha)$ are K_w -free and with no $k \times k$ -grid in any neighborhood $N^1[v]$. But, while these obstructions are ruled out, $\mathcal{G}(S, \alpha)$ has still no product structure if $\alpha > 0$ is too small.

► **Theorem 3.** *For every $n \geq 2$ and $\alpha < \|\square_{2n}^4\|$, the class of all intersection graphs of α -free homothetic regular $2n$ -gons does not admit product structure.*

In other words, we have $\alpha^*(\square_{2n}) \geq \|\square_{2n}^4\| > 0$ for every $n \geq 2$. As an interesting special case, we highlight that $\|\square_4^4\| = 1$ and hence $\alpha^*(\square_4) = 1$. That is, for every $\bar{\alpha} > 0$ we can construct collections $\mathcal{C} = \{S_v\}_{v \in V}$ of axis-aligned squares such that each square $S_v \in \mathcal{C}$ has at most an $\bar{\alpha}$ -fraction of its area covered by $\mathcal{C} - S_v$, i.e., \mathcal{C} is arbitrarily close to a contact representation, and still the intersection graphs of \mathcal{C} do not admit product structure³.

In fact, we may encounter the same situation among general (irregular) convex n -gons.

► **Theorem 4.** *For every $n \geq 3$ there is an n -gon S such that for all $\alpha < 1$ the class of intersection graphs of α -free homothetic copies of S does not admit product structure.*

For regular n -gons however, if α is large enough, we always have product structure.

► **Theorem 5.** *There is an $\alpha < 1$ such that for all $n > 6$, the class of intersection graphs of α -free homothetic regular n -gons admits product structure.*

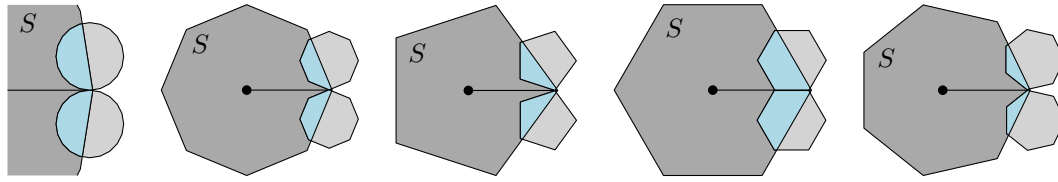
In other words, we have an $\hat{\alpha} < 1$ such that $\alpha^*(\square_n) \leq \hat{\alpha}$ for every $n > 6$. We prove Theorem 5 by showing that the graphs in $\mathcal{G}(\square_n, \hat{\alpha})$ are planar, and for this reason $\mathcal{G}(\square_n, \hat{\alpha})$ admits product structure [12]. To prove the planarity, we consider the *canonical drawing* of the intersection graph G , which is derived from an intersection representation with α -free homothetic regular n -gons by placing each vertex v at the center of its shape S_v and draw each edge uv as a short 1-bend polyline inside $S_u \cup S_v$. We define and discuss these drawings more detailed in Section 4. In fact, also many beyond-planar graph classes admit product structure (as we shall list below). And we actually suspect (cf. Conjecture 7 below) that whether or not $\mathcal{G}(\square_n, \alpha)$ admits product structure is equivalent to whether or not the canonical drawings of the graphs in $\mathcal{G}(\square_n, \alpha)$ belong to a novel type of beyond-planar drawing style.

► **Definition 6.** *For $k \geq 0$, a topological drawing⁴ Γ of a graph G in \mathbb{R}^2 is k -independent crossing if no edge e of G is crossed in Γ by more than k independent edges of G .*

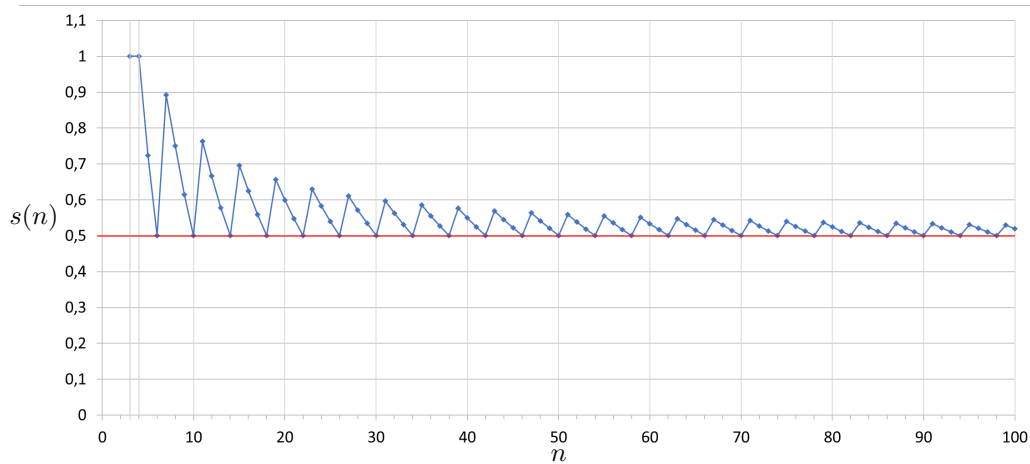
Clearly, 0-independent crossing drawings are precisely planar drawings. And 1-independent crossing drawings are precisely fan-crossing drawings [6]. In general, in a k -independent crossing drawing, every edge may for example be crossed by k stars of edges.

³ Hence, these graphs do not belong to any graph class with product structure as listed in **Related work**.

⁴ Vertices are points and edges are curves connecting their end-vertices. Any two edges have only finitely many points in common; each being either a common endpoint or a proper crossing.



■ **Figure 4** Two regular n -gons meeting in third, S , for large n and the cases $n \equiv 0, 1, 2, 3 \pmod{4}$.



■ **Figure 5** Values of $s(n)$ for $n = 3, \dots, 100$.

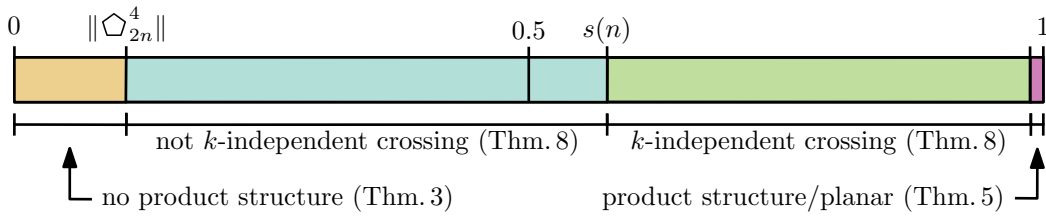
► **Conjecture 7.** *The class of intersection graphs of α -free homothetic regular n -gons admits product structure if and only if their canonical drawings are k -independent crossing for a global constant k (possibly depending on n).*

Our final contribution is to exactly determine α in terms of n for which the canonical drawings of all graphs in $\mathcal{G}(\diamond_n, \alpha)$ are k -independent crossing for some k . For this, we define

$$s(n) = \begin{cases} \|\diamond_n^{n/2+2}\| & \text{if } n \equiv 0 \pmod{4} \\ \|\diamond_n^{\lceil n/2 \rceil + 1}\| & \text{if } n \equiv 1 \pmod{4} \\ \|\diamond_n^{n/2+1}\| = \frac{1}{2} & \text{if } n \equiv 2 \pmod{4} \\ \|\diamond_n^{\lceil n/2 \rceil + 2}\| & \text{if } n \equiv 3 \pmod{4}. \end{cases} \tag{1}$$

The function $s(n)$ is defined as the tipping point whether or not two regular α -free n -gons can meet in a third regular n -gon S without containing a corner of S . See Figure 4 for an illustration and Figure 5 for a plot of $s(n)$. In particular note that $s(n) \geq \frac{1}{2}$ for all n and $\lim_{n \rightarrow \infty} s(n) = \frac{1}{2}$. The four cases are due to whether or not the four corners closest to the boundary of S need to be inside S in order for the two n -gons to meet, e.g., in the case $n \equiv 0 \pmod{4}$ only $n/2$ of the corners are inside S (and therefore, at the tipping point where the two n -gons meet exactly at a corner of S , $n/2 + 2$ are outside or at the boundary), while half of the area is covered if $n/2 + 1$ corners are inside S as in the case $n \equiv 2 \pmod{4}$.

► **Theorem 8.** *Let $n \geq 3$, $\alpha \in [0, 1]$, and $s(n)$ be defined as in (1). Then there exists a constant $k = k(n)$ such that the canonical drawings of all graphs in $\mathcal{G}(\diamond_n, \alpha)$ are k -independent crossing if and only if $\alpha \geq s(n)$.*



■ **Figure 6** Our results for even $n > 6$. It holds that $\lim_{n \rightarrow \infty} s(n) = \frac{1}{2}$ and $\lim_{n \rightarrow \infty} \|\hat{\square}_{2n}^4\| = 0$.

Crucially, Theorem 8 and Conjecture 7 together would give $\alpha^*(\hat{\square}_n) = s(n)$, i.e., that $\alpha \geq s(n)$ is also the exact tipping point for the product structure of the class $\mathcal{G}(\hat{\square}_n, \alpha)$. An overview of our results for even $n > 6$ is given in Figure 6.

Organization of the paper. After some related work below, we quickly define in Section 2 treewidth and product structure. In Section 3 we prove Theorems 3 and 4 by presenting a new graph class called *nested grids* with *no* linear local treewidth (hence no product structure), and then constructing α -free intersection representations of nested grids for the claimed sets S . In Section 4 we define and discuss the canonical drawing of an intersection graph from an α -free intersection representation. The canonical drawings are then used to prove Theorem 5 in Section 5 and Theorem 8 in Section 6. We discuss conclusions in Section 7.

Related work. Since the introduction by Dujmović et al. [12], a variety of graph classes have been shown to admit product structure, including planar graphs, graphs with bounded Euler genus g , apex-minor-free graphs [12, 25], k -planar graphs, k -nearest-neighbor graphs, (g, k) -planar graphs, d -map graphs, (g, d) -map graphs [16], h -framed graphs [1], (g, δ) -string graphs [8, 16], k -th powers of planar graphs with bounded maximum degree [8, 20], fan-planar graphs, k -fan-bundle graphs [20], and K_w -free intersection graphs of unit disks in \mathbb{R}^2 [17]. In addition, product structure has been used to investigate different concepts in graphs; sometimes resolving long-standing conjectures. This includes adjacency labeling schemes [2, 9, 19], nonrepetitive colorings [10], p -centered colorings [7], clustered colorings [11, 14], vertex rankings [5], queue layouts [12], reduced bandwidth [3], comparable box dimension [18], neighborhood complexity [21], twin-width [1, 22], and odd-coloring numbers [13].

On the other hand, there are only very few results for the *non*-existence of product structure. Besides linear local treewidth (Definition 1), a necessary condition for product structure is having bounded layered treewidth [4]. In fact, bounded layered treewidth implies linear local treewidth [15], making the former the stronger condition. For proper minor-closed graph classes, both linear local treewidth and bounded layered treewidth are also sufficient conditions for product structure [12]. However, this does not hold for general graph classes as some graph classes with bounded layered treewidth admit no product structure [4].

2 Preliminaries

Treewidth. Treewidth is a graph parameter first introduced by Robertson and Seymour [24] measuring the similarity of a graph to a tree. Let us define the edge-maximal graphs of treewidth t : For an integer $t \geq 0$, a t -tree is a graph H that is either K_{t+1} , or obtained from a smaller t -tree H' by adding one new vertex v with neighborhood $N(v) \subset V(H')$ that induces a clique of size t in H' . Now, the *treewidth* of a graph G , denoted as $\text{tw}(G)$, is the minimum t such that $G \subseteq H$ for some t -tree H .

Strong product of graphs. The *strong product of graphs* is a combination of the Cartesian product of graphs and the tensor product of graphs. The vertex-set of the strong product $G \boxtimes H$ of two graphs G and H is defined as $V(G) \times V(H)$. The edge-set is the union of the edges in the Cartesian and the tensor product of G and H . That is, there is an edge in $G \boxtimes H$ between two vertices $(u, u'), (v, v') \in V(G \boxtimes H)$ if and only if

$$u = v, u'v' \in E(H) \quad \text{or} \quad u' = v', uv \in E(G) \quad \text{or} \quad uv \in E(G), u'v' \in E(H).$$

3 Intersection graphs without product structure

In this section we prove that, for some $\alpha \in [0, 1]$ and some $S \subseteq \mathbb{R}^2$, the class $\mathcal{G}(S, \alpha)$ of all intersection graphs of α -free homothetic copies of S does not admit product structure. In particular, we consider for S regular $2n$ -gons in Section 3.2 and irregular n -gons in Section 3.3. Both cases rely on the same general construction, which we describe first in Section 3.1.

3.1 Nested grids

We aim to construct a graph class \mathcal{G} that does *not* have linear local treewidth (cf. Definition 1). Then, by [12, 15, Lemma 6], \mathcal{G} admits no product structure. Note that if \mathcal{G} has linear local treewidth, for each graph $G = (V, E) \in \mathcal{G}$ its treewidth $\text{tw}(G)$ is linearly bounded by its radius $\text{rad}(G) = \min_{v \in V} \min\{k \mid N^k[v] = V\}$. We now aim to construct a sequence G_1, G_2, \dots of graphs with $\text{rad}(G_k) \in O(k)$ but $\text{tw}(G_k) \in \Omega(k^2)$, i.e., where the treewidth is not linear in the radius. Here, we give a general description of $G_k, k \geq 1$, which is then completed in detail depending on the particular polygon S in Sections 3.2 and 3.3.

Step 1: Large grid to ensure small radius. We start with a $(k + 1) \times (k + 1)$ -grid with each edge subdivided twice, called the *large grid*. For an intersection representation, we use $(k + 1)^2$ large homothets of S , denoted $c_{i,j}$ with $i, j \in [k + 1]$, in a grid pattern representing the grid-vertices, and $4k^2$ smaller homothets of S , called the *subdivision shapes*, for the subdivision-vertices. The exact placement is chosen such that the subdivision shapes meet at their corners so that the resulting intersection graph is the desired subdivided grid as shown in Figure 7. Note that we do not require that the shapes $c_{i,j}$ have only a point contact with the subdivision shapes. We refer to this graph as $G_{k,1}$ and to the areas that are bounded by exactly twelve shapes as *cells*.

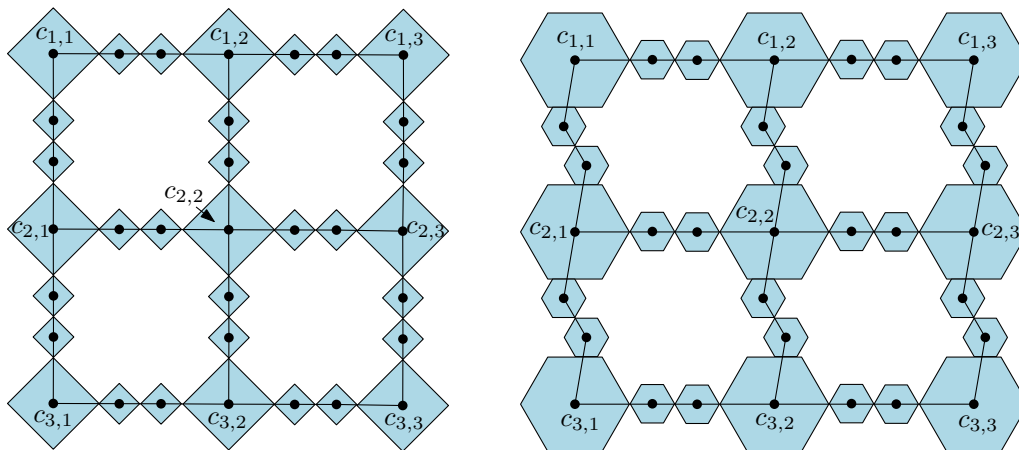
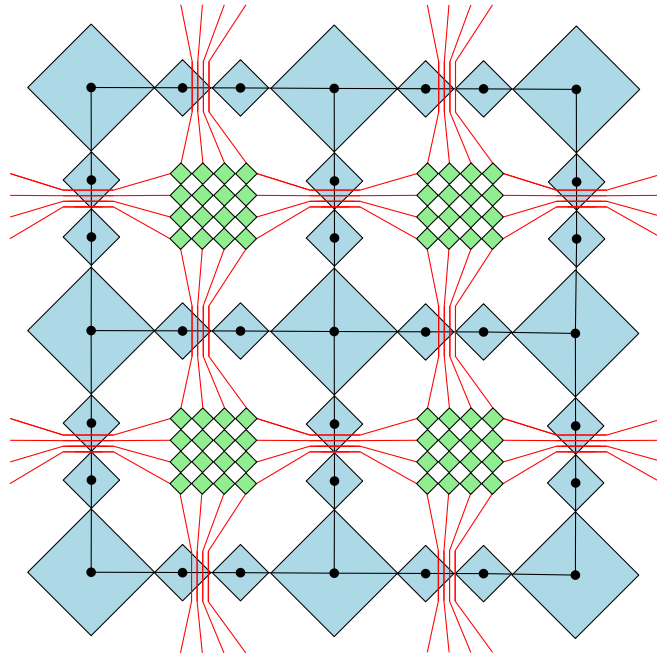


Figure 7 Examples of the large grid with homothetic squares or hexagons, as described in Step 1.



■ **Figure 8** Parts of a large grid (blue) with small grids in its cells (green) connected by paths (red).

Step 2: Small grids inside cells. Next, we insert a $k \times k$ -grid into each cell such that these *small grids* do not touch the large grid from Step 1. We denote the union of $G_{k,1}$ and all small grids by $G_{k,2}$. In the next step, we connect the small grids so that together they form a grid of size $k^2 \times k^2$ plus some additional edges and subdivisions, yielding quadratic treewidth. To do so, we also specify the placement of the small grid inside the cell more precisely in the next step. The large grid ensures that the radius of the resulting graphs is linear in k .

Step 3: Connecting the small grids to ensure large treewidth. We connect any two small grids in neighboring cells of $G_{k,1}$ by adding k pairwise disjoint paths, called *connecting paths*, as illustrated in Figure 8. Each set of connecting paths crosses the subdivided edge of $G_{k,1}$ at the contact point of the two corresponding subdivision shapes. To realize these crossings, we must ensure that a contact point of two same-sized homothets of S can be crossed by k independent edges, while keeping all shapes α -free. Furthermore, we connect the endpoints of these k independent edges to their corresponding small grid with pairwise disjoint paths, while keeping the radius small, that is in $O(k)$.

Leaving the shape-specific details of the crossings to Sections 3.2 and 3.3, we now show how to achieve the linear radius. As $G_{k,1}$ has radius $\mathcal{O}(k)$, it suffices to show that every vertex we add in a cell has distance $\mathcal{O}(k)$ to some vertex of $G_{k,1}$. We achieve this by placing the connecting paths inside a cell within a narrow corridor very close to the border of the cell (Figure 9). Such a corridor and connecting paths along the border of a cell can always be constructed using 1-free homothets of S of very small size. We start with a set \mathcal{P}_t of k paths connecting the top of the cell with the small grid, which is placed near the top of the cell for this purpose. All further connecting paths in the same cell can be placed iteratively by going along the new boundary. Note that the small grid is not necessarily placed in the center of the cell as the exact geometry inside the cell is shape-specific and the center might not be reachable while keeping the radius small.

As we create $4k$ paths per cell, every vertex on the paths has distance $O(k)$ from the grid $G_{k,1}$. In addition, the small grids are placed such that they touch the end of the constructed paths, yielding a distance of $O(k)$ for every vertex in each cell.

3.2 Regular $2n$ -gons

In this section we prove Theorem 3 by giving for every $n \geq 2$ an explicit $\alpha < 1$ such that intersection graphs of α -free homothetic copies of \diamond_{2n} admit no product structure. Recall that $\|\diamond_{2n}^m\| \in [0, 1]$ denotes the portion of the area of \diamond_{2n} within a segment with m corners.

► **Theorem 9.** *For every $n \geq 2$ and $\alpha < \|\diamond_{2n}^4\|$, the class of all intersection graphs of α -free homothetic regular $2n$ -gons does not admit product structure.*

Hence, as $\|\diamond_4^4\| = 1$ and $\|\diamond_6^4\| = \frac{1}{2}$, regular α -free squares do not admit product structure for any $\alpha < 1$, and regular α -free hexagons do not admit product structure for any $\alpha < \frac{1}{2}$.

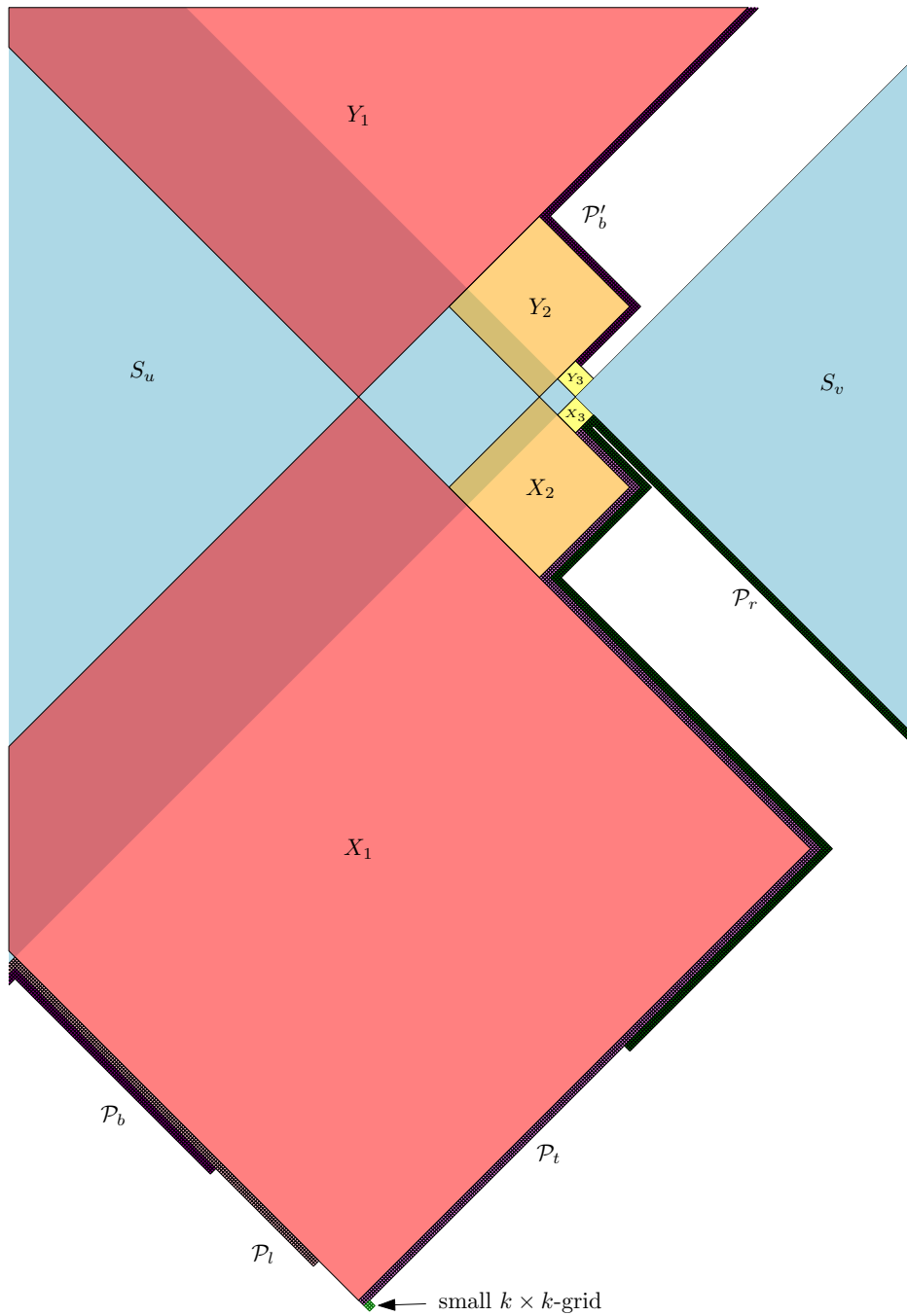
Proof. We prove the theorem by using the construction described in Section 3.1. Thus, we need to show that all three steps of the construction are feasible using α -free regular n -gons. Constructing the grid $G_{k,1}$ described in Step 1 using regular n -gons is clearly possible for all n , e.g., see Figure 7 for $n = 4$ and $n = 6$. For Step 2 place a $k \times k$ -grid inside each cell of $G_{k,1}$ yielding $G_{k,2}$. The main challenge is to show that Step 3 of the construction is feasible.

In Step 3 we connect the small grids to together contain a $k^2 \times k^2$ -grid subdivision. Let S_u, S_v be two adjacent subdivision shapes in the grid $G_{k,1}$ and recall that they meet at a corner. We aim to construct k pairwise disjoint paths crossing the S_u - S_v -contact with contact point q , see Figure 10 left. We thereby ensure that the inserted shapes do not intersect any shapes other than S_u and S_v . We iteratively place n -gons $X_1, Y_1, \dots, X_k, Y_k$ such that after iteration i with $i \in [k]$, the n -gons $X_1 Y_1, \dots, X_i Y_i$ form i independent edges crossing the line segment l from the center of S_u to q . Additionally, after iteration i there is an ε_i -ball $B_{\varepsilon_i}[q]$ around q for some $\varepsilon_i > 0$ such that $B_{\varepsilon_i}[q]$ does not intersect any X_j or Y_j for $j \leq i$. For ease of presentation, let $B_{\varepsilon_0}[q]$ be a ball around q that only intersects S_u and S_v .

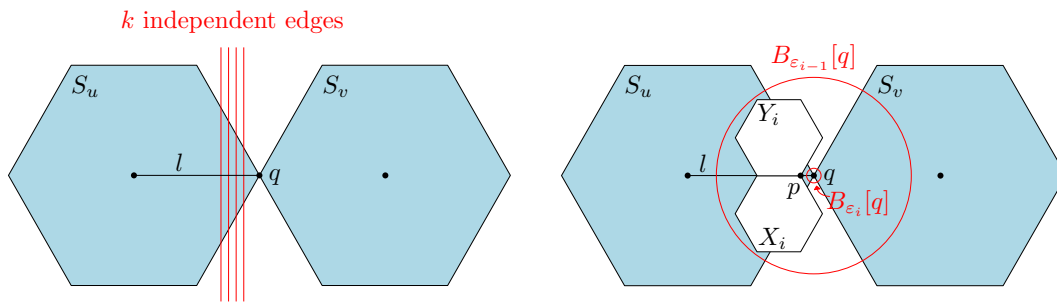
In iteration i we consider the ball $B_{\varepsilon_{i-1}}[q]$ that does not intersect any n -gons X_j, Y_j placed before. Let the corners of an n -gon S_z be c_1^z, \dots, c_n^z in clockwise order. Without loss of generality, let q be the corner c_1^u of S_u . Let $c_1^{X_i}$ be the corresponding corner of the n -gon X_i and $c_1^{Y_i}$ be the corresponding corner of Y_i . Further let p be a point close to q on l with ε' distance from q , where $0 < \varepsilon' < \varepsilon_{i-1}$. We place X_i, Y_i inside $B_{\varepsilon_{i-1}}[q]$ such that they share a side and meet with a corner at p , i.e., such that $c_n^{X_i} = p = c_2^{Y_i}$ as shown in Figure 10 right.

Note that two corners of X_i and Y_i , respectively, are placed outside of S_u and the two shapes do not intersect S_v . In addition, two corners are placed ε' -close to the border of S_u . Thus, for ε' small enough X_i and Y_i have arbitrarily less than $\|\diamond_{2n}^4\|$ area disjoint from S_u . As X_i and Y_i do not intersect any other shapes than S_u , we can choose ε' sufficiently small such that X_i and Y_i are α -free for every $\alpha < \|\diamond_{2n}^4\|$. Further observe that as $\varepsilon' > 0$, after placing X_i and Y_i there still exists a ball $B_{\varepsilon_i}[q]$ with positive radius ε_i around q that does not intersect any X_j or Y_j for $j \leq i$. Thus, the invariants hold and we can continue placing n -gons. After k iterations we have k independent edges crossing S_u and S_v , as required.

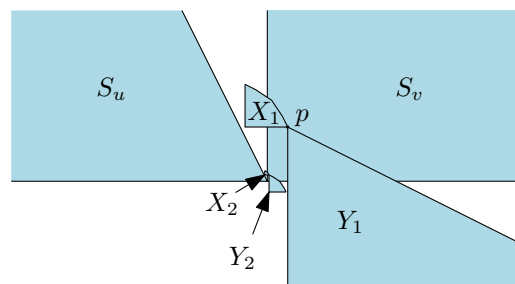
Finally, the edges $X_1 Y_1, \dots, X_i Y_i$ can be connected to the small grids as explained in Step 3 of the construction while keeping the radius in $\mathcal{O}(k)$. Thus, we have constructed a graph class that does not have linear local treewidth, which rules out product structure. ◀



■ **Figure 9** Example of how the paths connecting the grids can be constructed. S_u and S_v are subdivision shapes and X_1, Y_1, X_2, Y_2 , and X_3, Y_3 realize the independent edges crossing the contact point of S_u and S_v required in Step 3. \mathcal{P}_t connects the top of the cell, i.e. X_1, X_2, X_3 , with the small grid. $\mathcal{P}_l, \mathcal{P}_b, \mathcal{P}_r$ connect the left, bottom, and right side of the cell to the small grid and are shortened for improved readability. Note that the latter three sets of paths are not symmetric but walk along the boundary of the cell in order to reach the small grid while keeping the radius small.



■ **Figure 10** Left: We aim to cross two hexagons U, V with k independent edges. Right: Two hexagons X_i, Y_i inside the ball $B_{\epsilon_{i-1}}[q]$ with $c_n^{X_i} = p = c_2^{Y_i}$ that cross the line segment l from the center of S_u to q . In the next iteration, the hexagons X_{i+1}, Y_{i+1} are placed inside $B_{\epsilon_i}[q]$.



■ **Figure 11** Two shapes S_u, S_v that we cross with two independent edges x_1y_1, x_2y_2 . Note that to the bottom-left of X_2 and Y_2 , there is space for more crossing edges.

3.3 Triangles and irregular n -gons

This section is devoted to Theorem 4, which states that for every $n \geq 3$ there is a (possibly non-regular) n -gon S such that α -free intersection graphs of shapes homothetic to S do not admit product structure for any $\alpha < 1$. As all triangles are affinely equivalent, we conclude:

► **Corollary 10.** *The graph class of α -free intersection graphs of homothetic triangles does not admit product structure for any $\alpha < 1$.*

The following lemma specifies the shapes we use and immediately implies Theorem 4. We refer to Figure 11 for examples of shapes that satisfy the required properties.

► **Lemma 11.** *Let S be a convex shape with two orthogonal adjacent sides $l(S), b(S)$ such that S is contained in the rectangle spanned by $l(S)$ and $b(S)$ and no sides of S are parallel to $l(S)$ or $b(S)$. Then, for no $\alpha < 1$ does the class of all α -free intersection graphs of shapes homothetic to S admit product structure.*

The main difference to Section 3.2 is how we implement crossings, which is shown in Figure 11. We give a full proof in [23, Lemma 11].

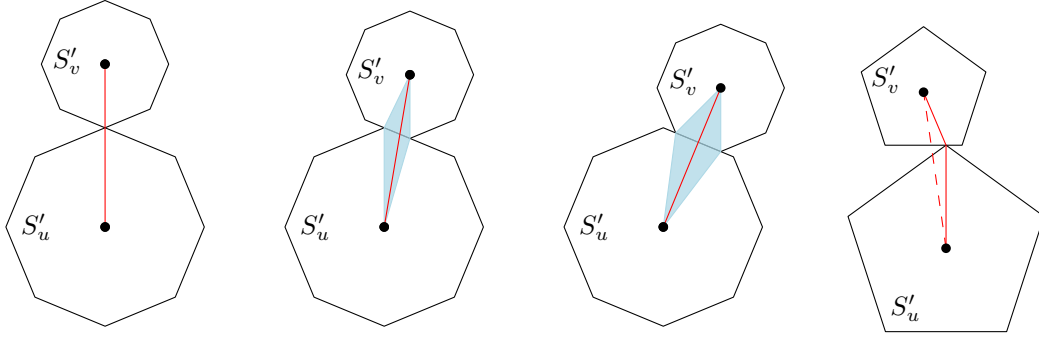
4 Canonical drawings

In this section, we describe how we derive a drawing of the corresponding intersection graph $G = (V, E)$ from a collection \mathcal{C} of α -free homothetic copies of \square_n (for some $\alpha > 0$). That is, we identify a point in \mathbb{R}^2 for each vertex $v \in V$ inside its corresponding set $S_v \in \mathcal{C}$, and route each edge $uv \in E$ as a polyline in \mathbb{R}^2 inside an ϵ -blowup of $S_u \cup S_v$. Both steps are quite

23:12 Intersection Graphs with and Without Product Structure

natural, but some care is needed in the details. While there is nothing surprising here, in the upcoming Sections 5 and 6 we prove that for specific choices of n and α , these drawings have interesting properties, such as being planar or k -independent crossing (cf. Definition 6).

Let $n \geq 3$ be fixed, and $\mathcal{C} = \{S_v\}_{v \in V}$ be a collection of α -free homothetic copies of \diamond_n for some $\alpha > 0$, and $G = (V, E)$ be its intersection graph. Choose $\varepsilon > 0$ small enough (to be discussed later). For each shape $S_v \in \mathcal{C}$ let c_v denote its center. We draw each vertex $v \in V$ as a point inside the ε -ball $B_\varepsilon(c_v)$ around c_v , such that all vertices lie in general position.



■ **Figure 12** Embeddings with the edges (thick red) inside the shapes.

Now, for every edge $uv \in E$ individually, we do the following. First, we scale down S_u at its center c_u to S'_u , and S_v at its center c_v to S'_v , such that $B_\varepsilon(c_u) \subset S'_u$, $B_\varepsilon(c_v) \subset S'_v$, and S'_u and S'_v touch but share no positive area. If the line segment $\overline{c_u c_v}$ intersects $S'_u \cap S'_v$, let p_{uv} denote the point of intersection. This is for example always the case when n is even (see Figure 12). Otherwise, let p_{uv} be the single point in $S'_u \cap S'_v$ (see Figure 12-right).

We now draw the edge uv as a 1-bend polyline connecting u to p_{uv} and p_{uv} to v . Observe that the edge uv , including its bend p_{uv} , is drawn inside $S'_u \cup S'_v$ and hence inside $S_u \cup S_v$. In case the bend points of several edges happen to coincide, we slightly move the bend points within their ε -balls such that no two such edges with a common endpoint cross. Similarly, we slightly move the bend points such that they are in general position together with the vertices. Hence, the drawing is simple⁵ except that edges may cross twice.

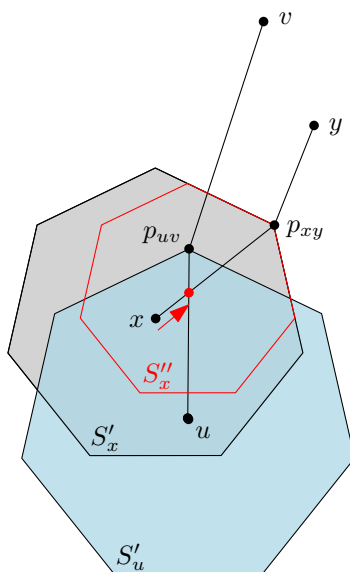
Let us list some crucial properties of the resulting drawing.

► **Observation 12.** *Given α -free homothetic copies of \diamond_n , the canonical drawing Γ of their intersection graph $G = (V, E)$ satisfies the following properties:*

- *Every vertex $v \in V$ is drawn ε -close to the center c_v of its shape S_v .*
- *For every edge $uv \in E$ there are scaled-down interiorly disjoint S'_u, S'_v with $B_\varepsilon(c_u) \subset S'_u \subseteq S_u$ and $B_\varepsilon(c_v) \subset S'_v \subseteq S_v$, with the edge drawn as a polyline with its only bend ε -close to $p_{uv} \in S'_u \cap S'_v$.*
- *The set of all bend points and all vertices is in general position.*

Note that we choose the ε -offsets sufficiently small so that if there is a crossing in our drawing, then the two edges also intersect in the possibly non-simple embedding obtained by choosing $\varepsilon = 0$. Hence, from now on we may assume the vertices and bends to be placed exactly at the centers, respectively contact points, for checking whether two edges cross.

⁵ An embedding is called *simple* if vertices and edges do not share points except for incident edges meeting at their common endpoint and non-adjacent edges may cross once but only two in a point (i.e., no touchings, no self-crossings, no crossings of adjacent edges, no three edges crossing in the same point).



■ **Figure 13** Situation in the proof of Lemma 14 with $p_{uv} \in S'_x \subseteq S_x$.

5 Planar drawings

Complementing the results in Section 3, we show here that for some $\alpha < 1$ and all $n > 6$, the α -free intersection graphs of homothetic regular n -gons admit product structure. But let us quickly discuss the case $\alpha = 1$ first. Here we have contact representations, i.e., the n -gons are interiorly disjoint and induce an edge if they touch. For every $n \neq 4$, these contact graphs are planar, and hence admit product structure [12]. For $n = 4$, we have contact graphs of axis-aligned squares, which are 1-planar, and hence also admit product structure [16].

► **Observation 13.** *For every $n \geq 3$, the class of 1-free intersection graphs of homothetic regular n -gons admits product structure.*

Turning back to the case $\alpha < 1$, i.e., the statement of Theorem 5, we shall use the canonical drawings defined in Section 4. To prove Theorem 5, we show that for appropriate $\alpha < 1$ and all $n > 6$ these canonical drawings are crossing-free and thus the corresponding class of intersection graphs admits product structure by [12].

► **Lemma 14.** *Let $n \geq 3$, $\alpha < 1$, and G be an intersection graph of α -free homothetic regular n -gons with canonical drawing Γ . If two edges $uv, xy \in E$ cross in Γ , then there is a point $p \in \mathbb{R}^2$ that is contained in at least three of S_u, S_v, S_x, S_y .*

Proof. Consider the scaled-down n -gons S'_u and S'_v used to draw the edge uv , as well as S'_x and S'_y used to draw the edge xy . In particular, consider $p_{uv} \in S'_u \cap S'_v$ and $p_{xy} \in S'_x \cap S'_y$. Further, we may assume that the crossing of uv and xy involves the segments $\overline{up_{uv}}$ and $\overline{xp_{xy}}$. Now observe that if $p_{xy} \in S_u$, then $p_{xy} \in S_u \cap S_x \cap S_y$ and we are done. Similarly, we are done if $p_{uv} \in S_x$. In the remainder, we aim to show that one of the two cases applies.

For this, let S''_x be obtained from S'_x by scaling it down at the point p_{xy} until the center of S''_x lies on the segment $\overline{up_{uv}}$ as shown in Figure 13. Since $p_{xy} \in S'_x$, we have $p_{xy} \in S''_x$ and $S''_x \subseteq S'_x \subseteq S_x$. As the center of S''_x lies on the segment $\overline{up_{uv}}$, we obtain that S''_x either contains p_{uv} or is completely contained in $S'_u \subseteq S_u$. In the first case, we have $p_{uv} \in S''_x \subseteq S_x$, while in the second we have $p_{xy} \in S''_x \subseteq S_u$, as desired. ◀

By Lemma 14, crossings in Γ are only possible if three homothetic copies of \diamond_n have a common point. However, this in turn (as long as $n > 6$) forces that one of the three shapes has some significant portion of its area covered by the other two.

► **Lemma 15.** *There is an $\alpha < 1$ such that for every $n > 6$ and every three homothetic copies S_u, S_v, S_w of \diamond_n that have a common point $p \in S_u \cap S_v \cap S_w$, one of the three copies is not α -free.*

Lemma 15, which we formally prove in [23, Lemma 15], together with Lemma 14 implies Theorem 5.

► **Theorem 5.** *There is an $\alpha < 1$ such that for all $n > 6$, the class of intersection graphs of α -free homothetic regular n -gons admits product structure.*

6 k -independent crossing drawings

In this section we again consider intersection graphs of α -free homothetic regular n -gons, and specifically, whether their canonical drawings are k -independent crossing (cf. Definition 6) for a global constant k (that might depend on n). For fixed $n \geq 3$, we let $\alpha \in [0, 1]$ vary. For $\alpha = 1$, one can show that the canonical drawings are planar (or 1-planar for $n = 4$) and in particular 1-independent crossing. For smaller α , we have a richer graph class, which is less likely to have k -independent crossing canonical drawings for any constant k . In fact, we shall prove that $s(n)$ as defined in (1) is the precise tipping point for α until which the canonical drawings for $\mathcal{G}(\diamond_n, \alpha)$ are k -independent crossing. That is, we prove Theorem 8.

6.1 Not k -independent crossing for $\alpha < s(n)$

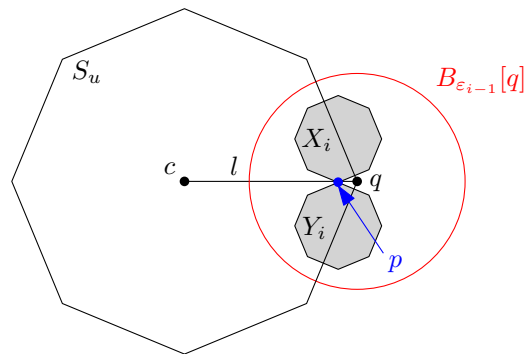
We show that for $\alpha < s(n)$, edges can be crossed by arbitrarily many independent edges.

► **Proposition 16.** *Let $n \geq 3$, $\alpha < s(n)$, and $k \geq 1$. Then there is a collection $\mathcal{C}_k = \{S_v\}_{v \in V}$ of α -free homothetic regular n -gons with intersection graph $G_k = (V, E)$ such that one particular edge $uv \in E$ is crossed in the canonical drawing Γ of G_k by k independent edges.*

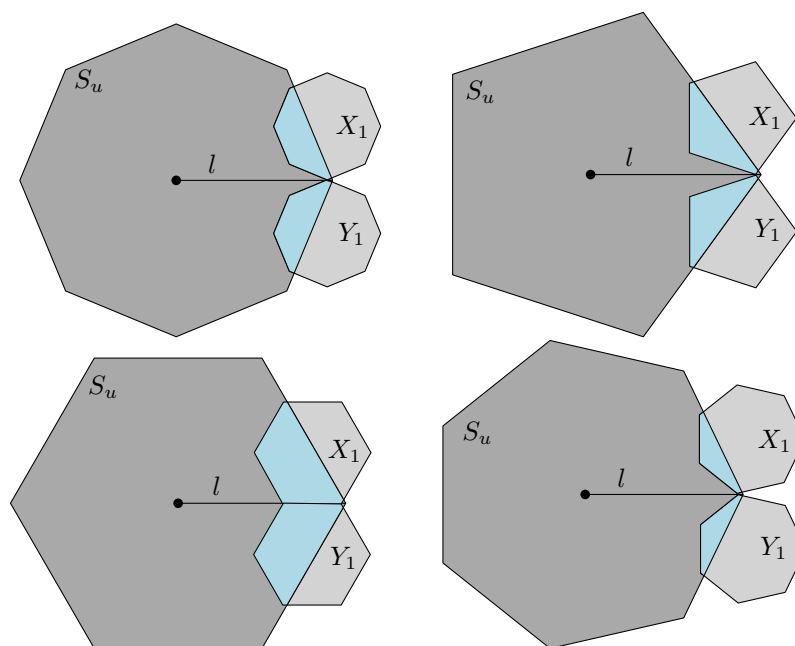
Proof. For the case $n \equiv 0 \pmod{4}$, we rotate the regular n -gon \diamond_n such that it has four corners at its extreme x - and y -coordinates; a top, a bottom, a left, and a right corner.

We start by placing a homothetic regular n -gon $S_u \subseteq \mathbb{R}^2$ with center c and right corner q . We iteratively place n -gons $X_1, Y_1, \dots, X_k, Y_k$ such that after step i , $1 \leq i \leq k$ the n -gons $X_1, Y_1, \dots, X_i, Y_i$ are placed and the i corresponding independent edges x_1y_1, \dots, x_iy_i all cross the line segment l from the center c of S_u to q . Additionally, there are $\varepsilon_0 > \varepsilon_1 > \dots > \varepsilon_k$ such that the ε_i -ball $B_{\varepsilon_i}[q]$ around q is disjoint from X_j and Y_j whenever $j \leq i$. Clearly, all invariants hold before step 1 with $\varepsilon_0 > 0$ being any value small enough such that at least an α -fraction of S_u is not covered by $B_{\varepsilon_0}[q]$.

In step i , we consider the ball $B_{\varepsilon_{i-1}}[q]$ around q that is disjoint from $X_1, Y_1, \dots, X_{i-1}, Y_{i-1}$. Let X_i and Y_i be (very small) homothetic n -gons inside $B_{\varepsilon_{i-1}}[q]$ such that the bottom corner of X_i and the top corner of Y_i coincide in a single point p on $l - q$, as shown in Figure 14. Note that X_i and Y_i have strictly more than $\|\diamond_n^{n/2}\|$ of their area covered by S_u , i.e., strictly less than $\|\diamond_n^{n/2+2}\|$ is free. Moreover, the closer p is to q , the closer is their free fraction is to $\|\diamond_n^{n/2} + 2\| = s(n)$. Now we move X_i, Y_i along l until at least $\alpha < s(n)$ of each of their areas is not covered by S_u and pick $\varepsilon_i > 0$ small enough so that $B_{\varepsilon_i}[q]$ is disjoint from X_i and Y_i . Observe that our invariants hold again.



■ **Figure 14** Placing X_i, Y_i inside the ϵ_{i-1} ball around q .



■ **Figure 15** Placement of X_1, Y_1 in the cases $n \equiv 0, 1, 2, 3 \pmod{4}$.

After step k , we can place a (tiny) homothetic n -gon S_v inside the ball $B_{\epsilon_k}[q]$ such that its left corner coincides with q , which completes the construction for $n \equiv 0 \pmod{4}$.

If $n \not\equiv 1 \pmod{4}$, then the regular n -gon \diamond_n does not have a corner in each of its four extreme directions. Instead, we may have corners or entire sides. Nevertheless, we can assume that \diamond_n has a right corner, and do the same construction as above for any n . However, depending on n , we get a different number of corners of X_i and Y_i inside S_u . For $n \equiv 1 \pmod{4}$ the number of corners of X_i inside of S_u is $\lceil n/2 \rceil$ and thus roughly $\|\diamond_n^{\lceil n/2 \rceil + 1}\|$ of the area of X_i is not covered by S_u . For $n \equiv 2 \pmod{4}$ the number of corners is $n/2 + 1$ and roughly $\|\diamond_n^{n/2 + 1}\| = \frac{1}{2}$ is free. Lastly, for $n \equiv 3 \pmod{4}$ the number of corners of X_i inside of S_u is $\lfloor n/2 \rfloor$ and thus roughly $\|\diamond_n^{\lfloor n/2 \rfloor + 2}\|$ of the area of X_i is not covered by S_u . An example of the placement of X_1, Y_1 in the various cases is given in Figure 15. ◀

6.2 k -independent crossing for $\alpha \geq s(n)$

We show that for every $n \geq 3$, $\alpha \geq s(n)$, and for every collection of α -free homothetic regular n -gons, the canonical drawing Γ of the corresponding intersection graph G is k -independent crossing (cf. Definition 6) for a global constant k that depends only on n .

The reader might recall Lemma 15 stating that for large enough α , no three α -free n -gons have a common point. In [23, Lemma 18], we use a similar strategy to show that for any large enough α (in this case $\alpha \geq \frac{1}{2}$), the number of n -gons containing a given point p is bounded.

► **Lemma 17.** *For a point $p \in \mathbb{R}^2$, $n \geq 5$, and $\alpha \geq \frac{1}{2}$, there are at most 13 regular n -gons $X \in \mathcal{C}$ with $p \in X$.*

The next lemma crucially exploits that $s(n)$ is chosen as the tipping point whether or not two regular n -gons can meet in a third regular n -gon without containing a corner of the latter.

► **Lemma 18.** *Let X be a homothetic copy of $\hat{\diamond}_n$. Let c be the center, Q be the set of corners, and p be any fixed point on the boundary of X . Further, let Y be another homothetic copy of $\hat{\diamond}_n$ such that $Y \cap \overline{cp} \neq \emptyset$. Then at least one of the following holds.*

- $\frac{\|Y-X\|}{\|Y\|} < s(n)$, i.e., less than an $s(n)$ -fraction of Y is not covered by X , or
- $Y \cap (Q \cup \{p\}) \neq \emptyset$, i.e., Y contains point p or a corner of X .

Proof. As we are done otherwise, we assume that Y contains no corner of X and that Y is not completely contained in X . Then Y intersects exactly one side s of X . Let h be the halfplane supported by s that contains X . Then $Y - X = Y - h$, and hence $\|Y - X\|/\|Y\| \leq \|\hat{\diamond}_n^m\|$ where m is the number of corners of Y that lie outside of X or on the boundary of X . For convenience let

$$m^* = \begin{cases} n/2 + 2 & \text{if } n \equiv 0 \pmod{4} \\ \lceil n/2 \rceil + 1 & \text{if } n \equiv 1 \pmod{4} \\ n/2 + 1 & \text{if } n \equiv 2 \pmod{4} \\ \lceil n/2 \rceil + 2 & \text{if } n \equiv 3 \pmod{4}. \end{cases}$$

I.e., $s(n) = \|\hat{\diamond}_n^{m^*}\|$, and we are done if $m < m^*$. So assume that Y has $m \geq m^*$ corners outside or on the boundary of X , and note that this implies that the center of Y lies outside or on the boundary of X . Also note that the radius of Y is smaller than the side length of X . Together, it follows that the point p either lies on the side s of X that is intersected by Y or on an adjacent side s' of X . Also we assume that $p \notin Y$, as otherwise we are done.

Next, we shall argue that we may assume that p is a corner of X . We actually only need p to satisfy the condition that Y contains neither a corner of X nor p but intersects a side of X adjacent to p and the line segment \overline{cp} . In case that p lies on a side $s' \neq s$ adjacent to s , then the corner p' of X where s and s' meet fulfills the same condition. In the other case that p lies on the side s that Y intersects, we can simultaneously move p and Y parallel to s until p coincides with the corner p' of s on the far end of Y (recall that $p \notin Y$). Seen from Y , this only changes the angle of the line segment \overline{cp} , making it lean even more towards Y . Again, the new situation fulfills the same condition.

Finally, we face the situation of p being a corner of X , while Y contains no corner of X but intersects a side s of X incident to p and intersects the line segment \overline{cp} . Then the portion of $\|Y\|$ outside X is, by definition, strictly less than $s(n)$, which concludes the proof. ◀

With Lemmas 17 and 18 at hand, we can now prove the following.

► **Proposition 19.** *Let $n \geq 3$ and $\alpha \geq s(n)$ be fixed. Then for any collection $\mathcal{C} = \{S_v\}_{v \in V}$ of α -free homothetic regular n -gons with intersection graph $G = (V, E)$, the canonical drawing Γ of G is $26(n+1)$ -independent crossing.*

Proof. For $n \in \{3, 4\}$, we have $s(n) = 1$. Hence, Γ is planar for $n = 3$ and 1-planar for $n = 4$, which is more than enough. For $n \geq 5$, let $uv \in E$ be any fixed edge in G . Our task is to bound the number of independent edges in G that cross uv in Γ .

Recall from Observation 12, that $S'_u \subseteq S_u$ and $S'_v \subseteq S_v$ are interiorly disjoint but touching homothetic n -gons, and edge uv is drawn as a polyline from u (ε -close to the center of S_u) to v (ε -close to the center of S_v) with one bend point ε -close to $p_{uv} \in S'_u \cap S'_v$. Actually, we may assume without loss of generality that u is the center of S'_u , v is the center of S'_v , and edge uv bends exactly at p_{uv} . Then uv consists of two line segments $\overline{up_{uv}}$ and $\overline{p_{uv}v}$, and it is enough to bound the number of edges that cross one of these line segments in Γ , say $\overline{up_{uv}}$.

For every edge xy that crosses $\overline{up_{uv}}$ in Γ , we have $S_x \cap \overline{up_{uv}} \neq \emptyset$ or $S_y \cap \overline{up_{uv}} \neq \emptyset$, or both. Let $A = \{y \in V \mid S_y \cap \overline{up_{uv}} \neq \emptyset\}$ be the subset of vertices of G whose corresponding sets intersect $\overline{up_{uv}}$. Crucially, all edges of G that cross uv in Γ along the line segment $\overline{up_{uv}}$ have an endpoint in A . Since $S'_u \subseteq S_u$, we have

$$\frac{\|S_y - S'_u\|}{\|S_y\|} \geq \frac{\|S_y - S_u\|}{\|S_y\|} \geq \alpha \geq s(n)$$

for each $y \in A$. Hence, by Lemma 18, each such S_y must contain the point p_{uv} or (at least) one of the n corners of S'_u . Lemma 17 says that at most 13 such S_y can contain the same point, and thus $|A| \leq 13 \cdot (n+1)$. In other words, at most $13(n+1)$ independent edges of G cross uv in Γ along the line segment $\overline{up_{uv}}$. Symmetrically, at most $13(n+1)$ independent edges cross $\overline{p_{uv}v}$, and thus Γ is k -independent crossing for $k = 26(n+1)$. ◀

Finally, Propositions 16 and 19 together prove Theorem 8.

► **Theorem 8.** *Let $n \geq 3$, $\alpha \in [0, 1]$, and $s(n)$ be defined as in (1). Then there exists a constant $k = k(n)$ such that the canonical drawings of all graphs in $\mathcal{G}(\diamond_n, \alpha)$ are k -independent crossing if and only if $\alpha \geq s(n)$.*

7 Conclusion

It remains an intriguing problem to determine for the regular n -gon \diamond_n the threshold $\alpha^*(\diamond_n)$ such that the class $\mathcal{G}(\diamond_n, \alpha)$ of intersection graphs of α -free homothetic copies of \diamond_n admits product structure for $\alpha > \alpha^*(\diamond_n)$ and no product structure for $\alpha < \alpha^*(\diamond_n)$.

With $s(n)$, as defined in (1), we determined the exact threshold for α such that for $\alpha < s(n)$ arbitrarily many independent edges can cross a single edge in canonical drawings. While this is exactly the crucial ingredient, we can not construct the nested grids for $\mathcal{G}(\diamond_n, \alpha)$ with $\alpha \approx s(n)$, unless $n \in \{4, 6\}$. Still, we suspect an alternative construction to work.

► **Conjecture 20.** *For every $\alpha < s(n)$, the class of intersection graphs of α -free homothetic regular n -gons does not have product structure.*

On the other hand, for $\alpha \geq s(n)$, the canonical drawings of $\mathcal{G}(\diamond_n, \alpha)$ are k -independent crossing. We prove this for $k = 26(n+1)$ (cf. Proposition 19) but suspect that a constant k independent of n should suffice. As already conjectured in the introduction, we believe that graph classes with k -independent crossing drawings have product structure.

► **Conjecture 21.** *The class of k -independent crossing graphs admits product structure.*

By Theorem 8 this would imply the following conjecture matching Conjecture 20.

► **Conjecture 22.** *For every $\alpha \geq s(n)$, the class of intersection graphs of α -free homothetic regular n -gons has product structure.*

This seems reasonable since similar beyond-planar graph classes, such as k -planar graphs [16], fan-planar graphs and k -fan-bundle graphs [20], have been shown to have product structure. In particular, Hickingbotham and Wood [20] show that if all graphs in \mathcal{G} are r -shallow minors of $H \boxtimes K_l$ with $r, l, \text{tw}(H) \in O(1)$, then \mathcal{G} has product structure. For example, they show this to be true for fan-planar graphs.

References

- 1 Michael A. Bekos, Giordano Da Lozzo, Petr Hliněný, and Michael Kaufmann. Graph Product Structure for h -Framed Graphs. In Sang Won Bae and Heejin Park, editors, *33rd International Symposium on Algorithms and Computation (ISAAC 2022)*, volume 248 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 23:1–23:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ISAAC.2022.23.
- 2 Marthe Bonamy, Cyril Gavoille, and Michał Pilipczuk. Shorter Labeling Schemes for Planar Graphs. *SIAM Journal on Discrete Mathematics*, 36(3):2082–2099, September 2022. Publisher: Society for Industrial and Applied Mathematics. doi:10.1137/20M1330464.
- 3 Édouard Bonnet, O-joung Kwon, and David R. Wood. Reduced bandwidth: a qualitative strengthening of twin-width in minor-closed classes (and beyond), 2022. arXiv:2202.11858, doi:10.48550/arXiv.2202.11858.
- 4 Prosenjit Bose, Vida Dujmović, Mehrnoosh Javarsineh, Pat Morin, and David R. Wood. Separating layered treewidth and row treewidth. *Discrete Mathematics & Theoretical Computer Science*, vol. 24, no. 1(Graph Theory), May 2022. doi:10.46298/dmtcs.7458.
- 5 Prosenjit Bose, Vida Dujmović, Mehrnoosh Javarsineh, and Pat Morin. Asymptotically optimal vertex ranking of planar graphs, 2020. arXiv:2007.06455, doi:10.48550/arXiv.2007.06455.
- 6 Franz J. Brandenburg. On fan-crossing graphs. *Theoretical Computer Science*, 841:39–49, 2020. doi:10.1016/j.tcs.2020.07.002.
- 7 Michał Dębski, Piotr Micek, Felix Schröder, and Stefan Felsner. Improved bounds for centered colorings. *Advances in Combinatorics*, August 2021. doi:10.19086/aic.27351.
- 8 Marc Distel, Robert Hickingbotham, Michał T. Seweryn, and David R. Wood. Powers of planar graphs, product structure, and blocking partitions. *European Conference on Combinatorics, Graph Theory and Applications*, pages 355–361, August 2023. doi:10.5817/CZ.MUNI.EUROCOMB23-049.
- 9 Vida Dujmović, Louis Esperet, Cyril Gavoille, Gwenaël Joret, Piotr Micek, and Pat Morin. Adjacency labelling for planar graphs (and beyond). *J. ACM*, 68(6), October 2021. doi:10.1145/3477542.
- 10 Vida Dujmović, Louis Esperet, Gwenaël Joret, Bartosz Walczak, and David Wood. Planar graphs have bounded nonrepetitive chromatic number. *Advances in Combinatorics*, March 2020. doi:10.19086/aic.12100.
- 11 Vida Dujmović, Louis Esperet, Pat Morin, and David R. Wood. Proof of the clustered hadwiger conjecture. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1921–1930, 2023. doi:10.1109/FOCS57990.2023.00116.
- 12 Vida Dujmović, Gwenaël Joret, Piotr Micek, Pat Morin, Torsten Ueckerdt, and David R. Wood. Planar graphs have bounded queue-number. *J. ACM*, 67(4), August 2020. doi:10.1145/3385731.
- 13 Vida Dujmović, Pat Morin, and Saeed Odak. Odd colourings of graph products, 2022. doi:10.48550/arXiv.2202.12882.

- 14 Vida Dujmović, Louis Esperet, Pat Morin, Bartosz Walczak, and David R. Wood. Clustered 3-colouring graphs of bounded degree. *Combinatorics, Probability and Computing*, 31(1):123–135, 2022. doi:10.1017/S0963548321000213.
- 15 Vida Dujmović, Pat Morin, and David R. Wood. Layered separators in minor-closed graph classes with applications. *Journal of Combinatorial Theory, Series B*, 127:111–147, 2017. doi:10.1016/j.jctb.2017.05.006.
- 16 Vida Dujmović, Pat Morin, and David R. Wood. Graph product structure for non-minor-closed classes. *Journal of Combinatorial Theory, Series B*, 162:34–67, 2023. doi:10.1016/j.jctb.2023.03.004.
- 17 Zdeněk Dvořák, Tony Huynh, Gwenaél Joret, Chun-Hung Liu, and David R. Wood. Notes on graph product structure theory. In Jan de Gier, Cheryl E. Praeger, and Terence Tao, editors, *2019-20 MATRIX Annals*, pages 513–533, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-62497-2_32.
- 18 Zdeněk Dvořák, Daniel Gonçalves, Abhiruk Lahiri, Jane Tan, and Torsten Ueckerdt. On comparable box dimension. In Xavier Goaoc and Michael Kerber, editors, *38th International Symposium on Computational Geometry (SoCG 2022)*, volume 224 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 38:1–38:14, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SoCG.2022.38.
- 19 Louis Esperet, Gwenaél Joret, and Pat Morin. Sparse universal graphs for planarity. *Journal of the London Mathematical Society*, 108(4):1333–1357, 2023. doi:10.1112/jlms.12781.
- 20 Robert Hickingbotham and David R. Wood. Shallow minors, graph products, and beyond-planar graphs. *SIAM Journal on Discrete Mathematics*, 38(1):1057–1089, 2024. doi:10.1137/22M1540296.
- 21 Gwenaél Joret and Clément Rambaud. Neighborhood complexity of planar graphs, 2023. doi:10.48550/arXiv.2302.12633.
- 22 Daniel Král, Kristýna Pekárková, and Kenny Štorgel. Twin-width of graphs on surfaces, 2024. doi:10.48550/arXiv.2307.05811.
- 23 Laura Merker, Lena Scherzer, Samuel Schneider, and Torsten Ueckerdt. Intersection graphs with and without product structure, 2024. doi:10.48550/arXiv.2409.01732.
- 24 Neil Robertson and Paul D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, 1986. doi:10.1016/0196-6774(86)90023-4.
- 25 Torsten Ueckerdt, David Wood, and Wendy Yi. An improved planar graph product structure theorem. *The Electronic Journal of Combinatorics*, pages P2–51, 2022. doi:10.37236/10614.


Upward Pointset Embeddings of Planar st -Graphs

Carlos Alegría 

Roma Tre University, Rome, Italy

Giordano Da Lozzo 

Roma Tre University, Rome, Italy

Giuseppe Di Battista 

Roma Tre University, Rome, Italy

Fabrizio Grosso 

Roma Tre University, Rome, Italy

Susanna Caroppo 

Roma Tre University, Rome, Italy

Marco D’Elia 

Roma Tre University, Rome, Italy

Fabrizio Frati 

Roma Tre University, Rome, Italy

Maurizio Patrignani 

Roma Tre University, Rome, Italy

Abstract

We study upward pointset embeddings (UPSEs) of planar st -graphs. Let G be a planar st -graph and let $S \subset \mathbb{R}^2$ be a pointset with $|S| = |V(G)|$. An UPSE of G on S is an upward planar straight-line drawing of G that maps the vertices of G to the points of S . We consider both the problem of testing the existence of an UPSE of G on S (UPSE TESTING) and the problem of enumerating all UPSEs of G on S . We prove that UPSE TESTING is NP-complete even for st -graphs that consist of a set of directed st -paths sharing only s and t . On the other hand, for n -vertex planar st -graphs whose maximum st -cutset has size k , we prove that it is possible to solve UPSE TESTING in $\mathcal{O}(n^{4k})$ time with $\mathcal{O}(n^{3k})$ space, and to enumerate all UPSEs of G on S with $\mathcal{O}(n)$ worst-case delay, using $\mathcal{O}(kn^{4k} \log n)$ space, after $\mathcal{O}(kn^{4k} \log n)$ set-up time. Moreover, for an n -vertex st -graph whose underlying graph is a cycle, we provide a necessary and sufficient condition for the existence of an UPSE on a given pointset, which can be tested in $\mathcal{O}(n \log n)$ time. Related to this result, we give an algorithm that, for a set S of n points, enumerates all the non-crossing monotone Hamiltonian cycles on S with $\mathcal{O}(n)$ worst-case delay, using $\mathcal{O}(n^2)$ space, after $\mathcal{O}(n^2)$ set-up time.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry; Theory of computation \rightarrow Design and analysis of algorithms; Mathematics of computing \rightarrow Graph algorithms

Keywords and phrases Upward pointset embeddings, planar st -graphs, st -cutset, non-crossing monotone Hamiltonian cycles, graph drawing enumeration

Digital Object Identifier 10.4230/LIPIcs.GD.2024.24

Related Version Full Version: <https://arxiv.org/abs/2408.17369> [1]

Funding This research was supported, in part, by MUR of Italy (PRIN Project no. 2022ME9Z78 – NextGRAAL and PRIN Project no. 2022TS4Y3N – EXPAND).

1 Introduction

Given an n -vertex upward planar graph G and a set S of n points in the plane, an *upward pointset embedding* (UPSE) of G on S is an upward planar drawing of G where the vertices are mapped to the points of S and the edges are represented as straight-line segments. The UPWARD POINTSET EMBEDDABILITY TESTING PROBLEM (UPSE TESTING) asks whether an upward planar graph G has an UPSE on a given pointset S .

Pointset embedding problems are classic challenges in Graph Drawing and have been considered for both undirected and directed graphs. For an undirected graph, a *pointset embedding* (PSE) has the same definition of an UPSE, except that the drawing must be planar, rather than upward planar. The POINTSET EMBEDDABILITY TESTING PROBLEM (PSE TESTING) asks whether a planar graph has a PSE on a given pointset S . Pointset embeddings have been studied by several authors. It is known that a graph admits a PSE on



© Carlos Alegría, Susanna Caroppo, Giordano Da Lozzo, Marco D’Elia, Giuseppe Di Battista, Fabrizio Frati, Fabrizio Grosso, and Maurizio Patrignani; licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 24; pp. 24:1–24:18



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

every pointset in general position if and only if it is outerplanar [12, 26]; such a PSE can be constructed efficiently [7, 8, 9, 10]. PSE TESTING is, in general, NP-complete [11], however it is polynomial-time solvable if the input graph is a planar 3-tree [35]. More in general, a polynomial-time algorithm for PSE TESTING exists if the input graph has a fixed embedding, bounded treewidth, and bounded face size [5]. PSE becomes NP-complete if one of the latter two conditions does not hold. PSEs have been studied also for dynamic graphs [15, 16].

The literature on UPSEs is not any less rich than the one on PSEs. From a combinatorial perspective, the directed graphs with an UPSE on a one-sided convex pointset have been characterized [6, 27]; all directed trees are among them. Conversely, there exist directed trees that admit no UPSE on certain convex pointsets [6]. Directed graphs that admit an UPSE on any convex pointset, but not on any pointset in general position, exist [3]. It is still unknown whether every digraph whose underlying graph is a path admits an UPSE on every pointset in general position, see, e.g., [33]. UPSEs where bends along the edges are allowed have been studied in [6, 24, 25, 30, 31]. From the computational complexity point of view [28, 29], it is known that UPSE TESTING is NP-hard, even for planar st -graphs and 2-convex pointsets, and that UPSE TESTING can be solved in polynomial time if the given pointset is convex.

Our contributions. We tackle UPSE TESTING for planar st -graphs. Let G be an n -vertex planar st -graph and S be a set of n points in the plane. We adopt the common assumption in the context of upward pointset embeddability, see e.g. [3, 6, 28, 29], that no two points of S lie on the same horizontal line. Our results are the following:

- In Section 3, we show that UPSE TESTING is NP-hard even if G consists of a set of internally-disjoint st -paths (Theorem 1). A similar proof shows that UPSE TESTING is NP-hard for directed trees consisting of a set of directed root-to-leaf paths (Theorem 2). This answers an open question from [4] and strengthens a result therein, which shows NP-hardness for directed trees with multiple sources and with a prescribed mapping for a vertex.
- In Section 4, we show that UPSE TESTING can be solved in $\mathcal{O}(n^{4k})$ time and $\mathcal{O}(n^{3k})$ space, where k is the size of the largest st -cutset of G (Theorem 7). This parameter measures the “fatness” of the digraph and coincides with the length of the longest directed path in the dual [18]. By leveraging on the techniques developed for the testing algorithm, we also show how to enumerate all UPSEs of G on S with $\mathcal{O}(n)$ worst-case delay, using $\mathcal{O}(kn^{4k} \log n)$ space, after $\mathcal{O}(kn^{4k} \log n)$ set-up time (Theorem 8).
- In Section 5, we provide a simple characterization of the pointsets in general position that allow for an UPSE of G , if G consists of two (internally-disjoint) st -paths. Based on that, we provide an $\mathcal{O}(n \log n)$ testing algorithm for this case (Theorem 9).
- Finally, in Section 6, inspired by the fact that an UPSE of a planar st -graph composed of two st -paths defines a non-crossing monotone Hamiltonian cycle on S , we provide an algorithm that enumerates all the non-crossing monotone Hamiltonian cycles on a given pointset with $\mathcal{O}(n)$ worst-case delay, and $\mathcal{O}(n^2)$ space usage and set-up time (Theorem 10).

Concerning our last result, we remark that a large body of research has considered problems related to enumerating and counting non-crossing structures on a given pointset [2, 13, 22, 32, 36]. Despite this effort, the complexity of counting the non-crossing Hamiltonian cycles, often called *polygonalizations*, remains open [20, 32, 34]. However, it is possible to enumerate all polygonalizations of a given pointset in singly-exponential time [37, 38]. Recently, an algorithm has been shown [21] to enumerate all polygonalizations of a given

pointset in time polynomial in the output size, i.e., bounded by a polynomial in the number of solutions. However, an enumeration algorithm with polynomial (in the input size) delay is not yet known, neither in the worst-case nor in the average-case acceptance. Our enumeration algorithm achieves this goal for the case of monotone polygonalizations.

Because of space limitations, some proofs are sketched or omitted. They can be found in the full version of the paper [1].

2 Preliminaries

We use standard terminology in graph theory [19] and graph drawing [17]. For an integer $k > 0$, let $[k]$ denote the set $\{1, \dots, k\}$. A *permutation with repetitions* of k elements from U is an arrangement of any k elements of a set U , where repetitions are allowed.

We denote by $\mathcal{CH}(S)$ the convex hull of a set S of points and by $\mathcal{B}(S)$ its boundary. The points of S with lowest and highest y -coordinates are the *south* and *north extreme* of S , respectively; together, they are the *extremes* of S . The *left envelope* of S is the subpath $\mathcal{E}_L(S)$ of $\mathcal{B}(S)$ to the left of the line through the extremes of S (including the extremes of S). The *right envelope* $\mathcal{E}_R(S)$ of S is defined analogously. We denote the subset of S in $\mathcal{E}_L(S)$ and $\mathcal{E}_R(S)$ by $\mathcal{H}_L(S)$ and $\mathcal{H}_R(S)$, respectively. A ray is *upward* if it passes through points whose y -coordinate is larger than the one of the starting point of the ray.

A polyline (p_1, \dots, p_k) is *y -monotone* if $y(p_i) < y(p_{i+1})$, for $i = 1, \dots, k - 1$. A *monotone path* on a pointset S is a y -monotone polyline (p_1, \dots, p_k) such that the points p_1, \dots, p_k belong to S . A *monotone cycle* on S consists of two monotone paths on S that share their endpoints. A *monotone Hamiltonian cycle* (p_1, \dots, p_k, p_1) on S is a monotone cycle on S such that each point of S is a point p_i (and vice versa).

A path (v_1, \dots, v_k) is *directed* if, for $i = 1, \dots, k - 1$, the edge (v_i, v_{i+1}) is directed from v_i to v_{i+1} ; the vertices v_2, \dots, v_{k-1} are *internal*. A *planar st -graph* is an acyclic digraph with one source s and one sink t , which admits a planar embedding in which s and t are on the boundary of the outer face. An *st -path* in a planar st -graph is a directed path from s to t . A drawing of a directed graph is *straight-line* if each edge is represented by a straight-line segment, it is *planar* if no two edges cross, and it is *upward* if every edge is represented by a Jordan arc monotonically increasing in the y -direction from the tail to the head. A digraph that admits an upward planar drawing is an *upward planar graph*. Every upward planar graph admits an upward planar straight-line drawing [18]. An *Upward Pointset Embedding* (*UPSE*, for short) of an upward planar graph G on a pointset S is an upward planar straight-line drawing of G that maps each vertex of G to a point in S . In this paper, we study the following problem.

UPWARD POINTSET EMBEDDABILITY TESTING PROBLEM (UPSE TESTING)

Input: An n -vertex upward planar graph G and a pointset $S \subset \mathbb{R}^2$ with $|S| = n$.
Question: Does there exist an UPSE of G on S ?

In the remainder, we assume that not all points in S lie on the same line, as otherwise there is an UPSE if and only if the input is a directed path. Recall that no two points in S have the same y -coordinate. Unless otherwise specified, we do not require points to be in *general position*, i.e., we allow three or more points to lie on the same line.

3 NP-Completeness of UPSE Testing

In this section we prove that UPSE TESTING is NP-complete. The membership in NP is obvious, as one can non-deterministically assign the vertices of the input graph G to the points of the input pointset S and then test in polynomial time whether the assignment results in an upward planar straight-line drawing of G . In the remainder of the section, we prove that UPSE TESTING is NP-hard even in very restricted cases.

We first show a reduction from 3-PARTITION to instances of UPSE in which the input is a planar st -graph composed of a set of internally-disjoint st -paths. An instance of 3-PARTITION consists of a set $A = \{a_1, \dots, a_{3b}\}$ of $3b$ integers, where $\sum_{i=1}^{3b} a_i = bB$ and $B/4 \leq a_i \leq B/2$, for $i = 1, \dots, 3b$. The 3-PARTITION problem asks whether A can be partitioned into b subsets A_1, \dots, A_b , each with three integers, so that the sum of the integers in each set A_i is B . Since 3-PARTITION is strongly NP-hard [23], we may assume that B is bounded by a polynomial function of b . Given an instance A of 3-PARTITION, we show how to construct in polynomial time, precisely $\mathcal{O}(b \cdot B)$, an equivalent instance (G, S) of UPSE TESTING.

The n -vertex planar st -graph G is composed of $4b + 1$ internally-disjoint st -paths. Namely, for $i = 1, \dots, 3b$, we have that G contains an a_i -path, i.e., a path with a_i internal vertices, and $b + 1$ additional k -paths, where $k = 2B + 1$. Note that $n = 2 + (b + 1)k + \sum_{i=1}^{3b} a_i = 2 + (b + 1)k + bB$.

The points of S lie on the plane as follows (see Figure 1a):

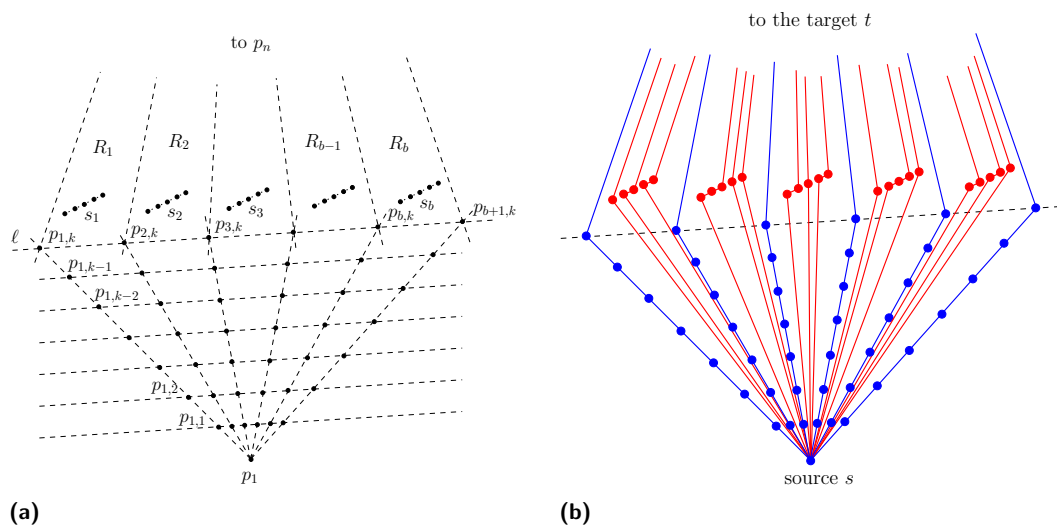
- p_1 is the origin, with coordinates $(0, 0)$.
- Consider $b + 1$ upward rays $\rho_1, \dots, \rho_{b+1}$, whose starting point is p_1 , such that the angles $\alpha_1, \dots, \alpha_{b+1}$ that they respectively form with the x -axis satisfy $3\pi/4 > \alpha_1 > \dots > \alpha_{b+1} > \pi/4$. Let ℓ be a line intersecting all the rays, with a positive slope smaller than $\pi/4$. For $j = 1, \dots, b + 1$, place k points $p_{j,1}, \dots, p_{j,k}$ (in this order from bottom to top) along ρ_j , so that $p_{j,k}$ is on ℓ and no two points share the same y -coordinate. Observe that $p_{b+1,k}$ is the highest point placed so far.
- Place p_n at coordinates $(0, 10 \cdot y(p_{b+1,k}))$.
- Finally, for $j = 1, \dots, b$, place B points along a non-horizontal segment s_j in such a way that: (i) s_j is entirely contained in the triangle with vertices $p_{j,k}, p_{j+1,k}$, and p_n , (ii) for any point p on s_j , the polygonal line $\overline{p_1 p} \cup \overline{p p_n}$ is contained in the region R_j delimited by the polygon $\overline{p_1 p_{j,k}} \cup \overline{p_{j,k} p_n} \cup \overline{p_n p_{j+1,k}} \cup \overline{p_{j+1,k} p_1}$, and (iii) no two distinct points on any two segments s_i and s_j share the same y -coordinate.

Note that S has $2 + (b + 1)k + bB = n$ points. The described reduction is the main ingredient for the proof of the following theorem.

► **Theorem 1.** UPSE TESTING is NP-hard even for planar st -graphs consisting of a set of directed internally-disjoint st -paths.

Proof. First, the construction of G and S takes polynomial time. In particular, the coordinates of the points in S can be encoded with a polylogarithmic number of bits. In order to prove the NP-hardness, it remains to show that the constructed instance (G, S) of UPSE TESTING is equivalent to the given instance A of 3-PARTITION. Refer to Figure 1b.

Suppose first that A is a positive instance of 3-PARTITION, that is, there exist sets A_1, \dots, A_b , each with three integers, such that the sum of the integers in each set A_j is B . We construct an UPSE of G on S as follows. We map s to p_1 and t to p_n . For $j = 1, \dots, b + 1$, we map the k internal vertices of a k -path to the points $p_{j,1}, \dots, p_{j,k}$, so that vertices that come first in the directed path have smaller y -coordinates. Furthermore, for $j = 1, \dots, b$, let $A_j = \{a_{j_1}, a_{j_2}, a_{j_3}\}$. Then we map the a_{j_1} internal vertices of an a_{j_1} -path, the a_{j_2} internal vertices of an a_{j_2} -path, and the a_{j_3} internal vertices of an a_{j_3} -path to the set of B points in the triangle with vertices $p_{j,k}, p_{j+1,k}$, and p_n , so that vertices that come first in the directed



■ **Figure 1** Illustration for the proof of Theorem 1. (a) The pointset S . (b) The UPSE of G on S .

paths have smaller y -coordinates and so that the internal vertices of the a_{j_1} -path have smaller y -coordinates than the internal vertices of the a_{j_2} -path, which have smaller y -coordinates than the internal vertices of the a_{j_3} -path. This results in an UPSE of G on S .

Suppose next that (G, S) is a positive instance of UPSE TESTING. Trivially, in any UPSE of G on S , we have that s is drawn on p_1 and t on p_n . Consider the points $p_{1,1}, \dots, p_{b+1,1}$. The paths using them use all the $(b+1)k$ points $p_{j,i}$, with $j = 1, \dots, b+1$ and $i = 1, \dots, k$. Indeed, if these paths left one of such points unused, no other path could reach it from s without passing through $p_{1,1}, \dots, p_{b+1,1}$, because of the collinearity of the points along the rays $\rho_1, \dots, \rho_{b+1}$. Hence, there are at most $b+1$ paths that use $(b+1)k$ points. Since the a_i -paths have less than k internal vertices, these $b+1$ paths must all be k -paths. Let P_1, \dots, P_{b+1} be the left-to-right order of the k -paths around p_1 . For $j = 1, \dots, b+1$, path P_j uses all points $p_{j,i}$ on ρ_j , as if P_j used a point $p_{h,i}$ with $h > j$, then two among P_j, \dots, P_{b+1} would cross each other. Note that, after using $p_{j,k}$, path P_j ends with the segment $\overline{p_{j,k}p_n}$. Hence, for $j = 1, \dots, b$, the region R_j is bounded by P_j and P_{j+1} ; recall that R_j contains the segment s_j . The a_i -paths must then use the points on s_1, \dots, s_b . Since $B/4 < a_i < B/2$, no two a_i -paths can use all the B points in one region and no four a_i -paths can lie in the same region. Hence, three a_i -paths use the B points in each region, and this provides a solution to the given 3-PARTITION instance. ◀

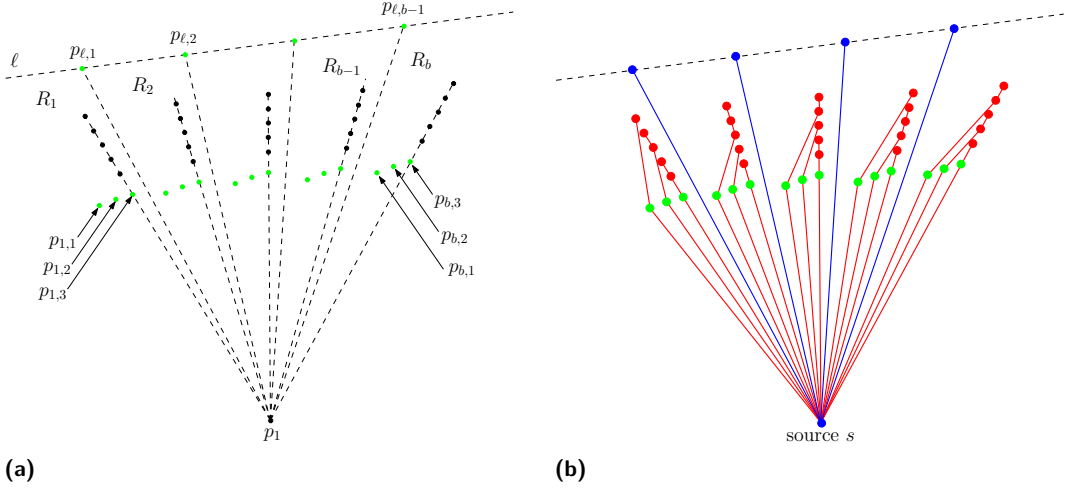
A similar reduction, illustrated in Figures 2a and 2b, allows us to state the following.

▶ **Theorem 2.** UPSE TESTING is NP-hard even for directed trees consisting of a set of directed root-to-leaf paths.

4 Testing and Enumeration Algorithms for Planar st -Graphs with Maximum st -Cutset of Bounded Size

An *st-cutset* of a planar st -graph $G = (V, E)$ is a subset W of E such that:

- removing W from E results in a graph consisting of exactly two connected components C_s and C_t ,
- s belongs to C_s and t belongs to C_t , and
- any edge in W has its tail in C_s and its head in C_t .



■ **Figure 2** Illustration for the proof of Theorem 2. (a) The pointset S . The points of S visible from p_1 (green points) are as many as the children of the root of the tree T . (b) The UPSE of T on S corresponding to a solution to the original instance 3-PARTITION (red vertices).

In this section, we consider instances (G, S) where G is a planar st -graph, whose maximum st -cutset has bounded size k . In Theorem 7, we show that UPSE TESTING can be solved in polynomial time for such instances (G, S) . Moreover, in Theorem 8, we show how to enumerate all UPSEs of (G, S) with linear delay. The algorithm for Theorem 7 is based on a dynamic programming approach. It exploits the property that, for an st -cutset W defining the connected components C_s and C_t , the extensibility of an UPSE Γ' of $C_s \cup W$ on a subset S' of S to an UPSE of G on S only depends on the drawing of the edges of W , and not on the embedding of the remaining vertices of C_s , provided that in Γ' there exists an horizontal line that crosses all the edges of W . The algorithm for Theorem 8 leverages a variation of the dynamic programming table computed by the former algorithm to efficiently test the extensibility of an UPSE of $C_s \cup W$ (in which there exists a horizontal line that crosses all the edges of W) on a subset S' of S to an UPSE of G on S .

The proofs of Theorems 7 and 8 exploit two dynamic programming tables T and Q defined as follows. Each entry of T and Q is indexed by a *key* that consists of a set of $h \leq k$ triplets $\langle e_i, p_i, q_i \rangle$, where, for any $i = 1, \dots, h$, it holds that $e_i \in E(G)$, $p_i, q_i \in S$, and $y(p_i) < y(q_i)$. Moreover, each key $\chi = \bigcup_{i=1}^h \langle e_i, p_i, q_i \rangle$ satisfies the following constraints:

- the set $E(\chi) = \bigcup_{i=1}^h e_i$ is an st -cutset of G and, for every i, j , with $i \neq j$, it holds true that $e_i \neq e_j$ (that is, $|E(\chi)| = h$);
- for every i, j , with $i \neq j$, it holds true that $p_i = p_j$ (resp. that $q_i = q_j$) if and only if e_i and e_j have the same tail (resp. the same head); and
- let ℓ_χ be the horizontal line passing through the tail with largest y -coordinate among the edges in $E(\chi)$, i.e., $\ell_\chi := y = y(p_i)$ s.t. $y(p_j) \leq y(p_i)$ for any $\langle e_j, p_j, q_j \rangle \in \chi$; then ℓ_χ intersects all the segments $\overline{p_j q_j}$, possibly at an endpoint.

For brevity, we sometimes say that the edge e_i has its tail (resp. its head) *mapped by* χ on p_i (resp. on q_i). We also say that e_i is *drawn as in* χ if its drawing is the segment $\overline{p_i q_i}$.

Let $\chi = \bigcup_{i=1}^h \langle e_i, p_i, q_i \rangle$ be a key of T and of Q ; see Figure 3a. Let G_χ be the connected component containing s of the graph obtained from G by removing the edge set $E(\chi)$.

The entry $T[\chi]$ contains a Boolean value such that $T[\chi] = \text{True}$ if and only if there exists an UPSE of $G_\chi^+ = G_\chi \cup E(\chi)$ on some subset $S' \subset S$ with $|S'| = |V(G_\chi^+)|$ such that:

- the lowest point p_s of S belongs to S' and s lies on it, and
- for $i = 1, \dots, h$, the edge e_i is drawn as in χ .

If $T[\chi] = \text{False}$, the entry $Q[\chi]$ contains the empty set \emptyset . If $T[\chi] = \text{True}$ and $E(\chi)$ coincides with the set of edges incident to s , then $Q[\chi]$ stores the set $\{\perp\}$. If $T[\chi] = \text{True}$ and $E(\chi)$ does not coincide with the set of edges incident to s , $Q[\chi]$ stores the set Φ of keys with the following properties. Let e_τ be any edge whose tail v_τ has maximum y -coordinate among the edges in $E(\chi)$, i.e., $\langle e_\tau, p_\tau, q_\tau \rangle$ is such that $y(p_\tau) \geq y(p_j)$ for any $\langle e_j, p_j, q_j \rangle \in \chi$. For each $\varphi \in \Phi$, we have that:

- $T[\varphi] = \text{True}$;
- $E(\chi) \cap E(\varphi)$ contains all and only the edges in $E(\chi)$ whose tail is not v_τ , and each edge $e_i \in E(\chi) \cap E(\varphi)$ is drawn in φ as it is drawn in χ ; and
- all the edges in $E(\varphi) \setminus E(\chi)$ have v_τ as their head.

Additionally, we store a list Λ of the keys σ such that $T[\sigma] = \text{True}$ and $E(\sigma)$ is the set of edges incident to t . Note that each edge in $E(\sigma)$ has its head mapped by σ to the point $p_t \in S$ with largest y -coordinate.

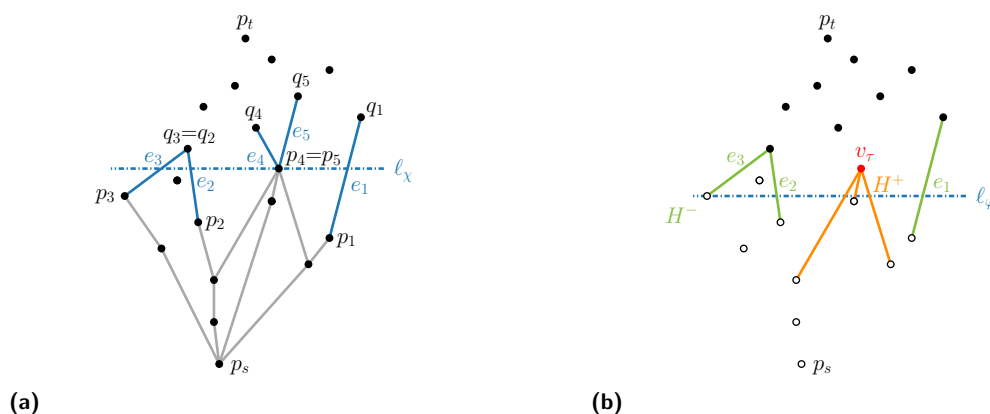
We use dynamic programming to compute the entries of T and Q in increasing order of $|V(G_\chi)|$. By the definition of T , we have that G admits an UPSE on S if and only if $\Lambda \neq \emptyset$.

First, we initialize all entries of T to **False** and all entries of Q to \emptyset .

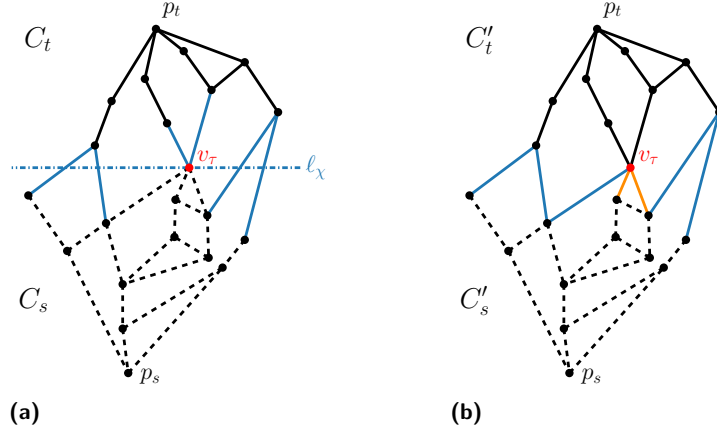
If $|V(G_\chi)| = 1$, then G_χ only consists of s . We set $T[\chi] = \text{True}$ and $Q[\chi] = \{\perp\}$ for every key $\chi = \bigcup_{i=1}^h \langle e_i, p_i, q_i \rangle$ such that:

- e_1, \dots, e_h are the edges incident to s ;
- $p_1 = \dots = p_h = p_s$; and
- for every distinct i and j in $\{1, \dots, h\}$, we have that p_s, q_i , and q_j are not aligned.

If $|V(G_\chi)| > 1$, we compute $T[\chi]$ and $Q[\chi]$ as follows, see Figure 3b. If two segments $\overline{p_i q_i}$ and $\overline{p_j q_j}$, with $i \neq j$, cross (that is, they share a point that is internal for at least one of the segments), then we leave $T[\chi]$ and $Q[\chi]$ unchanged; in particular, $T[\chi] = \text{False}$ and $Q[\chi] = \emptyset$. Otherwise, we proceed as follows. Let e_τ be any edge whose tail v_τ has maximum y -coordinate among the edges in $E(\chi)$. Let H^- be the set of edges obtained from $E(\chi)$ by removing all the edges having v_τ as their tail, and let H^+ be the set of edges of G having v_τ as their head. We define the set $H := H^- \cup H^+$. We have the following claim, which is illustrated in Figure 4.



■ **Figure 3** (a) An entry $\chi = \bigcup_{i=1}^5 \langle e_i, p_i, q_i \rangle$ with $T[\chi] = \text{True}$ and a corresponding UPSE of G_χ on a subset of S that includes p_s . (b) An entry φ from which χ stems; the points in S_\downarrow are filled white.



■ **Figure 4** Illustrations for Claim 3. (a) The connected components C_s (dashed) and C_t (solid black) defined by the st -cutset $E(\chi)$. (b) The connected components C'_s (dashed) and C'_t (solid black) defined by the st -cutset H (blue and orange edges).

\triangleright **Claim 3.** H is an st -cutset of G .

Consider the set S_\downarrow consisting of the points in S whose y -coordinates are smaller than $y(p_\tau)$. We have the following crucial observation.

► **Observation 4.** $T[\chi] = \text{True}$ if and only if there exists some key φ , with $E(\varphi) = H$, such that $T[\varphi] = \text{True}$, the edges in H^- are drawn in φ as in χ , the edges in H^+ have their heads mapped by φ on p_τ and their tails on a point in S_\downarrow .

In view of Observation 4, we can now define a procedure to compute $T[\chi]$ and $Q[\chi]$. Assume that the edges $e_1, \dots, e_{|H^-|}, \dots, e_{|H|} \in H$ are ordered so that the edges of H^- precede those of H^+ . By Observation 4, if $|S_\downarrow| < |H^+|$, then we leave $T[\chi]$ and $Q[\chi]$ unchanged, i.e., $T[\chi] = \text{False}$ and $Q[\chi] = \emptyset$. In fact, in this case, there are not enough points in S_\downarrow to map the tails of the edges in H^+ . Otherwise, let D be the set of all permutations with repetitions of $|H^+|$ points from S_\downarrow . We define a set Φ of keys that, for each $(d_1, \dots, d_{|H^+|}) \in D$, contains a key φ such that:

- (i) $E(\varphi) = H$;
- (ii) for any $i = 1, \dots, |H^-|$, the triple containing e_i in φ is the same as the triple containing e_i in χ (note that $e_i \in H^-$);
- (iii) for any $j = |H^-| + 1, \dots, |H|$, the triple containing e_j in φ has $q_j = p_\tau$, and $p_j = d_{j - |H^-|}$ (note that $e_j \in H^+$); and
- (iv) for every $i = 1, \dots, |H^-|$ and $j = |H^-| + 1, \dots, |H|$, it holds $p_i = p_j$ if and only if e_i and e_j have the same tail.

Let $\Phi^T = \{\varphi \in \Phi \wedge T[\varphi] = \text{True}\}$. By Observation 4, we have $T[\chi] = \text{True}$ if and only if $|\Phi^T| \geq 1$. Thus, we set $T[\chi] = \bigvee_{\varphi \in \Phi} T[\varphi]$ and $Q[\chi] = \Phi^T$. We say that χ *stems from* any key $\varphi \in \Phi$ with $T[\varphi] = \text{True}$.

We now upper bound the sizes of T and Q and the time needed to compute them.

\triangleright **Claim 5.** Tables T and Q have size in $\mathcal{O}(n^{3k})$ and $\mathcal{O}(kn^{4k} \log n)$, respectively.

The proof of Claim 5 is based on the fact that the number of entries of T (and, thus, of Q) is bounded by $\binom{m}{k} \cdot n^k \cdot n^k \leq (mn^2)^k$. This is because an st -cutset $E(\chi)$ has size at most $h \leq k$ and because the number of permutations with repetitions of the points describing a

mapping of the tails (or of the heads) of $E(\chi)$ on them is n^k . Further, $\binom{a}{b} \leq a^b$. Since each entry of T stores a single bit, we immediately have that T has $\mathcal{O}(n^{3k})$ size. Instead, each entry of Q stores at most $\mathcal{O}(n^k)$ keys of size $\mathcal{O}(k \log n)$; thus, Q has $\mathcal{O}(kn^{4k} \log n)$ size.

Computing $T[\chi]$ requires accessing the values of up to $|S_\downarrow|^{|H^+|} < n^k$ entries of T . Also, the time used to compute each entry $Q[\chi]$ is upper bounded by the time needed to write the $\mathcal{O}(n^k)$ keys in $Q[\chi]$, each of which has $\mathcal{O}(k \log n)$ size. Hence, we have the following.

▷ **Claim 6.** Tables T and Q can be computed in $\mathcal{O}(n^{4k})$ and $\mathcal{O}(kn^{4k} \log n)$ time, respectively.

Finally, recall that in order to verify whether G admits an UPSE on S , we need to check whether $\Lambda \neq \emptyset$. Computing the maximum size of an st -cutset of a planar st -graph G can be done in linear time, as it reduces to the problem of computing the length of a shortest path in the dual of any embedding of G (between the vertices representing the left and right outer faces of this embedding) [14, 18]. Therefore, the overall running time to test whether G admits an UPSE on S is dominated by the time needed to compute T , that is, $\mathcal{O}(n^{4k})$ time.

If the algorithm terminates with a positive answer, we obtain an UPSE Γ of G on S by exploiting table T . Let σ be a key in Λ . We initialize Γ to a drawing of the edges in $E(\sigma)$ as they are drawn in σ . Then we search in T a key χ with $T[\chi] = \text{True}$ such that σ stems from χ , and update Γ accordingly, until a key α is reached such that $T[\alpha] = \text{True}$ and $E(\alpha)$ is the set of edges incident to s . As the depth of the recursion is linear in the size of G and a key χ can be searched in $\mathcal{O}(n^k)$ time, we have the following.

► **Theorem 7.** *Let G be an n -vertex planar st -graph whose maximum st -cutset has size k and let S be a set of n points. UPSE TESTING can be solved for (G, S) in $\mathcal{O}(n^{4k})$ time and $\mathcal{O}(n^{3k})$ space; if an UPSE of G on S exists, it can be constructed within the same bounds.*

We describe the algorithm to enumerate all UPSEs of G on S that exploits table Q and set Λ . The algorithm defines and explores an acyclic digraph \mathcal{D} . The nodes of the digraph correspond to the keys χ of the table Q such that $Q[\chi] \neq \emptyset$, plus a source n_S and a sink n_T . Let χ_i and χ_j be two keys of Q such that $Q[\chi_i] \neq \emptyset$ and $Q[\chi_j] \neq \emptyset$, and let $n(\chi_i)$ and $n(\chi_j)$ be the nodes corresponding to χ_i and χ_j in \mathcal{D} , respectively. There exists an edge from $n(\chi_i)$ to $n(\chi_j)$ in \mathcal{D} if $\chi_j \in Q[\chi_i]$. Also, there exists an edge from n_S to each node $n(\sigma)$ such that $\sigma \in \Lambda$ and an edge to n_T from each node $n(\chi)$ such that $Q[\chi] = \{\perp\}$. Then \mathcal{D} is an $n_S n_T$ -graph since n_S is its unique source and n_T is its unique sink.

The algorithm performs a depth-first traversal of \mathcal{D} , in which every distinct $n_S n_T$ -path corresponds to an UPSE of G on S . We initialize an UPSE Γ on S as $\Gamma = S$ (only S is drawn). When the visit traverses an edge $(n(\chi_i), n(\chi_j))$ of \mathcal{D} , it adds to Γ the edges in $E(\chi_j) \setminus E(\chi_i)$, drawn as in χ_j . When the traversal reaches n_T , an UPSE Γ of G on S is produced. Backtracking to a node $n(\chi_i)$ along an edge $(n(\chi_i), n(\chi_j))$, the edges in $E(\chi_j) \setminus E(\chi_i)$ are removed from Γ .

To prove the correctness (see the full version of the paper [1] for a complete proof), we show that:

- (i) Distinct paths from n_S to n_T in \mathcal{D} correspond to different UPSEs of G on S .
- (ii) For each UPSE of G on S , there exists in \mathcal{D} a path corresponding to it.

Item i can be proved by contradiction: if two distinct $n_S n_T$ -paths \mathcal{P}_1 and \mathcal{P}_2 yielded the same UPSE Γ , there would be a node $n(\chi_x)$ shared by \mathcal{P}_1 and \mathcal{P}_2 such that the nodes $n(\chi_1)$ and $n(\chi_2)$ of \mathcal{P}_1 and \mathcal{P}_2 following $n(\chi_x)$ are different. Since $n(\chi_x)$ is shared by \mathcal{P}_1 and \mathcal{P}_2 , the keys χ_1 and χ_2 have the same edge-set $E(\chi_1) = E(\chi_2)$ but the tails of the edges in $E(\chi_x) \setminus E(\chi_1)$ are mapped differently, implying that the UPSEs yielded by \mathcal{P}_1 and \mathcal{P}_2 are different. To prove Item ii, we show that, if Γ is an UPSE of G on S , then there exists an

$n_S n_T$ -path that yields Γ . For $i = 1, \dots, n$, let S_i be the set of the lowest i points of S . Also, for $i = 1, \dots, n - 1$, let Γ_i be the restriction of Γ to the vertices of G mapped to S_i and to all their incident edges. We proceed by induction on i , showing that, for each Γ_i and Γ_{i+1} there exists an edge $(n(\chi_i), n(\chi_{i+1}))$ in D that produces Γ_{i+1} . This involves finding a suitable st -cutset in Γ and proving that this corresponds to a key in D .

We now discuss the running time of the algorithm. Table Q can be constructed in $\mathcal{O}(kn^{4k} \log n)$ time, by Claim 6. Also, the digraph \mathcal{D} can be constructed in linear time in the size of Q , which is $\mathcal{O}(kn^{4k} \log n)$ by Claim 5. Finally, we discuss the delay of our algorithm. Since the paths from n_S to n_T have $\mathcal{O}(n)$ size and since between an UPSE and the next one at most two paths are traversed, the delay of our algorithm is $\mathcal{O}(n)$. We get the following.

► **Theorem 8.** *Let G be a n -vertex planar st -graph whose maximum st -cut has size k and let S be a set of n points. It is possible to enumerate all UPSEs of G on S with $\mathcal{O}(n)$ delay, using $\mathcal{O}(kn^{4k} \log n)$ space, after $\mathcal{O}(kn^{4k} \log n)$ set-up time.*

5 Planar st -Graphs Composed of Two st -Paths

In this section, we consider the special case of Theorem 7 in which the underlying graph of the given planar st -graph is an n -vertex cycle. Here, Theorem 7 would yield an $\mathcal{O}(n^8)$ -time testing algorithm. We give a much faster algorithm based on a characterization of the positive instances, provided that the points are in general position.

► **Theorem 9.** *Let G be an n -vertex planar st -graph consisting of two st -paths P_L and P_R , and let S be a pointset with n points in general position. We have that G admits an UPSE on S with P_L to the left of P_R if and only if $|P_L| \geq |\mathcal{H}_L(S)|$ and $|P_R| \geq |\mathcal{H}_R(S)|$. Also, it can be tested in $\mathcal{O}(n \log n)$ time whether G admits an UPSE on S .*

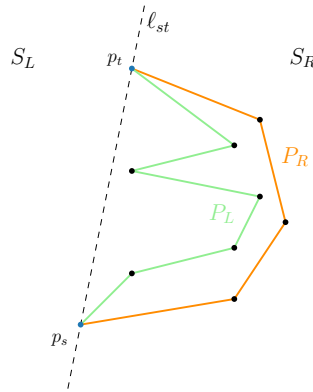
Proof. Provided the characterization in the statement holds, we can test whether G admits an UPSE on S by computing the convex hull $\mathcal{CH}(S)$ of S (in $\mathcal{O}(n \log n)$ time), deriving the sets $\mathcal{H}_L(S)$ and $\mathcal{H}_R(S)$ (in $\mathcal{O}(n)$ time, by scanning $\mathcal{CH}(S)$), and finally comparing their sizes with the ones of P_L and P_R (in $\mathcal{O}(1)$ time). Thus, we focus on proving the characterization.

The necessity is obvious. In the remainder we prove the sufficiency by induction on the size of S (and, thus, of $V(G)$). We give some preliminary definitions; see Figures 5–7. Let p_s (p_t) be the south (north) extreme of S and let ℓ_{st} be the line through p_s and p_t . Let S_L (S_R) be the subset of S in the closed half-plane to the left (right) of ℓ_{st} , including p_s and p_t . Note that $\mathcal{H}_L(S) \subseteq S_L$ and $\mathcal{H}_R(S) \subseteq S_R$. Also, since S is in general position, $S_L \cap S_R = \{p_s, p_t\}$.

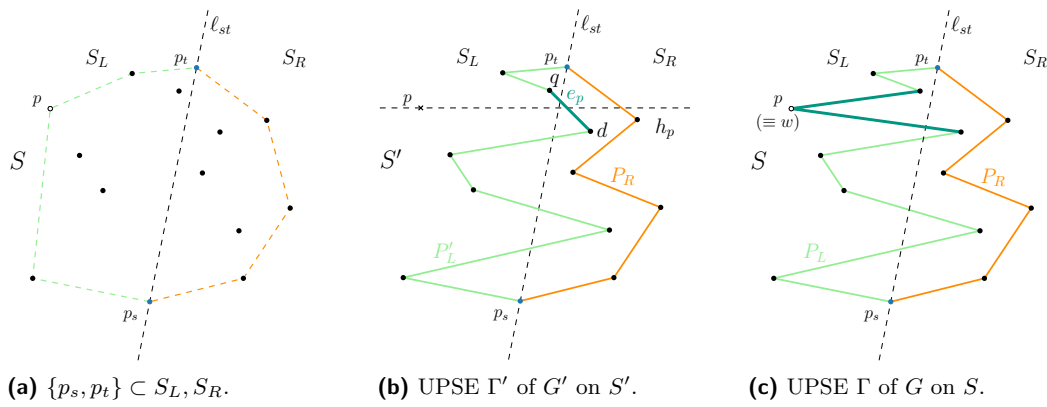
In the base case, it either holds that $S_L = \{p_s, p_t\}$ and $|\mathcal{H}_R(S)| = |P_R|$, or $S_R = \{p_s, p_t\}$ and $|\mathcal{H}_L(S)| = |P_L|$. We discuss the former case (see Figure 5), as the latter case is symmetric. In this case, an UPSE Γ of G on S can be constructed by drawing P_R as the right envelope $\mathcal{E}_R(S)$ and P_L as the y -monotone polyline connecting the point of $S_R \setminus \mathcal{H}_R(S)$.

If the base case does not hold, we distinguish two cases based on whether both S_L and S_R contain a vertex different from p_s and p_t (**Case A**), or only one of them does (**Case B**).

If **Case A** holds, assume $|P_L| \geq |S_L|$; the case $|P_L| < |S_L|$ is symmetric, as in that case it holds true that $|P_R| \geq |S_R|$. Refer to Figure 6. Then $\mathcal{H}_L(S)$ contains a point p different from p_s and p_t ; see Figure 6a. Since by the statement $|P_L| \geq |\mathcal{H}_L(S)|$ and $|\mathcal{H}_L(S)| \geq 3$, we have that P_L contains at least one internal vertex. Let $S' = S \setminus \{p\}$, let P'_L be an st -path with $|P'_L| = |P_L| - 1$, and let G' be the st -graph $P'_L \cup P_R$. Since $|\mathcal{H}_L(S')| \leq |S_L| - 1$ and $|S_L| \leq |P_L|$, we have that $|\mathcal{H}_L(S')| \leq |P_L| - 1 = |P'_L|$. Thus, the graph G' and the pointset S' satisfy the conditions of the statement. Since $|S'| = |S| - 1$ (and $|V(G')| = |V(G)| - 1$), by induction we have that the graph G' admits an UPSE Γ' on S' (see Figure 6b). Figures 6b and 6c show how to modify Γ' to obtain an UPSE Γ of G on S .



■ **Figure 5** Illustration for the base case of Theorem 9.

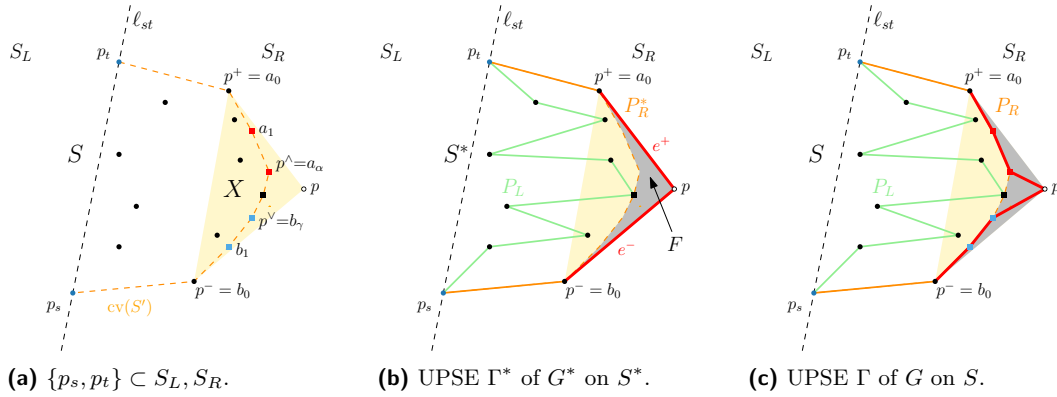


(a) $\{p_s, p_t\} \subset S_L, S_R$.

(b) UPSE Γ' of G' on S' .

(c) UPSE Γ of G on S .

■ **Figure 6** Illustrations for **Case A** in the proof of Theorem 9.



(a) $\{p_s, p_t\} \subset S_L, S_R$.

(b) UPSE Γ^* of G^* on S^* .

(c) UPSE Γ of G on S .

■ **Figure 7** Illustrations for **Case B2** in the proof of Theorem 9.

If **Case B** holds, recall that $S_L = \{p_s, p_t\} \subset S_R$, and since the base case does not apply, we have that $|P_R| > |\mathcal{H}_R(S)|$. Let p be any point in $\mathcal{H}_R(S) \setminus \{p_s, p_t\}$ and $S' = S \setminus \{p\}$. By the conditions of **Case B**, the path P_R contains at least one internal vertex. Let P'_R be an st -path with $|P'_R| = |P_R| - 1$, and let G' be the st -graph $P_L \cup P'_R$. We distinguish two cases based on the size of $\mathcal{H}_R(S')$. In **Case B1**, it holds $|P'_R| \geq |\mathcal{H}_R(S')|$, whereas in **Case B2**, it holds $|P'_R| < |\mathcal{H}_R(S')|$. In **Case B1**, we have that the pair (G', S') satisfies the conditions of the statement. In particular, the pair (G', S') either matches the conditions of the base

case or again those of **Case B**. Thus, since $|S'| = |S| - 1$ (and $|V(G')| = |V(G)| - 1$), we can inductively construct an UPSE Γ' of G' on S' , and obtain an UPSE of G on S with a redrawing similar to the one in Figure 6c. In **Case B2**, we proceed as follows; see Figure 7. Let p^+ (p^-) be the point of $\mathcal{H}_R(S)$ with the smallest y -coordinate and above p (with the largest y -coordinate and below p). Let X be the set of points of S in the interior of the triangle $\Delta p^+ p p^-$ together with p^+ and p^- (but not p). Clearly, the right envelope of $\mathcal{CH}(X)$ forms a subpath of the right envelope of $\mathcal{CH}(S')$; see Figure 7a. The set $\mathcal{H}_R(X)$ consists of p^- , p^+ , and of k vertices not belonging to $\mathcal{H}_R(S)$ (squares in Figure 7). We denote by $k^* = |P_R| - |\mathcal{H}_R(S)|$ the number of points in the interior of $\mathcal{CH}(S)$ that need to be the image of a vertex of P_R in an UPSE of G on S . Note that $k > k^* > 0$ holds. Let p^\wedge (p^\vee) be the point of $\mathcal{H}_R(S')$ with the smallest y -coordinate and above p (with the largest y -coordinate and below p). Up to renaming, let $a_0 = p^+$, $a_1, \dots, a_\alpha = p^\wedge$ be the subsequence of points of $\mathcal{E}_R(X)$ encountered when traversing $\mathcal{E}_R(X)$ from p^+ to p^\wedge (these points have decreasing y -coordinates). Similarly, let $b_0 = p^-$, $b_1, \dots, b_\gamma = p^\vee$ be the subsequence of points of $\mathcal{E}_R(X)$ encountered when traversing $\mathcal{E}_R(X)$ from p^- to p^\vee (these points have increasing y -coordinates). We define the set $X^* \subset \mathcal{H}_R(X)$ as follows. If $k^* \leq \alpha$, then $X^* = \{a_i | 1 \leq i \leq k^*\}$, otherwise $X^* = \{a_i | 1 \leq i \leq \alpha\} \cup \{b_i | 1 \leq i \leq k^* - \alpha\}$. Observe that, $|X^*| = k^*$. Also, by the definition of k^* , the path P_R contains $|\mathcal{H}_R(S)| - 2 + k^*$ internal vertices and since $\mathcal{H}_R(S) \geq 3$ in **Case B**, we have that P_R contains at least $k^* + 1$ internal vertices. Let $S^* = S \setminus X^*$, let P_R^* be an st -path with $|P_R| - k^*$ vertices, and let G^* be the st -graph $P_L \cup P_R^*$. Clearly, the pair (G^*, S^*) satisfies the statement, and in particular the base case. In fact, $|P_R^*| = |P_R| - k^*$, and by the definition of k^* , we have that $|P_R| - k^* = |\mathcal{H}_R(S)|$. Moreover, by construction, $\mathcal{H}_R(S) = \mathcal{H}_R(S^*)$, since the vertices of X^* lie in the interior of $\mathcal{CH}(S)$. Thus, since $|S^*| = |S| - k^*$, by induction G^* admits an UPSE Γ^* on S^* ; see Figure 7b. Moreover, as the base case applies to (G^*, S^*) , we have that the endpoints of the edges of P_R^* are consecutive along $\mathcal{E}_R(S)$. In particular, there exist two adjacent edges e^- and e^+ of P_R^* such that the tail of e^- is mapped to p^- , the head of e^- (i.e., the tail of e^+) is mapped to p , and the head of e^+ is mapped to p^+ . Thus, it is possible to obtain an UPSE Γ of G on S from Γ^* (see Figure 7c) by replacing the drawing of the edges e^+ and e^- with a y -monotone polyline that passes through all the points in X^* . Such a polyline lies inside the region F (shaded gray in Figures 7b and 7c) obtained by subtracting from the triangle $\Delta p^+ p p^-$ (interpreted as a closed region) all the points of $\mathcal{CH}(X)$. In particular, observe that, in Γ^* , the region F is not traversed by any edge and that the only points of S^* that lie on the boundary of F are p and the points in $\mathcal{H}_R(X) \setminus X^*$. ◀

6 Enumerating Non-crossing Monotone Hamiltonian Cycles

Theorem 9 allows us to test whether an n -vertex planar st -graph G composed of two st -paths can be embedded as a non-crossing monotone Hamiltonian cycle on a set S of n points. We now show an efficient algorithm for enumerating *all* the non-crossing monotone Hamiltonian cycles on S . Figure 8 shows two non-crossing monotone Hamiltonian cycles on a pointset.

► Theorem 10. *Let S be a set of n points. It is possible to enumerate all the non-crossing monotone Hamiltonian cycles on S with $\mathcal{O}(n)$ delay, using $\mathcal{O}(n^2)$ space, after $\mathcal{O}(n^2)$ set-up time.*

Let p_1, \dots, p_n be the points of S , ordered by increasing y -coordinates. This order can be computed in $\mathcal{O}(n \log n)$ time. For $i \in [n]$, let $S_i = \{p_1, \dots, p_i\}$. A *bipath* B on S_i consists of two non-crossing monotone paths L and R on S_i , each of which might be a single point, such that L and R start at p_1 , each point of S_i is the image of an endpoint of a segment of B , and if L and R both have at least one segment, then L is to the left of R (see Figure 9).

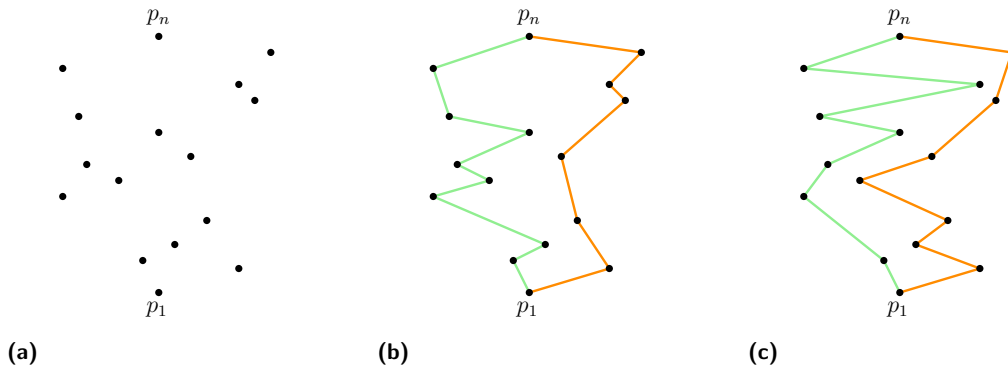


Figure 8 Two non-crossing monotone Hamiltonian cycles on the same pointset.

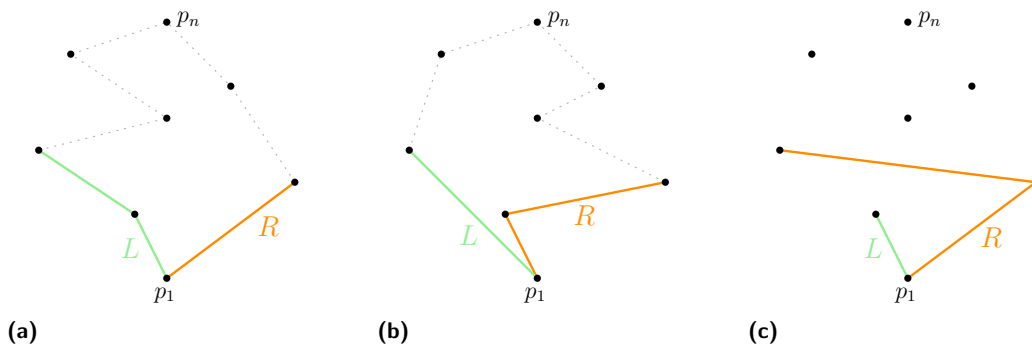


Figure 9 Three bipaths on S_4 . The first two bipaths are extensible, while the third one is not.

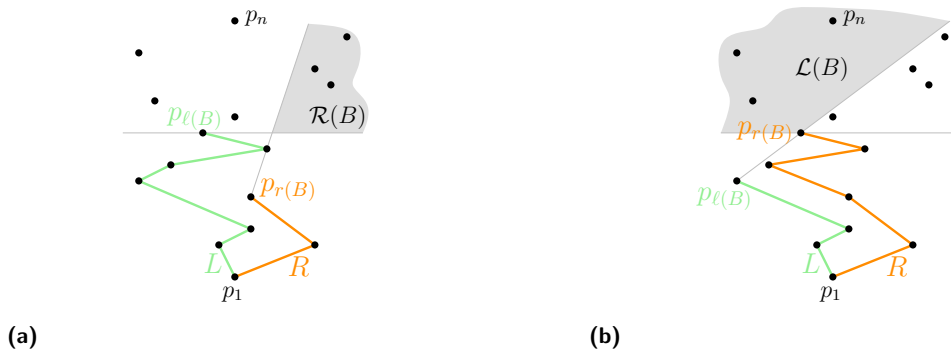


Figure 10 (a) Region $\mathcal{R}(B)$ for a bipath B . (b) Region $\mathcal{L}(B)$ for a bipath B .

We say that a bipath B is *extensible* if there exists a non-crossing monotone Hamiltonian cycle on S whose restriction to S_i is B . Consider a bipath B on S_i with $1 < i < n$. Let $p_{\ell(B)}$ and $p_{r(B)}$ be the endpoints of L and R with the highest y -coordinate, respectively. Suppose first that $\ell(B) > r(B)$. Consider the rightmost ray $\rho(p_{r(B)}, S_{\ell(B)} \setminus S_{r(B)})$ starting at $p_{r(B)}$ through a point of $S_{\ell(B)} \setminus S_{r(B)}$. We denote by $\mathcal{R}(B)$ the open region of the plane strictly to the right of $\rho(p_{r(B)}, S_{\ell(B)} \setminus S_{r(B)})$ and strictly above the horizontal line through $p_{\ell(B)}$; see Figure 10a. Similarly, if $p_{r(B)}$ is higher than $p_{\ell(B)}$, then $\mathcal{L}(B)$ is the open region of the plane strictly to the left of the leftmost ray $\ell(p_{\ell(B)}, S_{r(B)} \setminus S_{\ell(B)})$ from $p_{\ell(B)}$ through a point of $S_{r(B)} \setminus S_{\ell(B)}$ and strictly above the horizontal line through $p_{r(B)}$; see Figure 10b.

For any $i \in [n - 1]$, we say that a bipath B on S_i is *safe* if:

- (i) $i = 1$; or
- (ii) $i > 1$, $p_{\ell(B)}$ is higher than $p_{r(B)}$, and $|\mathcal{R}(B) \cap S| \geq 1$; or
- (iii) $i > 1$, $p_{r(B)}$ is higher than $p_{\ell(B)}$, and $|\mathcal{L}(B) \cap S| \geq 1$.

We have the following lemma which is proved in the full version of the paper [1].

► **Lemma 11.** *A bipath B is extensible if and only if it is safe.*

Our enumeration algorithm implicitly defines and explores a search tree T . Its leaves have level n and correspond to non-crossing monotone Hamiltonian cycles on S . The internal nodes at level i correspond to extensible bipaths on S_i and have at most two children each. The exploration of T performed by the algorithm is a depth-first traversal. When a node μ is visited, the number of its children is established. If μ has at least one child, the visit proceeds with any child of μ . Otherwise, μ is a leaf; then the visit proceeds with any unvisited child of the ancestor of μ that has largest level, among the ancestors of μ with unvisited children.

The algorithm starts at the root of T , which corresponds to the (unique) safe bipath on S_1 . At each node μ at level $i \in [n - 2]$, corresponding to a bipath $B(\mu)$, we construct either one or two bipaths on S_{i+1} , associated with either one or two children of μ , respectively. Let $L(\mu)$ and $R(\mu)$ be the left and right non-crossing monotone paths composing $B(\mu)$, respectively, and let $p_{\ell(B(\mu))}$ and $p_{r(B(\mu))}$ be the endpoints of $L(\mu)$ and $R(\mu)$ with the highest y -coordinate, respectively. If $\overline{p_{\ell(B(\mu))}p_{i+1}}$ does not cross $R(\mu)$, then let $B_L = B(\mu) \cup \overline{p_{\ell(B(\mu))}p_{i+1}}$. We test whether B_L is a safe bipath and, in the positive case, add to μ a child μ_L corresponding to B_L . Analogously, if $\overline{p_{r(B(\mu))}p_{i+1}}$ does not cross $L(\mu)$, we test whether $B_R = B(\mu) \cup \overline{p_{r(B(\mu))}p_{i+1}}$ is a safe bipath and, in the positive case, add to μ a child μ_R corresponding to B_R . Note that the algorithm guarantees that each non-leaf node of T is safe, and thus, by Lemma 11, extensible. Finally, at each node μ at level $n - 1$, we add a leaf λ to μ corresponding to the non-crossing monotone Hamiltonian cycle $B(\mu) \cup \overline{p_{\ell(B(\mu))}p_n} \cup \overline{p_{r(B(\mu))}p_n}$. Since μ is extensible, such a cycle is indeed non-crossing.

In order to complete the proof of Theorem 10, we show what follows:

- (i) Each node of T at level $i \neq n$ is internal.
 - (ii) Each leaf corresponds to a non-crossing monotone Hamiltonian cycle on S .
 - (iii) Distinct leaves correspond to different non-crossing monotone Hamiltonian cycles on S .
 - (iv) For each non-crossing monotone Hamiltonian cycle on S , there exists a leaf of T corresponding to it.
 - (v) Using $\mathcal{O}(n^2)$ pre-processing time and $\mathcal{O}(n^2)$ space, the algorithm enumerates each non-crossing monotone Hamiltonian cycle on S with $\mathcal{O}(n)$ delay.
- To prove Item i, we show that the leaves of T have all level n . Consider a node μ of T with level $i < n - 1$, we prove that it has a child in T . Recall that $B(\mu)$ is safe, otherwise it would not had been added to T , and thus, by Lemma 11, it is extensible. Hence, there exists a non-crossing monotone Hamiltonian cycle C on S whose restriction to S_i is $B(\mu)$. Also, the restriction of C to S_{i+1} is a bipath $B'(\mu)$ on S_{i+1} which coincides with $B(\mu)$, except that it contains either the segment $\overline{p_{\ell(B(\mu))}p_{i+1}}$ or the segment $\overline{p_{r(B(\mu))}p_{i+1}}$. Since $B'(\mu)$ is the restriction of C to S_{i+1} , it is extensible and thus, by Lemma 11, it is safe. It follows that μ has a child corresponding to $B'(\mu)$, which is inserted in T when adding either the segment $\overline{p_{\ell(B(\mu))}p_{i+1}}$ or the segment $\overline{p_{r(B(\mu))}p_{i+1}}$ to $B(\mu)$. The proof that a node with level $n - 1$ is not a leaf is analogous.
 - To prove Item ii, consider a leaf λ and its parent μ in T . Note that μ is associated with a safe bipath $B(\mu)$ on S_{n-1} ; by Lemma 11, we have that $B(\mu)$ is extensible. Since $B(\mu)$ is extensible, the (unique) monotone Hamiltonian cycle on S whose restriction to S_{n-1} is $B(\mu)$ is non-crossing. This cycle corresponds to λ and is added to T when visiting μ .

- To prove Item iii, suppose for a contradiction that there exist two leaves λ_1 and λ_2 associated with two monotone Hamiltonian cycles C_1 and C_2 , respectively, with $C_1 = C_2$. Let μ be the lowest common ancestor of λ_1 and λ_2 in T . Let j be the level of μ . Denote by μ_i the child of μ leading to λ_i , with $i \in \{1, 2\}$. By the construction of T , we have that exactly one of the bipaths $B(\mu_1)$ and $B(\mu_2)$ contains the segment $\overline{p_{\ell(B(\mu))}p_{j+1}}$, while the other one contains the segment $\overline{p_{r(B(\mu))}p_{j+1}}$. This contradicts the fact that $C_1 = C_2$.
- To prove Item iv, let C be a non-crossing monotone Hamiltonian cycle on S . Consider the safe bipath B on S_{n-1} obtained by removing from C the point p_n , together with its two incident segments. It suffices to show that T contains a node μ such that $B = B(\mu)$. In fact, in this case, μ is an extensible node of level $n - 1$ whose unique child in T is the leaf corresponding to C . To prove that T contains such a node μ , we prove by induction that, for every level $i = 1, \dots, n - 1$, the tree T contains a node corresponding to the restriction B_i of B to S_i . The base case trivially holds. For the inductive case, suppose that T contains a node ν whose associated bipath $B(\nu)$ is B_{i-1} . Then B_i is obtained by adding either the segment $\overline{p_{\ell(B(\nu))}p_i}$ or the segment $\overline{p_{r(B(\nu))}p_i}$ to B_{i-1} . Since B_i is extensible, by Lemma 11 it is safe, and hence ν has a child in T corresponding to B_i .
- Finally, we discuss Item v. To this aim, we compute in $\mathcal{O}(n^2)$ time two tables C and D of $\mathcal{O}(n^2)$ size that allow us to test in $\mathcal{O}(1)$ time whether a bipath B on S_i , with $i \in \{2, \dots, n - 1\}$, can be extended to a bipath on S_{i+1} and whether B is safe. The tables C and D are indexed by triples $\langle p_a, p_b, X \rangle$, where $p_a, p_b \in S$ with $a < b$ and $X \in \{L, R\}$. Each entry of C contains a Boolean value $C[p_a, p_b, X]$ that is set to **True** if and only if the segment $\overline{p_a p_{b+1}}$ does not cross any bipath B on S_b composed of two monotone st -paths L and R respectively ending at points p_a and p_b (if $X = L$) or respectively ending at points p_b and p_a (if $X = R$). Each entry of D contains a Boolean value $D[p_a, p_b, X]$ that is set to **True** if and only if the open region that is (i) strictly to the right of the rightmost (if $X = R$, or leftmost if $X = L$) ray starting at p_a and passing through a point in $S_b \setminus S_a$ and (ii) strictly above the horizontal line through p_b contains a point of S . For each fixed $a \in [n - 2]$ and $X \in \{L, R\}$, we compute all the entries $C[p_a, p_b, X]$ and $D[p_a, p_b, X]$ with $b = a + 1, a + 2, \dots, n - 1$ in overall $\mathcal{O}(n)$ time. This sums up to $\mathcal{O}(n^2)$ time over all the entries of C and of D . The query time of C and D , together with the fact that T has n levels, implies that the algorithm's delay is in $\mathcal{O}(n)$. More details can be found in the full version of the paper [1].

Items i–iv prove the correctness of the enumeration algorithm, while Item v proves its efficiency. This concludes the proof of Theorem 10.

7 Conclusions and Open Problems

We addressed basic pointset embeddability problems for upward planar graphs. We proved that UPSE testing is NP-hard even for planar st -graphs composed of internally-disjoint st -paths and for directed trees composed of directed root-to-leaf paths. For planar st -graphs, we showed that UPSE TESTING can be solved in $\mathcal{O}(n^{4k})$ time, where k is the maximum st -cutset of G , and we provided an algorithm to enumerate all UPSEs of G on S with $\mathcal{O}(n)$ worst-case delay. We also showed how to enumerate all monotone polygonalizations of a given pointset with $\mathcal{O}(n)$ worst-case delay. We point out the following open problems.

- Our NP-hardness proofs for UPSE TESTING use the fact that the points are not in general position. Given a directed tree T on n vertices and a set S of n points *in general position*, is it NP-hard to decide whether T has an UPSE on S ?

- Can UPSE TESTING be solved in polynomial time or does it remain NP-hard if the input is a *maximal* planar st -graph?
- We proved that UPSE TESTING for a planar st -graph is in XP with respect to the size of the maximum st -cutset of G . Is the problem in FPT with respect to the same parameter? Are there other interesting parameterizations for the problem?
- Let S be a pointset and \mathcal{P} be a non-crossing path on a subset of S . Is it possible to decide in polynomial time whether \mathcal{P} can be extended to a polygonalization of S ? A positive answer would imply an algorithm with polynomial delay for enumerating the polygonalizations of a pointset, with the same approach as the one we adopted in this paper for monotone polygonalizations.

References

- 1 Carlos Alegria, Susanna Caroppo, Giordano Da Lozzo, Marco D'Elia, Giuseppe Di Battista, Fabrizio Frati, Fabrizio Grosso, and Maurizio Patrignani. Upward pointset embeddings of planar st -graphs. *CoRR*, abs/2408.17369, 2024. [arXiv:2408.17369](https://arxiv.org/abs/2408.17369).
- 2 Victor Alvarez, Karl Bringmann, Radu Curticapean, and Saurabh Ray. Counting triangulations and other crossing-free structures via onion layers. *Discret. Comput. Geom.*, 53(4):675–690, 2015. doi:10.1007/S00454-015-9672-3.
- 3 Patrizio Angelini, Fabrizio Frati, Markus Geyer, Michael Kaufmann, Tamara Mchedlidze, and Antonios Symvonis. Upward geometric graph embeddings into point sets. In Ulrik Brandes and Sabine Cornelsen, editors, *18th International Symposium on Graph Drawing (GD 2010)*, volume 6502 of *LNCS*, pages 25–37. Springer, 2010. doi:10.1007/978-3-642-18469-7_3.
- 4 Elena Arseneva, Pilar Cano, Linda Kleist, Tamara Mchedlidze, Saeed Mehrabi, Irene Parada, and Pavel Valtr. Upward point set embeddings of paths and trees. In Ryuhei Uehara, Seok-Hee Hong, and Subhas C. Nandy, editors, *15th International Conference and Workshops on Algorithms and Computation (WALCOM 2021)*, volume 12635 of *LNCS*, pages 234–246. Springer, 2021. doi:10.1007/978-3-030-68211-8_19.
- 5 Therese Biedl and Martin Vatshelle. The point-set embeddability problem for plane graphs. In Tamal K. Dey and Sue Whitesides, editors, *28th ACM Symposium on Computational Geometry (SoCG 2012)*, pages 41–50. ACM, 2012. doi:10.1145/2261250.2261257.
- 6 Carla Binucci, Emilio Di Giacomo, Walter Didimo, Alejandro Estrella-Balderrama, Fabrizio Frati, Stephen G. Kobourov, and Giuseppe Liotta. Upward straight-line embeddings of directed graphs into point sets. *Comput. Geom.*, 43(2):219–232, 2010. doi:10.1016/J.COMGEO.2009.07.002.
- 7 Prosenjit Bose. On embedding an outer-planar graph in a point set. In Giuseppe Di Battista, editor, *5th International Symposium on Graph Drawing (GD '97)*, volume 1353 of *LNCS*, pages 25–36. Springer, 1997. doi:10.1007/3-540-63938-1_47.
- 8 Prosenjit Bose. On embedding an outer-planar graph in a point set. *Comput. Geom.*, 23(3):303–312, 2002. doi:10.1016/S0925-7721(01)00069-4.
- 9 Prosenjit Bose, Michael McAllister, and Jack Snoeyink. Optimal algorithms to embed trees in a point set. In Franz-Josef Brandenburg, editor, *Symposium on Graph Drawing (GD '95)*, volume 1027 of *LNCS*, pages 64–75. Springer, 1995. doi:10.1007/BFB0021791.
- 10 Prosenjit Bose, Michael McAllister, and Jack Snoeyink. Optimal algorithms to embed trees in a point set. *J. Graph Algorithms Appl.*, 1(2):1–15, 1997. doi:10.7155/JGAA.00002.
- 11 Sergio Cabello. Planar embeddability of the vertices of a graph using a fixed point set is NP-hard. *J. Graph Algorithms Appl.*, 10(2):353–363, 2006. doi:10.7155/JGAA.00132.
- 12 Netzahualcoyotl Castañeda and Jorge Urrutia. Straight line embeddings of planar graphs on point sets. In Frank Fiala, Evangelos Kranakis, and Jörg-Rüdiger Sack, editors, *8th Canadian Conference on Computational Geometry (CCCG 1996)*, pages 312–318. Carleton University Press, 1996. URL: http://www.cccg.ca/proceedings/1996/cccg1996_0052.pdf.

- 13 Gi-Sang Cheon, Hong Joon Choi, Guillermo Esteban, and Minh Song. Enumeration of bipartite non-crossing geometric graphs. *Discret. Appl. Math.*, 317:86–100, 2022. doi:10.1016/J.DAM.2022.04.008.
- 14 Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 4th Edition*. MIT Press, 2022. URL: <https://mitpress.mit.edu/9780262046305/introduction-to-algorithms/>.
- 15 Giuseppe Di Battista, Walter Didimo, Luca Grilli, Fabrizio Grosso, Giacomo Ortali, Maurizio Patrignani, and Alessandra Tappini. Small point-sets supporting graph stories. In Patrizio Angelini and Reinhard von Hanxleden, editors, *30th International Symposium on Graph Drawing and Network Visualization (GD 2022)*, volume 13764 of *LNCS*, pages 289–303. Springer, 2022. doi:10.1007/978-3-031-22203-0_21.
- 16 Giuseppe Di Battista, Walter Didimo, Luca Grilli, Fabrizio Grosso, Giacomo Ortali, Maurizio Patrignani, and Alessandra Tappini. Small point-sets supporting graph stories. *J. Graph Algorithms Appl.*, 27(8):651–677, 2023. doi:10.7155/JGAA.00639.
- 17 Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, 1999.
- 18 Giuseppe Di Battista and Roberto Tamassia. Algorithms for plane representations of acyclic digraphs. *Theor. Comput. Sci.*, 61:175–198, 1988. doi:10.1016/0304-3975(88)90123-5.
- 19 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.
- 20 David Eppstein. Counting polygon triangulations is hard. *Discret. Comput. Geom.*, 64(4):1210–1234, 2020. doi:10.1007/S00454-020-00251-7.
- 21 David Eppstein. Non-crossing hamiltonian paths and cycles in output-polynomial time. In Erin W. Chambers and Joachim Gudmundsson, editors, *39th International Symposium on Computational Geometry (SoCG 2023)*, volume 258 of *LIPICs*, pages 29:1–29:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICs.SOCG.2023.29.
- 22 Philippe Flajolet and Marc Noy. Analytic combinatorics of non-crossing configurations. *Discret. Math.*, 204(1-3):203–229, 1999. doi:10.1016/S0012-365X(98)00372-0.
- 23 M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, first edition edition, 1979.
- 24 Emilio Di Giacomo, Henry Förster, Daria Kokhovich, Tamara Mchedlidze, Fabrizio Montecchiani, Antonios Symvonis, and Anaïs Villedieu. On 1-bend upward point-set embeddings of st-digraphs. In José A. Soto and Andreas Wiese, editors, *16th Latin American Symposium on Theoretical Informatics (LATIN 2024), Part I*, volume 14578 of *LNCS*, pages 3–18. Springer, 2024. doi:10.1007/978-3-031-55598-5_1.
- 25 Francesco Giordano, Giuseppe Liotta, Tamara Mchedlidze, Antonios Symvonis, and Sue Whitesides. Computing upward topological book embeddings of upward planar digraphs. *J. Discrete Algorithms*, 30:45–69, 2015. doi:10.1016/J.JDA.2014.11.006.
- 26 Peter Gritzmann, Bojan Mohar, János Pach, and Richard Pollack. Embedding a planar triangulation with vertices at specified points. *The American Mathematical Monthly*, 98(2):165, 1991. doi:10.2307/2323956.
- 27 Lenwood S. Heath, Sriram V. Pemmaraju, and Ann N. Trenk. Stack and queue layouts of directed acyclic graphs: Part I. *SIAM J. Comput.*, 28(4):1510–1539, 1999. doi:10.1137/S0097539795280287.
- 28 Michael Kaufmann, Tamara Mchedlidze, and Antonios Symvonis. Upward point set embeddability for convex point sets is in P. In Marc J. van Kreveld and Bettina Speckmann, editors, *19th International Symposium on Graph Drawing (GD 2011)*, volume 7034 of *LNCS*, pages 403–414. Springer, 2011. doi:10.1007/978-3-642-25878-7_38.
- 29 Michael Kaufmann, Tamara Mchedlidze, and Antonios Symvonis. On upward point set embeddability. *Comput. Geom.*, 46(6):774–804, 2013. doi:10.1016/J.COMGEO.2012.11.008.

- 30 Michael Kaufmann and Roland Wiese. Embedding vertices at points: Few bends suffice for planar graphs. In Jan Kratochvíl, editor, *7th International Symposium on Graph Drawing (GD'99)*, volume 1731 of *LNCS*, pages 165–174. Springer, 1999. doi:10.1007/3-540-46648-7_17.
- 31 Michael Kaufmann and Roland Wiese. Embedding vertices at points: Few bends suffice for planar graphs. *J. Graph Algorithms Appl.*, 6(1):115–129, 2002. doi:10.7155/JGAA.00046.
- 32 Dániel Marx and Tillmann Miltzow. Peeling and nibbling the cactus: Subexponential-time algorithms for counting triangulations and related problems. In Sándor P. Fekete and Anna Lubiw, editors, *32nd International Symposium on Computational Geometry (SoCG 2016)*, volume 51 of *LIPICs*, pages 52:1–52:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICS.SOCG.2016.52.
- 33 Tamara Mchedlidze. Upward planar embedding of an n -vertex oriented path on $O(n^2)$ points. *Comput. Geom.*, 46(8):1003–1008, 2013. doi:10.1016/J.COMGEO.2013.05.004.
- 34 Joseph S. B. Mitchell and Joseph O'Rourke. Computational geometry column 42. *Int. J. Comput. Geom. Appl.*, 11(5):573–582, 2001. doi:10.1142/S0218195901000651.
- 35 Rahnuma Islam Nishat, Debajyoti Mondal, and Md. Saidur Rahman. Point-set embeddings of plane 3-trees. *Comput. Geom.*, 45(3):88–98, 2012. doi:10.1016/J.COMGEO.2011.09.002.
- 36 Andreas Razen and Emo Welzl. Counting plane graphs with exponential speed-up. In Cristian S. Calude, Grzegorz Rozenberg, and Arto Salomaa, editors, *Rainbow of Computer Science - Dedicated to Hermann Maurer on the Occasion of His 70th Birthday*, volume 6570 of *LNCS*, pages 36–46. Springer, 2011. doi:10.1007/978-3-642-19391-0_3.
- 37 Manuel Wettstein. Counting and enumerating crossing-free geometric graphs. *J. Comput. Geom.*, 8(1):47–77, 2017. doi:10.20382/JOCG.V8I1A4.
- 38 Katsuhisa Yamanaka, David Avis, Takashi Horiyama, Yoshio Okamoto, Ryuhei Uehara, and Tanami Yamauchi. Algorithmic enumeration of surrounding polygons. *Discret. Appl. Math.*, 303:305–313, 2021. doi:10.1016/J.DAM.2020.03.034.

Parameterized Algorithms for Beyond-Planar Crossing Numbers

Miriam Münch  

Faculty of Computer Science and Mathematics, University of Passau, Germany

Ignaz Rutter  

Faculty of Computer Science and Mathematics, University of Passau, Germany

Abstract

Beyond-planar graph classes are usually defined via forbidden configurations or patterns in a drawing. In this paper, we formalize these concepts on a combinatorial level and show that, for any fixed family \mathcal{F} of crossing patterns, deciding whether a given graph G admits a drawing that avoids all patterns in \mathcal{F} and that has at most c crossings is FPT w.r.t. c . In particular, we show that for any fixed k , deciding whether a graph is k -planar, k -quasi-planar, fan-crossing, fan-crossing-free or min- k -planar, respectively, is FPT with respect to the corresponding beyond-planar crossing number.

2012 ACM Subject Classification Theory of computation \rightarrow Fixed parameter tractability

Keywords and phrases FPT, Beyond-planarity, Crossing-number, Crossing Patterns

Digital Object Identifier 10.4230/LIPIcs.GD.2024.25

Funding Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 541433306.

1 Introduction

In practice, many graphs and networks are non-planar. Since the presence of crossings in a drawing has been identified as a major influence on the readability and also for its fundamental theoretical importance (see e.g. [38, 40, 39]), the crossing number of graphs, i.e., the minimum number of crossings one can achieve in a topological drawing of a graph, has been the subject of intensive research. It is one of the classical results of Garey and Johnson [24] that computing the crossing number is NP-complete, and this holds even under severe restrictions. Concerning approximation results, though some promising progress has been made, e.g., for graphs of bounded genus [27] and, most recently, for graphs of bounded degree [20], finding a constant-factor approximation currently seems elusive. On the other hand, Grohe [25] showed that the crossing number of a graph can be computed in FPT time, i.e., it can be decided whether a given graph G admits a drawing with at most c crossings in time $f(c)n^{O(1)}$ where f is a computable function and n denotes the size of the input. The polynomial dependency on the input is quadratic for Grohe’s algorithm and was later improved to linear by Kawarabayashi and Reed [32]. Recent years have seen a couple of extensions, particularly of Grohe’s approach. Pelsmajer et al. [36] showed that the approach can be extended to show that the odd crossing number is FPT. Very recently Hamm and Hliněný [26] showed that the problem of extending a partial drawing with the minimum number of crossings is FPT with respect to the achievable number of crossings.

More recent studies on the perception of drawings have somewhat refined the impact of crossings on readability [30, 33]. Rather than simply minimizing their number, there are several factors that influence the readability of drawings with crossings. Aside from geometric aspects, such as crossing angles, it has been identified as particularly relevant that the drawn graphs are sparse and close to planar in the sense that the crossings are well-distributed [29]. This has led to the research field of *beyond-planarity*, which is concerned



© Miriam Münch and Ignaz Rutter;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 25; pp. 25:1–25:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

with the study of graph classes that are defined by the existence of a drawing where certain bad configurations of crossings are avoided. The goal of beyond-planarity is to extend results concerning combinatorics, algorithms and geometric representations known for planar graphs to larger, more general classes of so-called beyond-planar graphs. A prime example are 1-planar graphs that have a drawing where each edge is involved in at most one crossing. By now there is a vast amount of literature that deals with various aspects of beyond-planarity; see e.g. [29, 22, 9]. Unfortunately, even the concept of 1-planarity is quite complicated. While efficient algorithms are known for testing whether a given graph is outer 1-planar [28, 3], testing whether a given graph G is 1-planar is NP-complete [34]. Also recognizing k -planar graphs is NP-hard for every $k \geq 1$ [44] and similar hardness results are known for several beyond-planar graph classes, e.g., k -gap-planar graphs [4, 5], fan-planar graphs and fan-crossing-free graphs [15]. Efficient algorithms are known only for few and rather special cases such as recognizing optimal 1-planar graphs [13], triangulated 1-planar graphs [12, 17] or optimal 2-planar graphs [23].

In light of these hardness results it is natural to consider FPT approaches for recognizing beyond-planar graphs. Bannister et al. [6] showed that testing 1-planarity is FPT when parameterized by the vertex cover number, treedepth or cyclomatic number, but is NP-complete even for graphs of bounded bandwidth, which in particular rules out FPT algorithms with respect to, e.g., pathwidth. Therefore the usefulness of structural parameters for recognizing beyond-planar graphs seems to be fairly restricted. One of the arguably most natural parameters to consider in the context of beyond-planarity is the distance of the input graph G from being planar. There are multiple ways to measure this; the *skewness* of G , denoted by $\text{skew}(G)$ is the minimum number of edges one can remove from G to turn it into a planar graph. The crossing number of G , denoted by $\text{cross}(G)$ is the minimum number of crossings in a topological drawing of G . Finally, given a beyond-planar drawing style \mathcal{D} , such as 1-planar drawings, one may consider the *beyond-planar crossing number* of a graph G , i.e., the minimum number $\text{cross}_{\mathcal{D}}(G)$ of crossings in any planar drawing of G according to style \mathcal{D} . For example, the 1-planar crossing number $\text{cross}_{1\text{-planar}}(G)$ is the minimum number of crossings in any 1-planar drawing of G and it is ∞ if no such drawing exists. Clearly, for any drawing style \mathcal{D} it holds that $\text{skew}(G) \leq \text{cross}(G) \leq \text{cross}_{\mathcal{D}}(G)$. It turns out that, for most beyond-planar graph classes the skewness and the crossing number are not a suitable parameter. By a well-known result of Cabello and Mohar [16] testing 1-planarity is NP-complete even for input graphs G with $\text{skew}(G) = 1$ and $\text{cross}(G) \leq 10$. In other words, testing 1-planarity is paraNP-hard parameterized by both the skewness and the crossing number.

Beyond-planar crossing numbers can deviate significantly from the crossing number [19, 45]. Hence it is natural to ask whether recognizing beyond-planar graphs is FPT with respect to the beyond-planar crossing number. To mention a specific example, for the class of 1-planar graphs: is it possible to test whether an input graph G admits a 1-planar drawing with at most c crossings in FPT time with respect to c ? Indeed, this is the case: the recent work of Hamm and Hliněný [26] shows that given a graph G and a partial drawing of a subgraph $H \subseteq G$, it can be decided in FPT time whether the given partial drawing can be completed into a 1-planar drawing of G with at most c crossings. In particular, if we assume that the partial drawing is empty, their result shows that 1-planarity is FPT by the 1-planar crossing number. The purpose of this paper is to understand which other beyond-planar graph classes admit an analogous result. As it turns out, Grohe's approach [25] for the crossing number, which is also behind the algorithm of Hamm and Hliněný [26] is fairly flexible and seems applicable to several notions of beyond-planarity. However, rather than crafting individual proofs for different classes of beyond-planar graphs, we seek to establish a meta theorem that allows to obtain such results for a wide variety of beyond-planar graph classes.

Large parts of the literature on beyond-planar graphs mention that beyond-planar graph classes are defined by the existence of a drawing that avoids certain crossing configurations or crossing patterns. In this sense, a beyond-planar graph class is defined by a set of forbidden crossing patterns. Interestingly, the idea of what a crossing configuration or pattern is precisely and what it means that a drawing contains such a pattern have not been formalized in the literature. In most papers, the forbidden configuration of crossings is given in a figure but it is usually followed by a more precise verbal definition, which then serves as a basis for theoretical considerations; see e.g. [2, 22, 31].

Contribution. In this paper, we present an attempt at formalizing the idea of forbidden patterns of crossings, where crossing patterns are modeled as small graphs and we seek a drawing whose planarization avoids these graphs. We note that there are various options to design such notions; we believe that our formalization provides an interesting trade-off: it is sufficiently general to express several known beyond-planar graph classes with a small number of patterns and on the other hand is amenable to algorithmic techniques for computing drawings that avoid a fixed list of patterns and have at most c crossings in FPT time. Our notion of patterns is simple and yet powerful enough that our results apply to the following notions of beyond-planar graphs: (simple) k -planar, (simple) k -quasi-planar, (simple) fan-crossing, (simple) fan-crossing free, (simple) min- k -planar.

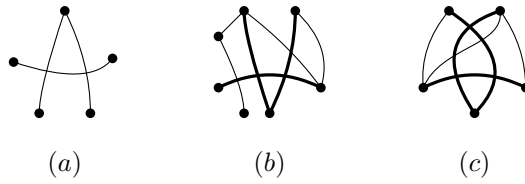
A set \mathcal{F} of forbidden crossing patterns defines a beyond-planar drawing style $\mathcal{D}(\mathcal{F})$, namely the drawings that avoid all patterns in \mathcal{F} . Our main result is the following meta theorem, which states that the beyond-planar crossing number $\text{cross}_{\mathcal{D}(\mathcal{F})}$ is FPT with respect to its natural parameterization.

► **Theorem 1.** *For any fixed set \mathcal{F} of crossing patterns, the problem of testing whether a given graph G admits a drawing with at most c crossings that avoids all patterns in \mathcal{F} is FPT with respect to c .*

In particular, since our notion of patterns is sufficiently general, this yields, among others, FPT results for the beyond-planar crossing number for several beyond-planar graph classes.

► **Corollary 2.** *The beyond-planar crossing number is FPT for the following notions of beyond-planarity and any fixed k : k -planar, k -quasi-planar, fan-crossing, fan-crossing free, min- k -planar.*

Outline. After discussing preliminaries in Section 2, we develop our notion of crossing patterns in Section 3. The next two sections together describe an FPT algorithm for testing the existence of a drawing with at most c crossings that avoids a fixed set \mathcal{F} of crossing patterns. It is based on the approach of Grohe [25] for the crossing number and works in the same two phases. In the first phase (Section 4), we bound the treewidth of the graph in terms of the beyond-planar crossing number, while the second phase solves the problem on graphs of bounded treewidth via Courcelle's theorem [21]. Section 6 presents some generalizations that allow us to cover additional beyond-planar graph classes such as simultaneous planarity and 2-layer drawings with various restrictions. We summarize our findings and discuss open questions in Section 7. Lemmas marked with (\star) are omitted due to space restrictions.



■ **Figure 1** (a) Forbidden configuration in a fan-crossing-free drawing. (b) – (c) Two drawings containing the forbidden configuration.

2 Preliminaries

A *drawing* Γ of a graph $G = (V, E)$ maps every vertex $v \in V$ to a point $\Gamma(v) \in \mathbb{R}^2$ and every edge $uv \in E$ to an open simple Jordan arc with endpoints $\Gamma(u)$ and $\Gamma(v)$ that does not pass through any $\Gamma(w)$ for $w \in V \setminus \{u, v\}$. A crossing is a common interior point of two edges. A graph is *planar* if it can be drawn in the plane without edge crossings. We say that a drawing Γ is *simple* if two edges cross at most once and no two adjacent edges cross.

Let Γ be a drawing of a graph G . The *planarization* of Γ is the drawing Γ' we obtain from Γ by replacing every crossing point by a crossing vertex. In this paper we do not allow self-crossings. For graphs considered in this paper we allow multi-edges and self-loops.

3 Combinatorial Description of Drawings and Crossing Patterns

Beyond-planar graph classes are usually defined via forbidden configurations or patterns in a drawing. Our goal is to formalize this concept on a combinatorial level. As an example, consider the class of *fan-crossing-free graphs*, which contains all graphs that admit a drawing that does not contain two adjacent edges that both cross a third edge. The corresponding forbidden pattern given in the literature (see e.g. [22]) is shown in Fig. 1(a). Clearly the drawing shown in Fig 1(b) contains the forbidden pattern. Similarly, also the drawing shown in Fig 1(c) contains an edge that is crossed by two edges e, f sharing an endpoint and is thus not fan-crossing-free. However, its planarization does not contain a subgraph isomorphic to the planarization of the forbidden pattern. The presence of the pattern is obscured by the additional crossing between the adjacent edges in the drawing. It is therefore natural to require for a drawing D to contain a given pattern P , that crossings that appear within the pattern P must appear also within an occurrence of P in D , whereas crossings that are absent in a pattern P but are present in the drawing can be ignored and do not help to avoid the pattern P .

Now consider 1-planar graphs. A natural forbidden pattern in this graph class is the one shown in Fig. 2(a). However, there are many more (see Fig. 2(b)–(d) for examples), as some of the involved edges may share an endpoint or edges may cross each other multiple times. This leads to the observation that, in many crossing patterns, e.g., for k -planarity, the crossings are of vital importance, whereas the vertices are often not. In fact, often drawing styles only specify the presence of crossings on some edges but are agnostic to whether the endpoints of the involved edges are identical or not and they therefore rather only require that there exists a part of some edge that is involved in certain crossings. To be able to express this, our patterns allow to contain *subdivision vertices* of degree 1, which signify that the corresponding edge of the pattern may be mapped to a part of an edge. In particular, the class of 1-planar graphs is defined by the absence of the single pattern shown in Fig. 2(e).

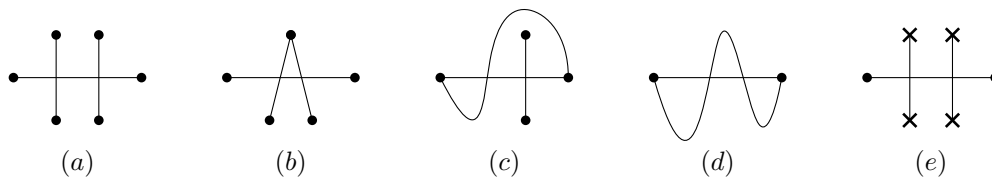


Figure 2 Forbidden patterns in a 1-planar drawing. Crosses represent subdivision vertices.

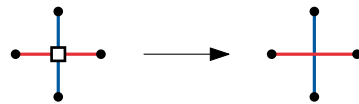


Figure 3 Illustrations of the smoothing operation at a crossing vertex.

► **Definition 3.** A crossing pattern is a graph $P = (V_P, E_P)$ with $V_P = R \cup C \cup S$ such that (i) each subdivision vertex in S has degree 1, (ii) each crossing vertex in C has degree 4 and its incidences are partitioned into two sets of size 2, and (iii) each real vertex in R and each subdivision vertex in S is adjacent to at least one crossing vertex.

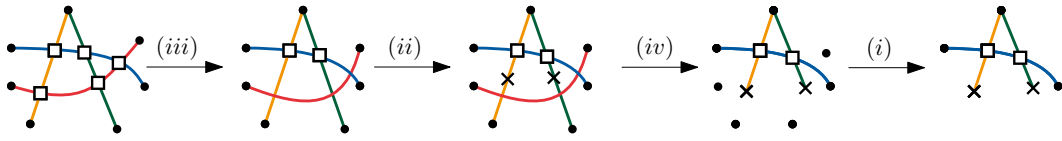
We note that condition (ii) allows us to uniquely identify the two edge segments that form a crossing. In our figures, we usually indicate the pairing by a planar edge-colored drawing where the incidences that are part of the same set in the partition are drawn oppositely in the rotation around the crossing vertex and have the same color. Condition (iii) is a technical condition that we will need later. It is however, a natural assumption, as it ensures that our patterns indeed express conditions on crossing configurations in drawings rather than structural conditions on the input graph.

Next, we need to define when a drawing Δ of a graph G contains a crossing pattern P . A simple idea would be to consider the planarization of Δ and to require that it contains a subgraph isomorphic to P , where crossing and real vertices of P are mapped to crossing and real vertices of the planarization, respectively. This is, however, not sufficient. In addition we want to be able to remove parts of an edge, where if the removed part does not start at a real vertex, we place a subdivision vertex. Finally, as explained above, it is crucial that we can ignore crossings that are present in the drawing but that are not relevant for forming the pattern. To formalize this, we define the concept of a drawing representation.

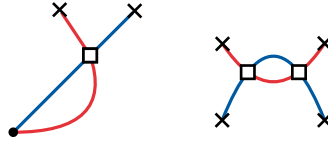
► **Definition 4.** A drawing representation is a graph $D = (V_R, E_R)$ whose vertex set $V_R = R \cup C \cup S$ consists of real vertices in R , crossing vertices in C and subdivision vertices in S such that (i) each subdivision vertex has degree at most 2, (ii) each crossing vertex has degree 4 and its edge incidences are partitioned into two sets of size 2.

Two drawing representations are *isomorphic* if there is an isomorphism between them that maps real vertices to real vertices, crossing vertices to crossing vertices and subdivision vertices to subdivision vertices. Observe that for any drawing Δ of a graph G , the planarization D of Δ is a drawing representation, where the partition of the incidences at each crossing vertex c is defined such that incidences that are opposite in the rotation around the crossing that corresponds to c are in the same set of the partition.

Let c be a crossing vertex in a drawing representation. To be able to ignore crossings, we define the operation of *smoothing* c as (i) inserting edges between those neighbors of c that are endpoints of edges whose incidences are in the same set of the partition for c and (ii) deleting c and its incident edges; see Fig. 3. In addition we allow the removal of isolated



■ **Figure 4** Illustration of the operations allowed on a drawing representation.



■ **Figure 5** Forbidden crossing patterns in simple drawings.

vertices, the addition of subdivision vertices on edges of the drawing representation and the removal of edges. To keep the degree of crossing vertices at 4, we restrict the last operation to edges that are not incident to a crossing vertex.

► **Definition 5.** *Let D, D' be drawing representations. Then D contains D' if and only if a drawing representation isomorphic to D' can be obtained from D by a sequence of the following operations*

- (i) *deleting an isolated vertex,*
- (ii) *subdividing an edge by introducing a subdivision vertex,*
- (iii) *smoothing a crossing vertex,*
- (iv) *deleting an edge that is not incident to a crossing vertex.*

Figure 4 shows an example of a sequence of such operations. We note that, while the planarization of a drawing of a graph G constitutes a drawing representation that is planar, the smoothing operation may turn such a representation into a non-planar graph. Additionally, observe that a crossing pattern as defined above is a drawing representation. In particular, we have now formally defined when a drawing representation contains a crossing pattern. We define that a drawing Δ of a graph G *contains* a crossing pattern P if and only if the planarization of Δ (considered as a drawing representation) contains P . If Δ does not contain P , we also say that Δ *avoids* P .

Note that a drawing of a graph G is simple if and only if it avoids the patterns shown in Fig. 5. The following lemma shows that several widely known beyond-planar drawing styles are captured by our notion of forbidden crossing patterns. For a definition of the mentioned graph-classes we refer to [35, 18, 14, 1, 7, 8]. Let P_k denote the crossing pattern consisting of an edge between two real vertices that is subdivided by $k + 1$ crossing vertices that are all adjacent to two private subdivision neighbors; see Fig. 6(a) for $k = 2$.

► **Lemma 6** (\star). *Let Δ be a drawing of a graph G . Then Δ is*

- (i) *k -planar if and only if it avoids P_k .*
- (ii) *fan-crossing free if and only if it avoids the pattern in Fig. 6(b).*
- (iii) *fan-crossing if and only if it avoids the pattern in Fig. 6(c).*
- (iv) *quasi-planar if and only if it avoids the pattern in Fig. 6(d).*
- (v) *min-1-planar if and only if it avoids the pattern in Fig. 6(e).*

Observe that also the number of forbidden crossing patterns in a k -quasi-planar or min- k -planar drawing is bounded by $f(k)$ for a computable function f , since the number of orders in which k edges can cross each other is bounded. Hence also min- k -planarity and k -quasi-planarity can be expressed by a finite set \mathcal{F} of forbidden patterns for any fixed k .

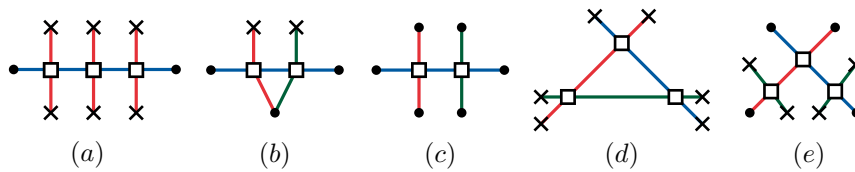


Figure 6 Examples of crossing patterns.

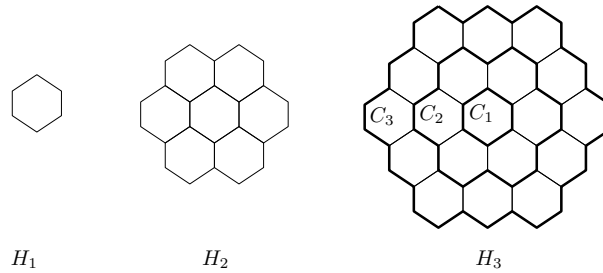


Figure 7 The hexagonal grids H_1, H_2, H_3 and the principal cycles of H_3 .

4 Bounding Treewidth

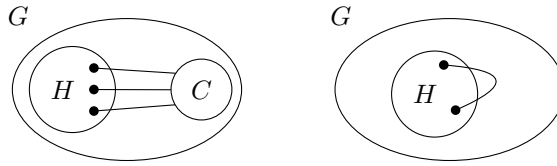
Let $G = (V, E)$ be a graph for the rest of this section and let $F \subseteq E$. For a fixed family \mathcal{F} of crossing patterns, we say that G has an (F, \mathcal{F}, c) -good drawing if there exists a drawing of G that (i) does not contain any pattern in \mathcal{F} , (ii) has at most c crossings and (iii) none of the crossings involve an edge from F . Let $c \geq 1$ be fixed for the rest of this section. The goal of the following two sections is to give an algorithm that solves the following problem in quadratic time. For a fixed family \mathcal{F} of crossing patterns, given a graph G , an integer c and a subset $F \subseteq E$, the question is whether there exists an (F, \mathcal{F}, c) -good drawing of G . Similarly to the algorithm given by Grohe [25] solving the generalized c -crossing number problem in quadratic time, our algorithm works in two phases. First the size of the input graph is reduced iteratively until we get a graph with treewidth bounded by a constant only depending on c . In the second phase the algorithm solves the problem on this graph of bounded treewidth.

To bound the treewidth of our input graph we use Grohe’s approach of an irrelevant-vertex strategy. However, several modifications are necessary, as Grohe only counts crossings, whereas we additionally have to ensure the absence of certain crossing patterns. In particular, Lemmas 10–12 are corresponding adaptations of Lemmas 5–7 in Grohe’s paper [25].

4.1 Fundamentals

Let $G = (V, E)$ and $H = (V', E')$ be two graphs. A *topological embedding* of G into H is a mapping h that maps every vertex $v \in V$ to a vertex in $h(v) \in V'$ and every edge $e \in E$ to a path $h(e)$ in H such that for all $v \neq w \in V$ we have $h(v) \neq h(w)$, for all $e \neq f \in E$ the paths $h(e)$ and $h(f)$ have at most their endpoints in common and for every edge $uv \in E$ the endpoints of $h(e)$ are $h(u)$ and $h(v)$, and $h(w) \notin h(e)$ for all $w \in V \setminus \{u, v\}$.

We denote the *hexagonal grid* with radius r by H_r ; see Figure 7. We number the r concentric cycles C_1, \dots, C_r from the interior to the exterior. Let G be a graph with subgraph H . An H -component of G is either a connected component C of $G - H$ together with all edges between C and H and their endpoints in H , or an edge $uv \in E(G) \setminus E(H)$ such that $u, v \in H$, together with its endpoints; see Figure 8. We call the vertices in



■ **Figure 8** Illustrations of H -components.

the intersection of an H -component with H the *vertices of attachment* of the component. Let $h : H_r \rightarrow G$ be a topological embedding. We call the subgraph $h(H_r \setminus C_r)$ the *interior* of $h(H_r)$. An $h(H_r)$ -component is *proper*, if it has at least one vertex of attachment in the interior of $h(H_r)$. The topological embedding h is *flat* if the union of $h(H_r)$ with all its proper components is planar.

For a fixed family \mathcal{F} of crossing patterns, the \mathcal{F} -crossing number of G is the minimum number of crossings over all drawings of G that do not contain a pattern in \mathcal{F} . The following theorem is due to Thomassen [43] who stated the result for the genus of a graph rather than its \mathcal{F} -crossing number. However, observe that the \mathcal{F} -crossing number of a graph is an upper bound for its genus and thus our modification of the theorem holds.

► **Theorem 7** ([43]). *For all $c, r \geq 1$ there is an $s \geq 1$ such that the following holds. If a graph G has \mathcal{F} -crossing number at most c and $h : H_s \rightarrow G$ is a topological embedding, then there is a subgrid $H_r \subseteq H_s$ such that the restriction $h|_{H_r}$ of h to H_r is flat.*

The following version of the well-known *Excluded Grid Theorem* will be a useful tool.

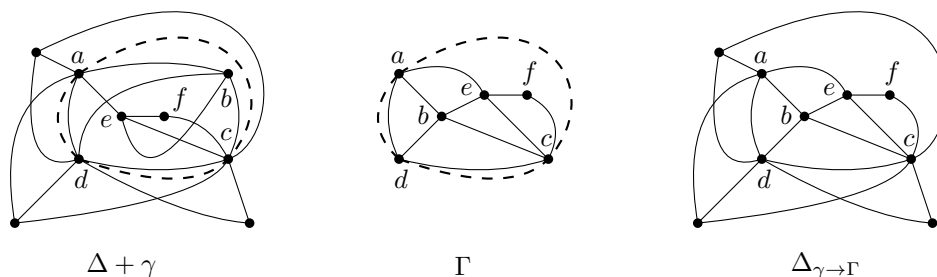
► **Theorem 8** ([25, 41, 11, 37]). *Let $s \geq 1$. Then there is a $w \geq 1$ and a linear-time algorithm that, given a graph G , either (correctly) recognizes that the treewidth of G is at most w or computes a topological embedding $h : H_s \rightarrow G$.*

4.2 The Algorithm

Let $G = (V, E)$ be a graph and let Δ be a drawing of G in the plane. Let γ be a Jordan curve in the plane that intersects Δ only in vertices; we call such a curve a *boundary curve*. Let G_I be the graph formed by the vertices and edges in the interior of γ and the vertices on γ . Let $G_I + \gamma$ denote the graph we obtain from G_I by adding an edge between two vertices u, v if u, v lie on γ and are connected via a subcurve of γ that does not contain other vertices of G_I as interior points. Assume there exists a crossing-free drawing Γ of $G_I + \gamma$ whose outer face is γ . We denote the drawing obtained from Δ by adding γ , replacing its interior with Γ and then removing γ by $\Delta_{\gamma \rightarrow \Gamma}$; see Figure 9. A drawing style \mathcal{D} is *patch-closed* if, whenever $\Delta \in \mathcal{D}$, then also $\Delta_{\gamma \rightarrow \Gamma} \in \mathcal{D}$ for any boundary curve γ and any crossing-free drawing Γ of $G_I + \gamma$.

► **Lemma 9** (*). *Every drawing style \mathcal{D} that is defined by the absence of a finite set of crossing patterns and the requirement that certain edges must not be involved in any crossing, is patch-closed.*

For the rest of this section let c be fixed and let $r := 4c + 3$. This choice becomes relevant in the proof of Lemma 11. Moreover, let s be sufficiently large, such that for every graph of \mathcal{F} -crossing number at most c and every topological embedding $h : H_s \rightarrow G$, there is a subgrid $H_r \subseteq H_s$ such that the restriction $h|_{H_r}$ of h to H_r is flat by Theorem 7. By Theorem 8 there is a w and a linear-time algorithm that, given a graph G of treewidth greater than w , computes a topological embedding $h : H_s \rightarrow G$. Note that w depends on s , which depends on r , which ultimately depends on c . Therefore w depends only on c .



■ **Figure 9** Illustration of patch-closeness. The dashed curve represents γ .

► **Lemma 10** (*). *Let \mathcal{F} be a finite family of crossing patterns and let w, r be fixed as above. There is a linear-time algorithm that, given a graph G , either recognizes that the \mathcal{F} -crossing number of G is greater than c , or recognizes that the treewidth of G is at most w , or computes a flat topological embedding $h: H_r \rightarrow G$.*

By applying this lemma, we either learn that G has treewidth at most w , or that the \mathcal{F} -crossing number of G is greater than c , or we obtain a flat embedding $h: H_r \rightarrow G$. In the first two cases we are done, as we can either move to the second phase or reject the instance immediately. Hence we now assume that we obtain a flat embedding $h: H_r \rightarrow G$. We denote the subgrid of H_r bounded by the second principal cycle C_2 as H^2 . The *kernel* of h is the subgraph of G consisting of $h(H^2)$ and all $h(H_r)$ -components whose vertices of attachment are all in $h(H^2)$. The *boundary of the kernel* is the subgraph of G consisting of $h(C_2)$ and all $h(H_r)$ -components whose vertices of attachment are all in $h(C_2)$. The subgraph containing the kernel without its boundary is called the *interior of the kernel*.

► **Lemma 11.** *Let G be a graph, $F \subseteq E(G)$ and let Δ be an (F, \mathcal{F}, c) -good drawing of G . Let $h: H_r \rightarrow G$ be a flat topological embedding. Then there is an (F, \mathcal{F}, c) -good drawing Δ' of G such that none of the edges of the kernel of h is involved in any crossing of Δ' .*

Proof. For $1 \leq i \leq r-1$, let B_i be the subgraph of G consisting of $h(C_i)$ and all $h(H_r)$ -components whose vertices of attachment are all in $h(C_i)$. Moreover, let R_i be the subgraph of G consisting of the images of the i -th and the $(i+1)$ -th principal cycle $h(C_i), h(C_{i+1})$ and the images of all edges in H_r connecting these two cycles. Observe that for i, j with $1 \leq i < j-1 \leq r-2$, the graphs $R_i \cup B_i$ and $R_j \cup B_j$ are disjoint. Note that at most two edges are involved in any crossing and recall that $r = 4c + 3$. For each crossing that involves two edges e, f where e belongs to B_i and f to B_j , we either have $i = j$ or $|i - j| = 1$. Thus, at most $2c$ distinct B_i 's contain a crossed edge and by the pigeonhole-principle, there is an i_0 with $2 \leq i_0 \leq r-1$ such that none of the edges in $B_{i_0} \cup R_{i_0}$ is involved in any crossing in Δ .

Then, there is a Jordan curve γ around $B_{i_0} \cup R_{i_0}$ that intersects Δ only in vertices of $B_{i_0} \cup R_{i_0}$ such that all edges of $B_{i_0} \cup R_{i_0}$ lie in the interior of γ . Let G_I be the graph consisting of the vertices and edges in the interior of γ . Since h is flat, there exists a planar drawing Γ of $G_I + \gamma$. Moreover, since (F, \mathcal{F}, c) -good drawings are patch-closed by Lemma 9, also $\Delta' = \Delta_{\gamma \rightarrow \Gamma}$ is (F, \mathcal{F}, c) -good. ◀

► **Lemma 12.** *There is a linear-time algorithm that, given a graph G and an edge set $F \subseteq E(G)$, either recognizes that the \mathcal{F} -crossing number is greater than c or recognizes that the treewidth of G is at most w or computes a graph G' and an edge set $F' \subseteq E(G')$ with $|V(G')| < |V(G)|$ such that G has an (F, \mathcal{F}, c) -good drawing if and only if G' has an (F', \mathcal{F}, c) -good drawing.*

Proof. We apply the algorithm of Lemma 10. If the \mathcal{F} -crossing number of G is greater than c or the treewidth of G is at most w we are done. Otherwise the algorithm returns a flat topological embedding $h : H_r \rightarrow G$. Let K be the kernel of h , I its interior, and B its boundary. Let G' be the graph obtained from G by contracting the connected subgraph I to a single vertex v_I . Let F' be the subset of F that is not contracted in G' together with the set of all edges of B and all edges incident to the new vertex v_I . Observe that G' and F' can be computed in linear time and $|V(G')| < |V(G)|$. It remains to show that G has an (F, \mathcal{F}, c) -good drawing if and only if G' has an (F', \mathcal{F}, c) -good drawing.

First assume that G has an (F, \mathcal{F}, c) -good drawing Δ . By Lemma 11 there exists an (F, \mathcal{F}, c) -good drawing Δ' such that none of the edges of the kernel of h is involved in any crossing of Δ' . Let Γ be the drawing of G' we obtain from Δ' by contracting the kernel of h into a single vertex v_I . Clearly, all edges of F' are uncrossed in Δ' . Note that since Δ' does not contain any pattern in \mathcal{F} , if Γ contains such a pattern $P \in \mathcal{F}$, then the occurrence of P must contain v_I . However, since all edges incident to v_I are crossing-free in Γ , v_I cannot be contained in an occurrence of a crossing pattern by property (iii) of crossing patterns.

Conversely, assume that G' has an (F', \mathcal{F}, c) -good drawing Γ . By the choice of F' , the wheel induced by the vertices of B and v_I is crossing-free in Γ . Since H is flat, there exists a planar drawing of the kernel with the vertices of B on the outer face and in the same order as they appear in Γ . Hence the planar graph I can be embedded into a small neighborhood of $\Gamma(v_I)$. Again, this does not introduce new crossing patterns, since all edges incident to a vertex in I are uncrossed and no edge in F is involved in any crossing in the resulting drawing of G . Thus we obtain an (F, \mathcal{F}, c) -good drawing of G . \blacktriangleleft

By repeatedly applying the algorithm from Lemma 12 to an input graph G , we either eventually obtain an equivalent instance whose \mathcal{F} -crossing number exceeds c , or we arrive at an equivalent instance that has bounded treewidth. In the former case, we can reject the instance, in the latter case we move on to Phase 2.

► **Corollary 13.** *There is a quadratic-time algorithm that, given a graph G , either recognizes that the \mathcal{F} -crossing number of G is greater than c or computes a graph G' and an edge set $F' \subseteq E(G)$ such that the treewidth of G' is at most w and G has an (F, \mathcal{F}, c) -good drawing if and only if G' has an (F', \mathcal{F}, c) -good drawing.*

5 Bounded Treewidth and Crossings

Now we describe Phase 2 of our algorithm. If in Phase 1 we do not find out that the \mathcal{F} -crossing number of G is greater than c , we end up with a graph G' of treewidth at most w , where w depends only on c , and an edge set $F' \subseteq E$ such that G has an (F, \mathcal{F}, c) -good drawing if and only if G' has an (F', \mathcal{F}, c) -good drawing. Now we use Courcelle's Theorem to show that we can decide in quadratic time whether G' has an (F', \mathcal{F}, c) -good drawing. To this end, we describe an MSO_2 -formula $\varphi_{\mathcal{F}}(F)$ such that $G \models \varphi_{\mathcal{F}}(F)$ if and only if G has an (F, \mathcal{F}, c) -good drawing. As drawings of graphs cannot be expressed in MSO_2 , we rather express this as the existence of a suitable planarization of G , or in fact rather a drawing representation that is planar.

We use the following primitive formulas as basic building blocks. The formula *conn* returns whether a vertex set $X \subseteq V$ is connected by edges from a set $E' \subseteq E$, *deg0*, *deg1*, *deg2* express that a given vertex x has degree 0, 1 or 2, respectively, in the subgraph formed by the edges in the set E' , *disjoint* expresses whether two edge sets or vertex sets are disjoint, *minor_H* expresses whether G contains a fixed graph H as a minor and *planar* expresses whether G is planar.

Let $G = (V, E)$ be a graph and let $F \subseteq E$ be a set of edges that must not receive a crossing. Assume we want to find a drawing representation with at most c crossings. Our first step is to construct from G a new graph G^+ by subdividing each edge in $E \setminus F$ with c *crossing dummies*. The idea is that a planarization of a drawing of G with at most c crossings can be described by identifying at most c pairs of crossing dummies of G^+ so that the resulting graph is planar. For the treatment of crossing patterns, we need to be able to place subdivision vertices on the edges of such a planarization. To enable this, we further modify G^+ into a graph G^* where we subdivide each edge that is incident to a crossing dummy by two *subdivision dummies*. The vertex set of G^* is $V \cup C \cup S$, where V contains the vertices of G , C contains the crossing dummies and S contains the subdivision dummies.

Expressing a Drawing Representation. The formula $\varphi_{\mathcal{F}}$ we construct applies to the graph G^* and receives as input the vertex sets V, C, S and $2c$ free variables $x_1, \dots, x_c, y_1, \dots, y_c$. The idea is that their values shall be vertices from C and the intended meaning is that the crossing dummy x_i is identified with the crossing dummy y_i to form an actual crossing. By smoothing all subdivision dummies and all crossing dummies that were not identified with another crossing dummy we can consider the result as a drawing representation, whose real vertices are the vertices of G and whose crossing vertices are formed by pairs of crossing dummies that were identified. It therefore makes sense to require that the values of the x_i are pairwise distinct, with the single exception that we allow $x_i = y_i$ to allow the use of fewer than c crossings. We express these requirements with the following formula.

$$\text{distinctCrossings} = \bigwedge_{i=1}^c \bigwedge_{j \neq i} (x_i \neq x_j) \wedge (x_i \neq y_j) \wedge (y_i \neq y_j)$$

As we work with G^* rather than the original graph G , paths whose interior vertices are all subdivision or crossing dummies play a particular role as such paths correspond to a part of an edge of G . The following predicate expresses that an edge set E' forms a path between two vertices x and y , whose interior vertices, if any, are subdivision or crossing dummies.

$$\begin{aligned} \text{path}(E', x, y) = & \exists I \subseteq C \cup S \text{ conn}(I, E') \wedge \forall v \in I \text{ deg}2(v, E') \wedge \\ & \forall v \in V (v \notin I \wedge v \neq x \wedge v \neq y \rightarrow \text{deg}0(v)) \wedge \text{deg}1(x, E') \wedge \text{deg}1(y, E') \end{aligned}$$

Using this we can easily formulate the requirement of having no self-crossings.

$$\text{noSelfCrossings} = \bigwedge_{i=1}^c \forall E' \subseteq E \neg \text{path}(E', x_i, y_i)$$

Constraining the Drawing Representation. Let G^\times denote the graph obtained from G^* by identifying x_i with y_i for $i \in \{1, \dots, c\}$. In what follows we want to express properties of G^\times in an MSO_2 -formula. However, our formula does not have access to the graph G^\times , instead we have to express all conditions in terms of the original graph G^* . A vertex $v \in V(G^\times)$ is incident to an edge $e \in E(G^\times)$ if and only if v is incident to e in G^* or if e is incident to a vertex v' that is identified with v in G^\times . This incidence relation in G^\times can be expressed as follows.

$$\text{inc}^\times(v, e) = \text{inc}(v, e) \vee \left(\bigvee_{i=1}^c (\text{inc}(x_i, e) \wedge v = y_i) \vee (\text{inc}(y_i, e) \wedge v = x_i) \right)$$

25:12 Parameterized Algorithms for Beyond-Planar Crossing Numbers

Now any formula φ that expresses a graph property can be transformed into a formula φ^\times that expresses that G^\times has the same property by replacing each occurrence of inc in φ by inc^\times and replacing any comparison of vertices $a = b$ by the expression $a = b \vee \bigvee_{i=1}^c ((x_i = a \wedge y_i = b) \vee (x_i = b \wedge y_i = a))$. In particular, planar^\times expresses that the graph G^\times is planar.

Avoiding Patterns. Assume x_1, \dots, x_c and y_1, \dots, y_c are chosen such that they satisfy $\text{distinctCrossings} \wedge \text{noSelfCrossings} \wedge \text{planar}^\times$. Then smoothing the subdivision dummies and the unused crossing dummies defines a planarization of a drawing of G with at most c crossings and hence a drawing representation of such a drawing. Checking whether this representation contains a fixed crossing pattern P would possibly require placing subdivision vertices. However, it is never necessary to place more than two subdivision vertices on an edge of the planarization, and therefore by construction G^* contains already a sufficient number of subdivision dummies that can be used as subdivision vertices. We also do not remove edges and vertices but rather specify the vertices and edges that we keep. Moreover, by working in the graph G^* , all crossings are automatically smoothed and we rather express explicitly, which crossings we want to keep.

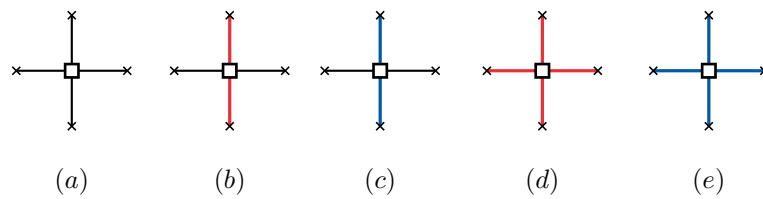
Let $P = (R_P \cup C_P \cup S_P, E_P)$ be a fixed crossing pattern. A *chain* in P is a path p that connects two vertices from $R_P \cup S_P$, whose interior vertices are in C_P , and for each interior vertex c of p the incidences with the two edges in p are in the same set of the partition of incidences of c . Note that there is a unique set of chains that cover P exactly, see Fig. 6, which we call the *chain cover* of P .

To express the presence of the pattern P , we choose a subdivision dummy in G^* for each subdivision vertex of P , a real vertex in G^* for each real vertex of P , as well as, for each chain c in the chain cover of P , a crossing dummy in G^* for each crossing vertex of p and a set of edges in G^* . Observe that for each crossing vertex we thus choose two crossing dummies, one for each of the two chains of the chain cover that contain it. We then express that the edge sets chosen for the chains are pairwise disjoint and for each chain p the corresponding chosen edge set forms a path in G^* that connects the two vertices chosen for the endpoints of p , does not contain real vertices in its interior and visits the crossing dummies chosen for its crossing vertices in the correct order. Finally, we express that all crossings of P are actually present by specifying for each crossing vertex c of P that the two chains p_1, p_2 of the chain cover that contain c have chosen a pair of crossing dummies for c that is identified by the choices of the x_i s and y_i s.

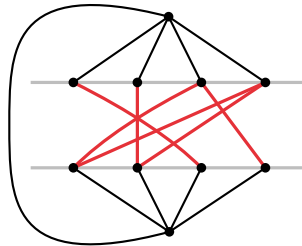
► **Lemma 14** (\star). *For every crossing pattern P there exists an MSO_2 -formula φ_P that determines for the graph G^* whether the graph G^\times defined by identifying the crossing dummies x_i and y_i for $i \in \{1, \dots, c\}$ contains P . The size of φ_P depends only on c and the size of P .*

Using the formula $\varphi_{\mathcal{F}} = \text{planar}^\times \wedge \bigwedge_{P \in \mathcal{F}} \neg \varphi_P$ it then follows that G admits an (F, \mathcal{F}, c) -good drawing if and only if $G^* \models \varphi_{\mathcal{F}}$. Observe that, since G^* is obtained from G by subdividing edges, it has bounded treewidth. Moreover, the size of $\varphi_{\mathcal{F}}$ depends only on the number of crossings c and the size of the patterns in \mathcal{F} . Using Courcelle's theorem, we obtain the following result, which together with Corollary 13 implies our main result Theorem 1.

► **Theorem 15.** *Let \mathcal{F} be a family of crossing patterns and let c be an integer. Let $G = (V, E)$ be a graph of treewidth at most c and let $F \subseteq E$. Then there is an algorithm that decides in quadratic FPT-time whether G admits an (F, \mathcal{F}, c) -good drawing.*



■ **Figure 10** Forbidden crossing patterns for SEFE.



■ **Figure 11** Sketch of the reduction for the 2-layered setting.

6 Extensions and Further Applications

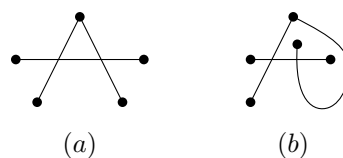
Our notion of drawing representations and crossing patterns can easily be extended to graphs whose edges are colored with a fixed number of colors. In this setting, two drawing representations are isomorphic if there is an isomorphism between them that for every color c maps c -colored edges to c -colored edges. As Phase 1 of our algorithm works irrespective of such a coloring and the MSO_2 formula from Phase 2 can be straightforwardly adapted to take into account the colors of the edges, Theorem 1 also holds for colored graphs and colored crossing patterns.

This enables several additional applications. For a graph whose edges are red and black, we can express that black edges must not be crossed by forbidding the crossing patterns in Figure 10(a), (b). By employing the reduction sketched in Figure 11 and further patterns on the red edges, this allows to compute the beyond-planar crossing number of 2-layered drawings in FPT time for all beyond-planar graph classes that can be expressed in terms of our crossing patterns.

► **Corollary 16.** *For any fixed set \mathcal{F} of crossing patterns testing whether a given bipartite graph admits a 2-layer drawing with at most c crossings that avoids all patterns in \mathcal{F} is FPT with respect to c .*

Considering graphs whose edges are black, red, or blue, and allowing only crossings between red and blue edges by forbidding the patterns in Fig. 10(a) – (e) allows to model the famous SEFE problem [10, 42].

► **Corollary 17.** *Testing whether a pair of graphs (G_1, G_2) admits a SEFE with at most c crossings is FPT with respect to c .*



■ **Figure 12** Two drawings that both contain the pattern shown in Fig. 6(b).

7 Conclusion

We introduced a combinatorial definition of a crossing pattern and showed that for every beyond-planar graph class \mathcal{C} that is defined by the absence of a finite number of such patterns, deciding whether a graph is in \mathcal{C} is FPT with respect to the corresponding beyond-planar crossing number. Our results are applicable to many beyond-planar graph classes, the ones we mentioned in the paper are only an extract.

However, since our current definition of crossing patterns does not take topological aspects into account, it is not able to distinguish the two drawings shown in Fig. 12. Recall that fan-planar graphs are defined by the absence of the pattern shown in Fig. 6(c) and the configuration in Fig. 12(b). Thus an interesting open question is, whether our notion of crossing patterns can be adapted such that also graph classes that rely on topological properties such as fan-planar or geometric k -planar graphs can be handled.

Another open question is whether bounding the treewidth also works for a definition of crossing patterns without condition (iii); i.e., if we do not require that real and subdivision vertices are adjacent to crossing vertices.




References

- 1 Pankaj K Agarwal, Boris Aronov, János Pach, Richard Pollack, and Micha Sharir. Quasi-planar graphs have a linear number of edges. *Combinatorica*, 17(1):1–9, 1997. doi:10.1007/BF01196127.
- 2 Patrizio Angelini, Michael A Bekos, Franz J Brandenburg, Giordano Da Lozzo, Giuseppe Di Battista, Walter Didimo, Giuseppe Liotta, Fabrizio Montecchiani, and Ignaz Rutter. On the relationship between k -planar and k -quasi-planar graphs. In *Graph-Theoretic Concepts in Computer Science: 43rd International Workshop, WG 2017*, pages 59–74. Springer, 2017. doi:10.1007/978-3-319-68705-6_5.
- 3 Christopher Auer, Christian Bachmaier, Franz J. Brandenburg, Andreas Gleißner, Kathrin Hanauer, Daniel Neuwirth, and Josef Reislhuber. Outer 1-planar graphs. *Algorithmica*, 74(4):1293–1320, 2016. doi:10.1007/s00453-015-0002-1.
- 4 Sang Won Bae, Jean-François Baffier, Jinhee Chun, Peter Eades, Kord Eickmeyer, Luca Grilli, Seok-Hee Hong, Matias Korman, Fabrizio Montecchiani, Ignaz Rutter, and Csaba D. Tóth. Gap-planar graphs. In *Graph Drawing and Network Visualization - 25th International Symposium, GD 2017*, volume 10692 of *Lecture Notes in Computer Science*, pages 531–545. Springer, 2017. doi:10.1007/978-3-319-73915-1_41.
- 5 Sang Won Bae, Jean-François Baffier, Jinhee Chun, Peter Eades, Kord Eickmeyer, Luca Grilli, Seok-Hee Hong, Matias Korman, Fabrizio Montecchiani, Ignaz Rutter, and Csaba D. Tóth. Gap-planar graphs. *Theoretical Computer Science*, 745:36–52, 2018. doi:10.1016/J.TCS.2018.05.029.
- 6 Michael J. Bannister, Sergio Cabello, and David Eppstein. Parameterized complexity of 1-planarity. *Journal of Graph Algorithms and Applications*, 22(1):23–49, 2018. doi:10.7155/jgaa.00457.

- 7 Carla Binucci, Aaron Büngener, Giuseppe Di Battista, Walter Didimo, Vida Dujmovic, Seok-Hee Hong, Michael Kaufmann, Giuseppe Liotta, Pat Morin, and Alessandra Tappini. Min- k -planar drawings of graphs. In *Graph Drawing and Network Visualization - 31st International Symposium, GD 2023*, volume 14465 of *Lecture Notes in Computer Science*, pages 39–52. Springer, 2023. doi:10.1007/978-3-031-49272-3_3.
- 8 Carla Binucci, Aaron Büngener, Giuseppe Di Battista, Walter Didimo, Vida Dujmovic, Seok-Hee Hong, Michael Kaufmann, Giuseppe Liotta, Pat Morin, and Alessandra Tappini. Min- k -planar drawings of graphs. *J. Graph Algorithms Appl.*, 28(2):1–35, 2024. doi:10.7155/JGAA.V28I2.2925.
- 9 Carla Binucci, Emilio Di Giacomo, Walter Didimo, Fabrizio Montecchiani, Maurizio Patrignani, Antonios Symvonis, and Ioannis G Tollis. Fan-planarity: Properties and complexity. *Theoretical Computer Science*, 589:76–86, 2015. URL: <https://www.sciencedirect.com/science/article/pii/S0304397515003515>, doi:10.1016/J.TCS.2015.04.020.
- 10 Thomas Bläsius, Stephen G. Kobourov, and Ignaz Rutter. Simultaneous embedding of planar graphs. In Roberto Tamassia, editor, *Handbook on Graph Drawing and Visualization*, pages 349–381. Chapman and Hall/CRC, 2013.
- 11 Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996. doi:10.1137/S0097539793251219.
- 12 Franz J Brandenburg. On 4-map graphs and 1-planar graphs and their recognition problem. *arXiv preprint*, 2015. arXiv:1509.03447.
- 13 Franz J. Brandenburg. Recognizing optimal 1-planar graphs in linear time. *Algorithmica*, 80(1):1–28, 2018. doi:10.1007/s00453-016-0226-8.
- 14 Franz J. Brandenburg. On fan-crossing graphs. *Theoretical Computer Science*, 841:39–49, 2020. doi:10.1016/j.tcs.2020.07.002.
- 15 Franz J Brandenburg. Fan-crossing free graphs and their relationship to other beyond-planar graphs. *Theoretical Computer Science*, 867:85–100, 2021. doi:10.1016/J.TCS.2021.03.031.
- 16 Sergio Cabello and Bojan Mohar. Adding one edge to planar graphs makes crossing number and 1-planarity hard. *SIAM Journal on Computing*, 42(5):1803–1829, 2013. doi:10.1137/120872310.
- 17 Zhi-Zhong Chen, Michelangelo Grigni, and Christos H Papadimitriou. Recognizing hole-free 4-map graphs in cubic time. *Algorithmica*, 45(2):227–262, 2006. URL: <https://link.springer.com/content/pdf/10.1007/s00453-005-1184-8.pdf>, doi:10.1007/S00453-005-1184-8.
- 18 Otfried Cheong, Sarel Har-Peled, Heuna Kim, and Hyo-Sil Kim. On the number of edges of fan-crossing free graphs. *Algorithmica*, 73(4):673–695, 2015. doi:10.1007/S00453-014-9935-2.
- 19 Markus Chimani, Max Ilsen, and Tilo Wiedera. Star-struck by fixed embeddings: Modern crossing number heuristics. In *Graph Drawing and Network Visualization - 29th International Symposium, GD 2021*, volume 12868 of *Lecture Notes in Computer Science*, pages 41–56. Springer, 2021. doi:10.1007/978-3-030-92931-2_3.
- 20 Julia Chuzhoy and Zihan Tan. A subpolynomial approximation algorithm for graph crossing number in low-degree graphs. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 303–316, 2022. doi:10.1145/3519935.3519984.
- 21 Bruno Courcelle. The monadic second-order theory of graphs I: Recognizable sets of finite graphs. *Information and Computation*, 1990.
- 22 Walter Didimo, Giuseppe Liotta, and Fabrizio Montecchiani. A survey on graph drawing beyond planarity. *ACM Computing Surveys*, 52(1):4:1–4:37, 2019. doi:10.1145/3301281.
- 23 Henry Förster, Michael Kaufmann, and Chrysanthi N Raftopoulou. Recognizing and embedding simple optimal 2-planar graphs. In *Graph Drawing and Network Visualization - 29th International Symposium, GD 2021*, pages 87–100. Springer, 2021. doi:10.1007/978-3-030-92931-2_6.
- 24 Michael R Garey and David S Johnson. Crossing number is NP-complete. *SIAM Journal on Algebraic Discrete Methods*, 4(3):312–316, 1983.
- 25 Martin Grohe. Computing crossing numbers in quadratic time. *Journal of Computer and System Sciences*, 68(2):285–302, 2004. doi:10.1016/j.jcss.2003.07.008.

- 26 Thekla Hamm and Petr Hliněný. Parameterised partially-predrawn crossing number. In Xavier Goaoc and Michael Kerber, editors, *38th International Symposium on Computational Geometry (SoCG '22)*, volume 224 of *LIPICs*, pages 46:1–46:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICs.SocG.2022.46.
- 27 Petr Hliněný and Markus Chimani. Approximating the crossing number of graphs embeddable in any orientable surface. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 918–927. SIAM, 2010.
- 28 Seok-Hee Hong, Peter Eades, Naoki Katoh, Giuseppe Liotta, Pascal Schweitzer, and Yusuke Suzuki. A linear-time algorithm for testing outer-1-planarity. *Algorithmica*, 72(4):1033–1054, 2015. doi:10.1007/s00453-014-9890-8.
- 29 Seok-Hee Hong and Takeshi Tokuyama, editors. *Beyond Planar Graphs*. Springer Singapore, 2020. doi:10.1007/978-981-15-6533-5.
- 30 Weidong Huang, Seok-Hee Hong, and Peter Eades. Effects of crossing angles. In *IEEE VGTC Pacific Visualization Symposium 2008, PacificVis 2008, Kyoto, Japan, March 5-7, 2008*, pages 41–46. IEEE, 2008. doi:10.1109/PACIFICVIS.2008.4475457.
- 31 Michael Kaufmann and Torsten Ueckerdt. The density of fan-planar graphs. *The Electronic Journal of Combinatorics*, 29(1), 2022. doi:10.37236/10521.
- 32 Ken-ichi Kawarabayashi and Bruce Reed. Computing crossing number in linear time. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 382–390, 2007. doi:10.1145/1250790.1250848.
- 33 Stephen G Kobourov, Sergey Pupyrev, and Bahador Saket. Are crossings important for drawing large graphs? In *Graph Drawing: 22nd International Symposium, GD 2014*, pages 234–245. Springer, 2014. doi:10.1007/978-3-662-45803-7_20.
- 34 Vladimir P Korzhik and Bojan Mohar. Minimal obstructions for 1-immersions and hardness of 1-planarity testing. *Journal of Graph Theory*, 72(1):30–71, 2013. URL: <https://onlinelibrary.wiley.com/doi/epdf/10.1002/jgt.21630>, doi:10.1002/JGT.21630.
- 35 János Pach and Géza Tóth. Graphs drawn with few crossings per edge. *Combinatorica*, 17(3):427–439, 1997. doi:10.1007/3-540-62495-3_59.
- 36 Michael J Pelsmajer, Marcus Schaefer, and Daniel Štefankovič. Crossing numbers and parameterized complexity. In *Graph Drawing: 15th International Symposium, GD 2007.*, pages 31–36. Springer, 2008.
- 37 Ljubomir Perkovic and Bruce A. Reed. An improved algorithm for finding tree decompositions of small width. *Int. J. Found. Comput. Sci.*, 11(3):365–371, 2000. doi:10.1142/S0129054100000247.
- 38 H. C. Purchase, R. F. Cohen, and M. I. James. An experimental study of the basis for graph drawing algorithms. *ACM Journal of Experimental Algorithmics*, 2:4, January 1997. doi:10.1145/264216.264222.
- 39 Helen C. Purchase. Which aesthetic has the greatest effect on human understanding? In Giuseppe Di Battista, editor, *Proceedings 5th International Symposium on Graph Drawing (GD '97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 248–261. Springer, 1997. doi:10.1007/3-540-63938-1_67.
- 40 Helen C. Purchase, Jo-Anne Alder, and David A. Carrington. Graph layout aesthetics in UML diagrams: User preferences. *Journal of Graph Algorithms and Applications*, 6(3):255–279, 2002. doi:10.7155/jgaa.00054.
- 41 Neil Robertson and Paul D. Seymour. Graph minors .xiii. the disjoint paths problem. *J. Comb. Theory, Ser. B*, 63(1):65–110, 1995. doi:10.1006/jctb.1995.1006.
- 42 Ignaz Rutter. Simultaneous embedding. In *Beyond Planar Graphs: Communications of NII Shonan Meetings*, pages 237–265. Springer, 2020. doi:10.1007/978-981-15-6533-5_13.
- 43 Carsten Thomassen. A simpler proof of the excluded minor theorem for higher surfaces. *J. Comb. Theory, Ser. B*, 70(2):306–311, 1997. doi:10.1006/jctb.1997.1761.
- 44 John C Urschel and Jake Wellens. Testing gap k-planarity is NP-complete. *Information Processing Letters*, 169:106083, 2021. doi:10.1016/J.IPL.2020.106083.
- 45 Nathan van Beusekom, Irene Parada, and Bettina Speckmann. Crossing numbers of beyond-planar graphs revisited. In *29th International Symposium on Graph Drawing and Network Visualization (GD '21)*, volume 12868, page 447. Springer Nature, 2021.

Storylines with a Protagonist

Tim Hegemann   

Universität Würzburg, Germany

Alexander Wolff  

Universität Würzburg, Germany

Abstract

Storyline visualizations show interactions between a given set of characters over time. Each *character* is represented by an x-monotone curve. A *meeting* is represented by a vertical bar that is crossed by the curves of exactly those characters that participate in the meeting. Therefore, character curves may have to cross each other. In the context of publication networks, we consider storylines where the characters are authors and the meetings are joint publications. We are especially interested in visualizing a group of colleagues centered around an author, the *protagonist*, who participates in all selected publications. For such instances, we propose a drawing style where the protagonist’s curve is drawn at a prominent position and never crossed by any other author’s curve.

We consider two variants of storylines with a protagonist. In the *one-sided* variant, the protagonist is required to be drawn at the top position. In this restricted setting, we can efficiently compute a drawing with the minimum number of *pairwise crossings*, whereas we show that it is NP-hard to minimize the number of *block crossings* (i.e., pairs of blocks of parallel curves that intersect each other). In the *two-sided* variant, the task is to split the set of co-authors of the protagonist into two groups, and to place the curves of one group above and the curves of the other group below the protagonist’s curve such that the total number of (block) crossings is minimized.

As our main result, we present an algorithm for bundling a sequence of pairwise crossings into a sequence of few block crossings (in the absence of meetings). It exploits a connection to a rectangle dissection problem. In the presence of meetings, it yields results that are very close to a lower bound. Based on this bundling algorithm and our exact algorithm for the one-sided variant, we present a new heuristic for computing two-sided storylines with few block crossings.

We perform an extensive experimental study using publication data of 81 protagonists from GD 2023 and their most frequent collaborators over the last ten years. Our study shows that, for two-sided storylines with a protagonist, our new heuristic uses fewer block crossings (and fewer pairwise crossings) than two heuristics for block crossing minimization in general storylines.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Human-centered computing → Graph drawings

Keywords and phrases Storyline visualization, storyline with a protagonist, crossing minimization, block crossings

Digital Object Identifier 10.4230/LIPIcs.GD.2024.26

Supplementary Material *Software (Source Code)*: <https://github.com/hegetim/publines> [14]
archived at `swh:1:dir:8b6a0dd881ed0b7a4b0e5d8dbfa38f82827aa641`

Funding *Tim Hegemann*: Supported by BMBF grant 01IS22012C.

Acknowledgements We thank Felix Klesen for many discussions and Tim Herrmann for implementing the first version of GreedyBlocks. We also thank the reviewers of this paper for many helpful comments.

1 Introduction

Storyline visualizations are a visual tool to convey information about interactions between a group of entities – usually people – over time. Arguably, storyline visualizations have a long history that was started by Minard’s startling visualization of Napoleon’s Russian campaign [24], but certainly they have become quite popular since Munroe [26] used them to cleverly visualize several cinema classics. Our use case is different; see Figure 1.



© Tim Hegemann and Alexander Wolff;

licensed under Creative Commons License CC-BY 4.0

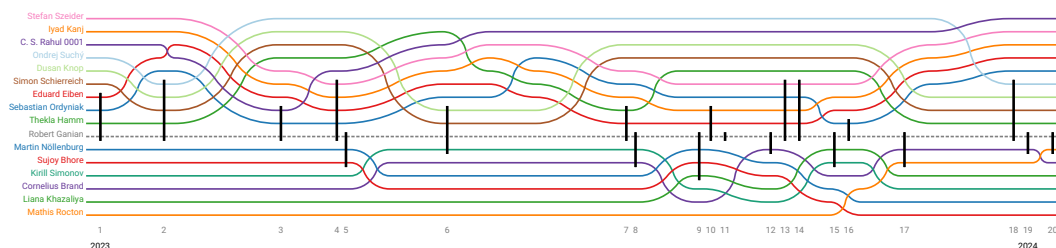
32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 26; pp. 26:1–26:22

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** Storyline visualization of Robert Galian and the 15 coauthors with whom he published most in the last year. In our drawing style, the protagonist (dashed line) has a prominent position – their curve is not crossed by any other curve. Make your own at <https://publines.github.io>!

More formally, a storyline \mathcal{S} is a pair (C, M) , where $C = \{1, \dots, k\}$ is a set of *characters* and $M = [m_1, \dots, m_n]$ is a sequence of *meetings*. If $1 \leq i < j \leq n$, we say that meeting m_i takes place before meeting m_j or that m_j is later than m_i . Every meeting is a subset of C of size at least 1. Note that we do not encode at what time a meeting happens exactly. In a *storyline visualization*, each character is represented by a continuous x-monotone curve. Let $i \in \{1, \dots, n\}$. At meeting $m_i \subseteq C$, the characters that participate in the meeting have to form an interval in the vertical order of the character curves. The meeting m_i is represented by a vertical line segment s_i at x-coordinate x_i such that (i) exactly the character curves in m_i cross s_i and (ii) $x_i < x_j$ if $i < j$. A storyline visualization thus maps every point $t \in [x_1, x_n]$ in time to a vertical order π_t of the characters.

In order to measure the quality of storyline visualizations (and to eventually design algorithms that produce readable drawings), various esthetic criteria have been suggested. Most works have focused on reducing the number of crossings of the character curves (simple pairwise crossings [12] or more complex types of crossings [31, 32]), others have also tried to minimize the number of wiggles [11] (that is, the number of turns) and/or the amount of vertical white-space [29, 23].

In this paper, we investigate a new variant of storylines that is motivated by the visualization of coauthor networks centered around a given main author or *protagonist*, that is, a character that is part of every meeting in a storyline. In order to stress the role of the protagonist, we disallow other curves to cross the protagonist’s curve which can thus be drawn as a straight horizontal line. We consider two variants concerning the placement of the protagonist: in the one-sided variant, the line of the protagonist is simply placed topmost, whereas in the two-sided variant, the line of the protagonist splits the other characters into two groups; those above and those below the line (see Figure 2). In both variants, we focus on minimizing crossings (but, in our experiments, we also keep track of the number of wiggles).

As we will show, minimizing pairwise crossings is easy in the one-sided variant. Therefore, we try to group these simple crossings into larger units, so-called *block crossings*. In a block crossing, two groups of curves are exchanged in the vertical ordering while no two curves within a group change their order. Such a grouping underlines the structure of the intersection pattern and leads to less visual clutter. Block crossings have also been investigated thoroughly beyond storyline visualization; in edge bundling [8], in metro maps [10], and due to their connection to the genus of a graph [3, 4, 5].

Our contribution. We introduce and formally define four variants of the new problem storyline crossing minimization with a protagonist; see Section 2. We show that one-sided storyline crossing minimization with a protagonist (1-SCM-P) can be solved efficiently; see Section 3. For the two-sided version of the problem (2-SCM-P), the characters have to

be split into two groups, which are then drawn above and below the horizontal line that represents the protagonist – using the algorithm for the one-sided case. We reduce the splitting problem to a max-cut problem and solve it heuristically; see Section 4. We then show that both variants of the storyline block crossing minimization problem (1-SBCM-P and 2-SBCM-P) are NP-hard; see Section 5.

In order to group pairwise crossings into blocks, we exploit a connection to partitioning the cells of crossing complexes into rectangular groups of cells that was observed by Fink et al. [8] in a more general topological setting. Note that in a purely geometric setting, the partitioning problem (namely of a simple orthogonal polygon into the minimum number of rectangles) can be solved efficiently, even in the presence of point holes [28]. In order to benefit from this, our bundling algorithm modifies the crossing complex heuristically such that it can be partitioned using algorithms for the geometric setting.

In our extensive experimental study we use publication data of 81 protagonists from GD 2023 and their most frequent collaborators over the last ten years; see Section 8. Somewhat unsurprisingly, a comparison shows that, for storylines with a protagonist, our specialized heuristic for 2-SBCM-P (combined with the bundling heuristic) uses fewer block crossings (and fewer pairwise crossings) than two heuristics for storyline block crossing minimization (SBCM) without a protagonist. On the other hand, these heuristics are free to allow the protagonist to cross other characters. From this point of view, it is indeed surprising that our 2-SBCM-P heuristic outperforms the two heuristics for SBCM.

We also evaluate the performance of our bundling heuristic against lower bounds that we obtain by partitioning the crossing complex without modifying it first. In general, this yields an optimal partitioning that cannot be realized geometrically, and hence, a lower bound. It turns out that the numbers of block crossings that our bundling heuristic produces are usually very close to this lower bound.

More related work. The *egocentric storylines* of Muelder, Crnovrsanin, Sallaberry, and Ma [25] probably come closest to our idea of storylines with a protagonist. However, their aim is to visualize large dynamic networks by selecting and drawing parts that are interesting for the user. As in our protagonist-setting, Muelder et al. allow the user to select a specific node p of the network, which is then displayed as a horizontal strip; in their case at the bottom of the layout. Then they select a subset of the nodes that are active in the current time step, namely nodes that are of relevance to p according to graph distance (combined with a time-based weight). Their input does not specify meetings; instead, more relevant nodes are placed closer to p . Nodes are shown as x-monotone strips between their first and last appearance; in time steps in which a node is not selected its strip becomes very thin and is placed behind the strips of selected nodes. Strips are colored according to a time-dependent clustering. Crossings are treated only implicitly, namely by reusing the vertical ordering of the nodes from the previous time step and inserting newly selected nodes according to their relevance.

Kim, Card, and Heer [17] visualized genealogical data using a storyline-like type of visualization. People are represented by x-monotone curves that start when they are born and end when they die (or when the diagram ends). The only type of meeting is marriage; that is, all meetings are of size 2. If a child is born, the start point of its curve is connected to its parents' curves by a dashed vertical line. Ancestors are placed by in-order, descendants by pre-order traversal of the family tree. Interestingly, when visualizing Elizabeth Taylor and her seven husbands [17, Fig. 9], the authors drew her as a protagonist and experiment with a 1- and a 2-sided layout. Again, crossings were not minimized; instead, after divorce, spouse curves simply return to the y-coordinate where they were before marriage.

Very recently, Kuo, Liu, and Ma [20] presented *SpreadLine*, a storyline-like visualization framework for egocentric networks that shares some properties with our two-sided approach. Instead of splitting the character set such that crossings are minimized, their split is based on an attribute of the characters given with the input. Hence, if the attribute of a character changes over time with respect to the protagonist’s attribute, the character’s curve will cross the protagonist’s curve. For generating the layout, Kuo et al. use the *StoryFlow* framework [23]. For crossing minimization, they use the well-known barycenter heuristic for one-sided crossing minimization in a sweeping fashion (i.e., starting on one side, and sweeping back and forth until the number of crossings no longer decreases).

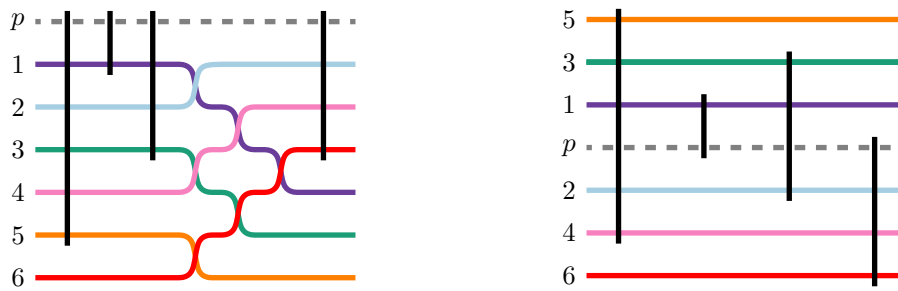
We now briefly review the existing (and not so closely related) approaches for storyline visualization in terms of computational techniques. Tanahashi and Ma [29] used a genetic algorithm to draw storylines with few crossings, few wiggles, and little white-space. Their algorithm is rather slow but produces esthetically pleasing results. Gronemann, Jünger, Liers, and Mambelli [12] used integer linear programming (ILP) to solve storyline crossing minimization (SCM) exactly. Fröschl and Nöllenburg [11] also used ILP, but in order to minimize the weighted number of wiggles (where each non-horizontal piece of a character’s curve is weighted by its height). Kostitsyna, Nöllenburg, Polishchuk, Schulz, and Strash [18] presented a fixed-parameter (FPT) algorithm for SCM. Van Dijk, Fink, Fischer, Markfelder, Ravsky, Suri, and Wolff [31] ([30]) gave the first FPT algorithm for SBCM and improved upon the running time of the FPT algorithm for SCM of Kostitsyna et al. Later, Van Dijk, Lipp, Markfelder, and Wolff [32] did an experimental study showing that, for SBCM, SAT-based algorithms are faster than ILP-based algorithms. Di Giacomo, Didimo, Liotta, Montecchiani, and Tappini [7] drew storylines where each character is represented by a plane tree rather than just by a curve, so a character can participate in several meetings simultaneously. They did two case studies, visualizing collaboration between scientists and work groups over time.

Another problem related to storyline visualization is metro-map layout, where metro lines are routed along the edges of an underlying graph whose vertices correspond to the metro stations. Note that other than the character curves in storyline visualization, the metro lines cannot go around a metro station if the station lies on the prescribed path that the metro line needs to follow. Fink and Pupyrev [9] showed that, given a fixed layout of the underlying graph, metro-line crossing minimization is NP-hard.

2 Preliminaries and Formal Problem Statement

We say that a meeting m *fits* a permutation π of C (or a permutation π *supports* a meeting m) if the characters in m form an interval in π . In order to enable support for all meetings in M , the order of characters may have to change at several points in time. Whenever this order changes, the character curves cross. Since crossings make it harder for an observer to follow a character curve, we aim to minimize the number of crossings.

We describe a *crossing* that swaps two characters by their position in π . If a crossing swaps the character at a with that at $a + 1$, it maps the permutation $\langle 1, \dots, a, a + 1, \dots, k \rangle$ to the permutation $\langle 1, \dots, a - 1, a + 1, a, a + 2, \dots, k \rangle$. When two disjoint blocks of curves cross all at once while staying parallel inside their respective blocks, we call this a *block crossing*. Let (a, b, c) with $a \leq b < c$ be a block crossing that swaps the consecutive blocks $\langle a, \dots, b \rangle$ and $\langle b + 1, \dots, c \rangle$. The permutation $\langle 1, \dots, a, \dots, b, \dots, c, \dots, k \rangle$ is therefore mapped to the permutation $\langle 1, \dots, a - 1, b + 1, \dots, c, a, \dots, b, c + 1, \dots, k \rangle$. Note that the block crossing $(a, a, a + 1)$ is equivalent to the pairwise crossing that swaps exactly the characters at a and $a + 1$. Let $\pi_{\text{id}} = \langle 1, 2, \dots, k \rangle$ be the identity permutation for a set of k elements. For $i \in \{1, \dots, k\}$, we write $\pi(i)$ for the position of character i in π . Analogously, for $j \in \{1, \dots, k\}$, we write $\pi^{-1}(j)$ for the character at position j in π .



(a) One-sided storyline drawing with p at the top. (b) Two-sided drawing of the same storyline.

■ **Figure 2** In one- and two-sided storylines, the protagonist is drawn as a straight line and does not participate in any crossings.

In our visualization style, the protagonist maintains a fixed position and therefore, it must not participate in any (block) crossing that would move it away from that position. In a further restricted variant, we require the protagonist to be placed either at the top or bottom of the stack of character lines.

► **Definition 1** (*Two-Sided Storyline Crossing Minimization with a Protagonist (2-SCM-P)*). Given a storyline instance (C, M) with a protagonist $p \in C$ and $M = [m_1, m_2, \dots, m_n]$, find a start permutation π_0 and a sequence $\mathcal{X} = [X_1, X_2, \dots, X_n]$ of (possibly empty) sequences of crossings such that (i) p is not involved in any crossing, (ii) for $1 \leq i \leq n$, $\pi_i = X_i(\pi_{i-1})$ supports m_i , and (iii) the total number of crossings is minimized.

The variant of the problem where we additionally require that $\pi_0(1) = p$ is called *One-Sided Storyline Crossing Minimization with a Protagonist (1-SCM-P)*. The variants of the problem where we count block crossings instead of pairwise crossings are called *(One-/Two-Sided) Storyline Block Crossing Minimization with a Protagonist (1-/2-SBCM-P)*.

A *realization* of (C, M) is a pair (π_0, \mathcal{X}) , where π_0 is a start permutation and \mathcal{X} is a sequence of sequences of (block) crossings such that every meeting is supported. Figure 2 shows drawings of a one- and a two-sided storyline with a protagonist.

3 Minimizing Pairwise Crossings in 1-SCM-P

1-SCM-P has fewer degrees of freedom than the more general SCM problem. Indeed, we will show that 1-SCM-P can be solved in polynomial time. We start with a simple observation.

► **Observation 2.** Given two characters c and d , if a meeting m contains c but not d and a later meeting m' contains d but not c , then c and d must cross between m and m' .

A crossing that fulfills the condition stated in the above observation is *unavoidable*. For each character $c \in C$, we define its *attendance vector* $v_c \in \{0, 1\}^{|M|}$ where $v_c(i) = 1$ if and only if $c \in m_i$. For any pair (c, d) of characters with $c < d$, we count the number U_{cd} of unavoidable crossings between c and d by removing entries from the attendance vectors until only those remain that inflict unavoidable crossings. First, remove all entries i where $v_c(i) = v_d(i)$. Then, remove all entries j where $(v_c(j), v_d(j)) = (v_c(j+1), v_d(j+1))$. Now U_{cd} is the number of remaining entries minus one, and the number of unavoidable crossings for the whole storyline is $\sum_{1 \leq c < d \leq k} U_{cd}$.

► **Lemma 3.** Every instance of 1-SCM-P can be realized with unavoidable crossings only.

Proof. The statement is trivial for storylines that can be realized without crossings. For a storyline (C, M) with protagonist p that requires crossings, we show the statement by contradiction. Let (π_0, \mathcal{X}) be a 1-SCM-P realization of (C, M) with the minimum number of crossings. Let c and d be two characters whose curves cross. Let c be the upper curve before the crossing, and let $t \in [0, n]$ be the point in time when they cross. Clearly, in order to cross, c and d must lie on the same side of p , say below. Now assume that this crossing is not unavoidable by our definition. Hence, there is a meeting m_i with $i < t$ that contains c but not d and there is no meeting m_j with $j > i$ that contains d but not c . (The case that the later meeting contains only d but not c , and there is no earlier meeting that contains c but not d is symmetric to the case that we study and can be handled analogously.)

We can safely remove the crossing of c and d at time t and swap the positions of c and d in all permutations after t . Any meeting m_j with $j > i$ that either contains both c and d or none of the two is not affected by this change. A meeting m_j that contains only c would require another crossing between c and d , which we could also remove safely. Hence, we obtain a realization with fewer crossings than (π_0, \mathcal{X}) , contradicting our choice of (π_0, \mathcal{X}) . ◀

► **Theorem 4.** *There is an algorithm that solves 1-SCM-P in $\mathcal{O}(k^2n)$ time, where k is the number of characters and n is the number of meetings.*

Proof. From Lemma 3 we know that the unavoidable crossings are sufficient to realize a storyline with a protagonist. Therefore, we present Algorithm 1 and prove that it produces exactly the unavoidable crossings.

We set the start permutation π_0 such that the characters are in descending lexicographic order with respect to their attendance vectors. Note that the first meeting m_1 fits π_0 . Let $i \in \{2, 3, \dots, k\}$ and assume that we have already computed a storyline up to and including meeting m_{i-1} . Let π_{i-1} be the permutation right after m_{i-1} . Given π_{i-1} , in order to support m_i , we have to move all characters in $S_1 = m_i \setminus m_{i-1}$ towards p (who attends all meetings) and all characters in $S_2 = m_{i-1} \setminus m_i$ away from p . Since we must not cross p , every character in S_1 must cross every character in S_2 . By Observation 2, these crossings are unavoidable. For S_1 and S_2 to form contiguous blocks, we may have to introduce additional crossings. Assume that we need such a crossing between characters c and d to move d closer to p . Let m_k be the latest meeting before m_{i-1} that contains c but not d . Such a meeting must exist, otherwise c and d would be swapped in π_0 . Hence, due to Observation 2, m_k and m_i induce an unavoidable crossing between c and d . Consequently, our algorithm produces only unavoidable crossings.

Any crossing introduced by one of the inner while-loops of Algorithm 1 moves character c upwards (that is, decreases $\pi^{-1}(c)$ in that loop). Therefore, the foreach-loops check at most k^2 pairs of characters, and Algorithm 1 runs in $\mathcal{O}(k^2n)$ time. ◀

4 Minimizing Pairwise Crossings in 2-SCM-P

In 1-SCM-P, we restrict the storyline such that the protagonist p always is at the topmost position. This may introduce a lot of crossings compared to storyline visualizations without this constraint. The variant 2-SCM-P drops this constraint but still requires that p does not participate in any crossing (i.e., can be drawn straight); see Figure 2b.

We map an instance (C, M) of 2-SCM-P to two instances of 1-SCM-P by splitting $C \setminus \{p\}$ into two sets C_1 and C_2 . For $j \in \{1, 2\}$, we set $C_j = C_j \cup \{p\}$ and let M_j be the restriction of M to characters in C_j . In order to find a split, we define the *crossing graph* of (C, M) to be the complete graph with vertex set $C \setminus \{p\}$, where, for characters c and d with $c < d$, edge

■ **Algorithm 1** One-sided storyline with minimum number of pairwise crossings.

Input: Instance (C, M) of one-sided SCM, attendance vectors v_1, v_2, \dots, v_k
Output: Realization $(\pi_0, \mathcal{X} = [X_1, \dots, X_{|M|}])$ of (C, M)

```

1  $\pi_0 \leftarrow$  characters in descending lexicographic order by attendance vectors
2  $\pi \leftarrow \pi_0$ 
3  $X_1 \leftarrow \emptyset$ 
4 for  $i \leftarrow 2$  to  $|M|$  do
5    $X_i \leftarrow \emptyset$ 
6   if  $m_i$  fits  $\pi$  then continue
7    $S_0 \leftarrow m_{i-1} \cap m_i$ 
8    $S_1 \leftarrow m_{i-1} \setminus m_i$ 
9    $S_2 \leftarrow m_i \setminus m_{i-1}$ 
10   $S_3 \leftarrow C \setminus (m_{i-1} \cup m_i)$ 
11  foreach  $c \in S_0$  ordered by  $\pi^{-1}$  do
12    while  $d \leftarrow \pi(\pi^{-1}(c) - 1) \in S_1$  do
13      switch  $c$  and  $d$  in  $\pi$ 
14      append  $(\pi^{-1}(c), \pi^{-1}(c), \pi^{-1}(d))$  to  $X_i$            // a pairwise crossing
15  foreach  $c \in S_2$  ordered by  $\pi^{-1}$  do
16    while  $d \leftarrow \pi(\pi^{-1}(c) - 1) \in S_3$  do
17      switch  $c$  and  $d$  in  $\pi$ 
18      append  $(\pi^{-1}(c), \pi^{-1}(c), \pi^{-1}(d))$  to  $X_i$            // a pairwise crossing
19  append  $(|S_0| + 1, |m_i|, |m_i| + |S_2|)$  to  $X_i$            // a block crossing
20  apply  $X_i$  to  $\pi$ 
21 return  $(\pi_0, \mathcal{X})$ 

```

$\{c, d\}$ has weight U_{cd} . Consider the problem (WEIGHTED) MIN-UNCUT which asks for a 2-coloring of the vertices of a graph such that the number (total weight) of the monochromatic edges is minimized. If two characters c and d are on different sides of p , then they do not cross. Otherwise they cause exactly U_{cd} unavoidable crossings, independently of the presence of other characters. By Theorem 4, we can solve the two 1-SCM-P instances resulting from the split optimally. Hence a split that minimizes the number of crossings corresponds to a solution of WEIGHTED MIN-UNCUT in the crossing graph.

Unfortunately, MIN-UNCUT is MaxSNP-hard [27]; it admits an $O(\sqrt{\log n})$ -approximation [1]. Note that MIN-UNCUT is the complement of MAX-CUT, which asks for a 2-coloring of the vertices of a graph such that the number of the bichromatic edges is maximized. In particular, the set of optimal solutions is the same for both problems.

We can efficiently detect instances of 2-SCM-P that can be drawn without crossings.

► **Theorem 5.** *Given a 2-SCM-P instance with k characters and n meetings, we can test in $\mathcal{O}(nk^2)$ time whether it admits a solution without crossings.*

Proof. Construct the crossing graph, test whether it is bipartite, and if yes, draw the two resulting one-sided instances using Theorem 4. ◀

Also if the crossing graph is planar, WEIGHTED MAX-CUT and hence 2-SCM-P can be solved efficiently [22]. Similarly, exact FPT algorithms for WEIGHTED MAX-CUT [6] carry over to 2-SCM-P. For our application where we require fast response for larges instances, we use a heuristic for WEIGHTED MAX-CUT [16] that is easy to implement and, although asymptotically $O(|C|^3)$, sufficiently fast.

5 Hardness of 1-SBCM-P and 2-SBCM-P

Van Dijk et al. [31] showed that SBCM is NP-complete by reducing from SORTING BY TRANSPOSITIONS (SBT). In SBT, given a permutation π and an integer t , the task is to decide whether π can be transformed into the identity permutation by applying a sequence of at most t transpositions (which we call block crossings). SBT is NP-hard [2]. Whereas in SBT the start permutation is given, for SBCM we need a gadget in order to ensure that there is a point in time where the characters are ordered as in π . We now construct such a gadget.

► **Lemma 6.** *Given a set C of k characters and a permutation π of C , there exists a sequence of meetings M of size $k - 1$ such that the one-sided storyline (C, M) with protagonist $\pi(1)$ can be drawn crossing-free and π is the only permutation that supports all meetings in M .*

Proof. For any given order of characters the sequence of meetings $[m_1, \dots, m_{k-1}]$ with $m_i = \{1, \dots, i + 1\}$ is supported by the identity permutation π_{id} . Any permutation $\pi' \neq \pi$ contains at least two characters c and d with $c < d$ whose positions are swapped compared to π . By construction, π' does not support m_{c-1} . ◀

► **Theorem 7.** *The problems 1-SBCM-P and 2-SBCM-P are NP-complete.*

Proof. It is easy to see that the decision variants of both 1-SBCM-P and 2-SBCM-P lie in NP. The number of (block) crossings necessary to support any meeting is bounded by $\binom{k}{2}$. So we can simply check a solution from left to right, in time polynomial in k and n .

Next we show that 1-SBCM-P is NP-hard. Using Lemma 6 we can build a one-sided storyline with a protagonist p such that $C \setminus \{p\}$ is in π -order just after m_{k-1} and in π_{id} -order just after m_k ; encoding the permutations for SBT (see the blue box in Figure 3).

Solving 1-SBCM-P for this instance gives us a start permutation $\pi_0 = \pi$ and a sequence $\mathcal{X} = [X_1, \dots, X_{k-1}, X_k, X_{k+1}, \dots, X_{2k-2}]$ of (possibly empty) sequences of block crossings. Let π_j denote the order of characters right after m_j . Meetings m_1, \dots, m_{k-1} allow us to maintain π from π_0 to π_{k-1} . Because of that, we can prepend (in order) all block crossings from X_1, \dots, X_{k-1} to X_k . The same is true for meetings m_k, \dots, m_{2k-2} and the identity permutation. So block crossings from X_{k+1}, \dots, X_{2k-2} can be appended (in order) to X_k . Now, X_k contains a minimum set of block crossings (a.k.a. transpositions) necessary in order to transform π to the identity.

For the NP-hardness of 2-SBCM-P, we reuse our proof for the one-sided variant. We use a simple gadget consisting of one additional author q and $2k^2$ meetings to fix an assignment of the authors to the sides that has all authors that are part of our one-sided instance in the same half of the drawing (see the red box in Figure 3). We add meetings $M' = [m'_1, \dots, m'_{2k^2}]$ where meetings with even index contain exactly p and q while those with odd index contain p and all characters $1, \dots, k$. The resulting storyline (C, M') can be drawn crossing-free if and only if $C \cup \{q\}$ is split $(C \setminus \{q\}, \{q, p\})$. Any other split introduces at least $2k^2 - 2$ block crossings (one for each meeting in M' except for the first and the last one). On the other hand, transforming π into identity takes at most $\binom{k}{2}$ crossings. Therefore, $(C \setminus \{q\}, \{q, p\})$ is the only optimal split, and the two-sided instance $(C \cup \{q\}, M' \circ M)$, where \circ denotes concatenation, encodes the one-sided instance (C, M) . ◀

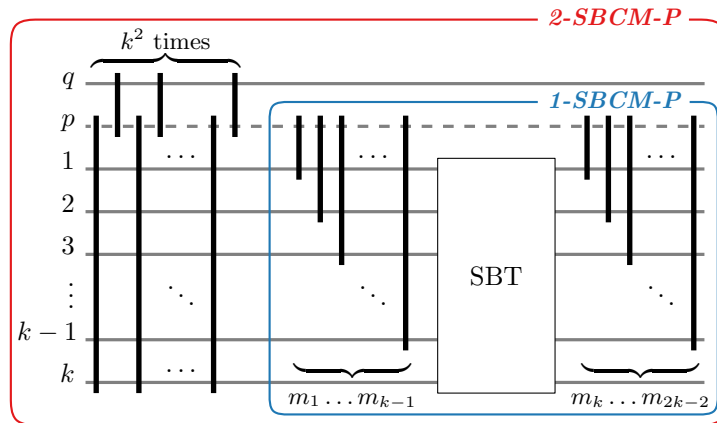


Figure 3 The NP-hardness construction for 2-SBCM-P reuses the one for 1-SBCM-P (blue box).

6 Bundling Pairwise Crossings in Storylines

While we can solve at least 1-SCM-P efficiently, we have seen that 1-SBCM-P is already NP-hard (Theorem 7). Still, we prefer block crossings from a cognitive point of view; they structure the set of pairwise crossings. Therefore, we now discuss the problem of covering a given set of pairwise crossings by the smallest number of block crossings. Fink et al. [8] introduced this problem for general graph embeddings and showed its NP-hardness.

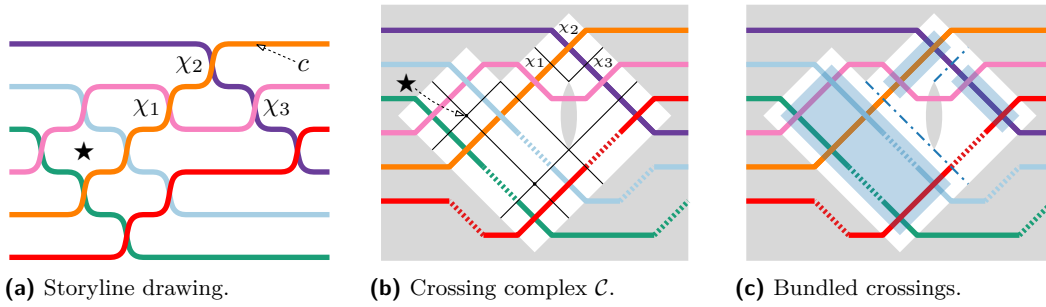
6.1 Bundling in the absence of meetings

Fortunately, in the special case of storylines, we can solve the following local version of the problem efficiently as shown below in Theorem 11. The version corresponds to the problem that needs to be solved between two consecutive meetings, given the sequence of pairwise crossings.

► **Definition 8** (Local Bundling for Storylines). *Given a permutation π and a sequence X of pairwise crossings, find the shortest sequence X' of block crossings such that, when applying X' to π , exactly the same lines cross as when applying X to π .*

Fink, Hershberger, Suri, and Verbeek [8] observed a connection between the bundled crossings problem and the minimum dissection problem for orthogonal polygons with holes. Soltan and Gorpinevich [28] showed that an orthogonal polygon with arbitrary holes can be dissected into the minimum number of rectangles in polynomial time. Their algorithm can be adapted to solve *Local Bundling for Storylines* as we will show now.

We say that two crossings χ_1 and χ_2 *touch* (and can hence be combined into a block crossing) if (i) they share exactly one character c and (ii) the part of the curve that represents c between the corresponding crossing points is y -monotone and is not crossed by any other curve. In Figure 4a, crossings χ_1 and χ_2 touch, but χ_1 and χ_3 do *not* touch and, unlike [8], we *cannot* combine them into a block crossing. We call the graph \mathcal{G} that has a vertex for every pairwise crossing in X and an edge for every pair of touching crossings the *touching graph* of X . Following Fink et al. [8], we define the *crossing complex* \mathcal{C} for a sequence X of pairwise crossings in a storyline as a special type of complex that consists of quadrilateral cells with sides and corners, where two respectively four cells touch:



■ **Figure 4** The crossing complex \mathcal{C} of a storyline drawing (a) contains a quadrilateral cell for each crossing. In (b) the cells are bounded by either the gray exterior (which includes holes) of \mathcal{C} or the thin black lines. Each cell shares a side (black line segments) with at most four other cells. A bundling corresponds to a dissection of \mathcal{C} into (blue shaded) rectangular groups of cells (c).

1. The complex \mathcal{C} contains a quadrilateral cell for each crossing in X . Each side of a cell corresponds to one “half” of a character curve; the one before and the one after the crossing.
2. If two crossings $\chi = \{b, c\}$ and $\psi = \{c, d\}$ in X touch each other, then their cells share the sides that correspond to the unique character in $\chi \cap \psi$. (Note that, in Figure 4b, the cells of χ_1 and χ_2 share a side, whereas the cells of χ_1 and χ_3 do not share a side.)
3. If the storyline drawing contains a quadrilateral face and the adjacent crossings form a cycle in \mathcal{G} , then the unique corner shared by the corresponding cells is part of \mathcal{C} (e.g., the starred face in Figure 4a becomes the starred corner in Figure 4b).

The crossing complex of the storyline drawing in Figure 4a is shown in Figure 4b. The exterior of \mathcal{C} is shaded in gray. Corners and sides that lie in the interior of \mathcal{C} are referred to as *internal*. In Figure 4b internal corners are marked by small black disks and internal sides by solid black line segments.

► **Lemma 9.** *The crossing complex \mathcal{C} of a storyline drawing can be laid out such that all internal sides are either falling or rising (i.e., have a slope of 45° or -45°).*

Proof. By item 2 of the definition above, an internal side can be drawn perpendicular to the curve of the common character of the two touching crossings. Since \mathcal{G} is a partial grid graph, it can be drawn orthogonally with respect to the embedding implied by the storyline drawing (see Figure 4b). Every crossing touches at most four other crossings, and the four arms of a crossing naturally yield a grid embedding. Therefore, internal sides can be drawn on a grid rotated by 45° . ◀

We assume that \mathcal{C} has been laid out as described in Lemma 9. In order to dissect \mathcal{C} into the minimum number of rectangular groups of cells, we cut \mathcal{C} along chords. A *chord* is a sequence of colinear internal sides that starts and ends in a *boundary corner* (i.e., a corner that is not internal) and has only internal corners in between. It is easy to see that cuts are necessary at every concave corner on the boundary $\partial\mathcal{C}$ of \mathcal{C} . Following Fink et al. [8], for each boundary corner z , we define its measure of “concaveness” $\kappa(z) = \lfloor (\alpha(z) - 1)/2 \rfloor$, where $\alpha(z)$ is the number of cells incident to z . This measure indicates how many cuts are necessary to make a corner convex. We will show that two cuts always suffice.

► **Lemma 10.** *Let z be a corner of \mathcal{C} . Then at most five cells are incident to z , and $\kappa(z) \leq 2$.*

Proof. Corners in \mathcal{C} (and on the boundary of \mathcal{C}) map to faces in the original storyline drawing. Internal corners in \mathcal{C} , by construction, result from a quadrilateral face in the storyline drawing; therefore, they have only four adjacent cells. All other corners lie on points on $\partial\mathcal{C}$ where internal sides have a common endpoint. This is the case when the crossings around a face in the storyline drawing form a path in \mathcal{G} (each touching pair of crossings implies an internal side). Since, at each touching pair of crossings, a face has a corner with an angle of 90° and the sum of angles of a face is 360° , such a path cannot be longer than five crossings. ◀

All corners with four adjacent cells are internal by definition. By Lemma 10, we know that concave boundary corners have three or five adjacent cells and thus, their measure is 1 or 2. Hence, every cut along a chord starting in a concave corner z decreases $\kappa(z)$ by 1.

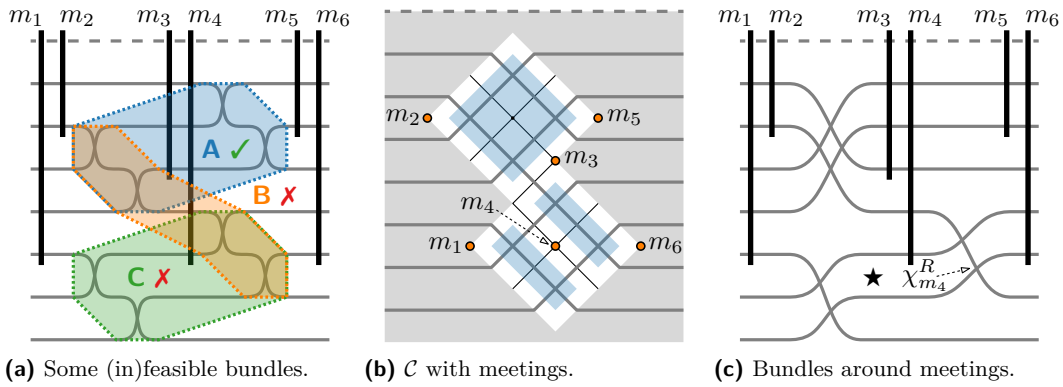
We identify two different types of chords; *effective chords*, which connect two concave corners and therefore decrease $\kappa(\mathcal{C}) = \sum_{z \in \partial\mathcal{C}} \kappa(z)$ by 2, and *simple chords*, which connect a concave and a convex corner. In order to minimize the number of cuts (and therefore rectangles), we must maximize the number of cuts along effective chords. Note that not all combinations of effective chords are feasible. Two effective chords are *in conflict* if they share a common internal corner. After cutting along one of them, the other one is no longer effective by our definition. Instead, it was cut into two simple chords. Recall that concave corners have measure 1 or 2. Let z be a corner. If $\kappa(z) = 1$, then every pair of effective chords incident to z is in conflict, because a single cut is sufficient to make z convex. If $\kappa(z) = 2$, then only those pairs of effective chords incident to z are in conflict if the two chords together do not reduce $\kappa(z)$ to 0. This is the case if, after cutting twice, z still has three consecutive cells.

In order to identify a maximum set of effective chords, we find a maximum independent set in their *conflict graph*. Note that due to the orthogonal nature of our chords, the conflict graph is bipartite and therefore a maximum independent set can be found in polynomial time (see König's theorem [21]). After applying cuts along a maximum set of effective chords, we handle the remaining concave corners by choosing an arbitrary simple chord starting at the corner and cutting along it. After that all connected components of \mathcal{C} are rectangular (i.e., their boundary corners are 90° or 180°). For the storyline depicted in Figure 4a, a dissection into rectangles is shown in Figure 4c.

It remains to translate the order of rectangles implied by the embedding of \mathcal{C} back into a feasible order for the bundles. Given two bundles A and B , A must happen before B if, for any two crossings χ_A in A and χ_B in B , (i) the cell of χ_A has a top-right or bottom-right edge to the cell of χ_B in \mathcal{G} (according to its grid embedding) or (ii) χ_A and χ_B are on the same *level* (i.e., if π_A is the permutation right before χ_A and π_B is the one right before χ_B , the character curves involved in χ_A have the same position in π_A as those of χ_B in π_B) and χ_A comes before χ_B in X . We then topologically sort the directed graph with a vertex for every bundle and arcs corresponding to the ordering constraints. This gives us a feasible order for the bundles.

► **Theorem 11.** *Given a storyline with a sequence X of pairwise crossings, Local Bundling for Storylines can be solved in $\mathcal{O}(|X|^2)$ time.*

Proof. The complex \mathcal{C} can be constructed in time linear in $|X|$ since the storyline yields the topology and, for each crossing in X , we introduce at most four sides and corners. We then find the effective chords in time linear in $|X|$ because at most four chords start, end, or go through any corner, and no two chords overlap. Since effective chords are in conflict only if they start at, end at, or go through the same corner, the number of nodes and edges in the conflict graph is linear in $|X|$. Hence, we can find a dissection of \mathcal{C} (corresponding to



■ **Figure 5** Not all bundles are feasible when it comes to meetings (a). Meetings correspond to point holes in the crossing complex \mathcal{C} (b). A possible bundling is shown in (c).

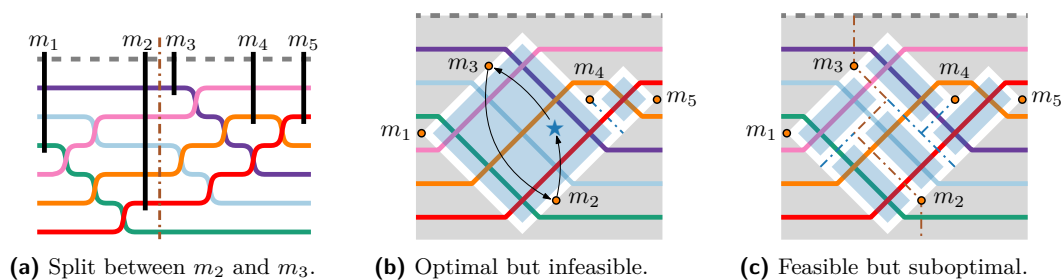
a maximum independent set in the conflict graph) in $\mathcal{O}(|X|^2)$ time using the algorithm of Soltan and Gorpinevich [28]. Applying the set of effective chords and adding the remaining simple chords is again bounded by the number of sides because every side is cut at most once. The number of bundles is bounded by $|X|$. We can find all ordering constraints and topologically sort the conflict graph in $\mathcal{O}(|X|^2)$ time. ◀

6.2 Bundling in the presence of meetings

So far we only bundled crossings between two neighboring meetings. It is obviously beneficial and often possible to bundle crossings across meetings. We allow a meeting only to happen before or after a block crossing (as shown in Figure 5c) but not inside. Hence, we may neither trap a meeting inside a bundle (see bundle C in Figure 5a) nor change the order of meetings. For example, in Figure 5a, meeting m_4 must happen before the bundled crossing B , whereas meeting m_3 must happen after B . Because we cannot change the order of meetings, such bundles are prohibited.

For a meeting m , let χ_m^R be the first crossing after m where one of the characters whose curves cross is part of m , whereas the other is not. We say that χ_m^R touches m from the right (see Figure 5c). Note that χ_m^R is adjacent to one of the faces where m ends (see the black star in Figure 5c). In order to prevent bundles (such as C in Figure 5a) to trap meetings inside, we prohibit faces where meetings end from being part of a bundle. For the crossing complex \mathcal{C} , this means that we remove the corresponding corner from \mathcal{C} . The dissection algorithm can be easily enhanced to support point holes [28]. Point holes get a measure of two and effective chords starting or ending at a point hole are in conflict if and only if they have different slopes (one falling one rising).

The last step of the bundling procedure, *ordering bundles*, must adhere to two further kinds of constraints, namely meeting–meeting and meeting–bundle constraints. A meeting must be placed to the right of (or after) another meeting as indicated by the sequence M of meetings. A bundle must be placed to the right of a meeting m if any crossing in the bundle touches m from the right. In Figure 6b, the constraints between meetings m_2 and m_3 and the bundle marked with the blue star are depicted as black arrows. An arrow starting at a and pointing at b indicates that a must be placed to the right of b . Unfortunately, sometimes these constraints form cycles and therefore the bundling cannot be realized geometrically (for example the bundle with the blue star in Figure 6b or bundle B in Figure 5a).



■ **Figure 6** With the optimal dissection this bundling has a dependency cycle between meetings m_2 , m_3 , and the bundle marked with the blue star (b). When we split the storyline between the conflicting meetings (a), the dissection becomes feasible but suboptimal (c) as three bundles would be sufficient.

We propose the following heuristic: whenever we encounter such a cycle, we split the storyline after the first meeting in the cycle (i.e., the meeting with the smallest index). See Figure 6a for an example. Note that every such cycle must involve at least one meeting. It remains open whether or not we can find a cut through \mathcal{C} that still produces an optimal result regarding the number of bundles. Cutting straight after a meeting and through the full height of the storyline can worsen the solution as shown in Figure 6c. We can insert the point holes and find the additional constraints within the same asymptotic runtime as above. We effectively run our algorithm once per split so theoretically our heuristic has a runtime of $O(|X|^3)$. However, in practice splits are rare and bundling is almost imperceptibly quick (see Section 8).

Summarizing, we can find an optimal bundling (i.e., with the minimum number of bundles) as long as no dependency cycles occur when ordering the resulting block crossings. Otherwise, we can use the number of bundles in the infeasible solution as a lower bound for the optimal number of bundles. Our experiments (see below) show that even our simple heuristic that splits the storyline whenever it encounters a conflict yields optimal results for many instances.

7 A Greedy Heuristic for SBCM

In this section we present a greedy algorithm to draw a storyline with few block crossings in $\mathcal{O}(k^2sn)$ time, where $|C| = k$ is the number of characters, $|M| = n$ is the number of meetings, and $\sum_i |X_i| = s$ is the number of block crossings. Because SBCM is NP-hard [31], we cannot hope for an optimal solution. Our algorithm repeatedly adds block crossings until all meetings fit. Our heuristic is based on previous work by Herrmann [15], but we use the following more involved scoring function. Our scoring function considers possible block crossings and chooses weights differently; moreover, we do not limit the number of meetings that are factored into the scoring. We use the attendance vector v_c of a character c (see Section 3) in order to find *t-conflict-free pairs*, that is, pairs that can stay in the same block for the next t meetings. A pair of characters (c, d) at a meeting m_i is *t-conflict-free* for a number t if $v_c(j) = v_d(j)$ for $i \leq j \leq i + t$.

We process the meetings in order. For $i \in \{2, 3, \dots, n\}$, let π be the permutation right after m_{i-1} . Let $\langle \pi(a), \dots, \pi(b-1) \rangle$ and $\langle \pi(c), \dots, \pi(d) \rangle$ be two maximal blocks of characters in π that all attend m_i and are separated by another block of characters $\langle \pi(b), \dots, \pi(c-1) \rangle$ that all do not attend m_i . Note that if no such configuration exists, m_i fits π and we continue with m_{i+1} . We now have several options to join the two blocks with a single block crossing. We can either merge the first block into the second or the second into the first and we can

Algorithm 2 SELECTBLOCKCROSSING.

Input: Permutation π , meeting m_i , attendance vectors v_1, \dots, v_k
Output: Block crossing

```

1 find  $a < b < c \leq d$  such that
2    $\{\pi(j) \mid j \in \{a, \dots, b-1, c, \dots, d\}\} \subseteq m_i$  and
3    $\{\pi(j) \mid j \in \{1, \dots, a-1, b, \dots, c-1, d+1\}\} \cap m_i = \emptyset$ 
4  $best \leftarrow \emptyset$ 
5  $\sigma_{best} \leftarrow -\infty$ 
6 foreach  $\chi \in \{(z, c-1, d) \mid z \in \{a, \dots, b\}\} \cup \{(a, b-1, z) \mid z \in \{c-1, \dots, d\}\}$  do
7    $\pi' \leftarrow \pi$  with  $\chi$  applied
8    $\sigma_\chi \leftarrow 0$ 
9   foreach  $(e, f) \in \{(\pi'(j), \pi'(j+1)) \mid a-1 \leq j \leq d\}$  do
10     $i' \leftarrow i$ 
11    while  $v_e[i'] == v_f[i']$  do  $i' \leftarrow i' + 1$ 
12     $\sigma_\chi \leftarrow \sigma_\chi + i' - i$ 
13  if  $\sigma_{best} < \sigma_\chi$  or  $(\sigma_{best} == \sigma_\chi$  and  $size(best) > size(\chi))$  then
14     $best \leftarrow \chi$ 
15     $\sigma_{best} \leftarrow \sigma_\chi$ 
16 return  $best$ 

```

choose any position inside one of the blocks where we insert the other block. We use the procedure described in Algorithm 2 in order to rank the possible block crossings and select the best based on adjacent conflict-free pairs and the size of the block crossing (i.e., number of characters involved). Let π' be the permutation after applying the block crossing that we found in the previous step. If π' does not support m_i , we repeat the described procedure with π' instead of π .

Our algorithm heavily depends on a good start permutation. We propose the following strategy to find one. Let M' be the sequence of meetings obtained by reversing M , and let $\tilde{\pi}$ be a random permutation. We apply our greedy algorithm to (C, M') with $\tilde{\pi}$ as a start permutation and record the permutation π'_{end} after the last meeting. Note that the last meeting in M' is the first meeting in M . We use π'_{end} as the start permutation for the actual run.

For each block crossing in the final drawing, we call SELECTBLOCKCROSSING (see Algorithm 2) once. The parameters a, b, c , and d can be determined in $\mathcal{O}(k)$ time. At most k block crossings are considered and each requires up to $k-1$ pairs of characters to be scored. The scoring function can be computed in $\mathcal{O}(n)$ time. Therefore, the total runtime is $\mathcal{O}(k^2sn)$, where s is the number of block crossings. Note that, in most cases, a lot fewer than n comparisons are required to score a pair of characters. If the worst-case runtime bound is a concern, the maximum number of comparisons can be limited by a constant, which reduces the runtime to $\mathcal{O}(k^2s)$ (and may slightly increase the number of block crossings).

8 Experiments

Our use case for storylines with a protagonist is to visualize how the peer group of a scientific author changes over time. We are interested in comparing heuristics for visualizing storylines with a protagonist to heuristics for visualizing storylines without a protagonist. We want to measure whether the latter benefit from the additional degree of freedom that allows them to choose realizations where the protagonist's curve is crossed by the curves of other characters. We also want to evaluate the performance of the bundling heuristic that we presented in Section 6.2.

Benchmark set. For our experiments, we identified the 81 authors of short and long papers in the GD 2023 proceedings that have at least 20 coauthors on publications over the last 10 years that are listed in the dblp computer science bibliography at <https://dblp.org>, a very reliable source of publication data (for computer science). For each among the 81 authors, we created storylines with their 5, 10, 15, and 20 most frequent coauthors. This yielded our benchmark set with 324 instances.

Metrics. Metrics for assessing the quality of a visualization look for measurable features that influence legibility and aesthetics. In the context of storylines, Tanahashi and Ma [29] discussed formative design criteria and suggested to measure line wiggles, line crossings, and white-space. Because our drawing style does not produce white-space, we measure *wiggles* and *crossings*. Additionally, we count the number of *block crossings* (or bundled crossings). We define the number of wiggles as the number of times a character curve changes position. This is equivalent to the sum of the sizes of all block crossings where the size of a block crossing is the number of lines involved. Note that the wiggles metric, as we defined it, is equivalent to what is known as passages in the context of metro maps [13].

Algorithms. We implemented the following four algorithms for drawing storylines with a protagonist with few (block) crossings. We call the algorithm that solves 1-SCM-P exactly *1-Sider* (see Section 3). Recall that our heuristic for the two-sided variant first splits the set of characters using a heuristic for MAX-CUT and then solves the resulting 1-SCM-P instances exactly (see Section 4). We call this heuristic for 2-SCM-P *2-Sider*. We call our greedy heuristic for SBCM *GreedyBlocks* (see Section 7).

As a baseline for our experiments we used a simple heuristic that we call *Median*. For each meeting, it selects the median of all attending characters (by their position in the current permutation) and join the remaining participants with the minimum number of crossings. For finding a decent start permutation, we use the same trick as with *GreedyBlocks* and draw the reversed storyline with a random start permutation.

We combine *1-Sider*, *2-Sider*, and *Median* with our algorithm for bundling pairwise crossings into block crossings (see Section 6). Note that our implementation uses Kuhn’s classic algorithm [19] for computing a maximum independent set in a bipartite graph; its cubic runtime is worse than what we stated in Theorem 11, but the algorithm is easy to implement and fast enough in our setting.

We implemented all algorithms as TypeScript web applications. We performed the tests under Fedora Linux 40 and node.js v20 on an AMD Ryzen 7 7840HS with 64 GB of RAM.

Results. We applied the four algorithms to each instance in our benchmark set and compared the results in terms of crossings, block crossings, and wiggles. In Table 1, μ denotes the average of that metric over all 81 realizations of that algorithm and number of characters. For every algorithm we also counted how often they produced the best result (with that metric) out of all algorithms tested; see the columns labeled β .

The results for the largest storylines with 21 authors are shown in detail in Figure 7. All measurements are relative to *GreedyBlocks*, so an algorithm with a measure below 1 performs better than *GreedyBlock* for that specific data set. For each algorithm, a horizontal line marks the median of its results. The medians and the detailed results in general support the overall trend that Table 1 shows.

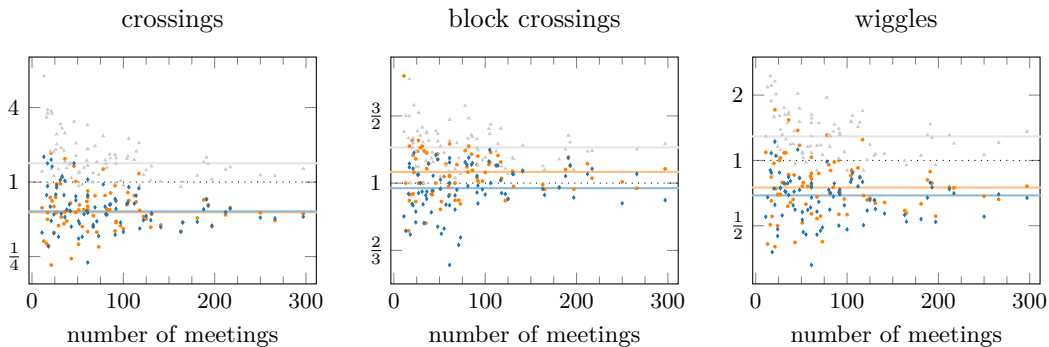
26:16 Storylines with a Protagonist

■ **Table 1** Experimental results on a dataset of 81 protagonists. The mean μ is calculated for each algorithm and each number of authors k . β measures the percentage of cases where an algorithm achieved the best result. Sums over 100% are possible due to ties.

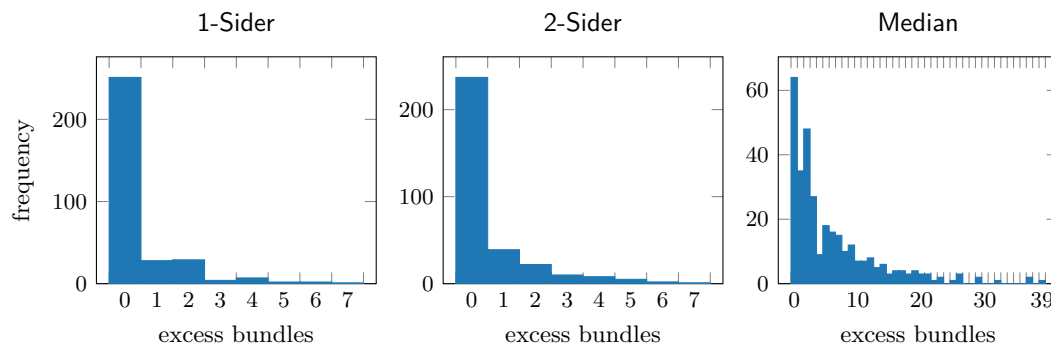
	k	1-Sider		2-Sider		Median		GreedyBlocks	
		μ	β	μ	β	μ	β	μ	β
crossings	6	85.1	0	24.3	88	32.8	21	47.0	11
	11	283.0	0	103.3	70	113.3	32	197.3	2
	16	536.9	0	211.8	59	222.6	33	418.0	10
	21	823.6	0	342.6	53	346.2	42	623.5	5
block crossings	6	33.8	0	18.7	68	21.1	32	23.0	26
	11	59.2	0	41.8	59	45.2	22	45.5	30
	16	75.9	0	58.9	63	63.4	22	62.8	27
	21	87.9	0	72.3	54	76.2	19	72.8	38
wiggles	6	112.6	0	42.9	81	53.5	26	67.3	11
	11	286.5	0	136.7	75	152.6	25	208.8	4
	16	469.0	0	239.3	68	266.6	26	388.3	6
	21	649.1	0	344.1	77	386.6	21	535.8	2

We expected the algorithms that implement our protagonist-focused style to be at a disadvantage compared to algorithms without that restriction. With about 30% more block crossings, 80% more wiggles, and more than twice as many crossings, for the 1-Sider algorithm this was clearly the case. The 2-sider algorithm in contrast was competitive in every metric. Both the 2-sider and Median algorithms match or even outperform the GreedyBlocks heuristic when it comes to block crossings, showing the effectiveness of our bundling algorithm.

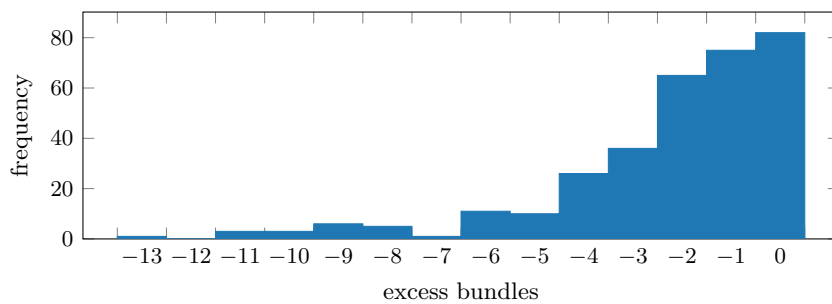
As discussed in Section 6, the bundling algorithm is not always optimal but we can determine a lower bound on the optimal number of bundles for any input. We evaluate the quality of our heuristic that splits the storyline whenever it encounters a conflict by comparing its results with the lower bound. The results can be found in Figure 8. For 1-Sider, 77% of our test set is bundled optimally; for 2-Sider, 73%. The results for Median are rather inconclusive. While most instances are close to the optimum, the overall gap between the lower bound and the actual number of block crossings is wider. Apparently, the protagonist benefits bundling.



■ **Figure 7** Results of 1-Sider \blacktriangle , 2-Sider \blacklozenge , and Median \bullet relative to GreedyBlocks when applied to the largest instances (with 21 characters) in our benchmark set. Horizontal lines mark the medians.



■ **Figure 8** For most of the 324 storylines in our benchmark set, the bundling heuristic operated at (or very close to) the lower bound.



■ **Figure 9** For most of the 324 storylines, rebundling reduced the number of block crossings.

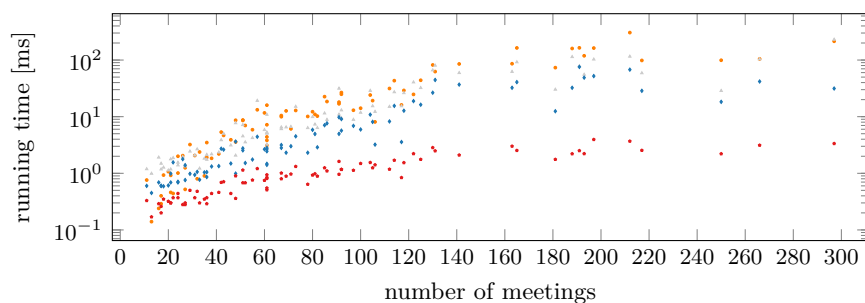
The bundling heuristic can also be applied to the results of GreedyBlocks as a post-processing step. So we run GreedyBlocks, resolve all block crossings into pairwise ones (we replace any block crossing with a block of a lines intersecting a block of b lines by $a \times b$ pairwise crossings), and then use the bundling heuristic to “rebundle” them again into block crossings. On the same dataset as used before (324 storylines derived from 81 protagonists) none of the instances in our dataset had more block crossings than without rebundling, despite the heuristic nature of our algorithm. See Figure 9 for detailed results. Rebundling decreased the number of block crossings in 75% of cases. As a post-processing step it showed positive results across the board.

Running times. For the dataset of large storylines (20 coauthors) we measured running times of 100 repetitions per input and algorithm. See Figure 10 for the results. We can clearly see that bundling takes a toll but overall, for our interactive use case, the running times are always tolerable and most of the time imperceptible.

9 Conclusion

Storylines with a protagonist arise naturally when visualizing how the peer group of a scientific author changes over time. Minimizing the number of (block) crossings helps to make such visualizations more readable. We have presented an efficient algorithm for minimizing *pairwise* crossings in a restricted case (1-SCM-P), and we have shown that it is NP-hard to minimize the number of *block* crossings (1-SBCM-P) even in the simpler protagonist setting. Our experimental evaluation has shown that our heuristic for bundling pairwise crossings

26:18 Storylines with a Protagonist

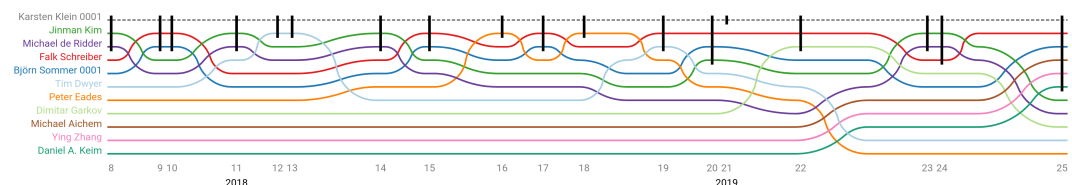


■ **Figure 10** Running times of 1-Sider \blacktriangle , 2-Sider \blacklozenge , Median \bullet , and GreedyBlocks \blacklozenge for the largest instances (with 21 characters) in our benchmark set. Each point is the average of 100 repetitions.

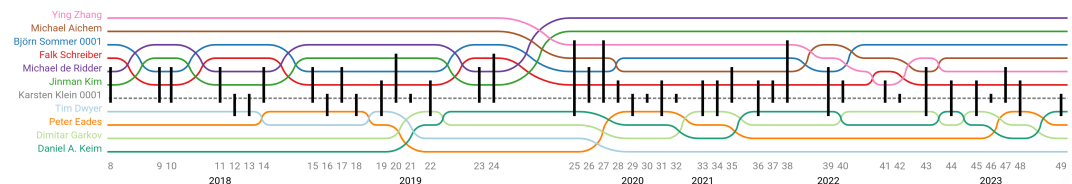
into block crossings performed close to optimal on our benchmark set. Our heuristics for 2-SBCM(-P) are fast enough for interactive applications. The fact that 2-Sider outperformed GreedyBlocks and Median underlines that more cleverness is needed to exploit the additional freedom that 2-SBCM offers compared to 2-SBCM-P. On the other hand, having a designated protagonist can be beneficial in use cases other than the visualization of publication histories; Kuo et al. [20] used a somewhat less strict notion of a protagonist to visualize (i) interactions among actors in social media and (ii) disease propagation centered around a primary outbreak.

Still, some questions remain open. Is 2-SCM-P NP-hard? Can bundling in the presence of meetings (see Section 6.2) be solved efficiently? Can we efficiently minimize the weighted number of wiggles [11] in the one-sided setting?

Figures 11–14 show some storyline visualizations produced by our algorithms. The storylines depicted are not necessarily part of our benchmark set. They were specifically chosen (and some of them cropped) in order to highlight noteworthy properties of the algorithms. Clicking on the star in a caption opens an interactive storyline visualization of the same setting using *current* publication data on dblp.

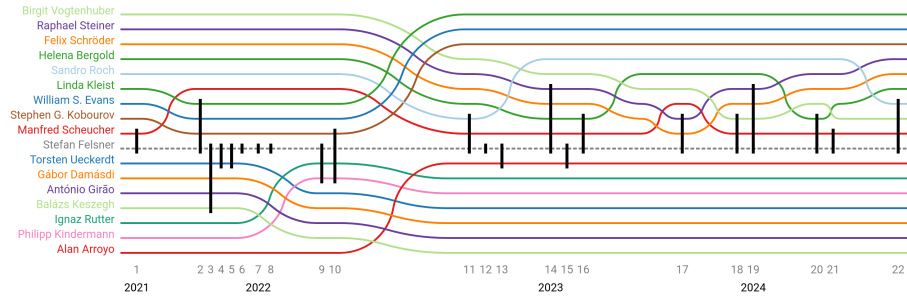


(a) Drawn by 1-Sider with bundling. ★

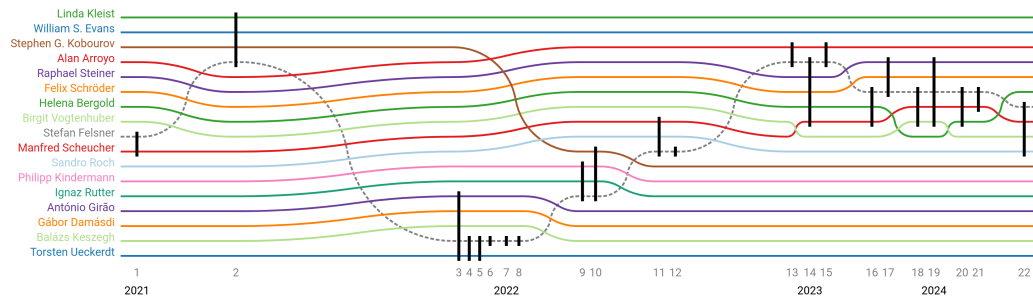


(b) Drawn by 2-Sider with bundling. ★

■ **Figure 11** Clippings of two storylines visualizing Karsten Klein and his 10 most frequent coauthors in the past 7 years. Note that 2-Sider yields a much more compact drawing (more meetings in the same scene space).

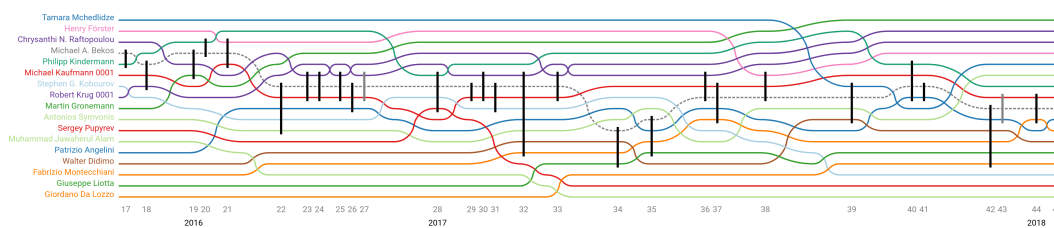


(a) Drawn by 2-Sider with bundling (55 pairwise crossings, 11 block crossings, and 52 wiggles). ★

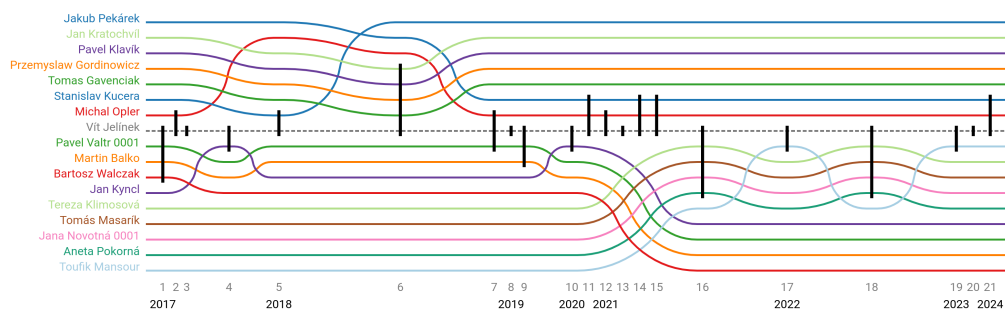


(b) Drawn by GreedyBlocks (44 pairwise crossings, 11 block crossings, and 55 wiggles). ★

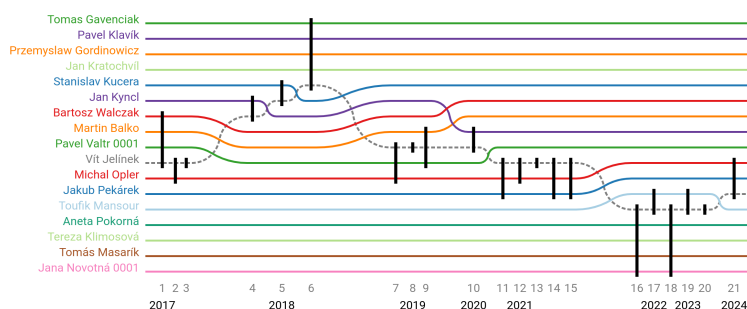
■ **Figure 12** Two complete storylines visualizing Stefan Felsner and his 16 most frequent coauthors in the past 3 years. Note that GreedyBlocks produces larger block crossings than 2-Sider.



■ **Figure 13** A clipping of a storyline visualizing Michael Bekos and his 16 most frequent coauthors in the past 10 years. This realization uses the Median algorithm with bundling. Note that, at meetings 32 and 33, our bundling heuristic missed an obvious opportunity to merge two 2×1 block crossings into a 2×2 block crossing. ★



(a) Drawn by 2-Sider with bundling (58 pairwise crossings, 10 block crossings, and 53 wiggles). ★



(b) Drawn by GreedyBlocks (16 pairwise crossings, 8 block crossings, and 24 wiggles). ★

■ **Figure 14** Two complete storylines visualizing Vít Jelínek and his 16 most frequent coauthors in the past 7 years. Note that when more than two disjoint groups collaborate, allowing the protagonist's curve to cross other curves can reduce the number of (block) crossings. This happened, however, not very often in the data set we analyzed.

References

- 1 Amit Agarwal, Moses Charikar, Konstantin Makarychev, and Yury Makarychev. $O(\sqrt{\log n})$ approximation algorithms for Min UnCut, Min 2CNF deletion, and directed cut problems. In *ACM Symp. Theory Comput. (STOC)*, pages 573–581, 2005. doi:10.1145/1060590.1060675.
- 2 Laurent Bulteau, Guillaume Fertin, and Irena Rusu. Sorting by transpositions is difficult. *SIAM J. Discrete Math.*, 26(3):1148–1180, 2012. doi:10.1137/110851390.
- 3 Steven Chaplick, Myroslav Kryven, Giuseppe Liotta, Andre Löffler, and Alexander Wolff. Beyond outerplanarity. In Fabrizio Frati and Kwan-Liu Ma, editors, *Graph Drawing & Network Vis. (GD)*, volume 10692 of *LNCS*, pages 546–559. Springer, 2018. doi:10.1007/978-3-319-73915-1_42.
- 4 Steven Chaplick, Thomas C. van Dijk, Myroslav Kryven, Ji-won Park, Alexander Ravsky, and Alexander Wolff. Bundled crossings revisited. In Daniel Archambault and Csaba D. Tóth, editors, *Graph Drawing & Network Vis. (GD)*, volume 11904 of *LNCS*, pages 63–77. Springer, 2019. doi:10.1007/978-3-030-35802-0_5.
- 5 Steven Chaplick, Thomas C. van Dijk, Myroslav Kryven, Ji-won Park, Alexander Ravsky, and Alexander Wolff. Bundled crossings revisited. *J. Graph Alg. Appl.*, 24(4):621–655, 2020. doi:10.7155/JGAA.00535.
- 6 Markus Chimani, Christine Dahn, Martina Juhnke-Kubitzke, Nils Kriege, Petra Mutzel, and Alexander Nover. Maximum cut parameterized by crossing number. *J. Graph Alg. Appl.*, 24(3):155–170, 2020. doi:10.7155/jgaa.00523.
- 7 Emilio Di Giacomo, Walter Didimo, Giuseppe Liotta, Fabrizio Montecchiani, and Alessandra Tappini. Storyline visualizations with ubiquitous actors. In David Auber and Pavel Valtr, editors, *Graph Drawing & Network Vis. (GD)*, volume 12590 of *LNCS*, pages 324–332. Springer, 2020. doi:10.1007/978-3-030-68766-3_25.

- 8 Martin Fink, John Hershberger, Subhash Suri, and Kevin Verbeek. Bundled crossings in embedded graphs. In Evangelos Kranakis, Gonzalo Navarro, and Edgar Chávez, editors, *Latin Amer. Symp. Theoret. Inform. (LATIN)*, volume 9644 of *LNCS*, pages 454–468. Springer, 2016. doi:10.1007/978-3-662-49529-2_34.
- 9 Martin Fink and Sergey Pupyrev. Metro-line crossing minimization: Hardness, approximations, and tractable cases. In Stephen K. Wismath and Alexander Wolff, editors, *Graph Drawing (GD)*, volume 8242 of *LNCS*, pages 328–339. Springer, 2013. doi:10.1007/978-3-319-03841-4_29.
- 10 Martin Fink, Sergey Pupyrev, and Alexander Wolff. Ordering metro lines by block crossings. *J. Graph Alg. Appl.*, 19(1):111–153, 2015. doi:10.7155/JGAA.00351.
- 11 Theresa Fröschl and Martin Nöllenburg. Minimizing wiggles in storyline visualizations. In Fabrizio Frati and Kwan-Liu Ma, editors, *Graph Drawing & Network Vis. (GD)*, volume 10692 of *LNCS*, pages 585–587. Springer, 2018. URL: <https://www.ac.tuwien.ac.at/files/pub/fn-mwsv-18.pdf>.
- 12 Martin Gronemann, Michael Jünger, Frauke Liers, and Francesco Mambelli. Crossing minimization in storyline visualization. In Yifan Hu and Martin Nöllenburg, editors, *Graph Drawing & Network Vis. (GD)*, volume 8901 of *LNCS*, pages 367–381. Springer, 2016. doi:10.1007/978-3-319-50106-2_29.
- 13 Herman Haverkort. Metaphorical metro maps: Design challenges. Schematic Mapping Workshop, 2022. URL: <https://www.ruhr-uni-bochum.de/schematicmapping/papers/smw-haverkort.pdf>.
- 14 Tim Hegemann and Alexander Wolff. Publines. Software, swbId: swb:1:dir:8b6a0dd881ed0b7a4b0e5d8dbfa38f82827aa641 (visited on 2024-10-14). URL: <https://github.com/hegetim/publines>.
- 15 Tim Herrmann. Storyline-Visualisierungen für wissenschaftliche Kollaborationsgraphen. Master’s thesis, University of Würzburg, 2022. URL: <https://www1.pub.informatik.uni-wuerzburg.de/pub/theses/2022-herrmann-masterarbeit.pdf>.
- 16 Sera Kahruman, Elif Kolotoglu, Sergiy Butenko, and Illya V. Hicks. On greedy construction heuristics for the max-cut problem. *Int. J. Comput. Sci. Eng.*, 3(3):211–218, 2007. doi:10.1504/IJCSE.2007.017827.
- 17 Nam Wook Kim, Stuart K. Card, and Jeffrey Heer. Tracing genealogical data with timenets. In Giuseppe Santucci, editor, *Advanced Visual Interfaces (AVI)*, pages 241–248. ACM Press, 2010. doi:10.1145/1842993.1843035.
- 18 Irina Kostitsyna, Martin Nöllenburg, Valentin Polishchuk, André Schulz, and Darren Strash. On minimizing crossings in storyline visualizations. In Emilio Di Giacomo and Anna Lubiw, editors, *Graph Drawing & Network Vis. (GD)*, volume 9411 of *LNCS*, pages 192–198. Springer, 2015. doi:10.1007/978-3-319-27261-0_16.
- 19 Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval Res. Logist. Quart.*, 2(1–2):83–97, 1955. doi:10.1002/nav.3800020109.
- 20 Yun-Hsin Kuo, Dongyu Liu, and Kwan-Liu Ma. SpreadLine: Visualizing egocentric dynamic influence. *IEEE Trans. Visual. Comput. Graphics*, 2024. To appear. arXiv:2408.08992.
- 21 Dénes König. Gráfok és mátrixok. *Matematikai és Fizikai Lapok*, 38:116–119, 1931.
- 22 Frauke Liers and Gregor Pardella. Partitioning planar graphs: A fast combinatorial approach for max-cut. *Comput. Optim. Appl.*, 51(1):323–344, 2012. doi:10.1007/S10589-010-9335-5.
- 23 Shixia Liu, Yingcai Wu, Enxun Wei, Mengchen Liu, and Yang Liu. StoryFlow: Tracking the evolution of stories. *IEEE Trans. Visual. Comput. Graphics*, 19(12):2436–2445, 2013. doi:10.1109/TVCG.2013.196.
- 24 Charles Joseph Minard. The Russian campaign 1812–1813. Diagram available at <https://commons.wikimedia.org/wiki/File:Minard.png>, 1869. Accessed 2017/04/03.
- 25 Chris W. Muehler, Tarik Crnovrsanin, Arnaud Sallaberry, and Kwan-Liu Ma. Egocentric storylines for visual analysis of large dynamic graphs. In *Proc. IEEE Int. Conf. Big Data*, pages 56–62, 2013. doi:10.1109/BigData.2013.6691715.

26:22 Storylines with a Protagonist

- 26 Randall Munroe. Movie narrative charts. Diagram available at <https://xkcd.com/657/>, 2009. Accessed 2017/04/03.
- 27 Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *J. Comput. Syst. Sci.*, 43(3):425–440, 1991. doi:10.1016/0022-0000(91)90023-X.
- 28 Valeriu Soltan and Alexei Gorpinevich. Minimum dissection of a rectilinear polygon with arbitrary holes into rectangles. *Discrete Comput. Geom.*, 9:57–79, 1993. doi:10.1007/BF02189307.
- 29 Yuzuru Tanahashi and Kwan-Liu Ma. Design considerations for optimizing storyline visualizations. *IEEE Trans. Visual. Comput. Graphics*, 18(12):2679–2688, 2012. doi:10.1109/TVCG.2012.212.
- 30 Thomas C. van Dijk, Martin Fink, Norbert Fischer, Fabian Lipp, Peter Markfelder, Alexander Ravsky, Subhash Suri, and Alexander Wolff. Block crossings in storyline visualizations. In Yifan Hu and Martin Nöllenburg, editors, *Graph Drawing & Network Vis. (GD)*, volume 9801 of *LNCS*, pages 382–398. Springer, 2016. doi:10.1007/978-3-319-50106-2_30.
- 31 Thomas C. van Dijk, Martin Fink, Norbert Fischer, Fabian Lipp, Peter Markfelder, Alexander Ravsky, Subhash Suri, and Alexander Wolff. Block crossings in storyline visualizations. *J. Graph Alg. Appl.*, 21(5):873–913, 2017. doi:10.7155/jgaa.00443.
- 32 Thomas C. van Dijk, Fabian Lipp, Peter Markfelder, and Alexander Wolff. Computing storylines with few block crossings. In Fabrizio Frati and Kwan-Liu Ma, editors, *Graph Drawing & Network Vis. (GD)*, volume 10692 of *LNCS*, pages 365–378. Springer, 2018. doi:10.1007/978-3-319-73915-1_29.

On k -Planar Graphs Without Short Cycles

Michael A. Bekos   

Department of Mathematics,
University of Ioannina, Greece

Aaron Büngener 

Department of Computer Science,
University of Tübingen, Germany

Michael Hoffmann   

Department of Computer Science,
ETH Zürich, Switzerland

Pat Morin  

School of Computer Science,
Carleton University, Ottawa, Canada

Alexandra Weinberger   


Faculty of Mathematics and Computer Science,
FernUniversität in Hagen, Germany

Prosenjit Bose  

School of Computer Science,
Carleton University, Ottawa, Canada

Vida Dujmović  

School of Electrical Engineering and Computer
Science, University of Ottawa, Canada

Michael Kaufmann   

Department of Computer Science,
University of Tübingen, Germany

Saeed Odak 

School of Electrical Engineering and Computer
Science, University of Ottawa, Canada

Abstract

We study the impact of forbidding short cycles to the edge density of k -planar graphs; a k -planar graph is one that can be drawn in the plane with at most k crossings per edge. Specifically, we consider three settings, according to which the forbidden substructures are 3-cycles, 4-cycles or both of them (i.e., girth ≥ 5). For all three settings and all $k \in \{1, 2, 3\}$, we present lower and upper bounds on the maximum number of edges in any k -planar graph on n vertices. Our bounds are of the form cn , for some explicit constant c that depends on k and on the setting. For general $k \geq 4$ our bounds are of the form $c\sqrt{kn}$, for some explicit constant c . These results are obtained by leveraging different techniques, such as the discharging method, the recently introduced density formula for non-planar graphs, and new upper bounds for the crossing number of 2- and 3-planar graphs in combination with corresponding lower bounds based on the Crossing Lemma.

2012 ACM Subject Classification Mathematics of computing \rightarrow Combinatorics; Mathematics of computing \rightarrow Graph theory; Human-centered computing \rightarrow Graph drawings

Keywords and phrases Beyond-planar Graphs, k -planar Graphs, Local Crossing Number, Crossing Number, Discharging Method, Crossing Lemma

Digital Object Identifier 10.4230/LIPIcs.GD.2024.27

Related Version Full Version: <https://arxiv.org/abs/2408.16085v1> [6]

Funding Alexandra Weinberger: Part of this work was done while A. W. was at the University of Technology in Graz and supported by the Austrian Science Fund (FWF): W1230.

Acknowledgements This work was started at the Summer Workshop on Graph Drawing (SWG D 2023) in Caldana, Italy.

1 Introduction

“What is the minimum and maximum number of edges?” is one of the most fundamental questions one can ask about a finite family of graphs. In some cases the question is easy to answer; for instance, for the class of all graphs on n vertices the answer is even trivial. Another such family is the one of planar graphs. More precisely, for planar graphs on n vertices we know from Euler’s Formula that they have at most $3n - 6$ edges. Furthermore,



© Michael A. Bekos, Prosenjit Bose, Aaron Büngener, Vida Dujmović, Michael Hoffmann, Michael Kaufmann, Pat Morin, Saeed Odak, and Alexandra Weinberger; licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 27; pp. 27:1–27:17

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

every planar graph on n vertices can be augmented (by adding edges) to a maximal planar graph with exactly $3n - 6$ edges. Important advances have recently been made for non-planar graphs in the context of graph drawing beyond-planarity [15]. But often an answer is much harder to come by. Specifically, there exist graph classes that are relevant in Graph Drawing where exact bounds on their edge density are difficult to derive.

The family of graphs that can be embedded on the Euclidean plane with at most k crossings per edge, called k -planar, is a notable example. Tight bounds on the edge density of these graphs, for small values of k , are crucial as they lead to improvements on the well-known Crossing Lemma [3]. This was first observed by Pach and Tóth [22], who back in 1997 presented one of the early improvements of the Crossing Lemma by introducing tight bounds on the edge density of 1- and 2-planar graphs. Since then, only two improvements emerged; one by Pach, Radoičić, Tardos, and Tóth [17, 18] in 2004 and one by Ackerman [1] in 2019, both by introducing corresponding bounds on the edge density of 3- and 4-planar graphs, respectively. On the other hand, it is worth noting that these progressive refinements on the Crossing Lemma led to corresponding improvements also on the upper bound on the edge density of general k -planar graphs with the best one being currently $3.81\sqrt{kn}$ due to Ackerman [1]. To the best of our knowledge, for 5-planar graphs a tight bound is missing from the literature, even though it would yield further improvements both on the Crossing Lemma and on the upper bound of the edge density of general k -planar graphs. Variants of the Crossing Lemma have also been proposed for specific classes of graphs, e.g., bipartite graphs [5, 12, 13].

In this work, we continue the study of this line of research focusing on special classes of graphs; in particular, on graphs not containing some fixed, so-called *forbidden substructures*. We consider three settings, according to which the forbidden substructures are 3-cycles (C_3 -free), 4-cycles (C_4 -free) or both of them (girth ≥ 5). For each of these settings, the problem of finding edge density bounds has been studied both in general and assuming planarity. In particular, while C_3 -free n -vertex graphs may have $\Theta(n^2)$ edges, C_4 -free graphs and graphs of girth 5 have at most $O(n^{\frac{3}{2}})$ edges; see e.g. [14, 23]. For C_3 -free planar graphs and planar graphs of girth 5, one can easily derive upper bounds on their edge density using Euler's Formula; see, e.g., Table 1. For C_4 -free planar graphs, Dowden [11] proved that every such graph has at most $\frac{15}{7}(n - 2)$ edges, and that this bound is best possible. For k -planar graphs, Pach, Spencer and Tóth [19, 20] provided a lower bound on the crossing number of C_4 -free k -planar graphs, which can be used to obtain an asymptotic upper bound of $O(\sqrt[3]{kn})$ on the edge density of such graphs with n vertices. Another related research branch focuses on bipartite graphs (that avoid all odd-length cycles). For this setting, Angelini, Bekos, Kaufmann, Pfister, and Ueckerdt [5] have proposed lower and upper bounds on the edge density of several classes of graphs beyond-planarity, including 1- and 2-planar graphs.

Our contribution

We study the class of k -planar graphs in the absence of 3-cycles, 4-cycles and both of them. Our results are summarized as follows:

- For each of the aforementioned settings, we present lower and upper bounds on the maximum number of edges of k -planar graphs with n vertices when $k \in \{1, 2, 3\}$. Our findings are summarized in Table 1.
- We next use these bounds to derive corresponding lower bounds on the crossing numbers of the graphs that avoid the forbidden patterns studied. For a summary refer to Table 2.
- We use the two-way dependency between edge density and Crossing Lemma to derive new bounds on the edge density of k -planar graphs for values of k greater than 3.

■ **Table 1** Maximum number of edges in k -planar graph classes, ignoring additive constants; results from the literature are shown in blue square brackets, results from this paper are shown in red angle brackets, bounds without a citation are derived from Euler’s formula. The lower bound for 2-planar C_4 -free graphs trivially holds for 3-planar C_4 -free graphs.

k	unrestricted		C_3 -free		C_4 -free		Girth 5	
	lower	upper	lower	upper	lower	upper	lower	upper
0	$3n$	$3n$	$2n$	$2n$	$\frac{15n}{7}$ [11]	$\frac{15n}{7}$ [11]	$\frac{5n}{3}$	$\frac{5n}{3}$
1	$4n$ [8]	$4n$ [8]	$3n$ [10]	$3n$ (5)	$2.4n$ (7)	$2.5n$ (6)	$\frac{13n}{6}$ (9)	$2.4n$ (8)
2	$5n$ [22]	$5n$ [22]	$3.5n$ [5]	$4n$ (10)	$2.5n$ (14)	$3.93n$ (11)	$\frac{16n}{7}$ (17)	$3.597n$ (15)
3	$5.5n$ [18]	$5.5n$ [18]	$4n$ [5]	$5.12n$ (18)	–	$4.933n$ (20)	$2.5n$ (22)	$4.516n$ (21)
k	$\Omega(\sqrt{k})n$ [22]	$3.81\sqrt{kn}$ [1]		$3.19\sqrt{kn}$ (19)		$3.016\sqrt{kn}$ (12) $O(\sqrt[3]{k})n$ [20]		$2.642\sqrt{kn}$ (16) $O(\sqrt[3]{k})n$ [20]

■ **Table 2** Bounds on the crossing numbers, ignoring additive constants; hold for sufficiently large m .

Graph class	unrestricted		C_3 -free	C_4 -free	Girth 5
	lower	upper	lower	lower	lower
2-planar		$\frac{10n}{3}$ (3)			
3-planar		$\frac{33n}{5}$ (4)			
general	$0.034\frac{m^3}{n^2}$ [1]		$0.049\frac{m^3}{n^2}$ (18)	$0.054\frac{m^3}{n^2}$ (11)	$0.071\frac{m^3}{n^2}$ (15)

To obtain the above results, we leverage different techniques from the literature, such as the discharging method, the recently introduced density formula for non-planar graphs [16], and new upper bounds for the crossing number of 2- and 3-planar graphs (Theorems 3 and 4) in combination with corresponding lower bounds based on the Crossing Lemma.

2 Preliminary Techniques and Tools

In this section, we describe techniques that we use in our proofs, namely, the discharging method [1, 2] (Section 2.1) and a method derived from a well-known probabilistic proof [3] of the Crossing Lemma (Section 2.2), which we formalise in the following. This section is concluded with two theorems of independent interest providing upper bounds on the number of crossings of (general) 2- and 3-planar graphs (Section 2.3).

2.1 The Discharging Method

In some of our proofs, we employ the discharging method [1, 2], which is summarised as follows. Consider a biconnected graph $G = (V, E)$ on $|V| = n$ vertices drawn in \mathbb{R}^2 and its planarization $G' = (V', E')$, where at every crossing both edges are subdivided using a new vertex of degree four. We denote the set of faces of G' by F' and call them *cells*. For a face $f \in F'$ we denote by $\mathcal{V}(f)$ and $\mathcal{V}'(f)$ the set of vertices from V and V' , respectively, that appear on the boundary ∂f of f . Furthermore, let $|f| = |\mathcal{V}'(f)|$ denote the *size* of f .

To each face $f \in F'$ we assign a charge $\text{ch}(f) = |\mathcal{V}(f)| + |f| - 4$. Using Euler’s formula $|V'| - |E'| + |F'| = 2$, it is not difficult to check (see [1]) that $\sum_{f \in F'} \text{ch}(f) = 4n - 8$.

We then distribute these charges so as to collect a discharge of at least α , for some $\alpha > 0$, for every pair $(v, f) \in V \times F'$ such that $v \in \mathcal{V}(f)$.

27:4 On k -Planar Graphs Without Short Cycles

Then $4n - 8 = \sum_{f \in F'} \text{ch}(f) \geq \sum_{v \in V} \alpha \deg_G(v) = 2\alpha|E|$ which implies

$$m = |E| \leq \frac{2}{\alpha}(n - 2). \quad (1)$$

The main challenge when applying this discharging method is to manage the redistribution of charges so that every vertex receives its due, for α as large as possible. As a natural first attempt, we may have each $f \in F'$ discharge α to each $v \in \mathcal{V}(f)$. This leaves f with a *remaining* charge of

$$\text{ch}^-(f) = \text{ch}(f) - \alpha|\mathcal{V}(f)| = (1 - \alpha)|\mathcal{V}(f)| + |f| - 4. \quad (2)$$

If $\text{ch}^-(f) \geq 0$, for all $f \in F'$, then we are done. However, in general, we may have $\text{ch}^-(f) < 0$, for some $f \in F'$. In such a case we have to find some other face(s) that have a surplus of remaining charge they can send to f .

2.2 The Crossing Lemma

We can obtain upper bounds on the density also using the Crossing Lemma [4]. As a basis, we need both an upper and a lower bound for the crossing number in terms of the number of vertices and edges. Upper bounds are discussed in Section 2.3. In this section we derive a lower bound using the Crossing Lemma, along the lines of its well-known probabilistic proof [3, Chapter 40].

► **Theorem 1.** *Let \mathcal{X} be a hereditary¹ graph family and $a, b \in \mathbb{R}$ such that for every $H \in \mathcal{X}$ with ν vertices and μ edges we have $\text{cr}(H) \geq a\mu - b\nu$. Then for every graph $G \in \mathcal{X}$ with n vertices and m edges with $2am \geq 3bn$ we have*

$$\text{cr}(G) \geq \frac{4a^3}{27b^2} \cdot \frac{m^3}{n^2}.$$

Proof. Let Γ be a minimum-crossing drawing of G . We take a random induced subgraph $G_p = (V_p, E_p)$ of G by selecting every vertex independently at random with probability p and consider the drawing Γ_p of G_p defined by Γ . Then any such graph G_p is in \mathcal{X} , and so the lower bound on $\text{cr}(G_p)$ from above holds for G_p and thus also in expectation:

$$\mathbb{E}(\text{cr}(\Gamma_p)) \geq a \cdot \mathbb{E}(E_p) - b \cdot \mathbb{E}(V_p).$$

We have $\mathbb{E}(V_p) = pn$ and $\mathbb{E}(E_p) = p^2m$. Furthermore, note that Γ is a minimum-crossing drawing of G and, therefore, no pair of adjacent edges crosses. Thus, for a crossing to be present in Γ_p , all four endpoints of the crossing edge pair need to be selected. Therefore, we have $\mathbb{E}(\text{cr}(\Gamma_p)) = p^4\text{cr}(\Gamma) = p^4\text{cr}(G)$. Putting everything together yields

$$\text{cr}(G) \geq \frac{am}{p^2} - \frac{bn}{p^3}. \quad (3)$$

The function on the right hand side of the above inequality has its unique maximum at $p = \frac{3bn}{2am}$. Setting $p = \frac{3bn}{2am}$ to (3) yields:

$$\text{cr}(G) \geq \frac{4a^3}{27b^2} \cdot \frac{m^3}{n^2}. \quad (4)$$

As a sanity check, we need $p \leq 1$. So the bound holds for $2am \geq 3bn$. ◀

¹ Closed under taking induced subgraphs.

The simple observation that one can remove relatively few edges from a k -planar graph to obtain a $(k - 1)$ -planar graph allows to lift density bounds for i -planar graphs, with $i < k$, to bounds for k -planar graphs. By iteratively removing edges from the graph and a drawing of it with maximum number of crossings, we can show the following. (The proof can be found in the full version of this paper [6].)

► **Theorem 2.** *Let \mathcal{X} be a monotone² graph family, let k be a positive integer, and let $\mu_i(n)$ be an upper bound on the number of edges for every i -planar graph from \mathcal{X} on n vertices, for $0 \leq i \leq k - 1$. Then for every $G \in \mathcal{X}$ with $n \geq 4$ vertices and m edges we have*

$$\text{cr}(G) \geq km - \sum_{i=0}^{k-1} \mu_i(n).$$

2.3 Upper Bounds on the Crossing Number of 2- and 3-planar graphs

The Crossing Lemma provides us with pretty good lower bounds for crossing numbers. As a complement, we also need corresponding upper bounds. For a k -planar graph G , we have a trivial bound of $\text{cr}(G) \leq km/2$. So if G is 2-planar, then $\text{cr}(G) \leq m \leq 5n - 10$. But we can do better, as the following theorem demonstrates.

► **Theorem 3.** *Every 2-planar graph on $n \geq 2$ vertices can be drawn with at most $(10n - 20)/3$ crossings.*

Proof. Let $G = (V, E)$ be a 2-planar graph on n vertices, and let Γ be any 2-plane drawing of G with a minimum number of crossings (among all 2-plane drawings of G). We allow multiple edges between the same pair of vertices in Γ , but no loops nor homotopic edge pairs (that is, for each pair e_1, e_2 of edges between the same two vertices, neither of the two parts of the plane bounded by the simple closed curve $e_1 \cup e_2$ is empty). Without loss of generality we assume that Γ is maximal 2-plane, that is, adding any edge to Γ results in a graph that is not 2-plane anymore. We may assume that adjacent edges do not cross in Γ [18, Lemma 1.1]. We claim that a $1/3$ -fraction of the edges in Γ is uncrossed.

Let us first argue how the claim implies the statement of the theorem. Denote by x the number of edges that have at least one crossing in Γ . The number γ of crossings in Γ is upper bounded by $2x/2 = x$ because every edge has at most two crossings and every crossing is formed by exactly two edges. Every 2-planar graph on $n \geq 3$ vertices has at most $5n - 10$ edges [21, 22], and this bound also holds for 2-plane multigraphs without loops or parallel homotopic edges [7]. It follows that $\gamma \leq x \leq \frac{2}{3}(5n - 10) = (10n - 20)/3$.

So it remains to prove the claim. Consider a vertex v and denote by $X(v)$ the set of edges incident to v that have at least one crossing in Γ . Let $e \in X(v)$, let c denote the crossing of e closest to v , let e^- denote the part of e between v and c , and let $\chi(e)$ denote the edge that crosses e at c . As $\chi(e)$ has at most two crossings, at least one of the two curves that form $\chi(e) \setminus c$ is uncrossed. Pick such a curve and denote it by $\chi(e)^-$. The curve $\chi(e)^-$ has two endpoints, one of which is c and the other is a vertex of G , which we denote by $\psi(e)$. As adjacent edges do not cross in Γ , we have $\psi(e) \neq v$. By closely following e^- and $\chi(e)^-$ we can draw a curve between v and $\psi(e)$ in Γ that does not cross any edge of Γ . Thus, by the maximality of Γ we conclude that $v\psi(e)$ is an edge in Γ , and it is uncrossed because Γ is crossing-minimal by assumption. In this way, we find an uncrossed edge $\eta(e) = v\psi(e)$ for

² Closed under taking subgraphs and disjoint unions.

each $e \in X(v)$. Different edges $e \neq f$ in $X(v)$ may yield the same edge $\eta(e) = \eta(f)$. But in this case by construction $\eta(e) = \eta(f)$ is homotopic to both $e^- \cup \chi(e)^-$ and $f^- \cup \chi(f)^-$, that is, the simple closed curve $e^- \cup \chi(e)^- \cup f^- \cup \chi(f)^-$ bounds a face in $\Gamma \setminus \eta(e)$. It follows that there is no other edge $g \in X(v) \setminus \{e, f\}$ for which $\eta(g) = \eta(e)$, that is, for every uncrossed edge u incident to v in Γ we have $|\eta^{-1}(u) \cap X(v)| \leq 2$. Therefore, at least a $1/3$ -fraction of the edges incident to v in Γ is uncrossed. As this holds for every vertex v , it also holds globally, which completes the proof of the claim and of the theorem. \blacktriangleleft

In a similar fashion, we can obtain an improved upper bound for 3-planar graphs, as the following theorem demonstrates. The proof can be found in the full version of this paper [6]. We remark that the argument used in the proof of Theorem 4 does not work for larger $k > 3$.

► **Theorem 4.** *Every 3-planar graph on $n \geq 2$ vertices can be drawn with at most $(33n - 66)/5$ crossings.*

3 1-planar graphs

In this section we focus on 1-planar graphs and we present lower and upper bounds on their edge density assuming that they are either C_3 -free (Section 3.1) or C_4 -free (Section 3.2) or of girth 5 (Section 3.3).

3.1 C_3 -free 1-planar graphs

We start with the case of C_3 -free 1-planar graphs, where we can derive an upper bound of $3(n - 2)$ on their edge density (see Theorem 5); for a matching lower bound (up to a small additive constant) refer to [10].

► **Theorem 5.** *Every C_3 -free 1-planar graph with $n \geq 4$ vertices has at most $3(n - 2)$ edges.*

Proof. We derive the upper bound by an application of the recently introduced edge-density formula for non-planar graphs [16] given as follows:

$$|E| \leq t(|V| - 2) - \sum_{c \in \mathcal{C}} \left(\frac{t-1}{4} \|c\| - t \right) - |\mathcal{X}|, \tag{5}$$

where \mathcal{C} and \mathcal{X} denote the sets of cells and crossings, respectively. By setting $t = 3$ to (5), one gets $|E| \leq 3(n - 2) + \frac{1}{2}|\mathcal{C}_5| - \frac{1}{2}|\mathcal{C}_6| - \dots - |\mathcal{X}|$, where C_i denotes the set of cells of size i with the size of a cell being the number of vertices and edge-segments on its boundary. Since each crossing is incident to at most two cells of size 5 (as otherwise a C_3 is inevitably formed), it follows that $\frac{1}{2}|\mathcal{C}_5| \leq |\mathcal{X}|$, which by the formula given above implies that $|E| \leq 3(n - 2)$. \blacktriangleleft

3.2 C_4 -free 1-planar graphs

We continue with the case of C_4 -free 1-planar graphs. As in the case of C_3 -free 1-planar graphs, we can again derive an upper bound of $3(n - 2)$ for the edge-density using the density formula of (5), since each crossing is incident to at most two cells of size 5 (as otherwise a C_4 is formed). In the following theorem, we present an improved upper bound.

► **Theorem 6.** *Every C_4 -free 1-planar graph with $n \geq 4$ vertices has at most $\frac{5}{2}(n - 2)$ edges.*

Proof. We apply the discharging method with $\alpha = 4/5$ so that the statement follows by (1). By (2) we have

$$\text{ch}^-(f) = \frac{1}{5}|\mathcal{V}(f)| + |f| - 4. \tag{6}$$

In particular, we have $\text{ch}^-(f) > 0$ for all faces with at least four edge segments on the boundary. It remains to handle triangles.

As the graph G is 1-planar, every edge of G' is incident to at least one vertex in V . It follows that

$$|\mathcal{V}(f)| \geq \lceil |f|/2 \rceil, \tag{7}$$

for each $f \in F'$. So every triangle $f \in F'$ has either three vertices in V and $\text{ch}^-(f) = -2/5$ (type-1) or two vertices in V and one vertex in $V' \setminus V$ with $\text{ch}^-(f) = -3/5$ (type-2).

We will argue how to make up for the deficits at triangles by transferring charges from neighboring faces.

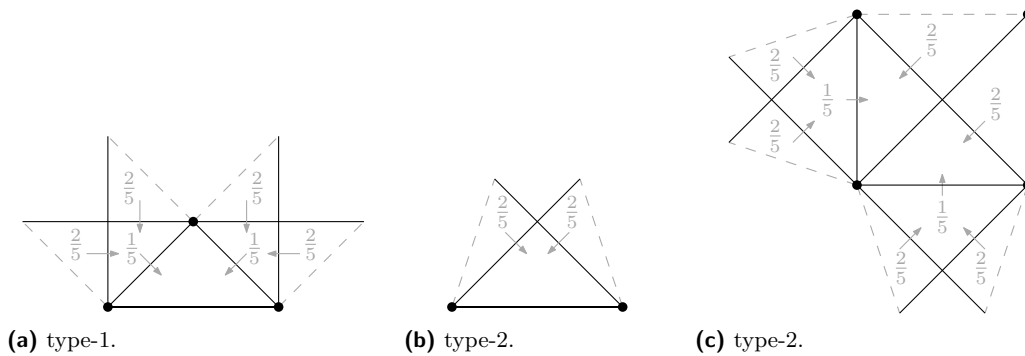
First, let us discuss faces of size at least five. So consider $f \in F'$ with $|f| \geq 5$, and let k denote the number of triangles adjacent to f in the dual of G' . Then for any vertex $v \in \mathcal{V}(f) \setminus \mathcal{V}(f)$, at most one of the two edges incident to v along ∂f can be incident to a triangle of F' (because otherwise the two edges of G that cross at v induce a C_4). Thus,

$$k \leq |\mathcal{V}(f)| + \frac{|f| - |\mathcal{V}(f)|}{2} = \frac{|f| + |\mathcal{V}(f)|}{2}.$$

Together with (6) we obtain

$$\text{ch}^-(f) = \frac{1}{5}|\mathcal{V}(f)| + |f| - 4 = \frac{|f| + |\mathcal{V}(f)|}{5} + \frac{4}{5}|f| - 4 \geq \frac{2}{5}k,$$

which shows that f can send a charge of $2/5$ to every adjacent triangle.



■ **Figure 1** Triangles in the planarization of C_4 -free 1-planar graphs.

Next, consider a face f with $|f| = 4$. Combining (2) and (7) we obtain $\text{ch}^-(f) = |\mathcal{V}(f)|/5 \geq 2/5$. We claim that f can send a charge of $2/5$ to every triangle that is adjacent to f via an edge of $E' \setminus E$ and a charge of $1/5$ to every triangle that is adjacent to f via an edge of E . To see this, let us consider the three different types of quadrangles in F' . By (7) we have $|\mathcal{V}(f)| \geq 2$.

If $|\mathcal{V}(f)| = 2$, then there is at most one triangle adjacent to f because any two triangles adjacent to f induce a C_4 . So in this case f can send a charge of $2/5$ to every adjacent triangle.

If $|\mathcal{V}(f)| = 3$, then any triangle adjacent to f via an edge of $E' \setminus E$ induces a C_4 in G . Thus, there exist at most two triangles adjacent to f and every such triangle is adjacent via an edge of E . So in this case f can send a charge of $1/5$ to every adjacent triangle.

Finally, if $|\mathcal{V}(f)| = 4$, then every triangle adjacent to f is adjacent via an edge of E . As $\text{ch}^-(f) = \frac{4}{5}$, also in this case f can send a charge of $1/5$ to every adjacent triangle. This completes the proof of our claim.

So let us consider the incoming charges at triangles. For a type-1 triangle f , neither of the adjacent faces is a type-1 triangle because such a pair would induce a C_4 in G . If at least two adjacent faces are type-2 triangles, then for each such triangle g , neither of the other ($\neq f$) two faces adjacent to g are triangles because together with f and g they would induce a C_4 . It follows that g receives a charge of $2 \cdot 2/5 = 4/5$ from its two other ($\neq f$) neighbors, see Figure 1a. As $\text{ch}^-(g) = -3/5$, the remaining charge of $1/5$ can be passed on to f . Then f receives a charge of $2 \cdot 1/5 = 2/5 = -\text{ch}^-(f)$ overall. Otherwise, at least two of the three faces adjacent to f have size at least four. Each passes a charge of $1/5$ across the joint edge, which is in E , to f . So the deficit of $\text{ch}^-(f) = -2/5$ is covered in this case as well.

It remains to consider type-2 triangles. Let f be a type-2 triangle, and consider the two faces g_1, g_2 that are adjacent to f via an edge of $E' \setminus E$. If both g_1 and g_2 are triangles, then they induce a C_4 in G , in contradiction to G being C_4 -free. If both g_1 and g_2 have size at least four, then f receives a charge of $2 \cdot 2/5 = 4/5$ from them, which covers $\text{ch}^-(f) = -3/5$ and even leaves room to send a charge of $1/5$ across its third edge, which is in E , see Figure 1b.

Hence, we may assume that without loss of generality g_1 is a type-2 triangle and $|g_2| \geq 4$. The third face $g_3 \notin \{g_1, g_2\}$ adjacent to f is not a type-1 triangle because then g_3 together with g_1 would induce a C_4 in G . If g_3 is a type-2 triangle, then neither of its two other ($\neq f$) neighbors is a triangle because together with f and g_1 there would be a C_4 in G . Therefore, we are in the case discussed above, where g_3 receives a charge of $4/5$ from its neighbors and passes on $1/5$ to f . Otherwise, we have $|g_3| \geq 4$ and thus g_3 sends a charge of $1/5$ to f across the joint edge, which is in E . Together with the charge of $2/5$ that f receives from g_2 via the joint edge, which is in $E' \setminus E$, this suffices to cover $\text{ch}^-(f) = -3/5$, see Figure 1c. ◀

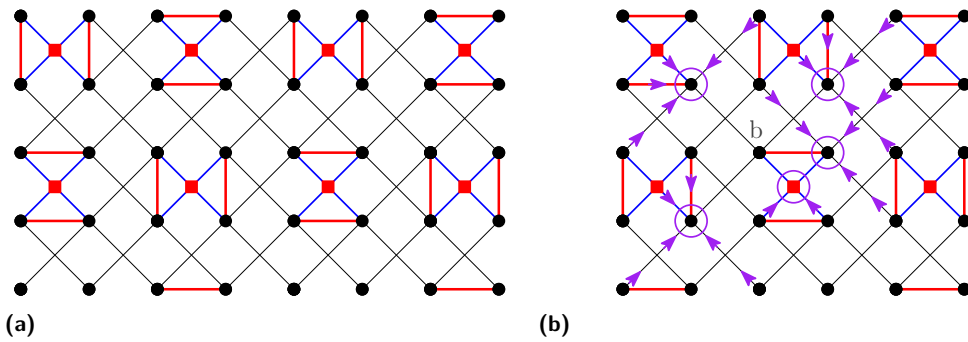
► **Theorem 7.** *For every sufficiently large n , there exists a C_4 -free 1-planar graph on n vertices with $2.4n - O(1)$ edges.*

Proof sketch. It can be observed that the construction in Figure 2a is a 1-plane drawing and achieves the required number of edges. To show that it is C_4 -free, we first observe that all (red) degree four vertices behave symmetrically and all (black) degree five vertices behave symmetrically. Further, all neighbors of degree four vertices have degree five, thus any cycle must contain a degree five vertex. We consider an arbitrary degree five vertex b and show that each neighbor of a vertex in $N(b)$ (circled in Figure 2b) has one unique neighbor in $N(b)$ (indicated by arrows in Figure 2b); see Figure 2b. Thus, the construction is C_4 -free. ◀

3.3 1-planar graphs of girth 5

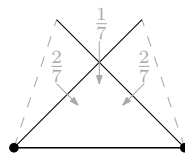
► **Theorem 8.** *Every 1-planar graph of girth 5 on n vertices has at most $2.4n$ edges.*

Proof. We go through the proof of the 1-planar C_4 -free case, and note that the arrow case as well as the type-1 triangles do not occur. If we choose $\alpha = \frac{5}{6}$, the type-2 triangles have a negative charge of $-\frac{4}{6}$, and can get charges of $\frac{2}{6}$ from their immediate neighboring cells which are of size at least 4. Note that for the case that those neighboring cells are of size 4, they have only one type-2 triangle by the C_4 -freeness property, which suffices to provide



■ **Figure 2** (a) A dense C_4 -free 1-plane graph. (b) The neighborhood of a vertex b .

enough charge. If a neighboring cell c has size 5, then, by 1-planarity, it shares with at least one neighbor a planar edge, through which it does not have to contribute charge. Since the remaining charge of c is at least $3 \cdot \frac{1}{6} + 5 - 4 = 1.5$, it can contribute to four neighbours $\frac{2}{6}$ charge each. If a neighboring cell c is of size larger than 5, then its remaining charge is at least $\frac{1}{6}|\mathcal{V}(c)| + |c| - 4 \geq |c| - 3 \geq \frac{2}{6}|c|$, and therefore there is enough charge to provide $\frac{2}{6}$ charge to every neighboring type-2 triangle. This immediately gives that an n -vertex 1-planar graph of girth 5 has at most $\frac{12}{5}n = 2.4n$ edges. ◀



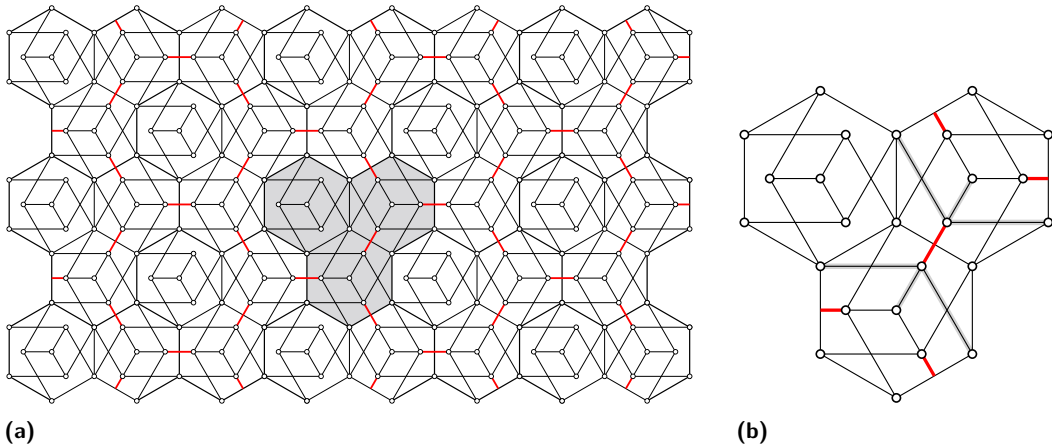
■ **Figure 3** Triangles in the planarization of 1-planar graphs of girth 5.

► **Theorem 9.** *For every sufficiently large n , there exists a 1-planar graph of girth 5 on n vertices with $(2 + \frac{1}{6})n - O(1) \approx 2.167n - O(1)$ edges.*

Proof sketch. It can be observed that the construction in Figure 4a is a 1-plane drawing and has the required number of edges. To show that it has girth 5, we first observe the subgraph within each hexagonal tile has girth 5. It is a Petersen graph, which is known to have girth 5. It follows that every C_3 or C_4 , if any, uses vertices from at least two different tiles. Second, we argue that no C_3 or C_4 uses a red edge. The neighbors of the two endpoints of a red edge are at pairwise distance at least two; see Figure 4b. As red edges are the only edges that cross tile boundaries and boundary edges are shared among adjacent tiles, it follows that every C_3 or C_4 , if any, uses at least two nonadjacent vertices u, v on the boundary ∂T of a tile T and exactly one vertex z in the interior of T . Then u and v are at distance three along ∂T and thus do not form a C_3 . Further, the tile T is the unique common tile of u and v , so there is no common neighbor of u and v outside of T . As z is the only common neighbor of u and v inside T , it follows that there is no C_4 through u, v, z . ◀

4 2-planar graphs

In this section, we focus on 2-planar graphs and we present bounds on their edge density assuming that they are C_3 -free (Section 4.1) or C_4 -free (Section 4.2) or of girth 5 (Section 4.3).



■ **Figure 4** (a) A 1-planar graph of girth 5 with about $2.167n$ edges (Theorem 9). The construction consists of repeated triplets of hexagonal tiles (bordered by thick edges, also shown in (b)).

4.1 C_3 -free 2-planar graphs

For the maximum edge density of C_3 -free 2-planar graphs, we can derive an upper bound of $4(n - 2)$ (see Theorem 10); for a lower bound of $3.5(n - 2)$ refer to [5].

► **Theorem 10.** *C_3 -free 2-planar graphs with n vertices have at most $4(n - 2)$ edges.*

Proof. To derive the upper bound, we apply the discharging method with $\alpha = \frac{1}{2}$ so that the statement follows by (1). By (2) we have

$$\text{ch}^-(f) = \frac{1}{2}|\mathcal{V}(f)| + |f| - 4. \quad (8)$$

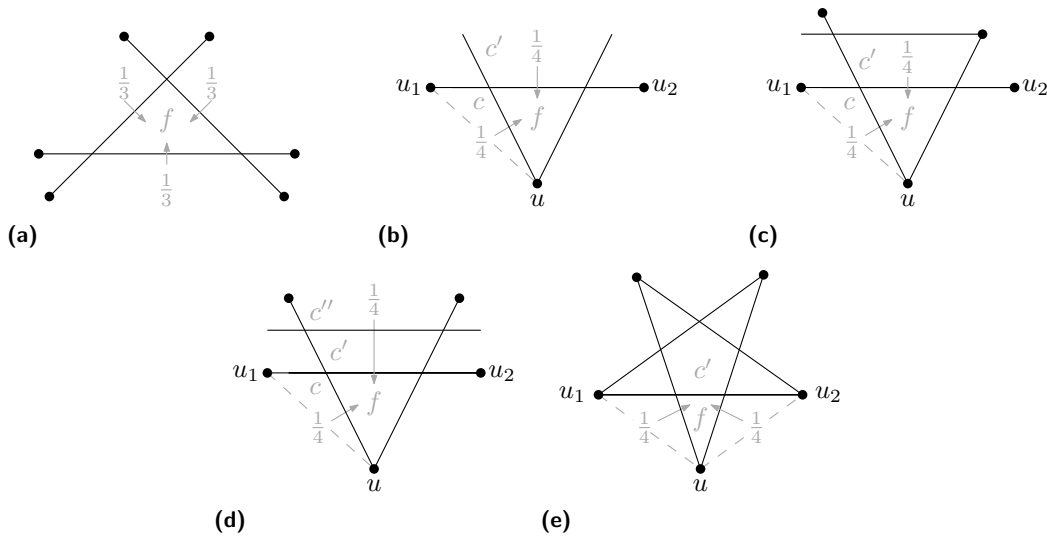
In particular, we have $\text{ch}^-(f) \geq 0$ for all faces with at least four edges on the boundary. It remains to handle triangles. Since we consider C_3 -free graphs, we distinguish between three types of triangles; those with 0, 1 and 2 vertices on their boundaries and it is not difficult to observe that the latter ones have zero charge, while the former ones have charge -1 and $-\frac{1}{2}$, respectively.

For each triangle f with zero or one vertices on its boundary, our strategy is to transfer at least $\frac{1}{4}$ and at most $\frac{1}{3}$ units of charge from the cells neighboring f . If f has no vertices on its boundary, then we will transfer $\frac{1}{3}$ units of charge from each neighboring cell. Otherwise, we will transfer $\frac{1}{4}$ units of charge each from two neighboring cells of f ; see Figure 5.

Assume first that $|\mathcal{V}(f)| = 0$; see Figure 5a. Since f is triangular, it follows that f is formed by three mutually crossing edges. Our strategy is to transfer $\frac{1}{3}$ units of charge from each cell neighboring f . Since f neighbors three such cells, this is enough to bring the remaining charge of f from -1 to 0. Let c be a neighboring cell of f . It follows that $|c| \geq 4$ with two vertices on its boundary. Thus its remaining charge is:

$$\frac{1}{2}|\mathcal{V}(c)| + |c| - 4 \geq 1 - 4 + |c| = |c| - 3$$

This implies that if $|c| \geq 5$, then the remaining charge of c is at least 2, in which case c can transfer $\frac{1}{3}$ units of charge to f and its remaining charge will be enough to distributed to the rest of its neighboring cells. For the second case, we assume $|c| = 4$. Since c has two vertices that appear consecutively on its boundary, it follows that one of the sides that bound c is



■ **Figure 5** Triangles in the planarization of C_3 -free 2-planar graphs.

crossing free. Denote this side by e and let c' be the cell on the other side of e . It follows that c' is neither a triangle with zero vertices nor a triangle with one vertex on its boundary. Hence, there is no need to transfer charge from c to c' according to our strategy. It follows that there are at most 3 neighboring cells that c may have to transfer charge to. Hence, c can transfer $\frac{1}{3}$ units of charge to f and its remaining charge will be enough to distributed to the rest of its neighboring cells, if needed.

To complete the proof of the theorem, we next consider the case in which $|\mathcal{V}(f)| = 1$; see Figure 5b. Let u be the vertex on the boundary of f and let (u_1, u_2) be the edge with one of its segments on the boundary of f . Let c_1 and c_2 be the two neighboring cells of f that share the two sides of f incident to its vertex. Since we consider C_3 -free graphs, it follows that (u, u_1) and (u, u_2) cannot be both in the graph. Assume that (u, u_i) with $i \in \{1, 2\}$ is not part of the graph. Then, the corresponding cell $c \in \{c_1, c_2\}$ neighboring f and having vertex u and u_i on its boundary has size at least 4, which means that its remaining charge is at least:

$$\frac{1}{2}|\mathcal{V}(c)| + |c| - 4 \geq 1 - 4 + |c| = |c| - 3 \geq \frac{1}{4}|c|$$

Hence, we can safely transfer $\frac{1}{4}$ units of charge from c to f , since the remaining charge of c would be enough for being distributed to the remaining cells neighboring c , if needed. This implies that if both (u, u_1) and (u, u_2) are not in the graph, then each of the cells c_1 and c_2 can transfer $\frac{1}{4}$ units of charge to f and then we are done. So, in the rest we can assume that this is not the case.

Let c' be the face neighboring f that is on the other side of the edge (u_1, u_2) . If c' has at least two vertices on its boundary, then as above we transfer $\frac{1}{4}$ units of charge from c' to f and the remaining charge of c' would be enough for being distributed to the remaining cells neighboring c' , if needed. So, it remains to consider the cases in which c' has either no or one vertex on its boundary.

Assume first that c' has one vertex on its boundary, that is, $\mathcal{V}(c') = 1$. Then:

$$\frac{1}{2}|\mathcal{V}(c')| + |c'| - 4 \geq \frac{1}{2} - 4 + |c'| = |c'| - 3.5$$

If c' is such that $|c'| \geq 5$, then $|c'| - 3.5 \geq \frac{1}{4}|c'|$ holds and as above we can safely transfer $\frac{1}{4}$ units of charge from c' to f . So, it remains to argue for the cases in which $|c'| \in \{3, 4\}$. First, we observe that $|c'| \neq 3$, as otherwise the two edges incident to u bounding f would form a pair of parallel edges. Hence, we may assume that $|c'| = 4$; see Figure 5c. Since c' has one vertex on its boundary, its remaining charge is $\frac{1}{2}$. In this case, we argue that at most two neighboring cells, namely, f and another one, may need additional charge from c' . In particular, the two cells neighboring c' that have the vertex of c' on their boundary do not need additional charge, since none of them can be a triangle with zero or one vertex on its boundary. This means that we can safely transfer $\frac{1}{4}$ units of charge from c' to f , as desired.

To complete the case analysis, we need to consider the case that c' has no vertex on its boundary. In this case, the remaining charge of c' is $|c'| - 4$. If $|c'| \geq 6$, then the remaining charge of c' is at least 2, which implies that $\frac{1}{4}$ units of charge can be safely transferred to f and the remaining charge of c' will be enough for being distributed to the rest of the cells neighboring c' , if needed. So, we may assume that $|c'| \in \{3, 4, 5\}$. First, we observe that $|c'| \neq 3$, as otherwise the two edges incident to u bounding f would form a pair of crossing edges, which is not possible in simple drawings. Hence, $|c'| \in \{4, 5\}$. If $|c'| = 4$, then its remaining charge is 0 and clearly it cannot transfer charge to f . In this case, we consider the cell c'' neighboring c' , which does not share a crossing point with f ; see Figure 5d. It follows that $|c''| \geq 4$ and c'' has two vertices on its boundary. Since c' does not require a transfer of charge, we transfer $\frac{1}{4}$ units of charge from c'' to f and as in the first case of the proof the remaining charge of c'' for being distributed to the rest of the cells neighboring c'' .

To complete the proof of the case $|\mathcal{V}(f)| = 1$, consider now the case $|c'| = 5$. In this case, the remaining charge of c' is 1 and this is enough to contribute a $\frac{1}{4}$ to at most four neighboring cells. Hence, we may assume that c' has to transfer $\frac{1}{4}$ units of charge to exactly five neighboring cells; see Figure 5e. In this case, it follows that none of the edges (u, u_1) and (u, u_2) is part of the graph (as otherwise there is a C_3 ; a contradiction). However, we have assumed that one of these edges belongs to the graph. ◀

4.2 C_4 -free 2-planar graphs

We continue with the case of C_4 -free 2-planar graphs, deriving an upper bound $3.929n$ on their maximum edge density (Theorem 11); for a lower bound of $2.5n - O(1)$ refer to Theorem 14.

► **Theorem 11.** *Every C_4 -free 2-planar graph on $n \geq 2$ vertices has at most*

$$\sqrt[3]{\frac{190,125}{3,136}}n < 3.929n$$

edges.

Proof. Let G be a C_4 -free graph with n vertices and m edges, and let Γ be a minimum-crossing drawing of G . Then Theorem 2 in combination with the upper bound of $\frac{15}{7}(n-2)$ by Dowden [11] regarding the edge density of C_4 -free planar graphs and Theorem 6 yields:

$$\text{cr}(G) \geq 2m - \frac{5}{2}n - \frac{15}{7}n = 2m - \frac{65}{14}n.$$

By applying Theorem 1 for $a = 2$ and $b = \frac{65}{14}$, we obtain the following lower bound on the number of crossings of G when $m \geq \frac{195n}{56} \approx 3.482n$.

$$\text{cr}(G) \geq \frac{6,272m^3}{114,075n^2}. \quad (9)$$

Assume now that G is additionally 2-planar. Then by Theorem 3, we obtain $\text{cr}(G) \leq \frac{10n}{3}$. Hence, by (9) we have:

$$\frac{6,272m^3}{114,075n^2} \leq \frac{10}{3}n \iff m^3 \leq \frac{190,125}{3,136}n^3. \quad \blacktriangleleft$$

► **Corollary 12.** *Every C_4 -free k -planar graph on $n \geq 2$ vertices and $m \geq 3.483n$ edges has at most*

$$\sqrt{\frac{114,075}{12,544}} \cdot \sqrt{k} \cdot n < 3.016\sqrt{kn}$$

edges.

Proof. Let G be a C_4 -free k -planar graph with n vertices and $m \geq 3.483n$ edges. By (9), we know a lower bound on its number of crossings, namely,

$$\text{cr}(G) \geq \frac{6,272m^3}{114,075n^2}.$$

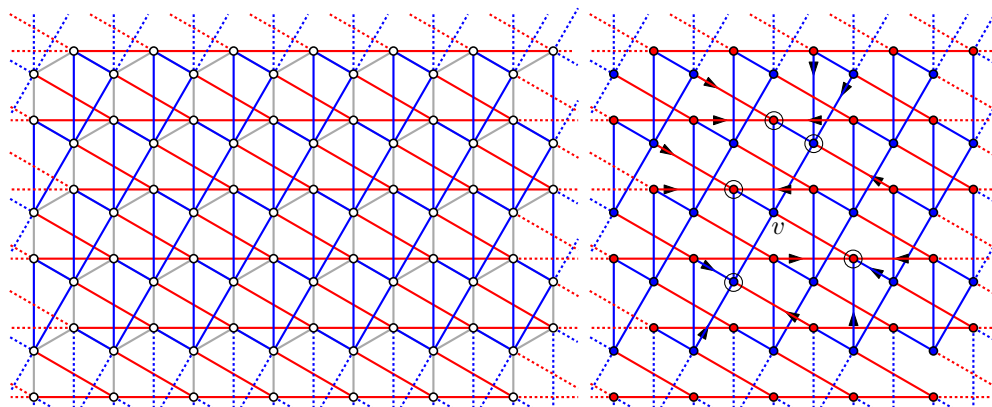
On the other hand, since G is k -planar, it holds $\frac{km}{2} \geq \text{cr}(G)$. Combining those, we get:

$$\frac{12,544m^2}{114,075n^2} \leq k \iff m \leq \sqrt{\frac{114,075}{12,544}kn}. \quad \blacktriangleleft$$

► **Remark 13.** An asymptotically better bound of $\Theta(\sqrt[3]{kn})$ edges, which however holds for significantly denser graphs only, can be obtained by combining an improved crossing lemma for C_4 -free graphs by Pach, Spencer, and Tóth [20, Theorem 3.1] with the trivial upper bound of at most $km/2$ crossings for k -planar graphs.

► **Theorem 14.** *For every sufficiently large n , there exists a C_4 -free 2-planar graph on n vertices with $2.5n - O(1)$ edges.*

Proof sketch. The construction is illustrated in Figure 6. The proof that it fulfills the claimed properties follows the same ideas as the proof of Theorem 7. The proof can be found in the full version of this paper [6]. ◀



■ **Figure 6** A 2-plane graph with $\approx 2.5n$ edges, shown red and blue. Gray shows the grid only.

4.3 2-planar graphs of girth 5

We conclude Section 4 with the case of 2-planar graphs of girth 5.

► **Theorem 15.** *Every 2-planar graph of girth 5 on n vertices has at most*

$$\sqrt[3]{\frac{11,163}{240}}n < 3.597n$$

edges.

Proof. Let G be a graph of girth 5 with n vertices and m edges. As a consequence of Euler's Formula, every planar graph of girth g on $n \geq 3$ vertices has at most $g(n-2)/(g-2)$ edges. Plugging this together with Theorem 8 into Theorem 2 we get

$$\text{cr}(G) \geq 2m - \frac{5}{3}n - \frac{12}{5}n = 2m - \frac{61}{15}n.$$

By applying Theorem 1 for $a = 2$ and $b = \frac{61}{15}$, we obtain the following lower bound on the number of crossings of G when $m \geq \frac{61n}{20}$.

$$\text{cr}(G) \geq \frac{800m^3}{11,163n^2}. \quad (10)$$

Assume now that G is additionally 2-planar. Then by Theorem 3, we obtain $\text{cr}(G) \leq \frac{10n}{3}$. Hence, by (10) we have:

$$\frac{800m^3}{11,163n^2} \leq \frac{10}{3}n \iff m^3 \leq \frac{11,163}{240}n^3. \quad \blacktriangleleft$$

The next corollary follows from (10) of the proof of Theorem 15; its proof is analogous to the one of Corollary 12.

► **Corollary 16.** *Every k -planar graph of girth 5 on $n \geq 2$ vertices and $m \geq 3.05n$ edges has at most*

$$\sqrt{\frac{11,163}{1,600}} \cdot \sqrt{k} \cdot n < 2.642\sqrt{kn}$$

edges.

Proof. Let G be a k -planar graph of girth 5 with n vertices and $m \geq 3.05n$ edges. By (10), we know a lower bound on its number of crossings, namely,

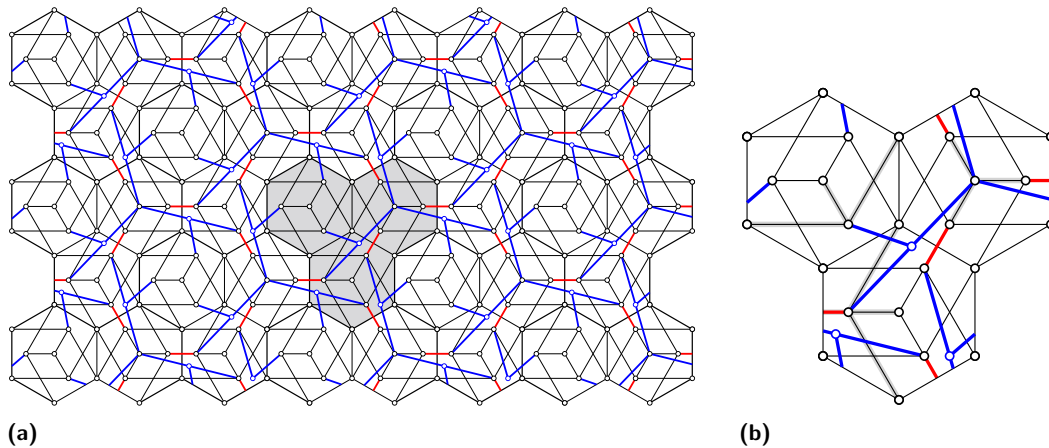
$$\text{cr}(G) \geq \frac{800m^3}{11,163n^2}.$$

On the other hand, since G is k -planar, it holds $\frac{km}{2} \geq \text{cr}(G)$. Combining those, we get:

$$\frac{1,600m^2}{11,163n^2} \leq k \iff m \leq \sqrt{\frac{11,163}{1,600}kn}. \quad \blacktriangleleft$$

► **Theorem 17.** *For every sufficiently large n , there exists a 2-planar graph of girth 5 on n vertices with $(2 + \frac{2}{7})n - O(1) \approx 2.286n - O(1)$ edges.*

Proof sketch. The construction in Figure 7 is an extension of the construction in Figure 4 that achieves the claimed properties. The proof of those properties follows the same ideas as the proof of Theorem 9. For details refer to the full version of this paper [6]. ◀



■ **Figure 7** Illustration for the proof of Theorem 17. The construction consists of repeated triplets of hexagonal tiles (bordered by thick edges, also shown in (b)).

5 3-planar graphs

This section is devoted to 3-planar graphs and is structured analogously to Section 4. For space reasons, the proofs of the upper bounds have been deferred to only the full version of this paper [6], as their proofs are very similar to the corresponding ones of Section 4.

5.1 C_3 -free 3-planar graphs

► **Theorem 18.** *Every C_3 -free 3-planar graph on $n \geq 2$ vertices has at most*

$$\sqrt[3]{\frac{2,673}{20}}n < 5.113n$$

edges.

► **Corollary 19.** *Every C_3 -free k -planar graph on $n \geq 2$ vertices and $m \geq \frac{9}{2}n$ edges has at most*

$$\sqrt{\frac{81}{8}} \cdot \sqrt{k} \cdot n < 3.182\sqrt{kn}$$

edges.

5.2 C_4 -free 3-planar graphs

► **Theorem 20.** *Every C_4 -free 3-planar graph on $n \geq 2$ vertices has at most*

$$\sqrt[3]{\frac{3,764,475}{31,360}}n < 4.933n$$

edges.

► **Theorem 21.** *Every 3-planar graph of girth 5 on n vertices has at most*

$$\sqrt[3]{\frac{368,379}{4,000}}n < 4.516n$$

edges.

► **Theorem 22.** *For every sufficiently large n , there are 3-planar graphs of girth 5 with $2.5n - O(1)$ edges.*

6 Conclusions and open problems

In this work, we continued an active research branch in Graph Drawing seeking for new edge density bounds for k -planar graphs that avoid certain forbidden substructures, namely, cycles of length 3 or 4 or both of them. For each of these settings, our focus was on k -planar graphs, with $k \in \{1, 2, 3\}$, as well as on general k . Several open problems have been triggered:

- The first one is the obvious one, that is, to close the gaps between the lower and the upper bounds reported in Table 1. We believe that this is a challenging open problem.
- In particular, it seems to us that the lower bounds for 2- and 3-planar can be improved.
- Note that there is a lot of empty space to fill in Table 2 where we did not find any reasonably good bounds.
- Another promising research direction is to study the edge density of k -planar graphs that are either C_k free for $k > 4$ or are of girth r with $r > 5$.
- Even though we focused on k -planar graphs, we believe that extending the study to other beyond-planar graph classes is a challenging research direction that worth to follow.
- On the algorithmic side, the recognition problem is of interest; in particular, assuming optimality. A concrete question here, e.g., is whether the problem of recognizing if a graph is optimal C_3 -free 1-planar can be done in polynomial time. Recall that in the general setting this problem can be solved in linear time [9].

References


- 1 Eyal Ackerman. On topological graphs with at most four crossings per edge. *Comput. Geom.*, 85, 2019. doi:10.1016/J.COMGEO.2019.101574.
- 2 Eyal Ackerman and Gábor Tardos. On the maximum number of edges in quasi-planar graphs. *J. Comb. Theory, Ser. A*, 114(3):563–571, 2007. doi:10.1016/J.JCTA.2006.08.002.
- 3 Martin Aigner and Günter M. Ziegler. *Proofs from THE BOOK*. Springer, Berlin, 4th edition, 2010. doi:10.1007/978-3-642-00856-6.
- 4 Miklós Ajtai, Václav Chvátal, Monroe M. Newborn, and Endre Szemerédi. Crossing-free subgraphs. *Ann. Discrete Math.*, 12:9–12, 1982. doi:10.1016/S0304-0208(08)73484-4.
- 5 Patrizio Angelini, Michael A. Bekos, Michael Kaufmann, Maximilian Pfister, and Torsten Ueckerdt. Beyond-planarity: Turán-type results for non-planar bipartite graphs. In *Proc. 29th Annu. Internat. Sympos. Algorithms Comput.*, volume 123 of *LIPICs*, pages 28:1–28:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICs.ISAAC.2018.28.
- 6 Michael A. Bekos, Prosenjit Bose, Aaron Büngener, Vida Dujmović, Michael Hoffmann, Michael Kaufmann, Pat Morin, Saeed Odak, and Alexandra Weinberger. On k -planar graphs without short cycles. *CoRR*, abs/2408.16085, 2024. arXiv:2408.16085.
- 7 Michael A. Bekos, Michael Kaufmann, and Chrysanthi N. Raftopoulou. On optimal 2- and 3-planar graphs. In *Proc. 33rd Internat. Sympos. Comput. Geom.*, volume 77 of *LIPICs*, pages 16:1–16:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPICs.SOCG.2017.16.
- 8 Rainer Bodendiek, Heinz Schumacher, and Klaus Wagner. Bemerkungen zu einem Sechsfarbenproblem von G. Ringel. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 53:41–52, 1983. doi:10.1007/BF02941309.
- 9 Franz J. Brandenburg. Recognizing optimal 1-planar graphs in linear time. *Algorithmica*, 80(1):1–28, 2018. doi:10.1007/S00453-016-0226-8.

- 10 Július Czap, Jakub Przybylo, and Erika Skrabul'áková. On an extremal problem in the class of bipartite 1-planar graphs. *Discuss. Math. Graph Theory*, 36(1):141–151, 2016. doi:10.7151/DMGT.1845.
- 11 Chris Dowden. Extremal C_4 -free/ C_5 -free planar graphs. *J. Graph Theory*, 83(3):213–230, 2016. doi:10.1002/JGT.21991.
- 12 Jacob Fox, János Pach, and Csaba D. Tóth. A bipartite strengthening of the crossing lemma. In *Proc. 15th Int. Sympos. Graph Drawing Network Visualization (GD 2007)*, volume 4875 of *Lecture Notes in Computer Science*, pages 13–24. Springer, 2007. doi:10.1007/978-3-540-77537-9_4.
- 13 Jacob Fox, János Pach, and Csaba D. Tóth. A bipartite strengthening of the crossing lemma. *J. Comb. Theory B*, 100(1):23–35, 2010. doi:10.1016/J.JCTB.2009.03.005.
- 14 Zoltán Füredi. New asymptotics for bipartite Turán numbers. *J. Comb. Theory, Ser. A*, 75(1):141–144, 1996. doi:10.1006/JCTA.1996.0067.
- 15 Seok-Hee Hong and Takeshi Tokuyama, editors. *Beyond Planar Graphs*. Springer, 2020. doi:10.1007/978-981-15-6533-5.
- 16 Michael Kaufmann, Boris Klemz, Kristin Knorr, Meghana M. Reddy, Felix Schröder, and Torsten Ueckerdt. The density formula: One lemma to bound them all. In *Proc. 32nd Int. Sympos. Graph Drawing Network Visualization (GD 2024)*, volume 320 of *LIPICs*, pages 5:1–5:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICs.GD.2024.5.
- 17 János Pach, Rados Radoicic, Gábor Tardos, and Géza Tóth. Improving the crossing lemma by finding more crossings in sparse graphs: [extended abstract]. In *Proc. 20th Annu. Sympos. Comput. Geom.*, pages 68–75. ACM, 2004. doi:10.1145/997817.997831.
- 18 János Pach, Radoš Radoičić, Gábor Tardos, and Géza Tóth. Improving the crossing lemma by finding more crossings in sparse graphs. *Discrete Comput. Geom.*, 36(4):527–552, 2006. doi:10.1007/s00454-006-1264-9.
- 19 János Pach, Joel Spencer, and Géza Tóth. New bounds on crossing numbers. In *Proc. 15th Annu. Sympos. Comput. Geom.*, pages 124–133. ACM, 1999. doi:10.1145/304893.304943.
- 20 János Pach, Joel Spencer, and Géza Tóth. New bounds on crossing numbers. *Discret. Comput. Geom.*, 24(4):623–644, 2000. doi:10.1007/S004540010011.
- 21 János Pach and Géza Tóth. Graphs drawn with few crossings per edge. In Stephen C. North, editor, *Proc. 4th Sympos. Graph Drawing (GD 1996)*, volume 1190 of *Lecture Notes in Computer Science*, pages 345–354. Springer, 1996. doi:10.1007/3-540-62495-3_59.
- 22 János Pach and Géza Tóth. Graphs drawn with few crossings per edge. *Combinatorica*, 17(3):427–439, 1997. doi:10.1007/BF01215922.
- 23 Rephael Wenger. Extremal graphs with no C^4 's, C^6 's, or C^{10} 's. *J. Comb. Theory, Ser. B*, 52(1):113–116, 1991. doi:10.1016/0095-8956(91)90097-4.

On the Edge Density of Bipartite 3-Planar and Bipartite Gap-Planar Graphs

Aaron Büngener ✉

Universität Tübingen, Germany

Maximilian Pfister ✉ 

Universität Tübingen, Germany

Abstract

We show that if a bipartite graph G with $n \geq 3$ vertices can be drawn in the plane such that (i) each edge is involved in at most three crossings per edge or (ii) each crossing is assigned to one of the two involved edges and each edge is assigned at most one crossing, then G has at most $4n - 8$ edges. In both cases, this bound is tight up to an additive constant as witnessed by lower-bound constructions. The former result can be used to improve the leading constant for the crossing lemma for bipartite graphs which in turn improves various results such as the biplanar crossing number or the maximum number of edges a bipartite k -planar graph can have.

2012 ACM Subject Classification Mathematics of computing → Combinatorics; Mathematics of computing → Graph theory

Keywords and phrases Edge Density, Beyond Planarity, bipartite Graphs, Discharging Method

Digital Object Identifier 10.4230/LIPIcs.GD.2024.28

Funding *Maximilian Pfister*: This research was supported by the DFG grant SCHL 2331/1-1.

Acknowledgements We thank the anonymous reviewers for their valuable feedback.

1 Introduction

There is a rich literature about planar graphs which includes both algorithmic and combinatorial results. Since minimizing the number of edge crossings in a drawing improves its readability [18], a drawing should be planar whenever possible. Using Euler's Formula, one can easily derive that a planar graph with n vertices can have at most $3n - 6$ edges, which, however, implies that most graphs are in fact not planar. Empirical studies showed that not only the number of crossings but also their topological/geometrical properties play a crucial role regarding the readability of a drawing [9, 12]. This gave rise to the research area of *beyond planarity*, where graph classes are defined in terms of forbidden crossing configurations – see [8] for a survey of the area. While there is a plethora of beyond-planar classes, we are here concerned with k -planar and k -gap-planar graphs. A graph G is k -planar if there exists a drawing of G in the plane such that each of its edges has at most k crossings. A graph G is k -gap-planar, if G can be drawn in the plane such that there exists an assignment of every crossing to one of the involved edges such that each edge is assigned at most k crossings.

While the research of k -planar graphs (in particular, 1-planar graphs) started already more than half a century ago [15], k -gap-planar graphs were introduced quite recently [5] and can, in some sense, be interpreted as an asymmetric version of k -planar graphs. The authors of [5] showed that every $2k$ -planar graph is k -gap-planar, but, for any fixed choice of k , there exists a 1-gap-planar graph which is not k -planar.

One of the most studied questions regarding a beyond-planar graph class is to determine its *edge density*, i.e., the maximum number of edges an n -vertex graph that belongs to this class can have, see e.g. [2, 7, 11] for some work in this direction. There is an additional motivation to study the edge density of k -planar graphs in particular: Improved results



© Aaron Büngener and Maximilian Pfister;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 28; pp. 28:1–28:21

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

■ **Table 1** Overview of related edge density bounds and summary of our results (red angle brackets).

Graph class	Bipartite		Lower bound
	Upper bound		
	Previous	Ours	
1-planar:	$3n - 8$ [10]	–	$3n - 8$ [10]
2-planar:	$3.5n - 7$ [3]	–	$3.5n - 12$ [3]
3-planar:	$5.205n$ [3]	$4n - 8$ (6)	$4n - 12$ (7)
k -planar:	$3.005\sqrt{kn}$ [3]	$2.871\sqrt{kn}$ (17)	–
1-gap-planar:	$4.25n$ [16]	$4n - 8$ (5)	$4n - 16$ [16]
k -gap-planar:	$4.25\sqrt{kn}$ [16]	$4.06n$ (18)	–

in turn improve the leading constant of the celebrated *Crossing Lemma* which has various applications, see [1] – currently there are tight (up to a constant number of edges) bounds for $k \in \{1, 2, 3, 4\}$ [15, 14, 13, 1]. The edge density was also studied when one imposes additional restrictions on (i) the drawings (e.g., the drawing is *outer* [6]) or on (ii) the graphs, where the most common restriction is to consider bipartite graphs, i.e., graphs which do not contain any odd cycle. In Table 1, we highlight the important past results regarding the edge density of bipartite graphs for our work.

Our contribution

In Section 3 and Section 4, we show that both bipartite 1-gap-planar graphs as well as bipartite 3-planar graphs on n vertices have at most $4n - 8$ edges, thus (partially) answering an open problem posed in [3] and proving a conjecture posed in [16]. In Section 5, we use the result of Section 4 to improve the constant of the bipartite crossing lemma from $\frac{1}{18.1}$ to $\frac{1}{16.5}$ which in turn improves the upper bound on the edge density of bipartite k -planar graphs from $\approx 3.005\sqrt{kn}$ to $\approx 2.871\sqrt{kn}$.

2 Preliminaries

Throughout the paper, we will assume that all drawings are *simple* in the sense that no two adjacent edges cross and no edge crosses itself¹. We consider bipartite graphs $G = (A, B, E)$ with $A \cap B = \emptyset$ and let $n = |A \cup B|$ and $m = |E|$. We will usually denote a vertex of A (of B) by a_i (by b_i). We do not require that G is simple, i.e., several edges between the same two endpoints are allowed if they are non-homotopic in the corresponding drawing Γ , i.e., for any such pair of edges, both the interior and the exterior of the closed region (defined by the pair) contains at least one vertex of G .

Let \mathcal{B} be a beyond-planar graph class. Denote by \mathcal{G} the set of all tuples (G, Γ) where $G \in \mathcal{B}$ is a bipartite graph of n vertices and Γ is a drawing of G (satisfying the constraints of \mathcal{B}) where any two copies of an edge are non-homotopic. Let $\mathcal{G}' \subset \mathcal{G}$ consist of all elements (G, Γ) such that G has the maximum number of edges among all graphs contained in \mathcal{G} . Finally, let $\mathcal{G}'' \subset \mathcal{G}'$ consist of all elements (G, Γ) such that Γ has the minimum number of crossings among all drawings contained in \mathcal{G}' . In the remainder, we will refer to such a tuple $(G, \Gamma) \in \mathcal{G}''$ as a *MaxMin* tuple (since G has the maximum number of edges and since Γ has the minimum number of crossings).

¹ we explicitly allow that two edges cross more than once

3 Bipartite 1-gap-planar graphs

Since we only consider 1-gap-planar graphs, we will abbreviate it henceforth by *gap-planar*. The *crossing graph* of a drawing Γ of graph G has a vertex v_e for any edge e of G and an edge $(v_e, v_{e'})$ if and only if the edges e and e' intersect in Γ . The following lemma is directly derived from the definition of gap-planarity.

► **Lemma 1** ([5]). *Let G be a gap-planar graph. Then G admits a gap-planar drawing Γ such that the crossing graph of Γ is a pseudoforest.*

For the remainder of this section, we will fix (G, Γ) as a MaxMin tuple (regarding gap-planarity). In order to show that G has at most $4n - 8$ edges, we want to find a set of edges of G , denoted by \mathcal{X} , such that no two edges of \mathcal{X} cross in Γ and such that $|\mathcal{X}| \geq \frac{m}{2}$. The result then follows immediately since \mathcal{X} induces a bipartite planar subgraph of G , hence $|\mathcal{X}| \leq 2n - 4$ and thus $m \leq 4n - 8$ follows. To define \mathcal{X} , we will consider the components of the crossing graph \mathcal{I} of Γ . Recall that by definition, no two edges that belong to different components of \mathcal{I} intersect, thus we can consider the components separately.

► **Lemma 2.** *Let X be an arbitrary component of \mathcal{I} . If X (i) is a tree, (ii) contains an even-length cycle or (iii) contains an odd-length cycle C and at least one rooted tree at a vertex of C (that is edge-disjoint from C) is a path of odd length, then X contains an independent set of size at least $\frac{|X|}{2}$.*

Proof. If X is a tree or contains an even-length cycle, then by definition and Lemma 1 X is bipartite and thus its vertices can be colored using two colors. Each color induces an independent set, from which one has size at least $\frac{|X|}{2}$. For the third case, let $C = (v_1, v_2, \dots, v_k)$ be the unique odd cycle of X and w.l.o.g. assume that the tree rooted at v_1 is in fact a path $(u_1, u_2, \dots, u_{2j+1})$ such that $(v_1, u_1) \in \mathcal{I}$. Coloring u_i with i odd implies that we have colored $j + 1$ vertices in one color, which is enough to accommodate for all vertices of the path in addition to v_1 . Clearly, the only vertex that is not on the path which is influenced by the coloring is v_1 . Removing v_1 (and the whole path) from X yields a tree which has an independent set of the desired size as shown in the previous case. Combining both independent sets then concludes the proof. ◀

Components of \mathcal{I} which do not meet the criteria of Lemma 2 are called *critical*. By Lemma 2, any such component X is a pseudotree whose unique odd cycle C is of odd length and none of the trees rooted at the vertices of C are odd-length paths. For these critical components, we cannot directly find an independent set in \mathcal{I} of appropriate size. To be more precise, we can only find an independent set of size $\left\lceil \frac{|X|-1}{2} \right\rceil$. To overcome this issue, we will show that for any such component X , there exists an uncrossed edge in Γ (i.e., a singleton in \mathcal{I}) which we can uniquely assign to X . The next lemma follows by our choice of (G, Γ) .

► **Lemma 3.** *Let $e_1 = (a_1, b_1)$ and $e_2 = (a_2, b_2)$ be two edges of G that intersect in Γ . Then, (a_1, b_2) or (a_2, b_1) drawn along the curves of e_1 and e_2 is present in Γ .*

Proof. Suppose for a contradiction that neither (a_1, b_2) nor (a_2, b_1) drawn along e_1 and e_2 exist in Γ . Denote by x the intersection point between e_1 and e_2 . W.l.o.g. assume that the crossing between e_1 and e_2 was assigned to e_1 in Γ . This implies that no crossing on the (open) segments a_1x and xb_1 can be assigned to e_1 . Moreover, at most one crossing is assigned to e_2 by definition – w.l.o.g. assume this crossing is due to an edge that intersects e_2 on the segment a_2x . Now, consider the graph $G' = G \setminus \{(a_1, b_1), (a_2, b_2)\} \cup \{(a_1, b_2), (a_2, b_1)\}$

with corresponding drawing Γ' where the drawing of all edges but (a_1, b_2) and (a_2, b_1) is inherited from Γ , while the edges (a_1, b_2) and (a_2, b_1) are drawn along (the original curves of) e_1 and e_2 , refer to Fig. 1a. First observe that G' is a non-homotopic multigraph, as neither (a_1, b_2) nor (a_2, b_1) drawn along e_1 and e_2 were present by assumption. Further, Γ' is a valid gap-planar drawing as we do not need to assign (a_1, b_2) any crossing, while (a_2, b_1) is assigned at most one crossing. But now we have a contradiction to our choice of G and Γ , as G' contains the same number of edges as G , but Γ' contains less crossings than Γ . \blacktriangleleft

► **Lemma 4.** *Let S be the set of singletons in \mathcal{I} and let Z be the set of critical components of \mathcal{I} . Then $|S| \geq |Z|$.*

Proof. As we will argue about graph G and its crossing graph \mathcal{I} simultaneously, we will assume in the following that an edge $e_i = (a_i, b_i)$ of G corresponds to a vertex v_i of \mathcal{I} . Let $X \in Z$ be a critical component in \mathcal{I} and let $C = (v_1, v_2, \dots, v_{2j+1})$ be its unique odd cycle in \mathcal{I} . Pick two adjacent vertices v_1 and v_2 of C with corresponding edges $e_1 = (a_1, b_1)$ and $e_2 = (a_2, b_2)$ of G . Lemma 3 ensures that at least one of (a_1, b_2) or (a_2, b_1) exists in G such that its curve follows e_1 and e_2 in Γ . W.l.o.g. assume that (a_1, b_2) exists and denote by v the corresponding vertex of (a_1, b_2) in \mathcal{I} . We distinguish between the following two cases based on whether v is adjacent to a vertex of C or not.

Assume first that v is adjacent to some vertex of C , i.e., (a_1, b_2) intersects an edge of G associated with a vertex of C in Γ . As (a_1, b_2) is drawn along e_1 and e_2 , this edge is either e_{2j+1} or e_3 by construction. We first observe that (a_1, b_2) cannot intersect both e_3 and e_{2j+1} , as otherwise X is not a pseudoforest (this also holds in the case where $e_3 = e_{2j+1}$, i.e., C is a 3-cycle, in which case (a_1, b_2) intersects this edge at most once). W.l.o.g. assume that (a_1, b_2) crosses e_3 , the other case is symmetric. If there is an additional edge e' besides e_3 that is crossing (a_1, b_2) , then e' also crosses either e_1 or e_2 as (a_1, b_2) is drawn along e_1 and e_2 , but then X is not a pseudoforest as this crossing would close another cycle. In particular, if it crosses e_1 , then we obtain the cycle (v_1, v_2, v_3, v, v') , and otherwise we obtain (v_2, v_3, v, v') , where v' is the corresponding vertex of e' in \mathcal{I} . Hence, (a_1, b_2) only crosses e_3 – but then we have an odd-length path rooted at v_3 in \mathcal{I} (that only contains vertex v), in which case X is not critical, a contradiction.

Thus we can assume from now on that v is not adjacent to any vertex of C . This means that either (a_1, b_2) is crossing free in Γ , or there is an edge e' that intersects (a_1, b_2) and thus either e_1 or e_2 . We keep the former case in mind and consider the latter case. W.l.o.g. assume that e' intersects e_1 , the other case is symmetric. Denote by v' the corresponding vertex of e' in \mathcal{I} . Now, in \mathcal{I} , we have a tree $T \subset \mathcal{I}[X]$ rooted at v_1 such that $(v_1, v') \in T$ and $(v', v) \in T$. Let t be the depth of T , let u_k be a leaf of depth t and let u_{k-1} be the unique parent of u_k . Denote by u_k^1, \dots, u_k^r the children of u_k with $u_k = u_k^1$ and let (α, β) be the corresponding edge to u_{k-1} in G . By traversing (α, β) starting from α , the first intersection we encounter is either with an edge that corresponds to a leaf u_k^i or with the edge that corresponds to the unique parent of u_{k-1} . If the latter case occurs, observe that by traversing (α, β) starting at β we encounter a leaf first. Denote by (α', β') the corresponding edge to u_k^i in G . Hence, w.l.o.g. assume that the crossing x with (α', β') is the first one that we encounter when traversing (α, β) starting at α . This implies that the segment αx is crossing free. Moreover, since u_k^i is a leaf, both segments $\alpha'x$ and $x\beta'$ are crossing free – but then the edge (α, β') exists (crossing free) in Γ by maximality.

In both cases, we found an uncrossed edge for a fixed pair of consecutive vertices. By repeating this argumentation for any two consecutive vertices of C , we can find a set of edges P with $|P| = |C| \geq 3$ such that any edge in P is uncrossed in Γ , i.e., belongs to S .

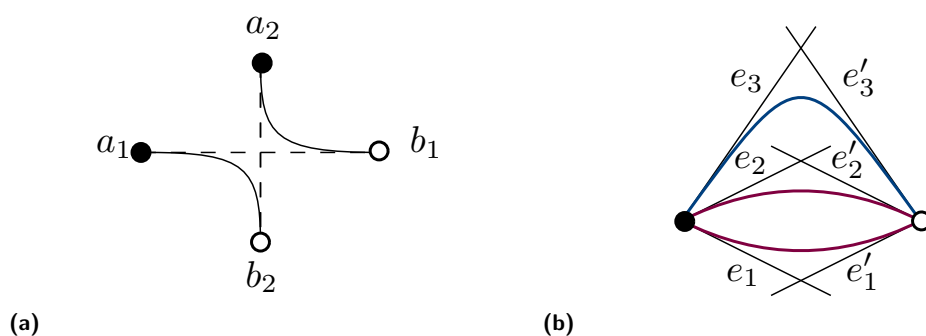


Figure 1 (a) Illustration for Lemma 3. (b) Subgraph formed by edges e_i, e'_i for $i \leq 3$. The blue edge is a non-homotopic copy of the red ones.

It remains to consider the interaction of different components of \mathcal{I} . Given two critical components X and Y of \mathcal{I} , it is possible that an edge occurs in both P_X and P_Y . We claim the following: any non-homotopic copy of such an edge occurs in at most two such sets. Assuming we have this claim at hand, the total number of uncrossed edges in Γ is at least

$$\frac{1}{2} \sum_{i=1}^{|Z|} X_i \geq \frac{1}{2} \sum_{i=1}^{|Z|} 3 = \frac{3|Z|}{2} \geq |Z|$$

since any odd cycle of \mathcal{I} has size at least three. It remains to prove the claim. Suppose for a contradiction that one copy of an edge (a, b) belongs to at least three (critical) components X_1, X_2 and X_3 . Denote by e_i and e'_i the two edges of X_i where (a, b) is drawn along. By definition, no edge of X_i crosses an edge of X_j for $i \neq j$. But then e_1 and e'_1, e_2 and e'_2 and e_3 and e'_3 need to bound the same cell of Γ which is impossible, as e_1, e_2 and e_3 are all incident to a , while e'_1, e'_2 and e'_3 are all incident to b , see Fig.1b. ◀

By Lemma 4, there exists a bijective mapping from S to Z . We will call a critical component with an additional (singleton) edge an *augmented* component. Observe that every augmented component X has an independent set of size $\lceil \frac{|X|-1}{2} \rceil + 1 \geq \frac{|X|}{2}$.

► **Theorem 5.** *An n -vertex bipartite gap-planar multigraph without homotopic parallel edges has at most $4n - 8$ edges.*

Proof. Let \mathcal{X} be the union of the maximum independent sets of every (augmented) component X of \mathcal{I} . Since we established that every component of \mathcal{I} , in particular also the augmented critical components, has an independent set of size at least half its size, it follows that $|\mathcal{X}| \geq \frac{m}{2}$. Clearly, no two edges of \mathcal{X} intersect in \mathcal{I} , hence the edges of \mathcal{X} induce a planar bipartite multi-graph $G_{\mathcal{X}}$. Since $G_{\mathcal{X}}$ does not contain any homotopic multiedges by construction, it still holds that any face of $G_{\mathcal{X}}$ has length at least four. Since Euler's formula can also be applied to non-simple graphs, we have that $G_{\mathcal{X}}$ has at most $2n - 4$ edges, and thus $m \leq 4n - 8$ which concludes the proof. ◀

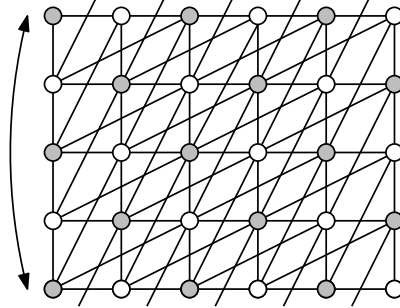
The lower-bound construction in [16], which yields n -vertex bipartite gap-planar graphs with $4n - 16$ edges, asserts that our bound is tight up to an additive constant.

4 Bipartite 3-planar graphs

In this section, we will establish an upper bound on the number of edges a bipartite 3-planar graph can have.

► **Theorem 6.** *A bipartite 3-planar graph G with n vertices has at most $4n - 8$ edges.*

Before we prove Theorem 6, we will provide a lower-bound construction to show its tightness (which has been suggested in [3]).



■ **Figure 2** Lower bound construction for bipartite 3-planar graphs. The vertices of the first and last row coincide.

► **Theorem 7.** *For infinitely many n , there exists a bipartite 3-planar graph G with n vertices and $4n - 12$ edges.*

Proof. Assume n is divisible by four. We arrange the vertices equally in four rows (i.e., every row contains $\frac{n}{4}$ vertices) and wrap the construction around a cylinder; i.e., the topmost and the bottommost row in Figure 2 coincide. Clearly, the drawing is 3-planar. In order to determine the number of edges, let us count the degrees of the vertices. The vertices in the first and last column have degree five, while all other vertices have degree eight. Hence, $2m = 8(n - 8) + 8 \cdot 5$ and thus $m = 4n - 12$. ◀

Similar to the previous section, we will fix (G, Γ) as a MaxMin tuple (w.r.t. 3-planarity) for the remainder of the section. In order to prove Theorem 6, we will use the *Discharging Method*, which was initially introduced in order to prove structural properties of planar graphs, e.g., for the proof of the Four Color Theorem [4]. Our proof will reuse parts of the notation and ideas from [1] where the author proved an upper bound on the number of edges of 4-planar graphs. We denote by $P(\Gamma)$ the so called *planarization* of Γ , i.e., the vertices and crossing points of Γ are the vertices of $P(\Gamma)$, while the edges of $P(\Gamma)$ are the crossing-free segments in Γ which are bounded by vertices and crossing points. We will refer to the vertices of $P(\Gamma) \cap G$ as *original*. We will denote by $e = (a, b)$ an edge of G while the segment of e restricted to a face f of $P(\Gamma)$ is denoted by $f[e]$, or, if the two endpoints x and y of the segment are known, we might also refer to the segment as xy . We will also use (B) and $(3P)$ to abbreviate the bipartite and 3-planar property, respectively.

We will prove Theorem 6 by induction on the number of vertices of G . Clearly, if $n \leq 6$, we have $4n - 8 > \binom{n}{2}$ and the theorem holds. Thus, we assume that $n \geq 7$. Moreover, we can assume that every vertex in G has degree at least 5, as otherwise the theorem follows by removing a vertex of small degree and applying the induction hypothesis. We begin with the following important observation for $P(\Gamma)$; its proof is analogous to the one in [1].

► **Property 8.** *If $P(\Gamma)$ is not 2-connected, then G has at most $4n - 8$ edges.*

Proof. Assume that $P(\Gamma)$ has a vertex x such that $P(\Gamma) \setminus \{x\}$ is not connected. The vertex x is either a vertex of G or a crossing point of two of its edges. Suppose first that x is vertex of G . Then, $G \setminus \{x\}$ is also not connected. Let G_1, \dots, G_k be the connected components

of $G \setminus \{x\}$, let G' be the graph induced by $V(G_1) \cup \{x\}$ and let G'' be the graph induced by $V(G_2) \cup \dots \cup V(G_k) \cup \{x\}$. Note that $4 \leq |V(G')|, |V(G'')| < n$, since we established earlier that the minimum degree of a vertex of G is at least 5. Therefore, it follows from the induction hypothesis that $|E(G)| \leq 4|V(G')| - 8 + 4|V(G'')| - 8 = 4(n+1) - 16 < 4n - 8$.

Suppose now that x is a crossing point of two edges e_1 and e_2 . Let \hat{G} be the graph we obtain by adding x as a vertex to G . Therefore, $|V(\hat{G})| = n + 1$ and $|E(\hat{G})| = |E(G)| + 2$. Let G_1, \dots, G_k be the connected components of $\hat{G} \setminus \{x\}$, let G' be the graph induced by $V(G_1) \cup \{x\}$ and let G'' be the graph induced by $V(G_2) \cup \dots \cup V(G_k) \cup \{x\}$. Again, note that $4 \leq |V(G')|, |V(G'')| < n$ by our observation about the minimum degree. It follows from the induction hypothesis that $|E(G)| \leq 4|V(G')| - 8 + 4|V(G'')| - 8 - 2 = 4(n+2) - 18 < 4n - 8$. ◀

Property 8 allows us to assume that $P(\Gamma)$ is 2-connected. The *boundary* δf of a face f in $P(\Gamma)$ consists of all the edges of $P(\Gamma)$ that are incident to f . Since $P(\Gamma)$ is 2-connected, the boundary of every face in $P(\Gamma)$ is a simple cycle. Thus, we can define the *size* of a face f , $|f|$, as the number of edges of $P(\Gamma)$ on its boundary. We will denote by $V(f)$ the set of original vertices on the boundary of f .

► **Observation 9.** *The boundary of every face in $P(\Gamma)$ is a simple cycle.*

Similar to [1], we begin by assigning a *charge* to every face of $P(\Gamma)$ such that the total charge is $4n - 8$. Then, we redistribute the charge in several steps such that the charge of every face is non-negative and the charge of every original vertex v is at least $\deg(v)/2$. Hence, $|E(G)| = \sum_{v \in V(G)} \deg(v)/2 \leq 4n - 8$ and we get the desired bound on $|E(G)|$.

Let V' , E' , and F' denote the vertex, edge, and face sets of $P(\Gamma)$, respectively. Clearly, $\sum_{f \in F'} |V(f)| = \sum_{v \in V(G)} \deg(v)$ and $\sum_{f \in F'} |f| = 2|E'| = \sum_{u \in V'} \deg(u)$ holds. Every vertex in $V' \setminus V(G)$ is a crossing point in G and therefore its degree in $P(\Gamma)$ is four. Hence,

$$\sum_{f \in F'} |V(f)| = \sum_{v \in V(G)} \deg(v) = \sum_{u \in V'} \deg(u) - \sum_{u \in V' \setminus V(G)} \deg(u) = 2|E'| - 4(|V'| - n).$$

Assigning every face $f \in F'$ a charge of $|f| + |V(f)| - 4$, we get a total charge of

$$\sum_{f \in F'} (|f| + |V(f)| - 4) = 2|E'| + 2|E'| - 4(|V'| - n) - 4|F'| = 4n - 8,$$

Recall that we will redistribute the initial charge s.t. the charge of every face of F' is non-negative, while every original vertex v has charge at least $\deg(v)/2$. An equivalent precondition is that

$$ch(f) \geq 0.5 \cdot |V(f)| \text{ for all } f \in F' \tag{1}$$

as we can then redistribute the excess charge from the faces to the original vertices in a final step. Let f be face of F' with $|V(f)| = x$ and $|f| = y$. We will then refer to f as an x - y face. To ease the notation, we will use the terms *triangles*, *quadrangles*, *pentagons*, *hexagons* and *heptagons* to refer to faces of size 3,4,5,6 and 7, respectively. For example, a 2-triangle is a face of size 3 whose boundary contains two original vertices. Since Γ is a non-homotopic drawing with the minimum number of crossings, there are no faces of size 2 in F' . Therefore, initially, the only faces which do not satisfy Equation (1) are 0-triangles and 1-triangles. In order to ensure that a face f still satisfies Equation (1) after it redistributed some of its charge to another face, we will introduce the notion of *local charge* for faces that contain sufficiently many original vertices. Let $f \in F'$ be a face of $P(\Gamma)$ with $|V(f)| \geq 2$. Let

v_1, v_2, \dots, v_k be the (ordered) sequence of vertices of $P(\Gamma)$ that are contained in δf . Let v_i and v_j be two consecutive original vertices of δf with $i < j$, i.e., all vertices v_k with $i < k < j$ are crossing points in Γ .

If $|j-i| > 1 \pmod{k-1}$ holds, then we split f along the hypothetical edge (v_i, v_j) and obtain a so called *subdivision face* $f_a = (v_i, v_{i+1}, \dots, v_j)$. Observe that $|f_a| \geq 3$ and $|V(f_a)| = 2$.

For example, let f be a 2-pentagon with vertices v_1, v_2, v_3, v_4, v_5 such that v_1 and v_3 are original vertices. By splitting along the pair $\{v_1, v_3\}$ we obtain the subdivision face $f_a = (v_1, v_2, v_3)$ and, as a remainder, $f_b = (v_3, v_4, v_5, v_1)$. Observe that in this particular case, f_b is also a subdivision face as $|f_b| \geq 3$ and $|V(f_b)| = 2$, but this is not the case in general. Let us now consider the charge distribution of such a subdivision. Since a subdivision always occurs at an edge between two original vertices, we have $|V(f_a)| + |V(f_b)| = |V(f)| + 2$ and $|f_a| + |f_b| = |f| + 2$. Hence, $ch(f) = ch(f_a) + ch(f_b)$ after the initial assignment holds. Since both f_a and f_b contain the two original vertices on their boundary which defined the subdivision edge and since the subdivision edges do not contribute to the degree of these two vertices, f_a and f_b have to retain less charge. In particular, we require that every subdivision face f_a satisfies

$$ch(f_a) \geq 0.5 \tag{2}$$

while the remainder f_b has to satisfy

$$ch(f_b) \geq 0.5(|V(f_b)| - s(f_b)) \tag{3}$$

where $s(f_b)$ is the number of subdivision edges on the boundary of f_b . Observe that Equation (2) is a special case of Equation (3) when $|V(f)| = 2$ and $s(f) = 1$. If all subdivision faces f_x of a face f (and its possible remainder) satisfy Equation (2) and Equation (3), it is immediate that f satisfies Equation (1), which allows us to argue about the charge in a more local way. In order to describe the way the charging of $\{0, 1\}$ -triangles works we will need the following definitions. Let f be a face, let e be one of its edges, and let f' be the other face that shares e with f . We say that f' is the *immediate neighbor* of f at e . Note that $f' \neq f$ since $P(\Gamma)$ is 2-connected. The following two definitions are taken from [1].

Wedge-neighbors. Let f_0 be a $\{0, 1\}$ -triangle in $P(\Gamma)$ and let x_1 and y_1 be two vertices of f_0 that are crossing points in Γ . Denote by e_x (resp., e_y) the edge of G that contains x_1 (resp., y_1) and does not contain y_1 (resp., x_1). Note that e_x and e_y intersect at the other vertex of f_0 . Let f_1 be the immediate neighbor of f_0 at x_1y_1 . For $i \geq 1$, if f_i is a 0-quadrangle, then denote by $x_{i+1}y_{i+1}$ the edge of $P(\Gamma)$ opposite to x_iy_i in f_i , such that e_x contains x_{i+1} and e_y contains y_{i+1} , and let f_{i+1} be the immediate neighbor of f_i at $x_{i+1}y_{i+1}$. Observe that $f_i \neq f_j$ for $i < j$, for otherwise x_j coincides with one of x_i and x_{i+1} (which implies that e_x crosses itself) or with one of y_i and y_{i+1} (which implies that e_x and e_y intersect more than once). Let j be the maximum index for which f_j is defined. We then call f_j the *wedge-neighbor* of f_0 at x_1y_1 (note that f_j is uniquely defined). We also say that f_0 is the wedge-neighbor of f_j at x_jy_j . Observe that since the relations being an immediate neighbor at a certain edge of $P(\Gamma)$ and being an opposite edge in a 0-quadrangle are both one-to-one, it follows that indeed there cannot be another triangle but f_0 that is a wedge-neighbor of f_j at x_jy_j . Note also that since e_x and e_y already intersect at a vertex of f_0 , and by definition f_j cannot be a 0-quadrangle, either $|f_j| \geq 5$ or $|f_j| = 4$ and $|V(f_j)| \geq 1$.

► **Observation 10** ([1]). *Let f be a face and let e be one of its edges. Then there is at most one triangle t such that t is a wedge-neighbor of f at e . If such a triangle exists, then either $|f| \geq 5$ or $|f| = 4$ and $|V(f)| \geq 1$.*

Vertex-neighbors. Let x be a crossing point in Γ and let f_0, f_1, f_2 and f_3 be the four faces that are incident to x in clockwise order around x . Note that these faces are distinct, since $P(\Gamma)$ is 2-connected. We say that f_0 and f_2 (resp., f_1 and f_3) are vertex-neighbors at x .

We will introduce one additional kind of neighbor relation.

Rich immediate neighbor. Let f be a face with edges e_0, e_1, \dots, e_k . We call an immediate neighbor f' of f at edge e_i *rich* if, in the facial walk of f' , we have e_{j-1}, e_j, e_{j+1} such that $e_i = e_j$, the common endpoint of e_{j-1} and e_j (e_{j+1} and e_j) is a crossing point in Γ , while the other endpoint of e_{j-1} (e_{j+1}) is an original vertex.

4.1 Step 1: Charging the 0-triangles

Consider a 0-triangle f and its immediate neighbors f_0, f_1 and f_2 . If one such f_i is a rich immediate neighbor, it charges one unit to f . Otherwise, f obtains $\frac{1}{3}$ units of charge from each of its three wedge-neighbors. Hence, in every case, $ch(f) = 0$ for all 0-triangles f .

► **Property 11.** *Let f be a 0-triangle. If one immediate neighbor f_i of f is a 0- x face, then f has a rich immediate neighbor f_j with $f_j \neq f_i$.*

Proof. Assume that the edges that define f are denoted by e_0, e_1 and e_2 such that face f_i shares the edge e_i with f . W.l.o.g. assume that f_0 is a 0- x face. Then, by definition, e_1 and e_2 have a crossing in $\delta f_0 \setminus \delta f$ and are thus crossed three times each. Since e_0 is crossed by e_1 and e_2 already, it can be crossed at most once more, w.l.o.g. assume that e_0 has a crossing in $\delta f_1 \setminus \delta f$. But then $f_2[e_0]$ and $f_2[e_1]$ each contain an original vertex by (3P) and hence f_2 is a rich immediate neighbor. ◀

► **Observation 12.** *No 0- x face has to pay in the initial charging step.*

Obviously, a sufficiently large face can be a wedge-neighbor and an immediate neighbor to several other faces. Fix an edge e . Since both the immediate neighbor relation and the wedge-neighbor relation is unique, e can be assigned to at most one wedge-neighbor (to at most one immediate neighbor). Further, if e is used for the rich immediate neighbor relation, it substitutes the wedge-neighbor relation. Hence, a face f is either a wedge-neighbor or a rich immediate neighbor over each of its edges. Since every rich immediate neighbor relation introduces three new edges and an original vertex (every wedge-neighbor relation introduces one new edge), our face gets an additional charge of four (one) units which clearly accommodates for the discharge if our face is sufficiently large. For example, a 2-pentagon that is wedge-neighbor to one rich immediate neighbor has, after discharging, still two units of charge left, which clearly satisfies Equation (1). We conclude with the following observation:

► **Observation 13.** *After the initial charging step, only the 1-triangles and (possibly) the 1-quadrangles do not satisfy Equation (1).*

Observe that the charge of every 1-triangle is 0, while the charge of a 1-quadrangle can be as low as $\frac{1}{3}$ (this occurs when a 1-quadrangle is a wedge-neighbor to two 0-triangles).

4.2 Step 2: Charging the 1-triangles

Every 1-triangle obtains 0.5 units of charge from its unique wedge-neighbor. Since a 1-triangle is not a wedge-neighbor to a 0-triangle by Observation 10, it did not loose charge in the previous step, hence we have $ch(f) = 0.5$ for any 1-triangle $f \in F'$. Observe that after these two rounds, every triangle satisfies Equation (1) (while this is explicit for $\{0, 1\}$ -triangles,

Observation 10 guarantees that no triangle is a wedge-neighbor and, by definition, they are also not a rich immediate neighbor, hence the charge of 2-triangles still satisfies Equation (1)). Moreover, by definition, 0-quadrangles are not a wedge-neighbor nor a rich neighbor. Since any x - y face can be wedge-neighbor to at most $y - x$ triangles, they loose at most $\frac{1}{2}(y - x)$ charge in the second step and thus always satisfy Equation (1) if $y \geq 8$. Recall that wedge-neighbor relations and rich immediate neighbor relations cannot interfere and rich immediate neighbors give a vast surplus of charge to our face. Thus, the only faces which potentially do not satisfy Equation (1) are 0-heptagons, 0-hexagons, 0-pentagons and 1-quadrangles. We will now establish that the first two in fact satisfy Equation (1):

1. Face f is a 0-heptagon: By Observation 12, f did not discharge to a 0-triangle. Since its initial charge of three is sufficient if f is a wedge-neighbor to at most six triangles, f needs to be a wedge-neighbor to seven 1-triangles. But this is impossible as the corresponding edges would form an odd cycle in G , a contradiction to (B).
2. Face f is a 0-hexagon: Again, we have by Observation 12 that f did not discharge to a 0-triangle. Its initial charge is sufficient if f is wedge-neighbor to at most four 1-triangles. If f is a wedge-neighbor to five or six 1-triangles, we again have an induced cycle of odd length in G , a contradiction to (B).

4.3 Step 3: Recharging 0-pentagons and 1-quadrangles

In order for a 0-pentagon (1-quadrangle) to not satisfy Equation (1), it requires a quite limited local structure which we will exploit to locally redistribute charges.

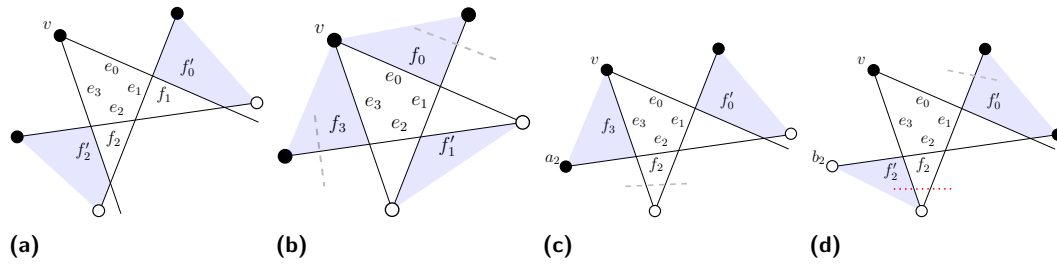
Throughout the analysis, we will denote by $e_0, e_1 \dots e_{k-1}$ the original edges (i.e., the edges in G) whose pairwise crossing points (and the corresponding segments) define face f . Let $e_i = (a_i, b_i)$ where $a_i \in A$ and $b_i \in B$ are original vertices. We will refer to the immediate neighbor to f at the edge e_i as f_i . We will denote by x_i the common endpoint of $f[e_i]$ and $f[e_{i+1}]$ which is part of the boundary δf of f . We will further denote by f'_i the vertex neighbor of f at x_i . Finally, we will denote by t_i the 1-triangle which is the wedge-neighbor to f at edge e_i (if it exists). Note that if t_i exists, then $t_i = f_i$ unless f_i is a 0-quadrangle. We denote the unique real vertex of t_i as v_i . Throughout the proof, our charging scheme will maintain the following invariant.

► **Invariant 14.** *Let x be an intersection point that belongs to the boundary δf of a face f and let e_1 and e_2 be the edges which define x . If neither $f[e_1]$ nor $f[e_2]$ contains an original vertex, then f does not discharge over x .*

4.3.1 f is a 1-quadrangle

With a slight abuse of notation, let $v = x_3$ be the real vertex of f . W.l.o.g. assume that $v \in A$.

1. f is wedge-neighbor to two 0-triangles. It follows that e_1 and e_2 are crossed three times each. Consider f'_0 . By assumption, $\delta f'_0$ contains $f'_0[e_0], f'_0[e_1]$ and $f'_0[e_2]$ such that the endpoints of $f'_0[e_1] \setminus f'_0[e_0]$ and $f'_0[e_2] \setminus f'_0[e_0]$ are original vertices by (3P), see Fig. 3a. Hence, f'_0 is a rich immediate neighbor to f_1 . A similar observation holds for f'_2 and f_2 . Thus, f did not loose any charge in the initial step and thus satisfies Equation (1), see Fig. 3a.
2. f is wedge-neighbor to two 1-triangles. Consider the edges e_1 and e_2 . They both have exactly two crossings which belong to δf .
 - a. $f_0[e_1]$ contains an original vertex. In particular, since f_2 is a 1-triangle, we have by (B) that the endpoint of $f_0[e_1]$ is of the same partition as v and thus it is a_1 . This implies that the segment of e_1 delimited by a_1 and x_0 is not crossed (in which case



■ **Figure 3** Illustrations used in the recharging of 1-quadrangles.

the gray-dashed edge in Fig. 3b that intersects e_0 is not present) and the sequence (a_1, x_0, v) of f_0 defines a subdivision face, which contains 0.5 excess charge which will be redistributed to f (over its vertex neighbor f_1 at x_0).

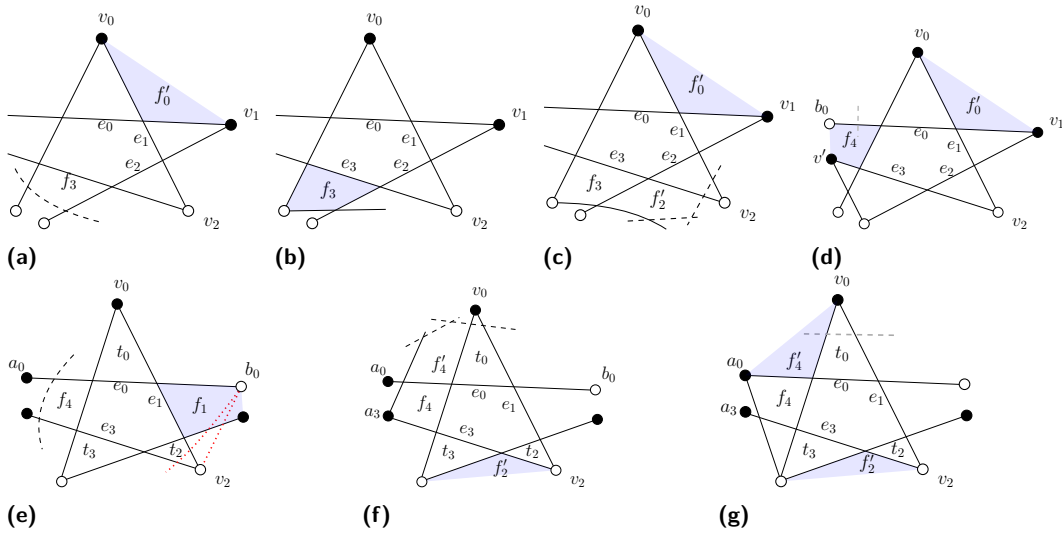
- b. $f_3[e_2]$ contains an original vertex. Following an analogous argument as before, this original vertex is a_2 and thus the sequence (v, x_2, a_2) forms a subdivision face and distributes its excess charge to f via f_2 .
 - c. Neither $f_0[e_1]$ nor $f_3[e_2]$ contains an original vertex. This setting can be observed in Fig. 3b if we assume that both gray-dashed edges are present. Consider the vertex neighbor f'_1 to f at x_1 . Since e_1 and e_2 are both crossed thrice, it follows that both $f'_1[e_1]$ and $f'_1[e_2]$ contain an original vertex. Thus (b_1, x_1, b_2) forms a subdivision face which will transfer its excess charge to its vertex-neighbor f at x_1 .
3. f is wedge-neighbor to one 0-triangle and one 1-triangle. W.l.o.g. assume that the wedge-neighbor to f via edge e_1 is a 0-triangle and hence the wedge-neighbor to f via edge e_2 is a 1-triangle. By assumption, edge e_2 has three crossings. Assume first that, $a_2 \in f_3[e_2]$, see Fig. 3c. Suppose first that $f_2 \neq t_2$, i.e., the gray-dashed edge in Fig. 3c is present. Observe that in this case, we have that e_1 and e_2 have three crossings each. But then f'_0 is a rich immediate neighbor to f_1 (as witnessed by the edges $f'_0[e_2]$, $f'_0[e_0]$ and $f'_0[e_1]$) and hence f did not charge f_1 to begin with, a contradiction to our assumption. If $f_2 = t_2$ holds, then we observe that (v, x_2, a_2) defines a subdivision face of f_3 , which will distribute its excess charge to its vertex-neighbor f_2 at x_2 , which will then propagate it to its unique wedge-neighbor f . Otherwise, we have $b_2 \in f_3[e_2]$ and thus $a_2 \in f'_0[e_2]$. If also $a_1 \in f'_0[e_1]$ holds, then f'_0 is a rich immediate neighbor to f_1 – hence f did not redistribute charge to f_1 and thus its charge was sufficient to begin with. Hence assume that $a_1 \notin f'_0[e_1]$, which implies e_1 has an additional crossing which belongs to $\delta f'_0$. In this case, consider f'_2 and observe that (b_2, x_2, b_3) lie consecutively on the boundary of f'_3 , as the dotted red edge in Fig. 3d cannot be present due to (3P), hence the sequence defines a subdivision face which distributes its excess charge to its vertex neighbor f at x_2 .

4.3.2 f is a 0-pentagon

Observation 12 establishes that f is not a (discharging) wedge-neighbor to any 0-triangle. By (B), f cannot be a wedge-neighbor to five 1-triangles.

f is a wedge-neighbor to exactly three 1-triangles

Observe that in this case, it is sufficient to distribute 0.5 units of charge to f . Assume first that the three 1-triangles appear consecutively. W.l.o.g. assume that t_0, t_1 and t_2 exist and that the common endpoint of e_0 and e_2 (i.e., v_1) is of the same partition as v_0 .



■ **Figure 4** Illustrations used in the recharging of 0-pentagons that are wedge-neighbors to three 1-triangles.

1. f_3 is a 0- x face. In this case, both e_2 and e_4 have a crossing in $\delta f_3 \setminus \delta f$ and are thus crossed three times each – this implies that $f_0 = t_0$ and $f_1 = t_1$ and thus the sequence (v_1, x_0, v_0) forms a subdivision face of f'_0 which can distribute its excess charge of 0.5 units to its vertex-neighbor f at x_0 , see Fig. 4a.
2. f_3 is a 1-quadrangle, hence either e_2 or e_4 has a crossing in $\delta f_3 \setminus \delta f$.
 - e_2 has a crossing in $\delta f_3 \setminus \delta f$. Again we observe that f_3 cannot be a wedge-neighbor at e_2 by (B). Thus, if f'_2 is not a 0-pentagon, f_3 has sufficient excess charge which can be distributed to its immediate neighbor f at e_3 , see Fig. 4b. If f'_2 is a 0-pentagon, then we necessarily have an additional edge that intersects both e_1 and e_3 . But then, by assumption both e_1 and e_2 have three intersections each – hence it holds that $f_0 = t_0$ and $f_1 = t_1$ and thus (v_1, x_0, v_0) forms a subdivision face of f'_0 which distributes its excess charge to its vertex-neighbor f at x_0 , see Fig. 4c.
 - e_4 has a crossing in $\delta f_3 \setminus \delta f$. In this case, the immediate neighbor of f_3 at e_4 can be a 1-triangle f' , see Fig. 4d (if this is not the case, we proceed as in the previous case). Let us denote the unique original vertex of f' by v' . If $b_0 \in f_4[e_0]$ then the sequence (b_0, x_4, x_3, v') either defines f_4 or forms a subdivision face – in both cases it has sufficient excess charge to distribute 0.5 units each to its vertex-neighbor f_0 at x_4 , which then propagates the charge to its unique wedge-neighbor f . Otherwise, $b_0 \notin f_4[e_0]$. Since e_4 and e_0 have three crossings each, it follows that $f_0 = t_0$ and $f_1 = t_1$ and hence again (v_1, x_0, v_0) forms a subdivision face of f'_0 which distributes its excess charge to its vertex-neighbor f at x_0 , again see Fig. 4d.
3. In the remaining cases, f_3 always has sufficient charge which it can distribute to f over edge e_3 . The crucial observation is that the edge e_3 is neither part of a wedge-neighbor relation nor of a rich immediate neighbor relation. Since we discharge 0.5 units over e_3 , which is the same quantity as a discharge over an edge that defines a wedge-neighbor relation in the second step, the analysis at the start of Step 2 still holds.

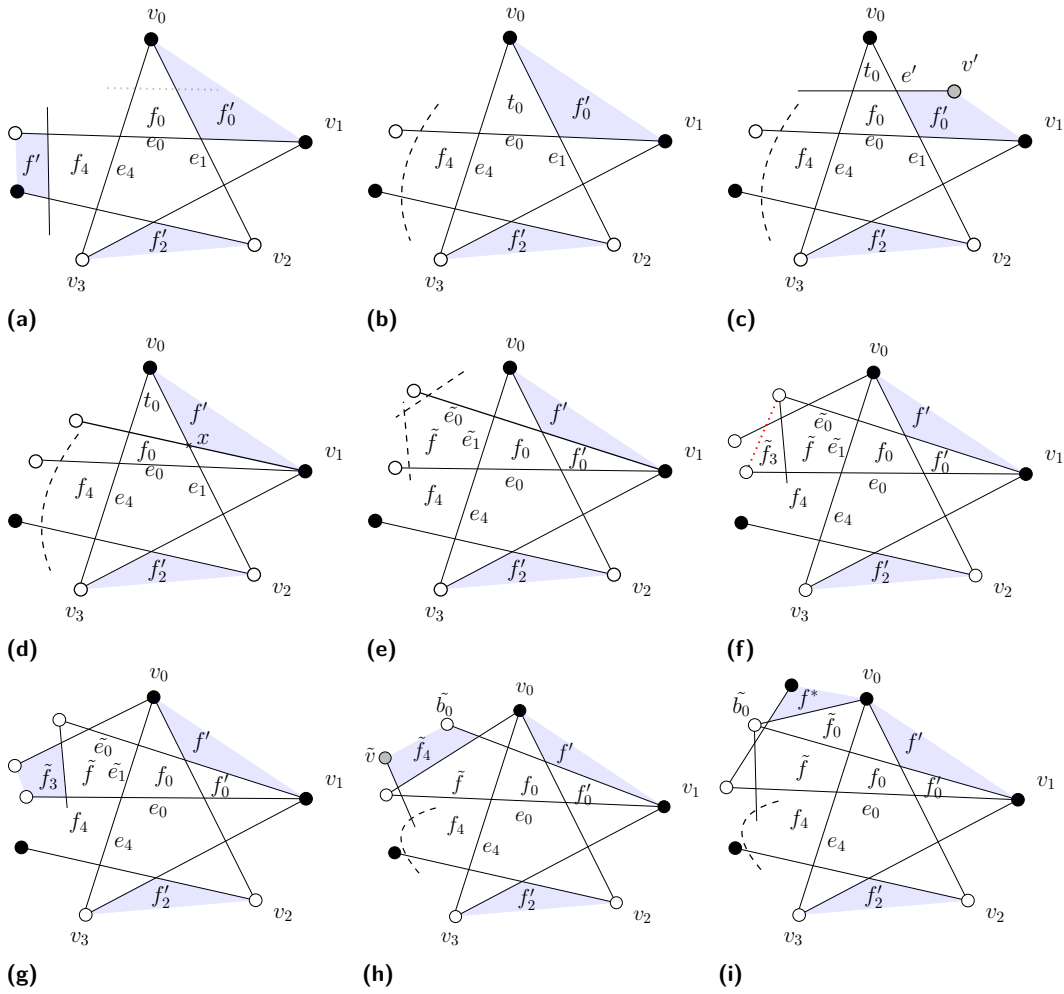
Assume now that the three 1-triangles do not appear consecutively. W.l.o.g. assume that t_0, t_2 and t_3 exist and that $v_0 \in A$ (which implies $v_2 \in B$ and $v_3 \in B$). Further, assume that, when traversing e_0 starting at a_0 , we encounter x_4 before x_0 , see Fig. 4e.

1. f_4 is a 0- x face. This implies that both e_0 and e_3 have a crossing in $\delta f_4 \setminus \delta f$. Consider f_1 . By construction $b_0 \in f_1[e_0]$, thus f_1 is not a 0- x face. By assumption, f_1 is not a 1-triangle. If f_1 is a 1-quadrangle, it is not a wedge-neighbor to any 1-triangle by (B) and it cannot be a wedge-neighbor to a 0-triangle due to (3P) (as it would necessarily have to cross e_1 and hence also e_3 , which already has three crossings), see Fig. 4e. The crucial observation here is that f'_1 cannot be a 0-pentagon, as this would imply $f_2 \neq t_2$ and thus an additional crossing of e_3 , a contradiction to (3P). Hence, f_1 does not have to distribute charge over the edge e_2 and hence has sufficient excess charge to distribute 0.5 units of charge to its immediate neighbor f . Finally, if f_1 is any larger face, it again has sufficient excess charge which can be distributed to f (the argument is analogous to the previous case).
2. f_4 is a 1-quadrangle. Assume first that $a_0 \notin f_4[e_0]$. Hence, a_3 is the unique vertex of f_4 , see Fig. 4f. By (B), f'_4 cannot be a 1-triangle. If f'_4 is a 0- x face, then there is an edge e' that intersects e_4 and e_1 such that $f_0 \neq t_0$. But then both e_1 and e_4 have three crossings each, and hence the sequence (v_3, x_2, v_2) forms a subdivision face of f'_2 which distributes 0.5 units of charge to its vertex-neighbor f , see Fig. 4f. If f'_4 is not a 0- x face (in particular, not a 0-pentagon), then f_4 has sufficient charge which it distributes to its immediate neighbor f at e_4 .
Assume now that $a_3 \notin f_4[e_3]$. Hence, a_0 is the unique vertex of f_4 . If $f_0 = t_0$, then the sequence (v_0, x_4, a_0) defines a subdivision face of f'_4 which distributes its charge to its vertex-neighbor f at x_4 . This situation is depicted in Fig. 4g when the dashed-gray edge does not exist. Otherwise, $f_0 \neq t_0$ and thus f_0 is a 0-quadrangle. This implies that $t_2 = f_2$ and $t_3 = f_3$ by (3P). But then the sequence (v_3, x_2, v_2) defines a subdivision face of f'_2 which distributes its excess charge to its vertex-neighbor f at x_2 , again see Fig. 4g.
3. f_4 is a larger face. Again, f_4 contains sufficient charge which it can distribute to f .

f is a wedge-neighbor to four 1-triangles

W.l.o.g. we assume that f_4 is not a 1-triangle and that $v_0 \in A$, which implies that $v_1 \in A, v_2 \in B$ and $v_3 \in B$.

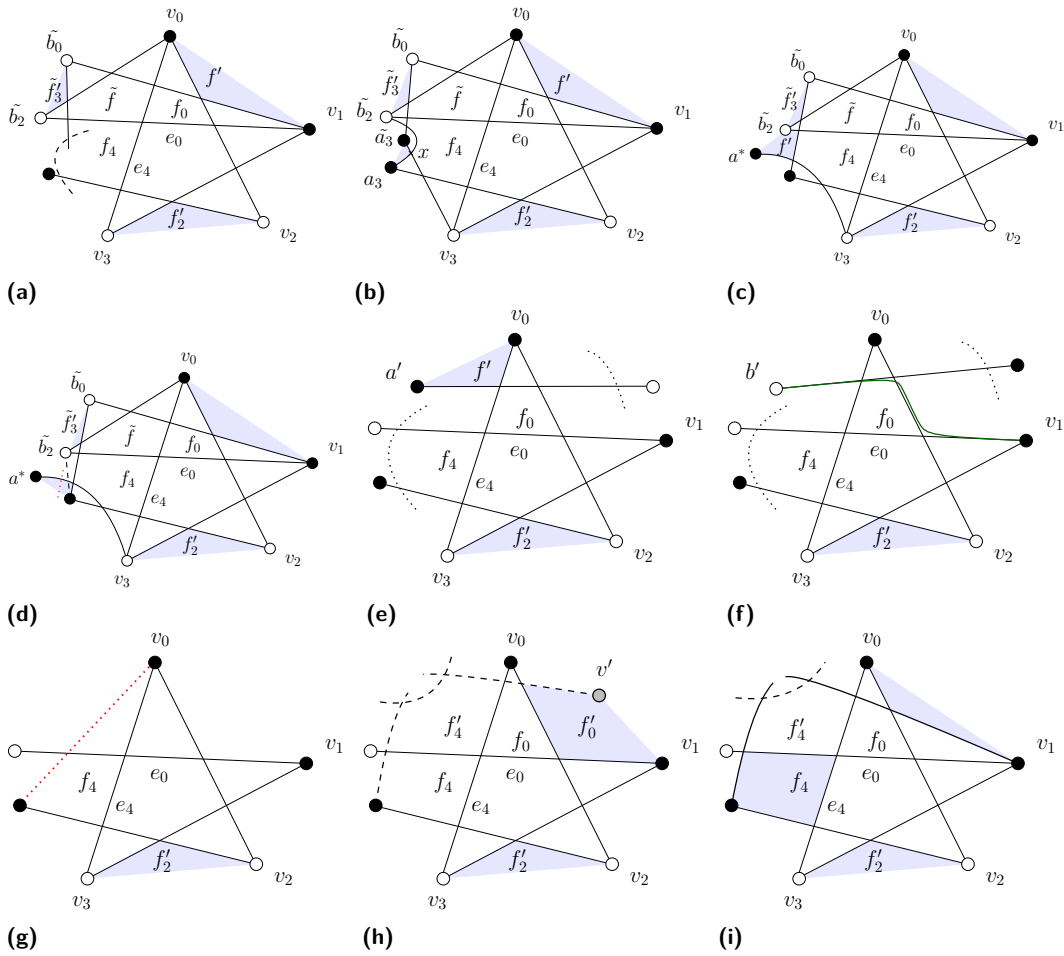
1. f_4 is a 0- x face. It follows that both e_0 and e_3 have three crossings each. Let us first consider the case where f_4 is a 0-quadrangle. Let e' be the edge of f_4 opposite to e_4 . Further, denote by x (y) the intersection of e' with e_0 (e_3). Let f' be the immediate neighbor of f_4 at e' . By (B) and (3P) we have $b_0 \in \delta f'$ and $a_3 \in \delta f'$. But then the sequence $f_a = (b_0, x, y, a_3)$ forms a subdivision of f' (note that f_a could coincide with f') which has an excess of at least one charge. Since it is not possible that both immediate neighbors of f_4 different from f' and f are 0-pentagons by (3P), f' loses at most 0.5 charge over a vertex-neighbor relation – thus it can distribute 0.5 charge to f through f_4 . In order to determine who supplies the remaining 0.5 charge, consider e_0 . If e_0 has no crossing in $\delta f_0 \setminus f$, then the sequence (v_1, x_0, v_0) forms a subdivision which distributes its excess charge to f ; otherwise, if e_0 has a crossing in $\delta f_0 \setminus f$, then by (3P) e_0 has no crossing in $\delta f_3 \setminus f$ and hence (v_3, x_2, v_2) forms the desired subdivision, see Fig. 5a. Henceforth, we can thus assume that f_4 is not a 0-quadrangle.
 - a. e_4 has no crossing outside of δf . Observe that this implies that also e_1 and e_2 do not have any additional crossings outside of δf , as an edge which is crossing e_1 or e_2 would necessarily also cross either e_0, e_3 or e_4 . We can thus identify two pairs of consecutive vertices of the same partition, namely (v_0, v_1) and (v_2, v_3) . Observe that $\delta f'_0$ ($\delta f'_2$) contains both v_0 and v_1 (v_2 and v_3). Hence, (v_1, x_0, v_0) and (v_3, x_2, v_2) each form a subdivision face which can distribute its excess charge to its vertex-neighbor f at x_0 and x_2 , respectively, see Fig. 5b.



■ **Figure 5** Illustrations used for the first part of the case analysis where face f is a 0-pentagons that is a wedge-neighbor to four 1-triangles.

- b. e_4 has a crossing in $\delta f_0 \setminus \delta f$ with an edge $e' = (a', b')$, the case where e_4 has a crossing in $\delta f_3 \setminus f$ is symmetric. This implies that $v_0 \notin f_0[e_4]$ and thus $f_0 \neq t_0$. We can then again observe that $f_3 = t_3$ and $f_2 = t_2$ holds and thus (v_3, x_2, v_2) forms a subdivision face of f'_2 , which distributes its excess charge of 0.5 to its vertex-neighbor f at x_2 . For the remaining 0.5 charge, we consider the following cases.
 - i. $a' \in f'_0[e']$ and $a' \neq v_1$ or $b' \in f'_0[e']$, see Fig. 5c. Let v' be the original vertex of $f'_0[e']$. Hence, by assumption $v' = b'$ or $v' = a'$, in both cases it holds that $v' \neq v_1$ (by (B) or by assumption). The sequence (v_1, x_0, x, v') , where x is the intersection point of e' and e_1 , then defines a subdivision face f_a (in the case of $v' = b'$, it is possible that f_a and f'_0 coincide). Observe that the vertex-neighbor of f_a at x is t_0 , which does not require any charge. Note that f_a is potentially a wedge-neighbor via the edge e_1 . However, since the initial charge of f_a is at least 2 (the extremal case occurs if $f_a = f'_0$ holds). In every case, f_a has an excess charge of at least one unit, it has sufficient charge such that it can distribute 0.5 to its wedge-neighbor and 0.5 to its vertex-neighbor f at x .

- ii. $a' \in f'_0[e']$ and $a' = v_1$. Again, let x be the intersection between e' and e_1 and let f' be the vertex-neighbor of f_0 at x . Since $a' \in f'_0[e']$ and since e_1 already has three crossings, it follows that (v_1, x, v_0) forms a subdivision of f' , see Fig. 5d, which charges 0.5 units to its vertex-neighbor f_0 at x . Before we propagate this charge from f_0 to f , we first have to consider f'_4 . If f'_4 is not a 0-pentagon, f'_4 has sufficient charge even before the third step, unless f'_4 is a 1-quadrangle. However, in the previous case analysis, we never required a 0-quadrangle to propagate the charge to a 1-quadrangle – hence, we can safely propagate 0.5 units of charge from f_0 to f . Hence, consider now the case that f'_4 is a 0-pentagon. Note that this implies that f_4 cannot be a 0-triangle by (3P). Since the following analysis is quite detailed, we set $\tilde{f} = f'_4$ and reuse the adjusted notation of f . In particular, we get $\tilde{e}_0 = e'$, $\tilde{e}_1 = e_4$ and $\tilde{e}_2 = e_0$, see Fig. 5e. Observe that $\tilde{t}_1 = f'_0$ and since $\tilde{f}_2 = f_4$ and since we covered the case where f_4 is a 0-quadrangle already, it follows that \tilde{t}_2 does not exist.
- A. \tilde{f} is wedge-neighbor to at most two 1-triangles. In this case, \tilde{f} does not require any charge and hence f_0 can propagate 0.5 units to f .
- B. \tilde{f} is wedge-neighbor to three 1-triangles.
- \tilde{t}_3 is missing. Assume first that \tilde{f}_3 is a 1-quadrangle. By (B), it is not a wedge-neighbor at \tilde{e}_4 to a 1-triangle, see Fig. 5f. Moreover, the immediate face at \tilde{e}_4 is not a 0-x face by (3P). Thus, \tilde{f}_3 does not loose charge over \tilde{e}_4 and we can therefore distribute the excess charge to \tilde{f} . If \tilde{f}_3 is not a 1-quadrangle, then the sequence $(\tilde{b}_4, \tilde{x}_3, \tilde{x}_2, \tilde{b}_2)$ forms a subdivision of face \tilde{f}_3 , see Fig. 5g. This subdivision has 1.5 units of excess charge and can therefore charge 0.5 to its immediate neighbor \tilde{f} at \tilde{e}_3 as well as 0.5 to its vertex-neighbors at \tilde{x}_2 and \tilde{x}_3 (if required).
 - \tilde{t}_4 is missing. By (3P), face \tilde{f}_4 is not a 1-quadrangle as it contains an original vertex of \tilde{e}_3 , which we denote by \tilde{v} as well as \tilde{b}_0 . The sequence $(\tilde{b}_0, \tilde{x}_4, \tilde{x}_3, \tilde{v})$ forms a subdivision face f_a of face \tilde{f}_4 (observe that $f_a = \tilde{f}_4$ is possible) – hence f_a has at least one unit of excess charge, see Fig. 5h. Since its vertex-neighbor at \tilde{x}_4 is a 1-triangle, it does not charge over \tilde{x}_4 (the same holds for \tilde{x}_3 , but this is not necessary), and thus its excess charge is sufficient to distribute 0.5 units to its immediate neighbor \tilde{f} at \tilde{f}_4 .
 - \tilde{t}_0 is missing. Assume first that \tilde{f}_0 is a 1-quadrangle. Note that in this case, it is possible that \tilde{f}_0 is a wedge-neighbor at \tilde{e}_4 to a 1-triangle t^* . If this is not the case, we proceed as in the first subcase, i.e., \tilde{f}_0 charges its excess of 0.5 units to its immediate neighbor \tilde{f} at \tilde{e}_0 . If \tilde{f}_0 is a wedge-neighbor at \tilde{e}_4 , then we consider the immediate neighbor of \tilde{f}_0 at the edge $e^* = (\tilde{a}_1, \tilde{b}_0)$, see Fig. 5i, which we denote by f^* . If we denote by x^* the intersection point between e^* and \tilde{e}_4 , then the sequence $(\tilde{a}_1, x^*, \tilde{a}_3)$ defines a subdivision face which distributes its excess charge to its vertex-neighbor t^* at x^* . Thus, \tilde{f}_0 can again distribute its excess charge to \tilde{f} as desired. The case where \tilde{f}_0 is not a 1-quadrangle is analogous to the previous ones, i.e., we can identify a suitable subdivision face.
- C. \tilde{f} is wedge-neighbor to four 1-triangles. In this extremal case, we have the setting depicted in Fig. 6a. We can again observe that the sequence $(\tilde{b}_0, \tilde{x}_3, \tilde{b}_2)$ forms a subdivision face of \tilde{f}_3 which can distribute 0.5 units of charge to \tilde{f} . In this case, we will evenly split the charge that f_0 obtained earlier such that each of \tilde{f} and f obtain 0.25 units. Note that both \tilde{f} and f need an additional charge of 0.25 units. Consider face f_4 . Recall that f_4 is a 0-x face by assumption, but



■ **Figure 6** Illustrations used for the second part of the case analysis where face f is a 0-pentagons that is a wedge-neighbor to four 1-triangles.

it cannot be a 0-triangle as observed earlier. If f_4 has size at least seven, it has sufficient charge to distribute 0.25 to each of \tilde{f} and f . To see this, let $x = |f_4|$ and observe that f_4 has an excess charge of $x - 4$. Since f_4 does not contain any original vertices, it cannot be a rich immediate neighbor nor a discharging vertex-neighbor by Inv. 14. Further, since f loses a combined charge of 0.5 over the edges e_0 and e_4 , it follows that it discharges at most $0.5(x - 1) \leq x - 4$ for $x \geq 7$. If f_4 is a 0-6 and does not have sufficient charge, then we have exactly the setting depicted in Fig. 6b. But then the sequence (\tilde{a}_3, x, a_3) is a subdivision face which distributes 0.5 charge to its vertex-neighbor f_4 at x , which can then be used to charge \tilde{f} and f . Hence, assume that f_4 is a 0-5 face, see Fig. 6c for the extremal case where f_4 is an immediate neighbor to two 0-pentagons \tilde{f} and f as well as a wedge-neighbor to two 1-triangles (the other possible arrangements of the two 1-triangles surrounding f_4 are symmetric). Let f' be the immediate neighbor of f_4 at the edge \tilde{e}_3 . By (3P) we have $\tilde{b}_2 \in \delta f'$. Now, consider the edge $e^* = (a^*, b^*)$ that is incident to v_3 (i.e., $v_3 = b^*$) and intersects the edge \tilde{e}_3 . If $a^* \in \delta f'$, then we can find a subdivision face, see Fig. 6c, which charges its immediate neighbor f_4 at \tilde{e}_3 . Otherwise, if $a^* \notin \delta f'$, then e^* is intersected by an

additional edge e'' . If $e'' = (a_3, \tilde{b}_2)$, then a_3 and a^* (together with an intersection point) again define a subdivision face, see Fig. 6d. We remark here that the local configuration which can be observed in Fig. 6d is dense, i.e., we do not have any excess charge left – this then yields, after an appropriate concatenation of the configurations, an alternative construction of bipartite 3-planar graphs on n vertices and $4n - \mathcal{O}(1)$ edges.

If $e'' \neq (a_3, \tilde{b}_2)$, we claim that $(G, \Gamma) \notin \mathcal{G}''$. Indeed, by substituting the edge \tilde{e}_3 by the edge (a_3, \tilde{b}_2) (which will be drawn along the old curve of \tilde{e}_3 and e_0) and inheriting the remainder of the drawing, we can construct a graph G' and a corresponding 3-planar drawing Γ' with strictly less crossings. Observe that by our construction, any copy of (a_3, \tilde{b}_2) , if it exists, is not homotopic to our new curve. Hence, we obtain a contradiction to our choice of (G, Γ) .

- iii. $a' \notin f'_0[e']$ and $b' \notin f'_0[e']$. Suppose first that, when traversing e' starting at a' , we encounter its intersection with e_4 before its intersection with e_0 , see Fig. 6e. Let x be the intersection between e' and e_4 and let f' be the vertex-neighbor of f_0 at x . The sequence (v_0, x, a') defines a subdivision face which charges 0.5 units to its vertex-neighbor f_0 at x . The crucial observation is that by (3P), a' is necessarily part of f'_4 – by a similar argument as in the previous case, f'_4 has sufficient charge and thus f_0 can propagate 0.5 to f .

Suppose now that, when traversing e' starting at b' , we encounter its intersection with e_4 before its intersection with e_0 , see Fig. 6f. But then our choice of (G, Γ) is a contradiction to our assumption unless b' and b_0 coincide, as the drawing Γ' of $G \setminus e' \cup \{(b', v_1)\}$, which inherits the curve of all edges of Γ and adds the edge (b', v_1) such that its curve is drawn along e', e_1 and e_0 , see the green edge in Fig. 6f, has strictly less crossings. Suppose therefore that b' and b_0 would coincide. But since f_4 is by assumption a 0- x face, we have that e_0 contains a crossing in $\delta f_4 \setminus f$ – but then by (3P) we necessarily have $b_0 = b' \in \delta f'_0$, a contradiction to our assumption.

2. f_4 is a x - y face with $x \geq 1$. By assumption, f_4 is not a 1-triangle and by construction it cannot be a 2-triangle.
 - f_4 is a quadrangle. Observe that it cannot be a wedge-neighbor via edges e_3 or e_0 by (B), see the red dotted edge in Fig. 6g for the case of e_0 , the other is symmetric. W.l.o.g. we assume that $f_4[e_0]$ does not contain an original vertex, i.e., the unique original vertex of f_4 is an endpoint of e_3 . Now, consider face f'_4 . If f'_4 is not a 0-pentagon, then f'_4 does not lose any charge over e_0 and hence can distribute its excess charge to its immediate neighbor f at e_4 . If f'_4 is a 0-pentagon, we necessarily have that $f_0 \neq t_0$, see Fig. 6h. But then e_1 and e_4 are crossed by an edge e' which has, by definition, an additional crossing in f'_4 . Consequently, by (3P), f'_0 contains an endpoint of e' . Let v' be this endpoint and denote by x' the intersection of e' with e_1 . If $v' \neq v_1$, we either have $f'_0 = (v_1, x_0, x', v')$ or (v_1, x_0, x', v') forms a subdivision face of f'_0 , see Fig. 6h. In either case, f'_0 has sufficient charge such that it can distribute 0.5 units to its vertex-neighbor f at x_0 . Otherwise, v' and v_1 coincide, see Fig. 6i and we are in the same setting as in Case 1(b)ii with the difference that f_4 is a 1-quadrangle instead of a 0- x face. Again, we can observe that (v_1, x, v_0) , where x is the intersection between e' and e_1 , forms a subdivision face and it charges 0.5 units to its vertex-neighbor f_0 at x_0 . Similar to the previous case, f_0 distributes 0.5 units to f unless f'_4 is a wedge-neighbor to exactly four 1-triangles. If f'_4 is a wedge-neighbor to four 1-triangles, then the remaining charge for f'_4 and f (we will cover the missing 0.5 charge for f right after) is provided by f_4 – as it has 0.5 excess charge, it can distribute 0.25 to each of its wedge-neighbors, see Fig. 6i.

- f_4 is a larger face. Similar to previous cases, f_4 then always has sufficient excess charge to distribute 0.5 to its immediate neighbor f at edge e_4 .

In order to determine who distributes the missing 0.5 charge to f , we consider face f_4 . By assumption, f_4 contains at least one original vertex. Assume first that both b_0 and a_3 belong to δf_4 . In this case f_4 is a 2-quadrangle (or a larger face) and has an excess charge of at least one. Since the vertex-neighbor at x_3 is a 1-triangle, it does not lose charge over x_3 – hence, it has sufficient charge to (potentially) distribute 0.5 via x_4 and to distribute 0.5 to its immediate neighbor f . Hence, assume that exactly one of b_0 or a_3 belongs to δf_4 . Assume w.l.o.g. that a_3 belongs to δf_4 which implies that e_0 has an additional crossing in $\delta f_4 \setminus f$. Now, if $f_0 = t_0$ holds, then (v_1, x_0, v_0) defines a subdivision face which distributes its excess charge to its vertex-neighbor f at x_0 . Otherwise, if $f_0 \neq t_0$, then e_1 and e_4 have an additional crossing. But then neither e_2 nor e_3 can have an additional crossing by (3P), hence $t_2 = f_2$ and $t_3 = f_3$ and hence (v_3, x_2, v_2) forms a subdivision face which distributes 0.5 units of charge to its vertex-neighbor f at x_2 .

After this final step, every face of F' satisfies Equation (1) which concludes the proof.

5 Implications

We will now use the main result of Section 4 to improve the lower-bound for the number of crossings, which consequently improves various other results. Note that, besides some numerical differences, the proof strategies for Sections 5.1 and 5.2 are identical to the ones of [3], while the proofs of Section 5.3 are identical to the ones of [17].

5.1 Crossing Lemma and Edge Density bounds

► **Theorem 15.** *Let G be a simple bipartite graph with $n \geq 3$ vertices and m edges. Then, the crossing number $cr(G)$ satisfies the following:*

$$cr(G) \geq 4m - \frac{25}{2}n + 27$$

Proof. The statements clearly hold when $m \leq 2n - 4$. Hence, we may assume w.l.o.g. that $m > 2n - 4$. It follows from [10] that if $m > 3n - 8$, then G has an edge that is crossed by at least two other edges. Also, by [3], we know that if $m > \frac{7}{2}n - 7$, then G has an edge that is crossed by at least three other edges. Finally, if $m > 4n - 8$, then Theorem 6 establishes that G has an edge that is crossed by at least four other edges. Hence we obtain by induction on the number of edges of G that the crossing number $cr(G)$ is at least:

$$cr(G) \geq (m - (2n - 4)) + (m - (3n - 8)) + (m - (\frac{7}{2}n - 7)) + (m - (4n - 8)) = 4m - \frac{25}{2}n + 27$$

◀

► **Theorem 16.** *Let G be a simple bipartite graph with n vertices and m edges, where $m \geq \frac{75}{16}n$. Then, the crossing number $cr(G)$ satisfies the following:*

$$cr(G) \geq \frac{1024}{16875} \cdot \frac{m^3}{n^2} \approx \frac{1}{16.5} \frac{m^3}{n^2}$$

Proof. Assume that G admits a drawing on the plane with $cr(G)$ crossings and let $p = \frac{75n}{16m} \leq 1$. Choose independently every vertex of G with probability p , and denote by G_p the graph induced by the vertices chosen in G_p . Let also n_p , m_p and c_p be the random variables corresponding to the number of vertices, of edges and of crossings of G_p . Taking expectations on the relationship $c_p \geq 4m_p - \frac{25}{2}n_p + 27$, which holds by Theorem 15, we obtain:

$$p^4 cr(G) \geq 4p^2m - \frac{25}{2}np \Rightarrow cr(G) \geq \frac{4m}{p^2} - \frac{25n}{2p^3}$$

The proof of the theorem follows by plugging $p = \frac{75n}{16m}$ (which is at most 1 by our assumption on m) to the inequality above. ◀

► **Theorem 17.** *Let G be a simple bipartite k -planar graph with n vertices and m edges, for some $k \geq 2$. Then:*

$$m \leq \sqrt{\frac{16875}{2048}kn} \approx 2.871\sqrt{kn}$$

Proof. For $k = 2$ and $k = 3$, the bounds of this theorem are weaker than the corresponding ones of [3], and of Theorem 6, respectively. So, we may assume w.l.o.g. that $k > 3$. We may also assume that $m \geq \frac{75}{16}n$, as otherwise there is nothing to prove. Combining the fact that G is k -planar with the bound of Theorem 16 we obtain:

$$\frac{1024}{16875} \cdot \frac{m^3}{n^2} \leq cr(G) \leq \frac{1}{2}mk$$

which implies:

$$m \leq \sqrt{\frac{16875}{2048}kn} \approx 2.871\sqrt{kn} \quad \blacktriangleleft$$

► **Theorem 18.** *Let G be a simple bipartite k -gap-planar graph with n vertices and $m \geq \frac{75}{16}n$ edges. Then:*

$$m \leq \sqrt{\frac{16875}{1024}kn} \approx 4.06\sqrt{kn}$$

Proof. By definition, we have that

$$cr(G) \leq k \cdot m$$

for any k -gap-planar graph G with m edges. On the other hand, Theorem 16 gives us

$$cr(G) \geq \frac{1024}{16875} \frac{m^3}{n^2}$$

since G is bipartite. Thus

$$\frac{1024}{16875} \frac{m^3}{n^2} \leq cr(G) \leq k \cdot m$$

and the result follows. ◀

5.2 Exclusion of complete bipartite graphs

► **Theorem 19.** *Let $K_{n,m}$ be a complete bipartite graph and let $n \leq m$. Then, K_{5,m_1} with $m_1 \geq 13$, K_{6,m_2} with $m_2 \geq 9$ and K_{7,m_3} with $m_3 \geq 7$ are not 3-planar and not gap-planar.*

Proof. $K_{5,13}$ has $5 \cdot 13 = 65$ edges, but any bipartite 3-planar (gap-planar) graph on 18 vertices has at most $4 \cdot 18 - 8 = 64$ edges, a contradiction. Similarly, we have that $K_{6,9}$ has $54 > 4 \cdot 15 - 8 = 52$ and $K_{7,7}$ has $49 > 4 \cdot 14 - 8 = 48$ edges. ◀

5.3 Biplanar crossing number

The biplanar (k -planar) crossing number of a graph G , denoted by $cr_2(G)$ ($cr_k(G)$), is the minimum number of crossings over all possible drawings of the edges of G in two (k) disjoint planes.

► **Theorem 20.** *Let $K_{p,q}$ be a complete bipartite graph with $p, q \geq 30$. Then*

$$cr_2(K_{p,q}) \geq \frac{p(p-1)q(q-1)}{204}$$

Proof. Applying [17, Lemma 1] together with Lemma 15 yields $cr_2(G) \geq 4m - (\frac{25}{2}n - 27) \cdot 2$ and thus $cr_2(K_{17,17}) \geq 360$. Using the recurrence relation

$$cr_2(K_{p+1,p+1}) \geq \left\lceil \frac{p+1}{p-1} \left\lceil \frac{p+1}{p-1} cr_2(K_{p,q}) \right\rceil \right\rceil$$

repeatedly as in [17], we obtain $cr_2(K_{30,30}) \geq 3723$ and thus

$$cr_2(K_{p,q}) \geq \frac{p(p-1)q(q-1)}{30 \times 29 \times 30 \times 29} cr_2(K_{30,30})$$

which yields the desired result. ◀

► **Theorem 21.** *For all $p, q \geq 9k + 2$,*

$$cr_k(K_{p,q}) \geq \frac{p(p-1)q(q-1)}{66 \cdot 3k^2}$$

Proof. Using Lemma 15, we obtain

$$cr_k(K_{9k+2,9k+2}) \geq 99k^2 + 121k + 16$$

Following the proof of Theorem 7 in [17] we then obtain the desired result. ◀

6 Conclusions and Open Problems

We have established tight upper bounds on the number of edges of bipartite gap-planar and bipartite 3-planar graphs. The following questions follow naturally:

- What is the density of bipartite k -planar graphs, in particular for $k = 4$? One could most likely apply the discharging method in a similar way for any fixed k – the issue that arises for larger k is just the sheer number of cases one has to consider. Hence, we ask as an open problem if one can (partially) automate such a charging proof in a similar way to [4]. This is of course also an interesting question in the normal (non-bipartite) setting.
- A graph is quasi-planar if there is a drawing in which no three edges mutually cross. What is the edge density of bipartite quasi-planar graphs?

References


- 1 Eyal Ackerman. On topological graphs with at most four crossings per edge. *Comput. Geom.*, 85:101574, 2019. doi:10.1016/j.comgeo.2019.101574.
- 2 Eyal Ackerman and Gábor Tardos. On the maximum number of edges in quasi-planar graphs. *J. Comb. Theory A*, 114(3):563–571, 2007. doi:10.1016/j.jcta.2006.08.002.

- 3 Patrizio Angelini, Michael A. Bekos, Michael Kaufmann, Maximilian Pfister, and Torsten Ueckerdt. Beyond-planarity: Turán-type results for non-planar bipartite graphs. In *ISAAC*, volume 123 of *LIPICs*, pages 28:1–28:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICs.ISAAC.2018.28.
- 4 K. Appel and W. Haken. Every planar map is four colorable. Part I: Discharging. *Illinois Journal of Mathematics*, 21(3):429–490, 1977. doi:10.1215/ijm/1256049011.
- 5 Sang Won Bae, Jean-François Baffier, Jinhee Chun, Peter Eades, Kord Eickmeyer, Luca Grilli, Seok-Hee Hong, Matias Korman, Fabrizio Montecchiani, Ignaz Rutter, and Csaba D. Tóth. Gap-planar graphs. *Theor. Comput. Sci.*, 745:36–52, 2018. doi:10.1016/j.tcs.2018.05.029.
- 6 Steven Chaplick, Myroslav Kryven, Giuseppe Liotta, Andre Löffler, and Alexander Wolff. Beyond outerplanarity. In *GD*, volume 10692 of *Lecture Notes in Computer Science*, pages 546–559. Springer, 2017. doi:10.1007/978-3-319-73915-1_42.
- 7 Otfried Cheong, Sariel Har-Peled, Heuna Kim, and Hyo-Sil Kim. On the number of edges of fan-crossing free graphs. *Algorithmica*, 73(4):673–695, December 2015. doi:10.1007/s00453-014-9935-z.
- 8 Walter Didimo, Giuseppe Liotta, and Fabrizio Montecchiani. A survey on graph drawing beyond planarity. *ACM Comput. Surv.*, 52(1), February 2019. doi:10.1145/3301281.
- 9 Weidong Huang, Peter Eades, and Seok-Hee Hong. Larger crossing angles make graphs easier to read. *J. Vis. Lang. Comput.*, 25(4):452–465, 2014. doi:10.1016/j.jvlc.2014.03.001.
- 10 Dmitri V Karpov. An upper bound on the number of edges in an almost planar bipartite graph. *Journal of Mathematical Sciences*, 196:737–746, 2014.
- 11 Michael Kaufmann and Torsten Ueckerdt. The density of fan-planar graphs. *Electron. J. Comb.*, 29(1):P1–29, 2022. doi:10.37236/10521.
- 12 Petra Mutzel. An alternative method to crossing minimization on hierarchical graphs. In Stephen C. North, editor, *Graph Drawing, Symposium on Graph Drawing, GD '96, Berkeley, California, USA, September 18-20, Proceedings*, volume 1190 of *Lecture Notes in Computer Science*, pages 318–333, Berlin, Heidelberg, 1996. Springer. doi:10.1007/3-540-62495-3_57.
- 13 János Pach, Rados Radoicic, Gábor Tardos, and Géza Tóth. Improving the crossing lemma by finding more crossings in sparse graphs. *Discret. Comput. Geom.*, 36(4):527–552, 2006. doi:10.1007/s00454-006-1264-9.
- 14 János Pach and Géza Tóth. Graphs drawn with few crossings per edge. *Combinatorica*, 17(3):427–439, 1997. doi:10.1007/BF01215922.
- 15 Gerhard Ringel. Ein Sechsfarbenproblem auf der Kugel. *Abh. Math. Sem. Univ. Hamb.*, 29:107–117, 1965.
- 16 Thomas Schneck. *New Parameters for Beyond-Planar Graphs*. dissertation, University Tübingen, 2020. URL: <https://publikationen.uni-tuebingen.de/xmlui/handle/10900/109680>.
- 17 Alireza Shavali and Hamid Zarrabi-Zadeh. On the biplanar and k-planar crossing numbers. In *CCCG*, pages 293–297, 2022.
- 18 Colin Ware, Helen Purchase, Linda Colpoys, and Matthew McGill. Cognitive measurements of graph aesthetics. *Information Visualization*, 1:103–110, June 2002. doi:10.1057/palgrave.ivs.9500013.

Improving the Crossing Lemma by Characterizing Dense 2-Planar and 3-Planar Graphs

Aaron Büngener ✉

Universität Tübingen, Germany

Michael Kaufmann ✉ 

Universität Tübingen, Germany

Abstract

The classical Crossing Lemma by Ajtai et al. and Leighton from 1982 gave an important lower bound of $c \frac{m^3}{n^2}$ for the number of crossings in any drawing of a given graph of n vertices and m edges. The original value was $c = 1/100$, which then has gradually been improved. Here, the bounds for the density of k -planar graphs played a central role. Our new insight is that for $k = 2, 3$ the k -planar graphs have substantially fewer edges if specific local configurations that occur in drawings of k -planar graphs of maximum density are forbidden. Therefore, we are able to derive better bounds for the crossing number $\text{cr}(G)$ of a given graph G . In particular, we achieve a bound of $\text{cr}(G) \geq \frac{73}{18}m - \frac{305}{18}(n - 2)$ for the range of $5n < m \leq 6n$, while our second bound $\text{cr}(G) \geq 5m - \frac{407}{18}(n - 2)$ is even stronger for larger $m > 6n$.

For $m > 6.79n$, we finally apply the standard probabilistic proof from the BOOK and obtain an improved constant of $c > 1/27.61$ in the Crossing Lemma. Note that the previous constant was $1/29$. Although this improvement is not too impressive, we consider our technique as an important new tool, which might be helpful in various other applications.

2012 ACM Subject Classification Mathematics of computing → Graph theory; Mathematics of computing → Combinatorics

Keywords and phrases Crossing Lemma, k -planar graphs, discharging method

Digital Object Identifier 10.4230/LIPIcs.GD.2024.29

1 Introduction

The classical Crossing Lemma by Ajtai et al. [4] and Leighton [10] has been considerably improved constant-wise from $\frac{1}{100}$ in many subsequent works [3, 11, 13] and for many variants [16], such as bipartite graphs [5], graphs of bounded girth [12], multigraphs [9, 14], etc. Székely [17] gave an collection of applications of the Crossing Lemma in discrete geometry.

The gradual improvement of the above mentioned constant has been mainly done by using the linear bounds for the number of edges for planar, 1-planar, 2-planar, etc. graphs. k -planar graphs have a drawing where each edge is crossed at most k times. Density bounds for k -planar n -vertex graphs have been subject to intensive research in the past. While planar graphs have at most $3n - 6$ edges, the best known upper bounds for 1-planar, 2-planar and 3-planar graphs are $4n - 8$ [18], $5n - 10$ [13] and $5.5n - 11.5$ [7] respectively; for the corresponding non-simple versions the bounds might slightly differ [7]. They have been directly applied for better bounds for the crossing lemma. The current best constant of $\frac{1}{29}$ uses even the bound for 4-planar graphs [1], which is $6n - 12$.

We will perform a more refined analysis by considering drawings that are in some sense between k -planar and $k + 1$ -planar drawings for $k = 1, 2$. In their paper from 2006 [11], Pach, Radoicic, Tardos and Tóth used a similar approach to improve the corresponding constant of the Crossing Lemma. They considered the density of 1-planar drawings with a fixed number of crossing-free triangles, a class of drawings between planar and 1-planar in general.



© Aaron Büngener and Michael Kaufmann;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 29; pp. 29:1–29:22

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

A similar road has been taken in the paper [2] about simple quasi-planar graphs. While the general density bound here is $6.5n$, the authors consider drawings without triangular cells that have no vertex on the boundary. For such a more general class, a bound of $7n$ can be derived. This bound has not been applied for the Crossing Lemma, though. We will apply such a refined look to 2- and 3-planar drawings: It turns out that either we can prove much smaller bounds for the edge density than provided by the upper bounds of the corresponding k -planar classes (which is per se good for the Crossing Lemma) or we can characterize the drawing in a very good way, which simplifies the way of counting the crossings.

The idea has been motivated by some results in the literature. (Non-simple) optimal 2-planar and 3-planar graphs have been characterized [7], and there is very limited flexibility for the structure of such graphs. We know that with much less restrictions on the drawings, the limits of the maximum density for some superclasses for 1-planar and 2-planar graphs are still roughly at the same value. Examples for this effect are the min-1-planar and min-2-planar graphs [8] as superclasses of 1-planar and 2-planar graphs, as well as gap-planar graphs as a superclasses of 2-planar graphs [6].

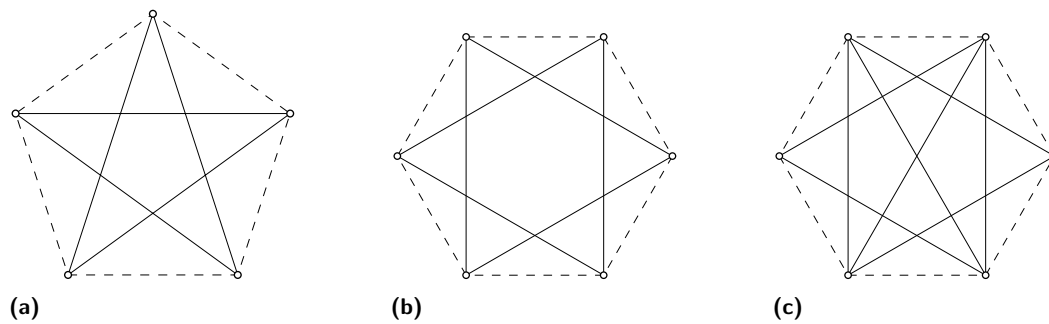
To use the concept of k -planarity for various values of k , we planned to specify at which point between k - and $k + 1$ -planarity the density is changing. This turned out to be difficult, and hence we go the other way around and forbid local configurations that have to occur in optimal k -planar drawings. That leads to nice insights on the density bounds and surprising results. Note that all our results hold for non-simple graphs and non-simple drawings.

2 Definitions and Notation

A *drawing* or *topological graph* D is a graph drawn in the plane such that the vertices are pairwise distinct points and the edges are represented as Jordan arcs connecting the corresponding endpoints. We assume simplicity in the sense that edges do not overlap other vertices in the interior. Two edges might cross, but we do not allow that more than two edges cross at a single point. We also assume that two edges have only a finite number of common interior points and no two edges meet tangentially. Remark that we will consider not necessarily simple drawings, i.e., we will allow non-homotopic multiple edges as well as adjacent crossing edges, while loops are forbidden. Since we mostly assume that the number of crossings will be minimal, there will be no empty lenses, i.e., empty regions having a boundary that is being defined by two edges; c.f. Proposition 10.

The *crossing number* $\text{cr}(D)$ is defined to be the total number of crossing points in D . For an abstract graph G , the *crossing number* $\text{cr}(G)$ is the minimum value of $\text{cr}(D)$ over all drawings D with D is a drawing of G . A drawing D is *k -planar* if no edge is crossed more than k times. A graph G is *k -planar* if it has a k -planar drawing.

Forbidden configurations. We now define three forbidden configurations that play a key role: A *full k -planar p -gon* F_p^k can be described by a p -cycle C_p of planar edges with no other vertices inside, which is then greedily extended by a maximal number of edges to be placed inside that are as short as possible observing this subgraph is still k -planar. To finally arrive at F_p^k , we delete the planar cycle C_p at the boundary. In this way, we define a *full 2-planar pentagon* F_5^2 to be the graph $K_5 - C_5$ drawn in the way described above (see Figure 1a). Similarly, we can define full 2-planar hexagons F_6^2 and full 3-planar hexagons F_6^3 as specific drawings of subgraphs of $K_6 - C_6$. More precisely, a *full 2-planar hexagon* consists of the six short, i.e., 2-hop edges inside a planar C_6 (see Figure 1b). A *full 3-planar hexagon* consists of all possible 2-hop and two 3-hop edges inside a planar C_6 (see Figure 1c).



■ **Figure 1** (a) A full 2-planar pentagon F_5^2 , (b) a full 2-planar hexagon F_6^2 and (c) a full 3-planar hexagon F_3^6 with their boundaries (dashed).

Clearly, a configuration F_p^k may be crossed by some other edges. But for full 2-planar pentagons and full 2-planar hexagons, this cannot happen in the case of 2-planar drawings, which motivates to define the planar 5-cycle resp. 6-cycle surrounding them as their *boundary* (even if not all of its edges may exist in a drawing). This implies that, for 2-planar drawings, full 2-planar pentagons and hexagons are edge-disjoint (while they may have common boundary edges). Similarly, in the case of 3-planarity and full 3-planar hexagons, the cycle surrounding them consists of uncrossed edges if there are no empty lenses. With this in mind, we analogously define the *boundary* of a full 3-planar hexagon, and observe that these configurations are edge-disjoint for 3-planar drawings.

Using the definitions above, we are able to state our main results in the next section.

3 Results

In this section, we present our results. The proofs of Theorem 1 and Theorem 3 use the discharging method and can be found in Section 4.

► **Theorem 1.** *Any graph G with $n \geq 3$ vertices that admits a 2-planar F_5^2 -free drawing has at most $4.5(n - 2)$ edges. If the drawing is also F_6^2 -free, then G has at most $\frac{13}{3}(n - 2)$ edges.*

Counting the number of edges in a drawing consisting of $0.5(n - 2)$ full 2-planar hexagons, we see that the first of the two bounds is tight. For the second bound, we refer to a pentagonalization of the plane, where four edges have been added within each pentagon.

► **Corollary 2.** *For every 2-planar drawing of any graph with $n \geq 3$ vertices and $\frac{13}{3}(n - 2) + x$ edges for $x \in [0, \frac{2}{3}(n - 2)]$, the number of F_5^2 and F_6^2 configurations is at least x .*

Note that G cannot be 2-planar for $x > \frac{2}{3}(n - 2)$ by the corresponding density bound.

Proof. Assume that drawing D is a 2-planar drawing of a graph with n vertices and $\frac{13}{3}(n - 2) + x$ edges such that the number of F_5^2 or F_6^2 configurations is $y < x$. We can destroy those configurations by removing one edge from each F_5^2 and F_6^2 . Hence, we still have more than $\frac{13}{3}(n - 2)$ edges, which is a contradiction to Theorem 1. ◀

This implies that drawings of optimal 2-planar graphs consist of $\frac{2}{3}(n - 2)$ full 2-planar pentagons, a fact that has been already known [7]. Similar results hold for 3-planar drawings.

► **Theorem 3.** *Any graph with $n \geq 3$ vertices that admits a 3-planar F_6^3 -free drawing has at most $5(n - 2)$ edges.*

29:4 Improving the Crossing Lemma

This bound is tight, which one can see by considering optimal 2-planar graphs.

The next corollary allows us to characterize drawings of dense 3-planar graphs very well. This extends the characterization of optimal 3-planar graphs, which must have a drawing consisting of $\frac{1}{2}(n-2)$ F_6^3 configurations and their boundaries [7].

► **Corollary 4.** *For every 3-planar drawing of any graph with $n \geq 3$ vertices and $5(n-2) + x$ edges for $x \in [0, 0.5(n-2)]$, the number of F_6^3 configurations is at least x .*

Note that G cannot be 3-planar for $x > 0.5(n-2)$ by the corresponding density bound.

Proof. Analogously to the proof of Corollary 2, we assume that there is a 3-planar drawing D of a graph with n vertices and $5(n-2) + x$ edges such that the number of F_5^2 or F_6^2 configurations is $y < x$. Those configurations can be destroyed by removing one edge from each F_6^3 , hence we still have more than $5(n-2)$ edges, which is a contradiction to Theorem 3. ◀

A consequence of this is a new upper bound for the edge density of simple 3-planar graphs, i.e., the case where multi-edges are forbidden. Note that the best known bound before was $5.5n - 11.5$ edges [7] and there exist examples with $5.5n - 15$ edges [11].

► **Corollary 5.** *There are no 3-planar graphs on $n \geq 3$ vertices with $5.5n - 11.5$ edges. Therefore, any simple 3-planar graph on $n \geq 3$ vertices has at most $5.5n - 12$ edges.*

Proof. Assume that there exists a (not necessarily simple) 3-planar graph G with $5.5n - 11.5$ edges. Then, by Corollary 4, we would find in any 3-planar drawing D of G at least $0.5(n-2) - 0.5$ full 3-planar hexagons. Let \mathcal{H} be any triangulation on the set of vertices that includes all the boundaries of all F_6^3 configurations in D . As F_6^3 configurations consist of four triangles, only $2(n-2) - 4(0.5(n-2) - 0.5) = 2$ triangles in \mathcal{H} do not belong to an F_6^3 .

Now we count the edges. Starting with the edges of \mathcal{H} , each F_6^3 consists of five additional edges. The other two triangles may contain one additional edge, which gives in total at most $3(n-2) + 2.5(n-2) - 2.5 + 1 = 5.5n - 12.5$ edges, contradicting the assumed density. ◀

From Theorem 1 and Theorem 3 we can also derive new lower bounds for the number of crossings in a graph. The proof can be found in Section 5.

► **Theorem 6.** *Let G be a graph with $n > 2$ vertices and m edges. Then*

- (a) $\text{cr}(G) \geq \frac{73}{18}m - \frac{305}{18}(n-2)$,
- (b) $\text{cr}(G) \geq 5m - \frac{407}{18}(n-2)$.

A slightly weaker bound than in (a) of $\text{cr}(G) \geq 4m - \frac{50}{3}(n-2)$ can be derived with a significantly shorter proof by only applying Theorem 3; we point this out in the proof.

That improves the best known results for $m > 5(n-2)$, which are $\text{cr}(G) \geq 4m - \frac{103}{6}(n-2)$ [11] respectively $\text{cr}(G) \geq 5m - \frac{139}{6}(n-2)$ [1]. Theorem 6 implies directly a better constant in the Crossing Lemma.

► **Theorem 7.** *Let G be a graph with n vertices and m edges. Then $\text{cr}(G) \geq \frac{6000}{165649} \frac{m^3}{n^2} - \frac{218351}{165649}n > \frac{1}{27.61} \frac{m^3}{n^2} - 1.32n$. If $m \geq 6.79n > \frac{407}{60}n$, then $\text{cr}(G) \geq \frac{6000}{165649} \frac{m^3}{n^2} > \frac{1}{27.61} \frac{m^3}{n^2}$.*

Proof. Let G be a graph with n vertices and m edges. For the case $m \geq \frac{407}{60}n$, we construct a random subgraph G' by selecting every vertex of G independently with probability $p = \frac{407}{60}n/m \leq 1$. We denote the number of edges and vertices in G' by m' and n' . By Theorem 6 and linearity of expectation, we obtain $\mathbb{E}[\text{cr}(G')] \geq 5\mathbb{E}[m'] - \frac{407}{18}\mathbb{E}[n']$. We replace $\mathbb{E}[n'] = pn$, $\mathbb{E}[m'] = p^2m$ and $\mathbb{E}[\text{cr}(G')] = p^4 \text{cr}(G)$, and get

$$\text{cr}(G) \geq \frac{5m}{p^2} - \frac{407n}{18p^3} = \frac{6000}{165649} \frac{m^3}{n^2}.$$

For the case $m < \frac{407}{60}n$ we compare the bound $\text{cr}(G) \geq \frac{6000}{165649} \frac{m^3}{n^2} - \frac{218351}{165649}n$ with the corresponding best known linear bounds $\text{cr}(G) \geq m - 3(n - 2)$, $\text{cr}(G) \geq \frac{7}{3}m - \frac{25}{3}(n - 2)$ [11] and Theorem 6. ◀

One direct application of the improved Crossing Lemma is a new bound on the edge density for k -planar graphs.

▶ **Corollary 8.** *For $k \geq 2$, any simple k -planar graph with n vertices has at most $3.72\sqrt{kn}$ edges.*

Proof. As in [13], the new bound for k -planar graphs can be derived directly from the new Crossing Lemma and the fact that each edge can be crossed at most k times:

$$\frac{1}{27.61} \frac{m^3}{n^2} \leq \text{cr}(G) \leq km/2,$$

which then leads to $m \leq \sqrt{13.805kn} \leq 3.72\sqrt{kn}$. ◀

The best previous constant in the bound was 3.81.

4 Proof of Theorems 1 and 3

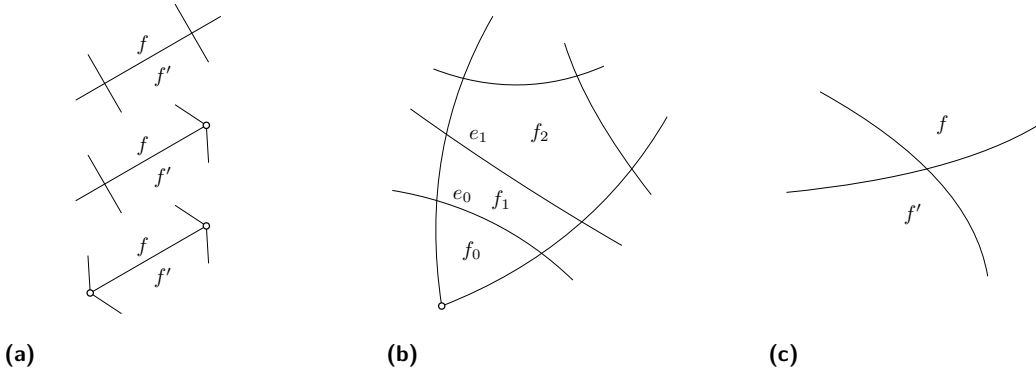
In this section, we give the proofs of the two central theorems of our paper. First, we will introduce some necessary concepts, we basically adopted the notation by Ackerman [1].

Notation. We interpret a drawing D as a plane map $M(D) = (V', E')$ whose vertices V' are either vertices $V(D)$ of D or crossing points of D . An edge e in E' connects two vertices of V' , i.e., it is a crossing-free segment of an edge of D , which we denote by \bar{e} . We call an edge of E' an r -edge, if $r \in \{0, 1, 2\}$ of its endpoints are vertices of D . For a vertex $v \in V(D)$, we write $\deg(v)$ for its *degree*. The degree of a crossing is always four.

Let F' be the set of faces of $M(D)$. For a face $f \in F'$, we write $|f|$ for the number of edges in E' that are incident to f . Similarly, $|V(f)|$ denotes the number of (real) vertices of D that are incident to f . Note that we will assume 2-connectivity, hence the boundary of every face is a simple cycle and we avoid double-counting of the vertices. A face with $|f| = s$ is called a s -gon. In the cases of $s = 3, 4, 5, 6, 7$ we write instead *triangle*, *quadrilateral*, *pentagon*, *hexagon* and *heptagon*. If we want to denote that $|V(f)| = r$ and $|f| = s$, we write r - s -gon and use this wording also for 2-triangles, 0-quadrilaterals, etc. for simplicity. If we only want to specify for a face that $|V(f)| = r$, then we call it an r -face.

Further, we need some definitions for relations between faces in F' . Two faces are r -neighbors if they share an r -edge. Let now be $e_0 \in E'$ a 0-edge of a face $f_0 \in F'$ and $f_1 \in F'$ the 0-neighbor of f_0 at e_0 . For $i \geq 1$, if $f_i \in F'$ is a 0-quadrilateral, then let be $f_{i+1} \in F'$ the 0-neighbor of f_i at the edge e_i opposite to f_{i-1} . The face f_i , for which i is maximal, is called the *wedge-neighbor* of f_0 at e_0 . Since D is 3-planar, we have $i \leq 3$. Notice the alternative definition of a wedge-neighbor by Ackerman [1]. Finally, we define two faces $f, f' \in F'$ to be *vertex-neighbors*, if f and f' share a crossing-vertex c , but not an edge in E' incident to c . See Figure 2 for an illustration of the defined terms.

Preliminaries for the proofs. We prove both theorems by induction. This will allow us, as in [1], to study only 2-connected drawings (see Proposition 9). For $n = 3$, independently from the forbidden configurations, there are at most three non-homotopic edges in any drawing and therefore both theorems hold. If $n > 3$ and there is a vertex $v \in G$ with $\deg(v) \leq 4$, then the theorems follow after removing v by induction.



■ **Figure 2** Illustrations of the defined neighborhood-relations. (a) From top to bottom: The faces f and f' are 0-neighbors, 1-neighbors, 2-neighbors resp. (b) The 0-pentagon f_2 is the wedge-neighbor of the 1-triangle f_0 at its edge e_0 . (c) The faces f and f' are vertex-neighbors.

► **Proposition 9.** *If D is not 2-connected, then Theorem 1 and Theorem 3 are true.*

Proof. The argument follows the lines of [1]. To argue for the different scenarios of Theorem 1 and Theorem 3 at the same time, let $a(n - 2)$ for $a \in \{\frac{13}{3}, 4.5, 5\}$ be an upper bound on the number of edges, which we want to prove. Assume that there is a vertex $x \in E'$ such that $M(D) \setminus \{x\}$ is not connected. Then x is either a vertex or a crossing of D .

If x is a vertex of D , then $D \setminus \{x\}$ is not connected, so let D_1, \dots, D_k be the connected components of $D \setminus \{x\}$. Let further D' be the drawing induced by $V(D_1) \cup \{x\}$ and D'' the drawing induced by $V(D_2) \cup \dots \cup V(D_k) \cup \{x\}$. Let $|V(D')| = n'$, $|V(D'')| = n''$ and observe $n' + n'' = n + 1$. Since every vertex has at least degree four, $4 < n', n'' < n$ holds. By induction, it follows $m \leq (an' - 2a) + (an'' - 2a) = a(n + 1) - 4a < a(n - 2)$.

Assume now that x is a crossing of D . Let \hat{D} be the drawing obtained by replacing x by a vertex. This increases the number of vertices by one and the number of edges by two. Let D_1, \dots, D_k be the connected components of $\hat{D} \setminus \{x\}$. Again, let D' be the drawing induced by $V(D_1) \cup \{x\}$ and D'' the drawing induced by $V(D_2) \cup \dots \cup V(D_k) \cup \{x\}$. For $|V(D')| = n'$, $|V(D'')| = n''$ we observe $4 < n', n'' < n$. By induction, we get $m \leq (an' - 2a) + (an'' - 2a) - 2 = a(n + 2) - 4a - 2 < a(n - 2)$. ◀

Therefore we will always assume that D is 2-connected. As both theorems consider upper bounds for the number of edges for the specific graph classes, we also assume that we consider graphs G that are edge-maximum for the specific class of graphs, and for such graphs a corresponding drawing D that is crossing-minimum. These assumptions will enable us to conduct a focused analysis of the bounds for the number of edges.

► **Proposition 10.** *Let D be a drawing that is either (1) 2-planar F_5^2 -free or (2) 2-planar F_5^2 -free and F_6^2 -free or (3) 3-planar F_6^3 -free and maximally-dense-crossing-minimal under this restriction. Then the following properties hold:*

- (a) *There are no empty lenses.*
- (b) *For all faces $f \in F'$ we have $|f| \geq 3$.*
- (c) *The wedge-neighbor of a 0-triangle or a 1-triangle is a face $f \in F'$ with $|f| \geq 4$ that is not a 0-quadrilateral.*
- (d) *If there are two vertices $u, v \in V(D)$ on the boundary of a face $f \in F'$, then the edge uv is part of the boundary of f . Therefore every face $f \in F'$ with $|V(f)| > 2$ is a 3-triangle.*

Proof.

- (a) Since there are no two homotopic edges, there are no empty lenses with two vertices. Any other empty lens can be destroyed by swapping the segments of the edges of D that define the empty lens (without creating one of the forbidden configurations). This reduces the number of crossings contradicting that D is crossing-minimal.
- (b) Loops and self-intersecting edges are forbidden, so there is no face $f \in F'$ with $|f| = 1$. Every face $f \in F'$ with $|f| = 2$ is an empty lens, which does not appear in D by (a).
- (c) Let f be an arbitrary face. By definition, the face f is never a 0-quadrilateral. If $|f| = 3$, then this would imply an empty lens.
- (d) For an arbitrary face f , assume that no edge $e = uv$ exists on the boundary of f . Therefore, we may insert e contradicting that D is maximally dense. By this, we cannot create one of the three forbidden configurations F_5^2, F_6^2 and F_6^3 , since they do not contain planar edges. This does not create homotopic edges as every other edge $e' = uv$ homotopic to e would have been already on the boundary of f or would have formed an empty lens with an edge of the boundary of f contradicting (a).
Assume now that a face f with $|V(f)| > 2$ exists that is not a 3-triangle. Then we find three vertices in $V(D)$ on the boundary of f , which do not all appear next to each other. We introduce a new edge between two of them, contradicting the maximality of D . ◀

In the following, we will use the *discharging method*. See [1, 2, 8, 15] for similar applications of this technique. We define a *charging function* $\text{ch} : F' \rightarrow \mathbb{R}$ that assigns an *initial charge* of

$$\text{ch}(f) = |f| + |V(f)| - 4 \quad (1)$$

to every face $f \in F'$. It is known that for the total charge $\sum_{f \in F'} \text{ch}(f) = 4n - 8$ holds (refer to [2] for details). The challenge now is to redistribute the charge so that in the end every face $f \in F'$ has a charge of $\text{ch}'(\cdot)$ that satisfies $\text{ch}'(f) \geq \alpha|V(f)|$ for a suitable $\alpha > 0$, while the total charge does not change. From this and the observation that $\sum_{f \in F'} |V(f)| = \sum_{v \in V(D)} \deg(v) = 2m$ holds, we can derive an upper bound of

$$m \leq \frac{2}{\alpha}(n - 2) \quad (2)$$

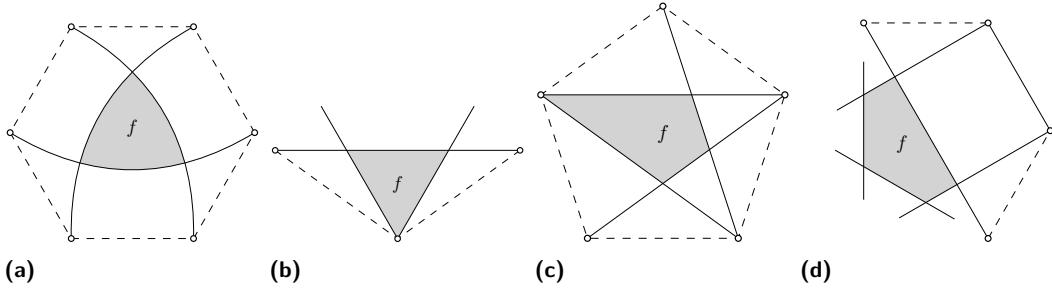
on the number of edges. For a given α and a face f with charge c , we say that $|c - \alpha|V(f)||$ is the *demand* of f , if $c - \alpha|V(f)|$ is negative, otherwise we call it the *excess* of f . If f has no demand, then we also say that f is satisfied.

4.1 Proof and Discharging for Theorem 1

► **Theorem 1.** *Any graph G with $n \geq 3$ vertices that admits a 2-planar F_5^2 -free drawing has at most $4.5(n - 2)$ edges. If the drawing is also F_6^2 -free, then G has at most $\frac{13}{3}(n - 2)$ edges.*

Proof. We start with the bound of $\frac{13}{3}(n - 2)$. Let D be a 2-planar, F_5^2 -free and F_6^2 -free drawing that is maximally-dense-crossing-minimal. Assign to every face $f \in F'$ the initial charge $\text{ch}(f)$ according to Equation (1). The initial charges are distributed in the following way:

- Step 1: Each 0-triangle receives $\frac{1}{3}$ charge from each of its wedge-neighbors.
- Step 2: Each 1-triangle receives $\frac{1}{26}$ charge from both 1-neighbors.
- Step 3: Each 1-triangle receives $\frac{5}{13}$ charge from its wedge-neighbor.
- Step 4: Each 2-quadrilateral contributes its excess to its wedge-neighbor.
- Step 5: For each 2-triangle f , let $\mathcal{C}(f)$ be the inclusion-minimal planar cycle of D enclosing f (i.e. the planar cycle that does not contain other planar edges). Then f distributes its excess equally over those faces that lie inside $\mathcal{C}(f)$ and have a demand.



■ **Figure 3** Discharging for Theorem 1. Planar edges that exist by Proposition 10 are dashed.

Denote the charges after the i -th step by $\text{ch}_i(\cdot)$. With this, we have $\text{ch}'(\cdot) = \text{ch}_5(\cdot)$.

► **Proposition 11.** *For all faces $f \in F'$, we have $\text{ch}'(f) \geq \frac{6}{13}|V(f)|$.*

Proof. We analyze the final charge $\text{ch}'(\cdot)$ for all faces. Note that a face contributes through each edge of its boundary in Step 1-3 at most once and the only contributing faces in Step 1 are 2-quadrilaterals (see Figure 3a) and in Step 2 2-triangles (see Figure 3b). Also $\text{ch}_3(f) \geq \frac{6}{13}|V(f)|$ already implies $\text{ch}'(f) \geq \frac{6}{13}|V(f)|$. Because of Proposition 10 there are only 3-triangles and faces f with $|f| \geq 3$ and $|V(f)| \leq 2$.

- *f is a 0-triangle.* Then f receives in Step 1 in $3 \cdot \frac{1}{3}$ charge and never contributes charge. Therefore $\text{ch}_3(f) = -1 + 1 = 0 \geq \frac{6}{13} \cdot 0$.
- *f is a 1-triangle.* Then f receives in Step 2 $2 \cdot \frac{1}{26}$ charge, in Step 3 $\frac{5}{13}$ charge and never contributes charge. Therefore $\text{ch}_3(f) = 0 + \frac{6}{13} \geq \frac{6}{13} \cdot 1$.
- *f is a 2-triangle.* Then f starts with 1 charge and contributes in Step 2 at most $2 \cdot \frac{1}{26}$ charge. Therefore $\text{ch}_3(f) \geq 1 - \frac{1}{13} = \frac{12}{13} \geq \frac{6}{13} \cdot 2$.
- *f is a 3-triangle.* Then f never receives or contributes charge. Thus $\text{ch}_3(f) = 2 \geq \frac{6}{13} \cdot 3$.
- *f is a 0-quadrilateral.* Then f starts with 0 charge and never receives or contributes charge as it cannot be the wedge-neighbor of another face. Therefore $\text{ch}_3(f) = 0 \geq \frac{6}{13} \cdot 0$.
- *f is a 1-quadrilateral.* Then f starts with 1 charge. If f contributes in Step 3 to less than two 1-triangles, we have $\text{ch}_3(f) \geq 1 - \frac{5}{13} = \frac{8}{13} \geq \frac{6}{13} \cdot 1$. Otherwise, we know that f is bounded by a 5-cycle of planar edges (Figure 3c). Here, charges do not change in Step 4, but we can find $\frac{3}{13}$ charge from the excesses of 2-triangles in this 5-cycle and move that to f in Step 5. Therefore, we have $\text{ch}'(f) = 1 - 2 \cdot \frac{5}{13} + \frac{3}{13} = \frac{6}{13} \geq \frac{6}{13} \cdot 1$.
- *f is a 2-quadrilateral.* Then f has one wedge-neighbor, to which it contributes either $\frac{1}{3}$ charge in Step 1 or $\frac{5}{13}$ charge in Step 3. So we have $\text{ch}_3(f) \geq 2 - \frac{5}{13} = \frac{21}{13} \geq \frac{6}{13} \cdot 2$.
- *f is a 0-pentagon.* Note that all wedge-neighbors of f are 1-triangles or 2-quadrilaterals, as otherwise there would be an edge with three crossings or a face with two real vertices that are not connected by an edge. If f contributes to five 1-triangles in Step 3, then we would have an F_5^2 configuration, which is forbidden. Otherwise, at least one 2-quadrilateral contributes its excess of $\frac{14}{13}$ to f in Step 4 (see Figure 3d). Therefore we have $\text{ch}_4(f) \geq 1 + \frac{14}{13} - 4 \cdot \frac{5}{13} = \frac{7}{13} \geq \frac{6}{13} \cdot 0$.
- *f is a 1-pentagon or a 2-pentagon resp.* Then f contributes to at most three or two 1-triangles resp. in Step 3. Therefore, we have $\text{ch}_3(f) \geq 2 - 3 \cdot \frac{5}{13} = \frac{11}{13} \geq \frac{6}{13} \cdot 1$ resp. $\text{ch}_3(f) \geq 3 - 2 \cdot \frac{5}{13} = \frac{29}{13} \geq \frac{6}{13} \cdot 2$.
- *f is a 0-hexagon.* If f contributes to six 1-triangles in Step 3, then we would have an F_6^2 configuration, which is forbidden. Otherwise, we have $\text{ch}_3(f) \geq 2 - 5 \cdot \frac{5}{13} = \frac{1}{13} \geq \frac{6}{13} \cdot 0$.
- *f is a 1-hexagon resp. 2-hexagon.* Then f contributes to at most four resp. three 1-triangles in Step 3 and we have $\text{ch}_3(f) \geq 3 - 4 \cdot \frac{5}{13} = \frac{19}{13} \geq \frac{6}{13} \cdot 2$.

- f is a face with $|f| \geq 7$. Then f may contribute charge to at most $|f|$ wedge-neighbors in Step 3. Therefore $\text{ch}_3(f) \geq |f| + |V(f)| - 4 - \frac{5}{13} \cdot |f| \geq \frac{8}{13} \cdot 7 + |V(f)| - 4 \geq \frac{6}{13}|V(f)|$. Therefore, all faces $f \in F'$ are satisfied, which proves the proposition. ◀

Combining Proposition 11 and Equation (2), $m \leq 2 \cdot \frac{13}{6}(n - 2)$ is implied, as claimed.

For drawings, where F_6^2 configurations are allowed, we can use similar discharging steps to prove the bound of $4.5(n - 2)$ on the number of edges. Here we set $\alpha = \frac{4}{9}$, and therefore 1-triangles can receive $\frac{1}{18}$ charge from both its 1-neighbors each in Step 2 without creating a demand for any 2-triangles. Therefore, faces have to contribute in Step 3 only $\frac{1}{3}$ charge to satisfy all 1-triangles. Now let f be a 0-hexagon that is the wedge-neighbor of six 1-triangles. Starting with 2 charge, it contributes at most $6 \cdot \frac{1}{3}$ in Step 3, and therefore ends with $0 \geq \frac{4}{9} \cdot 0$ charge. For all other faces we still have enough charge with the same analysis as above.

Therefore, there exists a function $\text{ch}'(\cdot)$ satisfying $\text{ch}'(f) \geq \frac{4}{9}|V(f)|$ for all $f \in F'$, while the total amount of charge is still $4n - 8$. By Equation (2) we get $m \leq 2 \cdot \frac{9}{4}(n - 2)$. ◀

4.2 Proof and Discharging for Theorem 3

► **Theorem 3.** *Any graph with $n \geq 3$ vertices that admits a 3-planar F_6^3 -free drawing has at most $5(n - 2)$ edges.*

Proof. Let D be a 3-planar F_6^3 -free drawing that is maximally-dense-crossing-minimal. As in the proof of Theorem 1, we assign the initial charges $\text{ch}(f)$ to the faces of $M(D)$ and redistribute them to achieve a function $\text{ch}'(\cdot)$. The discharging takes place in seven steps:

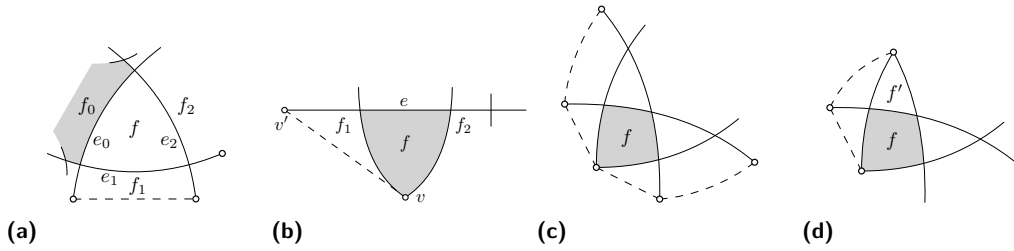
- Step 1: Each 0-triangle receives 1 charge from each 0-neighbor that is a 2-quadrilateral.
- Step 2: Each 0-triangle with a demand receives $\frac{1}{3}$ charge from all wedge-neighbors.
- Step 3: Each 2-triangle distributes its excess equally over all 1-neighbors that are 1-triangles.
- Step 4: Each 1-triangle receives its demand from its wedge-neighbor.
- Step 5: Each face distributes its excess equally over the wedge-neighbors that are 0-pentagons, but at most 0.3 to each of them, and keeps the rest.
- Step 6: Each face distributes its excess equally over all vertex-neighbors that are 0-quadrilaterals or 0-pentagons. The 0-quadrilaterals distribute this charge equally over their 0-neighbors that have a demand.
- Step 7: For each face f , let $\mathcal{C}(f)$ be the inclusion-minimal planar cycle of D enclosing f (i.e. the planar cycle that does not contain other planar edges). Then f distributes its excess equally over those faces that lie inside $\mathcal{C}(f)$ and have a demand.

Again, we denote by $\text{ch}_i(\cdot)$ the charges after the i -th step and by $\text{ch}'(\cdot)$ the final charges. Our goal is to show $\text{ch}'(f) \geq 0.4|V(f)|$ for all faces $f \in F'$. Note that this is already implied by $\text{ch}_4(f) \geq 0.4|V(f)|$, as in Step 5-7 faces contribute only their excesses. We structure the proof into several propositions, collecting statements about the discharging steps.

► **Proposition 12.** *After Step 2, 0-triangles are and remain satisfied.*

Proof. Let f be a 0-triangle. We have $\text{ch}(f) = -1$. If f receives in Step 1 charge, then $\text{ch}_1(f) = 0$. Otherwise, f receives $3 \cdot \frac{1}{3}$ charge in Step 2, so $\text{ch}_2(f) = 0$. 0-triangles do not contribute charge in Step 3-4, since they are not wedge-neighbors of 1-triangles. Therefore, $\text{ch}'(f) \geq 0.4 \cdot 0$ holds. ◀

► **Proposition 13.** *In Step 1-2, 0-faces contribute no charge.*



■ **Figure 4** Illustrations for the proofs of Proposition 13, Proposition 14 and Proposition 15.

Proof. No faces except 2-quadrilaterals contribute charge in **Step 1**, so we consider only **Step 2**. Assume that a 0-face f_0 contributes charge to a 0-triangle f in **Step 2, and f_0 and f are therefore 0-neighbors at an edge e_0 . Let e_1, e_2 be the other edges of f and f_1, f_2 the 0-neighbors at these edges (see Figure 4a). Since f_0 is a 0-face, it is incident to two crossings each with \bar{e}_1 and \bar{e}_2 and these edges also cross each other at f . Therefore \bar{e}_1 and \bar{e}_2 have already three crossings and end at f_1 resp. f_2 . The edge \bar{e}_0 ends also at one of f_1 or f_2 , as otherwise it would have four crossings. W.l.o.g. e_0 ends at f_1 and by Proposition 10 f_1 is a 2-quadrilateral. Hence, $\text{ch}_1(f) \geq 0.4 \cdot |V(f)|$, contradicting that f receives charge later. ◀**

► **Proposition 14.** *After Step 3, 1-triangles have a demand of at most 0.3 charge.*

Proof. Let f be a 1-triangle with the real vertex v and the 0-edge e . Let further f_1, f_2 be the 1-neighbors of f (see Figure 4b). Then \bar{e} ends at one of f_1 and f_2 , as otherwise it would have more than three crossings. W.l.o.g. let f_1 be that face with the vertex v' to which \bar{e} is incident. Then by Proposition 10 the edge vv' exists and f_1 is a 2-triangle. Therefore, f_1 starts with 1 charge and has an initial excess of 0.2. Thus, f receives 0.1 charge in **Step 3**. We have $\text{ch}_3(f) = 0.1$, which is equivalent to a demand of 0.3. ◀

► **Proposition 15.** *After Step 4, all 1-quadrilaterals are satisfied.*

Proof. Let f be a 1-quadrilateral. We have $\text{ch}(f) = 1$ and f contributes charge only in **Step 2** and **Step 4**. If f contributes to at most one wedge-neighbor or to two 1-triangles, then $\text{ch}_4(f) \geq 1 - 0.6 \geq 1 \cdot 0.4$. Otherwise, f contributes either to two wedge-neighbors that are both 0-triangles or to one 0-triangle and one 1-triangle. In the first case, both 0-triangles are already satisfied after **Step 1**, as they have wedge-neighbors that are 2-quadrilaterals (see Figure 4c). In the second case, f contributes not more than 0.2 charge to the 1-triangle f' , because one of its 1-neighbors is a 2-triangle contributing its excess of 0.2 charge only to f' in **Step 3** (see Figure 4d). Therefore, we have $\text{ch}_4(f) \geq 1 - \frac{1}{3} - 0.2 \geq 1 \cdot 0.4$. ◀

► **Proposition 16.** *After Step 4, all faces are and remain satisfied that are not 0-pentagons that are the wedge-neighbor of four or five 1-triangles.*

Proof. Note again that for a face f the charge $\text{ch}_4(f) \geq 0.4|V(f)|$ implies already that it has no demand in **Step 5-7**, since faces only there contribute their excesses.

To see that 0-triangles and 1-quadrilaterals are satisfied, we refer to Propositions 12 and 15. 1-triangles are satisfied by definition of **Step 4**. Remember that only 3-triangles and r - s -gons with $r \leq 2, s \geq 3$ can exist by Proposition 10. Now we discuss the other cases:

- *f is a 2-triangle.* We start with $\text{ch}(f) = 1$. As only wedge-neighbors contribute in **Step 1-2** and **Step 4** and f cannot be a wedge-neighbor of another face, the only critical step is **Step 3**. Here, f contributes in total at most its excess of 0.2 charge, and therefore $\text{ch}_4(f) \geq 1 - 0.2 \geq 2 \cdot 0.4$.

- *f* is a 3-triangle. We start with $\text{ch}(f) = 2$ and *f* never contributes charge. It follows that $\text{ch}_4(f) = 2 \geq 3 \cdot 0.4$ holds.
- *f* is a 0-quadrilateral. Again, *f* never contributes charge, and therefore $\text{ch}(f) = \text{ch}_4(f) = 0 \geq 0 \cdot 0.4$ holds.
- *f* is a 2-quadrilateral. We start with $\text{ch}(f) = 2$. Note that *f* contributes only once as it has only one wedge-neighbor, and therefore we have $\text{ch}_4(f) \geq 2 - 1 \geq 2 \cdot 0.4$.
- *f* is a 0-pentagon with at most three wedge-neighbors that are 1-triangles. We have $\text{ch}(f) = 1$ and *f* contributes only to three faces. With Proposition 13 and Proposition 14 $\text{ch}_4(f) \geq 1 - 3 \cdot 0.3 \geq 0 \cdot 0.4$ follows.
- *f* is a 1-pentagon or a 2-pentagon. *f* starts with $\text{ch}(f) \geq 2$ and we have $\text{ch}_4(f) \geq 2 - 3 \cdot \frac{1}{3} \geq 2 \cdot 0.4$.
- *f* is a face with $|f| \geq 6$. Then *f* may contribute to at most $|f|$ wedge-neighbors charge. Therefore, we have $\text{ch}_4(f) \geq |f| + |V(f)| - 4 - \frac{1}{3} \cdot |f| \geq |V(f)|$. ◀

It remains to prove that 0-pentagons with four or five wedge-neighbors that are 1-triangles have at least zero charge after Step 7. We show this by the following four propositions, which we only state here; the proofs can be found in Appendix A.

► **Proposition 17.** *In Step 5, each 0-pentagon receives 0.3 charge from all wedge-neighbors that are not 1-triangles, 0-triangles or 0-pentagons.*

► **Proposition 18.** *In Step 6, each 1-face and 2-face *f* with $|f| \geq 5$ and each 0-face *f* with $|f| \geq 7$ contributes at least 0.4 charge to the vertex-neighbors that are 0-quadrilaterals or 0-pentagons.*

► **Proposition 19.** *After Step 7, all 0-pentagons that are the wedge-neighbor of four 1-triangles are satisfied.*

► **Proposition 20.** *After Step 7, all 0-pentagons that are the wedge-neighbor of five 1-triangles are satisfied.*

By Propositions 16, 19 and 20 $\text{ch}'(f) \geq 0.4 \cdot |V(f)|$ holds for all faces $f \in F'$. Since charge is only moved, its total amount is still $4n - 8$ and Equation (2) implies $m \leq \frac{2}{0.4}(n - 2)$. ◀

5 Proof of Theorem 6

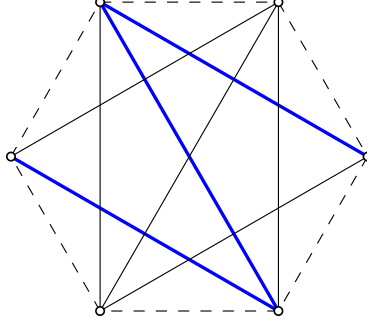
In this section, we present the proof of Theorem 6 that shows how to use the earlier stated observations and theorems and leads to a better bound for the Crossing Lemma.

► **Theorem 6.** *Let G be a graph with $n > 2$ vertices and m edges. Then*

- (a) $\text{cr}(G) \geq \frac{73}{18}m - \frac{305}{18}(n - 2)$,
- (b) $\text{cr}(G) \geq 5m - \frac{407}{18}(n - 2)$.

Proof. We start proving the bound in (a). If $m \leq 5(n - 2)$, then the bound follows from the linear bound $\text{cr}(G) \geq \frac{7}{3}m - \frac{25}{3}(n - 2)$ [11]. So assume $m > 5(n - 2)$ and let D be a crossing-minimal drawing of G . From D , we iteratively remove the edge with the most crossings until $5(n - 2)$ edges are left. In particular, as long as the maximum number of crossings is three, we always remove an edge from an F_6^3 configuration. By Theorem 3, we stop latest, when there are no F_6^3 configurations. By this process, edges are iteratively deleted until we reach $5(n - 2)$ edges, as following:

29:12 Improving the Crossing Lemma



■ **Figure 5** Three independent edges (blue) with three crossings in an F_6^3 configuration. In D_3 one of them is already deleted, in D_{3-} also the other two.

- m_{5+} edges with five or more crossings – denote the resulting drawing by D_4 ,
- then m_4 edges with four crossings – denote the resulting drawing by D_3 and the set of edges deleted in this step by E_4 ,
- then m_3 edges with three crossings from F_6^3 configurations – denote the resulting drawing by D_{3-} .

Note that m_4 or m_3 could be zero in the case that we reached $5(n-2)$ already during step (1) or (2). Afterwards we have m_3 edge-disjoint F_6^3 configurations with a missing edge in D_{3-} . So we are able to find $2m_3$ more independent edges with three crossings and delete them (see Figure 5). Continue the deletion process by still removing the edge with the most crossings until this edge no longer has three or more crossings; we denote the number of these deleted edges by m_{3-} . Call the achieved drawing D_2 . By applying the linear bound from [11] again, we have

$$\begin{aligned} \text{cr}(G) &\geq [5m_{5+} + 4m_4 + 3m_3] + [2 \cdot 3m_3 + 3m_{3-}] + \left[\frac{7}{3}(5(n-2) - 2m_3 - m_{3-}) - \frac{25}{3}(n-2) \right] \\ &= 5m_{5+} + 4m_4 + \frac{13}{3}m_3 + \frac{2}{3}m_{3-} + \frac{10}{3}(n-2). \end{aligned} \quad (3)$$

As all values are non-negative, it is not hard to see that this is at least

$$\geq 4(m_{5+} + m_4 + m_3 + 5(n-2)) - \frac{50}{3}(n-2) = 4m - \frac{50}{3}(n-2).$$

For the better bound of $\text{cr}(G) \geq \frac{73}{18}m - \frac{305}{18}(n-2)$ we have to elaborate on the value m_4 , as there was no slack in the last inequality.

As a preparation, we first consider the structure of D_2 . Let c_{pent} be the number of F_5^2 configurations and c_{hex} the number of F_6^2 configurations in D_2 . Let further E_0 be the set of crossing-free edges on the boundary of the forbidden configurations in D_3 resp. D_2 that do not exist in D_2 , and therefore may be added. We denote $|E_0| = m_0$ and state the following; the proof is in Appendix B.

► **Proposition 21.** *With the notation above, $c_{pent} + c_{hex} \geq \frac{2}{3}(n-2) - \frac{4}{3}m_3 - m_{3-} + m_0$.*

Next, we show how to limit the number of the edges of E_4 , i.e., the deleted edges that were accounted with four crossings in D_4 . For that, we introduce a triangulation \mathcal{H} on the set of the n vertices of D_4 that contains (1) the boundary of every F_6^3 configuration in D_3 , (2) the boundary of every F_5^2 and F_6^2 configuration in D_2 , (3) every edge in E_4 that lies completely outside of these forbidden configurations. This definition is refined in the proof of the next proposition. Note that it is always possible to achieve such a triangulation \mathcal{H} , because edges of E_4 cannot cross each other.

► **Proposition 22.** *Let $\mathcal{H}' \subseteq \mathcal{H}$ be the set of triangles that do not belong to the forbidden configurations and let $c_\Delta = |\mathcal{H}'|$. Then $m_4 \leq m_3 + c_{hex} + 4m_0 + 4c_\Delta$.*

The proof can be found in Appendix B. Combining the results, we can finish the first part of the proof. Proposition 21 implies

$$c_\Delta \leq 2(n-2) - 4m_3 - \left[\frac{2}{3}(n-2) - \frac{4}{3}m_3 - m_{3-} + m_0 \right] \cdot 3 - c_{hex} = 3m_{3-} - 3m_0 - c_{hex},$$

because the total number of triangles is $2(n-2)$ and a pentagon resp. hexagon contains three resp. four triangles. Together with Proposition 22, this gives

$$m_4 \leq m_3 + c_{hex} + 4m_0 + 4(3m_{3-} - 3m_0 - c_{hex}) \leq 5m_3 + 12m_{3-}.$$

Multiplying this term by $\frac{1}{18}$ and adding it to Equation (3), we get as desired

$$\begin{aligned} \text{cr}(G) &\geq 5m_{5+} + 4m_4 + \frac{13}{3}m_3 + \frac{2}{3}m_{3-} + \frac{10}{3}(n-2) + \frac{m_4 - (5m_3 + 12m_{3-})}{18} \\ &\geq \frac{73}{18}(m_{5+} + m_4 + m_3) + \frac{10}{3}(n-2) \\ &= \frac{73}{18}(m_{5+} + m_4 + m_3 + 5(n-2)) - \frac{305}{18}(n-2). \end{aligned}$$

For the bound in (b) see the following: If $m \leq 6(n-2)$, then we can apply the bound of (a). So let be $m > 6(n-2)$. Iteratively delete the edge with the most crossings in a crossing-minimal drawing D until $6(n-2)$ edges are left; these edges have at least five crossings, as the density of 4-planar graphs is $\leq 6(n-2)$ [1]. With the bound in (a), this implies

$$\text{cr}(G) \geq 5(m - 6(n-2)) + \frac{73}{18} \cdot 6(n-2) - \frac{305}{18}(n-2) = 5m - \frac{407}{18}(n-2). \quad \blacktriangleleft$$

6 Discussion

We have improved the leading constant of the lower bound for the crossing number of a given graph G . Although this improvement does not seem to be too impressive at first sight, we worked out some interesting observations for drawings with a limited number of crossings per edge. This leads to further improvements, conjectures and suggestions for future research.

In particular, we have improved for $m > 5(n-2)$ the lower bound of the crossing number, unfortunately we did not reach tightness. We confirm the conjecture by [11] that $\text{cr}(G) \geq \frac{25}{6}m - \frac{35}{2}(n-2)$ holds and highlight that this bound would follow from our proofs, if we were able to show a slightly stronger statement in Proposition 22, namely $m_4 \leq m_3 + c_{hex} + 4m_0 + \frac{4}{3}c_\Delta$. The corresponding upper bound can be obtained by a construction where the plane subgraph consists only of pentagonal and hexagonal faces [11].

Applying our technique to 4-planar drawings might show that these drawings without full hexagons F_6^4 have density $\leq 5.5(n-2)$. This would provide a characterization of optimal 4-planar graphs, which is a well-known open problem. Further, we can look at 5-planar graphs, a class that has been considered as too complex for actual research. Just applying Corollary 8 improves the current known density bound from $8.52n$ to $8.32n$.

It seems to be worthwhile to apply the idea to bipartite graphs to obtain improvements of the Crossing Lemma. Here, the corresponding linear bound $\text{cr}(G) \geq 3m - \frac{17}{2}n + 19$ used in the current proof in [5] is not tight.

Furthermore, we have indicated a way how to obtain the exact density bound of optimal simple 3-planar graphs. Note that we only did one step in this direction.

References

- 1 Eyal Ackerman. On topological graphs with at most four crossings per edge. *Comput. Geom.*, 85, 2019. doi:10.1016/J.COMGEO.2019.101574.
- 2 Eyal Ackerman and Gábor Tardos. On the maximum number of edges in quasi-planar graphs. *J. Comb. Theory, Ser. A*, 114(3):563–571, 2007. doi:10.1016/J.JCTA.2006.08.002.
- 3 Martin Aigner and Günter M. Ziegler. *Proofs from THE BOOK (3. ed.)*. Springer, 2004.
- 4 Miklós Ajtai, Vašek Chvátal, Monroe M Newborn, and Endre Szemerédi. Crossing-free subgraphs. In *North-Holland Mathematics Studies*, volume 60, pages 9–12. Elsevier, 1982.
- 5 Patrizio Angelini, Michael A. Bekos, Michael Kaufmann, Maximilian Pfister, and Torsten Ueckerdt. Beyond-planarity: Density results for bipartite graphs. *CoRR*, abs/1712.09855, 2017. arXiv:1712.09855.
- 6 Sang Won Bae, Jean-François Baffier, Jinhee Chun, Peter Eades, Kord Eickmeyer, Luca Grilli, Seok-Hee Hong, Matias Korman, Fabrizio Montecchiani, Ignaz Rutter, and Csaba D. Tóth. Gap-planar graphs. *Theor. Comput. Sci.*, 745:36–52, 2018. doi:10.1016/J.TCS.2018.05.029.
- 7 Michael A. Bekos, Michael Kaufmann, and Chrysanthi N. Raftopoulou. On optimal 2- and 3-planar graphs. In Boris Aronov and Matthew J. Katz, editors, *33rd International Symposium on Computational Geometry, SoCG 2017, July 4-7, 2017, Brisbane, Australia*, volume 77 of *LIPICs*, pages 16:1–16:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPICs.SOCG.2017.16.
- 8 Carla Binucci, Aaron Büngener, Giuseppe Di Battista, Walter Didimo, Vida Dujmovic, Seok-Hee Hong, Michael Kaufmann, Giuseppe Liotta, Pat Morin, and Alessandra Tappini. Min-k-planar drawings of graphs. In Michael A. Bekos and Markus Chimani, editors, *Graph Drawing and Network Visualization - 31st International Symposium, GD 2023, Palermo, Italy, September 20-22, 2023, Revised Selected Papers, Part I*, volume 14465 of *Lecture Notes in Computer Science*, pages 39–52. Springer, 2023. doi:10.1007/978-3-031-49272-3_3.
- 9 Michael Kaufmann, János Pach, Géza Tóth, and Torsten Ueckerdt. The number of crossings in multigraphs with no empty lens. *J. Graph Algorithms Appl.*, 25(1):383–396, 2021. doi:10.7155/JGAA.00563.
- 10 Frank Thomson Leighton. *Complexity issues in VLSI: optimal layouts for the shuffle-exchange graph and other networks*. MIT press, 1983.
- 11 János Pach, Rados Radoičić, Gábor Tardos, and Géza Tóth. Improving the crossing lemma by finding more crossings in sparse graphs. *Discret. Comput. Geom.*, 36(4):527–552, 2006. doi:10.1007/S00454-006-1264-9.
- 12 János Pach, Joel Spencer, and Géza Tóth. New bounds on crossing numbers. *Discret. Comput. Geom.*, 24(4):623–644, 2000. doi:10.1007/S004540010011.
- 13 János Pach and Géza Tóth. Graphs drawn with few crossings per edge. *Comb.*, 17(3):427–439, 1997. doi:10.1007/BF01215922.
- 14 János Pach and Géza Tóth. A crossing lemma for multigraphs. *Discret. Comput. Geom.*, 63(4):918–933, 2020. doi:10.1007/S00454-018-00052-Z.
- 15 Radoš Radoičić and Géza Tóth. The discharging method in combinatorial geometry and the pach–sharir conjecture. In *Surveys on Discrete and Computational Geometry: Twenty Years Later: AMS-IMS-SIAM Joint Summer Research Conference, June 18-22, 2006, Snowbird, Utah*, volume 453, page 319. American Mathematical Soc., 2008.
- 16 Marcus Schaefer. The graph crossing number and its variants: A survey. *The electronic journal of combinatorics*, pages DS21–Apr, 2012.
- 17 László A. Székely. Crossing numbers and hard erdős problems in discrete geometry. *Comb. Probab. Comput.*, 6(3):353–358, 1997. URL: <http://journals.cambridge.org/action/displayAbstract?aid=46513>.
- 18 R Von Bodendiek, Heinz Schumacher, and Klaus Wagner. Bemerkungen zu einem sechsfarbenproblem von g. ringel. In *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, volume 53, pages 41–52. Springer, 1983.

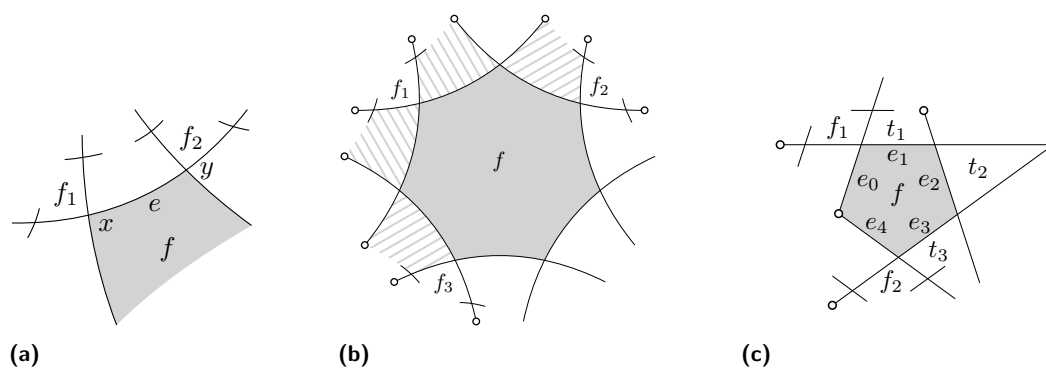


Figure 6 (a) No face contributes to two consecutive vertex-neighbors in Step 6. (b) If a 0-heptagon f contributes to three vertex-neighbors f_1, f_2, f_3 in Step 6, then it contributes not to all its seven wedge-neighbors in Step 1-5. (c) A 1-pentagon f contributing to all three wedge-neighbors in Step 1-5 and to two vertex-neighbors in Step 6 leads to a contradiction.

A Details for Section 4

► **Proposition 17.** *In Step 5, each 0-pentagon receives 0.3 charge from all wedge-neighbors that are not 1-triangles, 0-triangles or 0-pentagons.*

Proof. Note that in the calculations of Proposition 16, we assumed for all faces that are possibly a wedge-neighbor of a 0-pentagon except 0-triangles, 1-triangles, 1-quadrilaterals and 0-pentagons that they give at least 0.3 charge to all wedge-neighbors. If such a face f has a wedge-neighbor that is a 0-pentagon, then it did not contribute charge to it in Step 1-4, and therefore has 0.3 charge left for it in Step 5. The only critical case is a 1-quadrilateral f with a 0-pentagon and a 0-triangle f' as wedge-neighbors. Observe that in this case $\text{ch}_1(f') = 0$ already, because there is a 2-quadrilateral next to f' as in Figure 4c, and therefore $\text{ch}_4(f) = 1$. Thus, f can contribute 0.3 charge to the 0-pentagon. ◀

► **Proposition 18.** *In Step 6, each 1-face and 2-face f with $|f| \geq 5$ and each 0-face f with $|f| \geq 7$ contributes at least 0.4 charge to the vertex-neighbors that are 0-quadrilaterals or 0-pentagons.*

Proof. Let e be a 0-edge of a face f incident to the crossings x and y . Let f_1 be the vertex-neighbor of f at x and f_2 the vertex-neighbor at y . If f_1 and f_2 are 0-faces, then \bar{e} has more than three crossings, a contradiction. Therefore, no face can contribute charge through two consecutive crossings on its boundary in Step 6. For a face f , this implies that it can contribute to at most $\lfloor \frac{|f|}{2} \rfloor$ vertex-neighbors in this step.

Now we distinguish different cases for the face f that might contribute to vertex-neighbors. We start with the case that f is a 0-face. Here, after Step 5, f has an excess of

$$\text{ch}_5(f) \geq |f| - 4 - |f| \cdot 0.3 = 0.7|f| - 4,$$

which is at least $0.4 \cdot \lfloor \frac{|f|}{2} \rfloor$ for $|f| \geq 8$ and therefore enough. If f is a 0-heptagon, then, by the inequality above, there is enough charge if f contributes to at most two vertex-neighbors in Step 6. So assume that it contributes to three vertex-neighbors. It is not hard to see that there are wedge-neighbors of f with $|f| \geq 4$ and $|V(f)| \geq 1$ (see Figure 6a), so f contributes in Step 1-5 to at most six faces. Now we have $\text{ch}_5(f) \geq 3 - 6 \cdot 0.3 = 1.2$, which is sufficient to give three vertex-neighbors 0.4 charge each in Step 6.

29:16 Improving the Crossing Lemma

Let f now be a 1-face. Then f has an excess of

$$\text{ch}_5(f) - 0.4 \geq |f| + 1 - 4 - (|f| - 2) \cdot \frac{1}{3} - 0.4 = \frac{2}{3}|f| - 3.4 + \frac{2}{3},$$

which is at least $0.4 \cdot \lfloor \frac{|f|}{2} \rfloor$ if $|f| \geq 6$. If f is a 1-pentagon, then, by the inequality above, there is enough charge if f contributes to at most one vertex-neighbor in Step 6. So assume the opposite, i.e., f contributes to two vertex-neighbors f_1, f_2 in Step 6. If f contributes in Step 1-5 to only one or two wedge-neighbors, then its excess after Step 5 is at least $1.6 - \frac{2}{3}$ charge and therefore sufficient, so assume this is not the case either.

Walking along the boundary of f , let e_0 be a 1-edge of f , let e_1, e_2, e_3 be the 0-edges and let e_4 be the other 1-edge of f . Let further t_i be the wedge-neighbor of f at e_i for $i \in \{1, 2, 3\}$. W.l.o.g. f_1 lies at the crossing of e_0 , and therefore the face t_2 is a 1-triangle, as otherwise f would not contribute charge to t_2 in Step 1-5 (see Figure 6b). Therefore, f_2 lies at the crossing of e_4 . Note that \bar{e}_2 ends at t_1 or t_3 , say w.l.o.g. at t_1 . But then $|t_1| \geq 4$ and $|V(t_1)| \geq 1$, so f does not contribute charge to t_1 in Step 1-5, a contradiction to our assumption. Therefore, 1-pentagons can contribute 0.4 charge to the desired vertex-neighbors.

The last case is that f is a 2-face. Then f has an excess of

$$\text{ch}_5(f) - 0.8 \geq |f| + 2 - 4 - (|f| - 3) \cdot \frac{1}{3} - 0.8 \geq \frac{2}{4}|f| - 1.8,$$

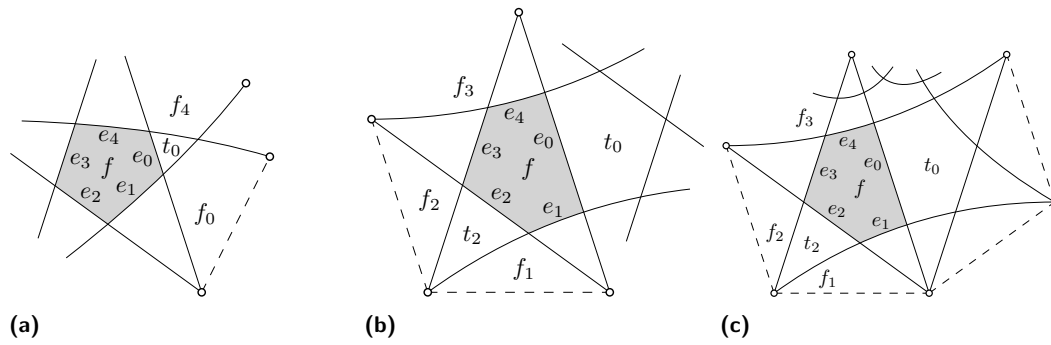
which is at least $0.4 \cdot \lfloor \frac{|f|}{2} \rfloor$ for $|f| \geq 5$. ◀

► **Proposition 19.** *After Step 7, all 0-pentagons that are the wedge-neighbor of four 1-triangles are satisfied.*

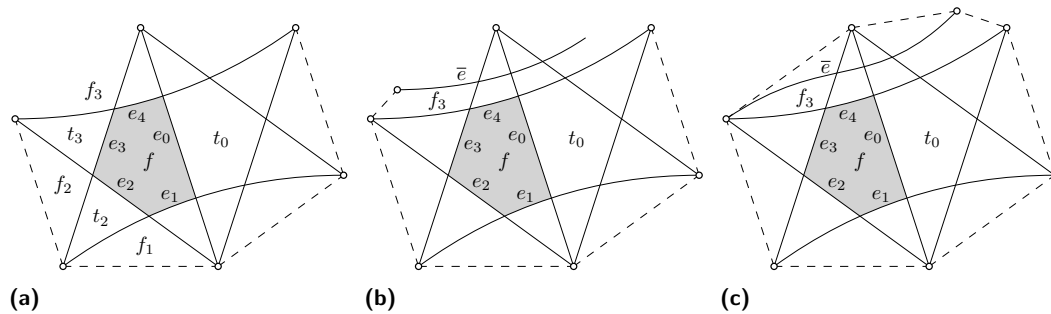
Proof. We introduce a notation for the edges and faces at a 0-pentagon f . Let $e_i, i \in \{0, \dots, 4\}$ be the edges forming the boundary of f , so that e_i and $e_{(i+1 \bmod 5)}$ have a crossing at f . Further we denote by t_i the wedge-neighbor of f at e_i and by f_i the vertex-neighbor of f at the crossing of e_i and $e_{(i+1 \bmod 5)}$.

Let f be a 0-pentagon with four wedge-neighbors that are 1-triangles. So we have $\text{ch}_4(f) \geq 1 - 4 \cdot 0.4 = -0.2$. Let w.l.o.g. t_0 be the wedge-neighbor of f that is not a 1-triangle. If t_0 is not a 0-triangle or 0-pentagon, then it contributes, by Proposition 17, 0.3 charge to f in Step 5 and f is satisfied. Otherwise, distinguish between the type of the face t_0 .

- *Case 1: t_0 is a 0-triangle.* Observe that \bar{e}_1 and \bar{e}_4 already have three crossings and \bar{e}_0 two crossings. Therefore, \bar{e}_0 ends at f_0 or f_4 , say w.l.o.g. f_0 , so f_0 is a 2-quadrilateral (see Figure 7a). Then f_0 has an excess of 0.2 after Step 5, as it only contributes in Step 1 charge. Note that the vertex-neighbors of f_0 are f and f_4 . Since f_4 is not a 0-face, f_0 contributes its excess of 0.2 charge in Step 6 only to f , and therefore $\text{ch}_6(f) \geq 0$.
- *Case 2: t_0 is a 0-pentagon.* Again, \bar{e}_1 and \bar{e}_4 already have three crossings and \bar{e}_0 two crossings. Therefore, f_2 is a 2-triangle and also one of f_1 and f_3 , say w.l.o.g. f_1 (see Figure 7b). So we have $\text{ch}_3(t_2) = -0.2$, and therefore $\text{ch}_4(f) \geq 1 - 3 \cdot 0.3 - 0.2 \geq -0.1$. Note that the only face besides f that may receive charge from t_0 in Step 5 is f_4 . Therefore, we distinguish two cases:
 - *Case 2.1: f_4 is a 0-pentagon.* If less than three wedge-neighbors of t_0 are 1-triangles, then $\text{ch}_4(t_0) \geq 0.4$ and f receives enough charge in Step 5. If three wedge-neighbors of t_0 are 1-triangles, then the 1-triangle at the vertex at which \bar{e}_1 ends has -0.2 charge after Step 3, as it lies between two 2-triangles (see Figure 7c). Therefore, we have $\text{ch}_4(t_0) = 1 - 2 \cdot 0.3 - 0.2 = 0.2$ and f can receive a half of it in Step 5, which is enough.



■ **Figure 7** Illustrations for the proof of Proposition 19 Case 1 and 2.1.



■ **Figure 8** Illustrations for the proof of Proposition 19 Case 2.2.

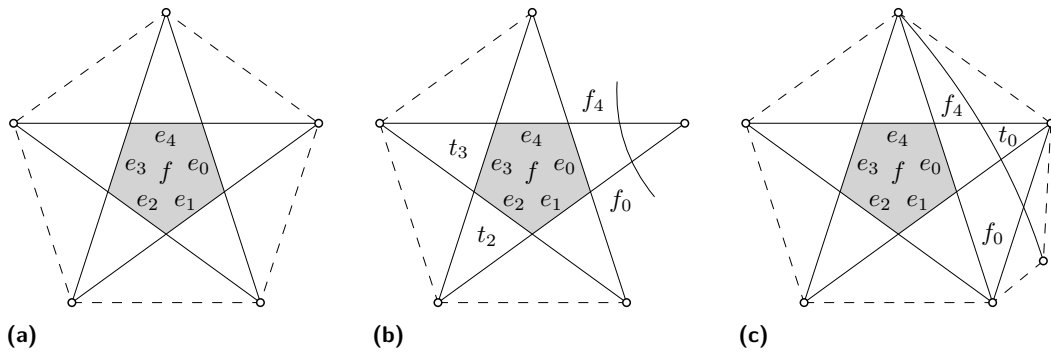
- *Case 2.2: f_4 is not a 0-pentagon.* If $\text{ch}_4(t_0) \geq 0.1$, then f receives its missing charge already in Step 5. So assume the opposite, which implies that four wedge-neighbors of t_0 are 1-triangles (see Figure 8a).

If now f_3 is a 2-triangle, then $\text{ch}_3(t_3) = -0.2$ and we have $\text{ch}(f) = 1 - 2 \cdot 0.3 - 2 \cdot 0.2 = 0$ and f never has a demand. Otherwise, there is an edge \bar{e} crossing \bar{e}_3 at f_3 that has already three crossings, and therefore f_3 is either a 2-quadrilateral or a 1-triangle. In the first case, f_3 has an excess of at least 0.2 and only contributes it to f in Step 6. In the second case, we have a planar cycle of length seven, in which all faces except f are satisfied after Step 6. Here, f receives its demand in Step 7 from a 2-quadrilateral that is a vertex-neighbor of t_0 (Figure 8c). In all cases f is satisfied after Step 6. ◀

► **Proposition 20.** *After Step 7, all 0-pentagons that are the wedge-neighbor of five 1-triangles are satisfied.*

Proof. We continue to use the notation introduced in the proof of Proposition 19. Let f be a 0-pentagon with five wedge-neighbors that are 1-triangles. We distinguish the number of 0-neighbors of f that are 0-quadrilaterals. Note that at most two such faces can exist next to a 0-pentagon.

- *Case 1: No 0-neighbor of f is a 0-quadrilateral.* Then all five vertex-neighbors are 2-triangles (see Figure 9a) and we have $\text{ch}'(f) = \text{ch}_4(f) = 1 - 0.5 \cdot 2 = 0$.
- *Case 2: Exactly one 0-neighbor of f is a 0-quadrilateral.* Assume w.l.o.g. that this 0-quadrilateral lies at e_0 . We consider the wedge-neighbors t_2 and t_3 of f (see Figure 9b). Observe that $\text{ch}_3(t_2) = \text{ch}_3(t_3) = -0.2$, and therefore $\text{ch}_4(f) \geq 1 - 3 \cdot 0.3 - 2 \cdot 0.2 = -0.3$. Note further that f_0 and f_4 cannot be 0-faces, and therefore, by Proposition 18, f receives the missing charge in Step 6 if $|f_0|$ or $|f_4|$ is at least five. The same holds if one of f_0



■ **Figure 9** Illustrations for the proof of Proposition 20 Cases 1 and 2.

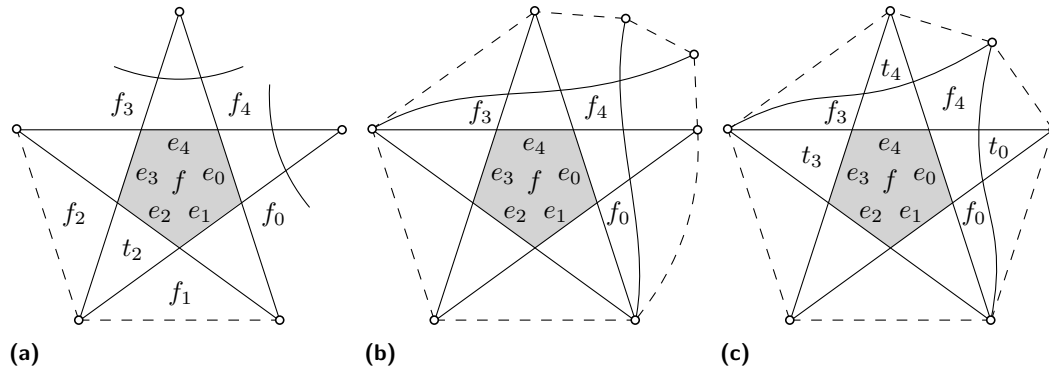
and f_4 is a 2-quadrilateral or both are 1-quadrilaterals, because in this case there is an excess of at least 0.3 charge after Step 5, which is only contributed to f (their other vertex-neighbor is t_0 , which does not receive charge in Step 5, see Figure 9b).

Further f_1 and f_2 cannot be 2-triangles or 3-triangles. If both are 1-triangles, then there would be homotopic multi-edges, which is not allowed. So the last case to consider is when one of them – w.l.o.g. f_4 – is a 1-triangle and the other – therefore f_0 – is a 1-quadrilateral. If f_4 is the only wedge-neighbor, to which f_0 contributes in Step 1-5, then it contributes its excess of 0.3 charge to f in Step 6 and f is satisfied. Otherwise, the second wedge-neighbor of f_0 is also a 1-triangle and we have a planar cycle of length six (see Figure 9c). Here, f_0 contributes 0.1 charge to f in Step 6 and the 1-neighbor of f_0 that is a 2-triangle can contribute its excess of 0.2 to f in Step 7. Therefore, we have $\text{ch}'(f) \geq 0$.

- *Case 3: Exactly two 0-neighbors of f are 0-quadrilaterals.* W.l.o.g. one 0-quadrilateral is at e_0 . If the other 0-quadrilateral would be at e_2 (resp. e_3), then e_1 (resp. e_4) would have four crossings. Therefore, we can assume w.l.o.g. that the second 0-quadrilateral is at e_4 . Here, we have $\text{ch}_3(t_2) = -0.2$ as f_1 and f_2 are 2-triangles, thus $\text{ch}_4(f) \geq 1 - 4 \cdot 0.3 - 0.2 = -0.4$ (see Figure 10a).

We distinguish the type of the vertex-neighbor f_4 . Note that $|f_4| \geq 4$ and f_4 cannot be a 2-quadrilateral. If f_4 is not a 0-quadrilateral, 1-quadrilateral, 0-pentagon or 0-hexagon, then, by Proposition 18, f_4 contributes 0.4 charge to f in Step 6, and therefore f is satisfied. The other cases are more complex, but they all have in common that if one of f_0 and f_3 is a 2-quadrilateral, then it has an excess of at least $2 - 2 \cdot 0.4 - 0.3 = 0.9$ charge after Step 5 and this is enough to ensure $\text{ch}_6(f) \geq 0$.

- *Case 3.1: f_4 is a 0-quadrilateral.* Then the only case to consider is that f_3 and f_4 are 1-triangles. This directly implies a planar cycle of length seven (see Figure 10b). Here, we make use of the second part of Step 6 and have two 2-quadrilaterals contributing 0.9 charge each to the 0-neighbors of f at e_0 and e_4 , which then is moved to f . Therefore, f is satisfied after Step 6.
- *Case 3.2: f_4 is a 1-quadrilateral.* Then t_0 and t_4 receive at least 0.2 charge in Step 3, and therefore we have $\text{ch}_4(f) \geq 1 - 2 \cdot 0.3 - 3 \cdot 0.2 = -0.2$. If now f_4 contributes to less than two 1-triangles in Step 4, f receives from f_4 enough charge in Step 6. Otherwise, f_0 and f_3 are 1-triangles, implying a planar cycle of length six (Figure 10c). Here, t_0 and t_4 have two 1-neighbors that are 2-triangles and $\text{ch}_3(t_0) = \text{ch}_3(t_4) = -0.1$ holds. Therefore, f contributes only $2 \cdot 0.1 + 0.2 + 2 \cdot 0.3 = 1$ charge and f never has a demand.



■ **Figure 10** Illustrations for the proof of Proposition 20 Case 3, 3.1 and 3.2.

- *Case 3.3: f_4 is a 0-pentagon.* We introduce some new notation for f_4 and its wedge-neighbors, likewise for the 0-pentagon f itself: Let $\tilde{f} := f_4$, \tilde{e}_0 the edge-segment of \tilde{e}_4 at \tilde{f} , \tilde{e}_1 the edge-segment of \tilde{e}_0 at \tilde{f} and so on (see Figure 11a). Analogously, we denote by \tilde{t}_i the wedge-neighbor of \tilde{f} at \tilde{e}_i and by \tilde{f}_i the vertex-neighbor at the crossing of \tilde{e}_i and $\tilde{e}_{(i+1) \bmod 5}$. Note that $\tilde{t}_0 = f_0$ and $\tilde{t}_1 = f_3$ are 1-triangles or 2-quadrilaterals and, as pointed out above, we only have to consider the case that both are 1-triangles.

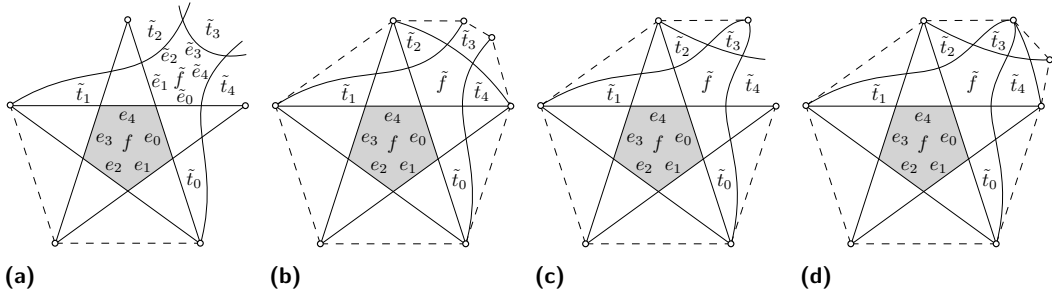
Observe that f is the only vertex-neighbor of f_4 that may receive charge from f_4 in Step 6, as all its other vertex-neighbors cannot be 0-faces. Distinguish the number of 1-triangles that are wedge-neighbors of f_4 . Note that the wedge-neighbors of f_4 can never be 0-triangles or 0-pentagons, so, by Proposition 17, they contribute 0.3 charge to f_4 if they are not 1-triangles. If three or less wedge-neighbors of f_4 are 1-triangles, then $\text{ch}_5(f) \geq 1 - 3 \cdot 0.3 + 2 \cdot 0.3 \geq 0.7$, which then is contributed to f in Step 6 implying $\text{ch}_6(f) \geq 0$. If all five wedge-neighbors of f_4 are 1-triangles, then we have the F_6^3 configuration, which is forbidden. So the case remains that four wedge-neighbors of f_4 are 1-triangles. Here, $\text{ch}_5(f_4) \geq 1 - 4 \cdot 0.3 + 0.3 = 0.1$ holds and this charge is contributed to f in Step 6, so there is only 0.3 charge missing for f .

By symmetry, \tilde{t}_2 is w.l.o.g. a 1-triangle. If \tilde{t}_3 is the wedge-neighbor of \tilde{f} that is not a 1-triangle, then it must be 2-quadrilateral and this implies a planar cycle of length seven, in which f is the only face with a demand after Step 6 (see Figure 11b). The 2-quadrilateral \tilde{t}_3 has an excess of 0.9 charge after Step 6 and contributes it in Step 7 to f . Therefore, f is satisfied.

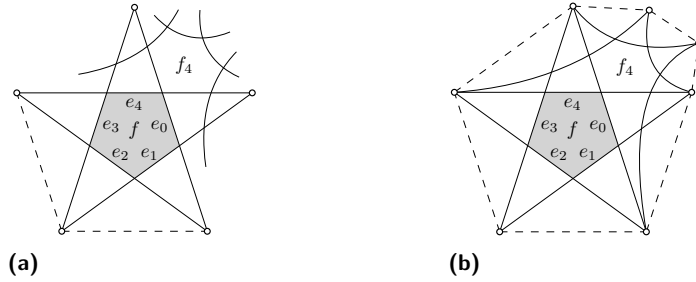
So assume now that \tilde{t}_3 is a 1-triangle and \tilde{t}_4 is the wedge-neighbor that is not a 1-triangle (see Figure 11c). Note that \tilde{t}_4 is not a 0-face. So for all cases, except that \tilde{t}_4 is a 1-quadrilateral or 2-quadrilateral, Proposition 18 guarantees that \tilde{t}_4 contributes in Step 6 0.4 charge to all its vertex-neighbors. In particular, the 0-neighbor of f at e_0 receives 0.4 charge and gives it completely to f . Thus, in this case, f is satisfied.

If \tilde{t}_4 is a 2-quadrilateral, then we have $\text{ch}_5(\tilde{t}_4) = 0.9$ and it contributes in the same way enough charge to f via the 0-neighbor of f at e_0 . This works also if \tilde{t}_4 is a 1-quadrilateral contributing to only one wedge-neighbor (namely \tilde{f}) in Step 1-5.

In the last case where \tilde{t}_4 is a 1-quadrilateral and contributes to \tilde{f} and another wedge-neighbor in Step 1-5, this second wedge-neighbor is \tilde{f}_3 and must be a 1-triangle. This implies a planar cycle of length seven (see Figure 11d). In this case, \tilde{t}_4 and its 1-neighbor that is a 2-triangle have an excess of 0.1 resp. 0.2 charge after Step 5 and contribute it to f in Step 6 and Step 7. Therefore, f is satisfied.



■ **Figure 11** Illustrations for Case 3.3 in the proof of Proposition 20.



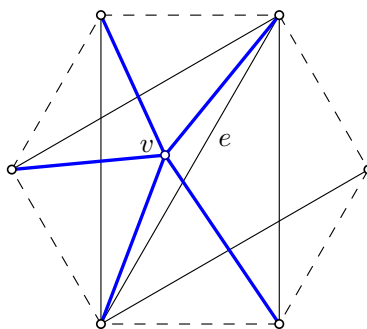
■ **Figure 12** Illustrations for Case 3.4 in the proof of Proposition 20.

- *Case 3.4: f_4 is a 0-hexagon.* Note that no wedge-neighbor of f_4 can be a 0-face, so f_4 contributes no charge in Step 1-3 and Step 5 (see Figure 12a). If at most four wedge-neighbors of f_4 are 1-triangles, then $\text{ch}_5(f_4) \geq 2 - 4 \cdot 0.3 = 0.8$ holds by Proposition 17. In this case, there is at most one other vertex-neighbor of f_4 besides f that can be a 0-quadrilateral or 0-pentagon and f_4 can contribute to both 0.4 charge in Step 6. That is enough to satisfy f .
If five wedge-neighbors of f_4 are 1-triangles, then no vertex-neighbor of f_4 except f is a 0-face. Therefore, f receives the excess of f_4 in Step 6, which is at least $2 - 5 \cdot 0.3 = 0.5$. So again f is satisfied.
Assume now that all six wedge-neighbors of f_4 are 1-triangles (see Figure 12b). Then two of them have a demand of only 0.2 after Step 3 as they have two 1-neighbors that are 2-triangles. Therefore, $\text{ch}_5(f_4) \geq 2 - 4 \cdot 0.3 - 2 \cdot 0.2 = 0.4$. Here, f is the only face to which f_4 contributes in Step 6 and we have $\text{ch}_6(f) \geq 0$. ◀

B Details for Section 5

► **Proposition 21.** *With the notation above, $c_{\text{pent}} + c_{\text{hex}} \geq \frac{2}{3}(n - 2) - \frac{4}{3}m_3 - m_{3-} + m_0$.*

Proof. Insert the m_0 missing planar edges to D_2 at the boundaries of the forbidden configurations. Further add a vertex v and five edges in every F_6^3 configuration from D_3 as shown in Figure 13. More precisely, notice that in D_2 three edges have been deleted from each F_6^3 configuration. Those three edges form a path consisting of a 2-hop edge, a 3-hop edge and a second 2-hop edge. Only one 3-hop edge e still exists and it is crossing-free in D_2 . We arbitrarily choose a side of e and place the new vertex v close to e at this side. We realize the five new edges by connecting v to the two vertices of the configuration that are on same side of e , further to the two endpoints of e , and to one of the two endpoints on the opposite side of e . We do not create new forbidden configurations by this operation, thus the number of F_5^2 and F_6^2 configurations in D_2 does not change.



■ **Figure 13** Illustration for the proof of Proposition 21. We augment each F_6^3 configuration after the deletion of the three blue edges in Figure 5 by one vertex and five edges for the drawing \tilde{D} .

As a next step, we remove one edge from each F_5^2 and F_6^2 configuration in D_2 and call this drawing \tilde{D} . Remark that \tilde{D} is 2-planar, F_5^2 -free, F_6^2 -free and has $5(n-2) - 2m_3 - m_{3-} + m_0 + 5m_3 - (c_{pent} + c_{hex})$ edges on $n + m_3$ vertices.

Assume we have fewer F_5^2 and F_6^2 configurations in D_2 than stated in the proposition. Then \tilde{D} would have more than

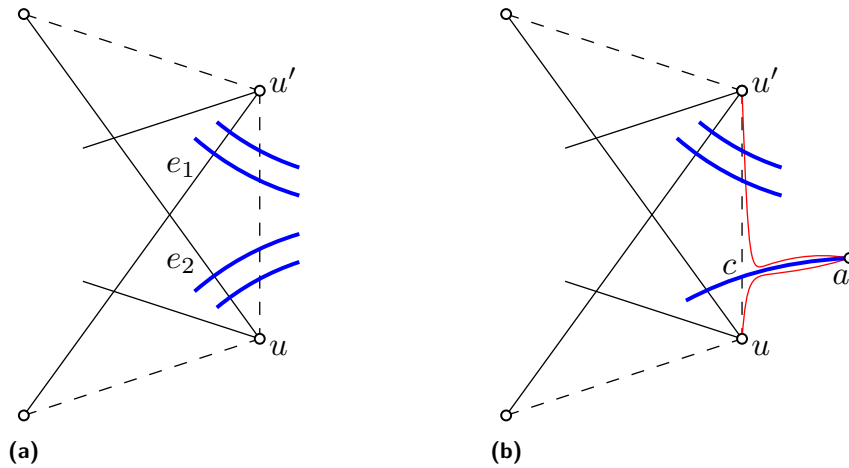
$$5(n-2) - 2m_3 - m_{3-} + m_0 + 5m_3 - \left[\frac{2}{3}(n-2) - \frac{4}{3}m_3 - m_{3-} + m_0 \right] = \frac{13}{3}(n-2 + m_3)$$

edges, which contradicts the statement of Theorem 1 for \tilde{D} . ◀

► **Proposition 22.** Let $\mathcal{H}' \subseteq \mathcal{H}$ be the set of triangles that do not belong to the forbidden configurations and let $c_\Delta = |\mathcal{H}'|$. Then $m_4 \leq m_3 + c_{hex} + 4m_0 + 4c_\Delta$.

Proof. Our strategy is to account for every edge in E_4 an unique F_6^2 or an unique F_6^3 configuration (there are $m_3 + c_{hex}$ of them) (Case 1) or to account for four non-assigned E_4 edges either an edge that might be inserted in D_2 in a planar way (Case 2) or a triangle in \mathcal{H}' (Cases 3 and 4). For the assignment, we use the fact that edges of E_4 do not cross each other and D_4 is 4-planar.

1. $e \in E_4$ lies completely in one of the forbidden configurations. This can only be the case in an F_6^2 or F_6^3 configuration as all five edges of an F_5^2 configuration still exist in D_2 . In each F_6^2 or F_6^3 configuration all 2-hops exist in D_3 . Therefore, e is a 3-hop and crosses the other 3-hops inside the hexagon, which therefore cannot be in E_4 . So e is the only edge in E_4 inside the forbidden configuration and can be assigned to it.
2. $e \in E_4$ starts in a forbidden configuration P and ends in another one, say P' . Let uu' be the edge on the boundary of P that e crosses. We will assign the edge e to uu' and argue that $uu' \in E_0$. Let e_1, e_2 and e'_1, e'_2 resp. be the 2-hop edges of P and P' that enclose the edge uu' (see Figure 14a). Each of these four edges is crossed at least twice by edges belonging to the same forbidden configurations P or P' . Edge e crosses at least two of those four edges. And since those edges must not be crossed more than four times, there are at most four edges of E_4 that will be assigned to the same boundary edge uu' . Note also that $uu' \in E_0$, as otherwise e has at least five crossings (two each in the forbidden configurations and one with uu'). Therefore, at most four edges of E_4 will be assigned to $uu' \in E_0$.
3. $e \in E_4$ is completely outside of any forbidden configuration. By the properties of triangulation \mathcal{H} , e is an edge of a triangle of the triangulation \mathcal{H}' . We assign e to that triangle. By this, at most three such edges of E_4 belong to the same triangle in \mathcal{H}' .



■ **Figure 14** (a) At most four edges of E_4 (blue) can leave a forbidden configuration P through the same edge of its boundary, as otherwise one of the 2-hops e_1, e_2 of P has more than four crossings. (b) For an edge uu' on the boundary of a forbidden configuration P that is crossed by edges in E_4 (blue), the neighboring triangle in \mathcal{H}' is defined by the edges ua and au' (red).

4. $e \in E_4$ lies partially in the faces of \mathcal{H}' and a forbidden configuration. This is the remaining case. Let uu' be a boundary edge of the forbidden configuration P that is crossed by edge $e \in E_4$. To define the triangle t , which is adjacent to the boundary edge uu' , we consider the crossing edge, say $e = ab$ that has the crossing c with uu' that is closest to u . Consider the two segments (a, c) and (c, b) of e , such that (a, c) is completely outside of the forbidden configuration P . We define the edges of the triangle t , which is adjacent to uu' , to be the edge that closely follows the two segments (u, c) and (c, a) ; as the third edge of t , we take the edge that closely follows the two segments (a, c) and (c, u') , see Figure 14b. Note that, by the choice of the triangulation, Case 4 can only occur on one of the edges of the triangle t , here the edge uu' (an edge in E_4 that enters t through another edge and ends at u or u' would cross e , a contradiction to the fact that edges of E_4 do not cross each other). We distinguish two cases:

- uu' is crossed by at most two edges of E_4 . We assign those edges to the triangle t . In the extreme case, we might have two more edges from Case 3 being assigned to t . Thus, not more than four edges are assigned to t in total.
- uu' is crossed more often. As in Case 2, we observe that there are at most four edges crossing uu' . We claim that neither the edge ua nor $u'a$ can be in E_4 . For that, observe that uu' is crossed already at least three times (by assumption) and e at least two times (in the forbidden configuration P). Every edge crossing ua or $u'a$ must cross either uu' or e and they cannot have four additional crossings in total since D_4 is 4-planar.

This implies that it is sufficient to assign the edges in E_4 that cross uu' to the triangle t . ◀

Flips in Colorful Triangulations

Rohan Acharya ✉

Department of Computer Science, University of Warwick, Coventry, UK

Torsten Mütze ✉ 

Institut für Mathematik, Universität Kassel, Germany

Department of Theoretical Computer Science and Mathematical Logic,

Charles University, Prague, Czech Republic

Francesco Verciani ✉

Institut für Mathematik, Universität Kassel, Germany

Abstract

The associahedron is the graph \mathcal{G}_N that has as nodes all triangulations of a convex N -gon, and an edge between any two triangulations that differ in a flip operation. A *flip* removes an edge shared by two triangles and replaces it by the other diagonal of the resulting 4-gon. In this paper, we consider a large collection of induced subgraphs of \mathcal{G}_N obtained by Ramsey-type colorability properties. Specifically, coloring the points of the N -gon red and blue alternatingly, we consider only *colorful* triangulations, namely triangulations in which every triangle has points in both colors, i.e., monochromatic triangles are forbidden. The resulting induced subgraph of \mathcal{G}_N on colorful triangulations is denoted by \mathcal{F}_N . We prove that \mathcal{F}_N has a Hamilton cycle for all $N \geq 8$, resolving a problem raised by Sagan, i.e., all colorful triangulations on N points can be listed so that any two cyclically consecutive triangulations differ in a flip. In fact, we prove that for an arbitrary fixed coloring pattern of the N points with at least 10 changes of color, the resulting subgraph of \mathcal{G}_N on colorful triangulations (for that coloring pattern) admits a Hamilton cycle. We also provide an efficient algorithm for computing a Hamilton path in \mathcal{F}_N that runs in time $\mathcal{O}(1)$ on average per generated node. This algorithm is based on a new and algorithmic construction of a tree rotation Gray code for listing all n -vertex k -ary trees that runs in time $\mathcal{O}(k)$ on average per generated tree.

2012 ACM Subject Classification Mathematics of computing → Combinatorics

Keywords and phrases Flip graph, associahedron, triangulation, binary tree, vertex coloring, Hamilton cycle, Gray code

Digital Object Identifier 10.4230/LIPIcs.GD.2024.30

Related Version *Full Version:* <https://arxiv.org/abs/2406.03783> [1]

Funding This work was supported by Czech Science Foundation grant GA 22-15272S. The second and third author participated in the workshop “Combinatorics, Algorithms and Geometry” in March 2024, which was funded by German Science Foundation grant 522790373.

1 Introduction

The *associahedron* is a polytope of fundamental interest and importance [6, 14, 17], as it lies at the heart of many recent developments in algebraic combinatorics and discrete geometry; see [21] and the references therein. In this paper we are specifically interested in its combinatorial structure, namely the graph of its skeleton; see Figure 1. This graph, which we denote by \mathcal{G}_N , has as nodes all triangulations of a convex N -gon ($N \geq 3$), and an edge between any two triangulations that differ in a *flip* operation, which consists of removing an edge shared by two triangles and replacing it by the other diagonal of the resulting 4-gon. The graph \mathcal{G}_N is isomorphic to the graph that has as nodes all binary trees with $N - 2$ vertices, and an edge between any two trees that differ in a tree rotation. Each binary tree arises as the geometric dual of a triangulation, with the root given by “looking through” a fixed outer edge, and flips translate to tree rotations under this bijection; see Figure 2.



© Rohan Acharya, Torsten Mütze, and Francesco Verciani;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

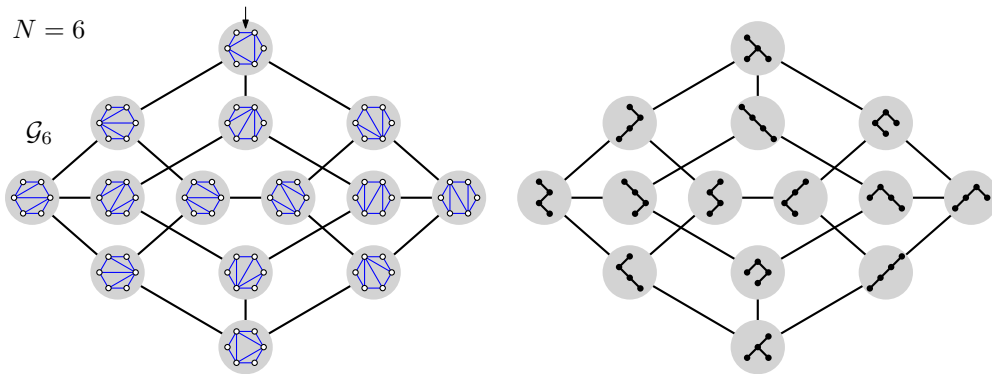
Editors: Stefan Felsner and Karsten Klein; Article No. 30; pp. 30:1–30:20

Leibniz International Proceedings in Informatics

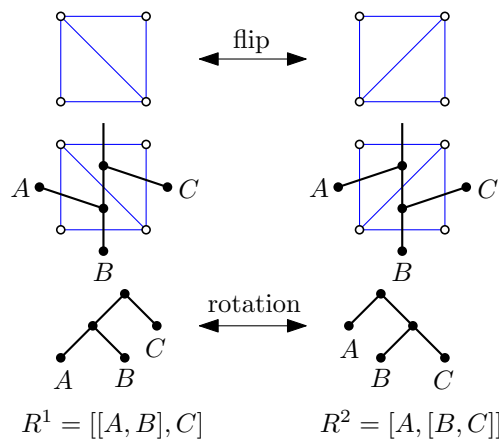


LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

30:2 Flips in Colorful Triangulations



■ **Figure 1** The graph of the 3-dimensional associahedron. The top edge of each triangulation is the outer edge that determines the root of the corresponding binary tree (see little arrow).

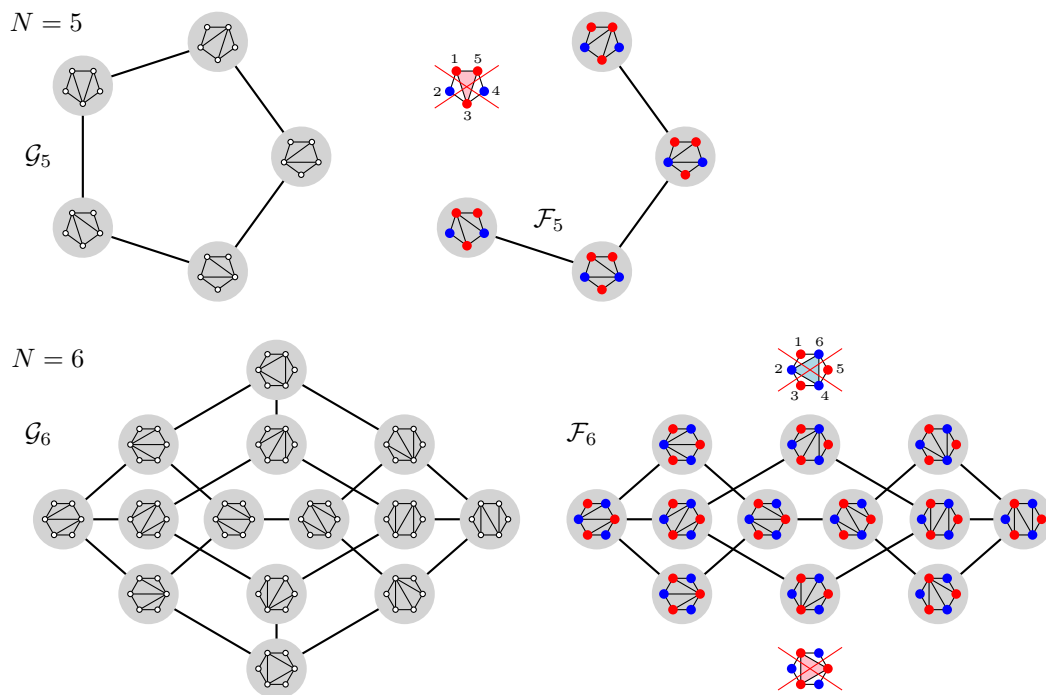


■ **Figure 2** Correspondence between flips in triangulations (top) and rotations in binary trees (bottom).

Properties of the graph \mathcal{G}_N have been the subject of extensive investigations in the literature. Most prominently, the diameter of \mathcal{G}_N was shown to be $2N - 10$ for all $N > 12$ [22, 24]. Furthermore, the graph \mathcal{G}_N is regular with degree $N - 3$, and this number is also its connectivity [16]. The chromatic number of \mathcal{G}_N is at most $\mathcal{O}(\log N)$ [3, 10], while the best known lower bound is only 4.

Another fundamental graph property that we focus on in this paper is Hamiltonicity. To this end, Lucas [18] first proved that \mathcal{G}_N admits a Hamilton cycle for $N \geq 5$, and a short proof was given by Hurtado and Noy [16]. A Hamilton path in \mathcal{G}_N can be computed efficiently and yields a Gray code ordering of all binary trees by rotations [19]. This algorithm is a special case of the more general Hartung-Hoang-Mütze-Williams permutation language framework [12, 13, 20, 5].

In this paper, we consider a large collection of induced subgraphs of \mathcal{G}_N obtained by Ramsey-type colorability properties. This line of inquiry was initiated by Sagan [23], following a sequence of problems posed by Propp on a mailing list in 2003. Specifically, we label the points of the convex N -gon by $1, \dots, N$ in counterclockwise order, and we color them red (**r**) and blue (**b**) alternately. It follows that point i is colored red if i is odd and blue if i is even. For even N , any two neighboring points have opposite colors, whereas for odd N this property is violated for the first and last point, which are both red.



■ **Figure 3** Induced subgraphs of the associahedron \mathcal{G}_5 (top left) and \mathcal{G}_6 (bottom left) obtained for the coloring sequence $\mathbf{rbrb}\dots$ by forbidding monochromatic triangles. The triangulations with monochromatic triangles are still shown, but they are not part of the graphs \mathcal{F}_5 and \mathcal{F}_6 (top right and bottom right, respectively) and hence crossed out.

We say that a triangulation is *colorful* if every triangle has points of both colors, i.e., no triangles in which all three points have the same color. We write \mathcal{F}_N for the subgraph of \mathcal{G}_N induced by all colorful triangulations. In other words, \mathcal{F}_N is obtained from \mathcal{G}_N by deleting all triangulations that have a monochromatic triangle; see Figure 3.

1.1 Sagan’s problem and its generalization

Sagan [23] proved that \mathcal{F}_N is a connected graph, and he asked [personal communication] whether \mathcal{F}_N admits a Hamilton path or cycle. Looking at the first two interesting instances $N = 5$ and $N = 6$ in Figure 3, we note that \mathcal{F}_5 has a Hamilton path, but no cycle, and \mathcal{F}_6 has no Hamilton path and hence no cycle either. Furthermore, \mathcal{F}_7 admits a Hamilton path (see Figure 12), but no Hamilton cycle, which seems rather curious (cf. Theorem 5 below). We prove the following result.

► **Theorem 1.** *For any $N \geq 8$, the graph \mathcal{F}_N has a Hamilton cycle.*

The resolution of Sagan’s question immediately gives rise to the following more general problem: We consider an arbitrary sequence α of coloring the points $1, \dots, N$ red or blue, and let \mathcal{F}_α be the corresponding induced subgraph of \mathcal{G}_N obtained by forbidding monochromatic triangles. For which sequences α does \mathcal{F}_α admit a Hamilton path or cycle?

Formally, a *coloring sequence* is a sequence $\alpha = (\alpha_1, \dots, \alpha_\ell)$ of even length $\ell \geq 2$ with $\alpha_i \geq 1$ for $i = 1, \dots, \ell$, and it encodes the coloring pattern

$$\mathbf{r}^{\alpha_1} \mathbf{b}^{\alpha_2} \mathbf{r}^{\alpha_3} \mathbf{b}^{\alpha_4} \dots \mathbf{r}^{\alpha_{\ell-1}} \mathbf{b}^{\alpha_\ell} \tag{1}$$

30:4 Flips in Colorful Triangulations

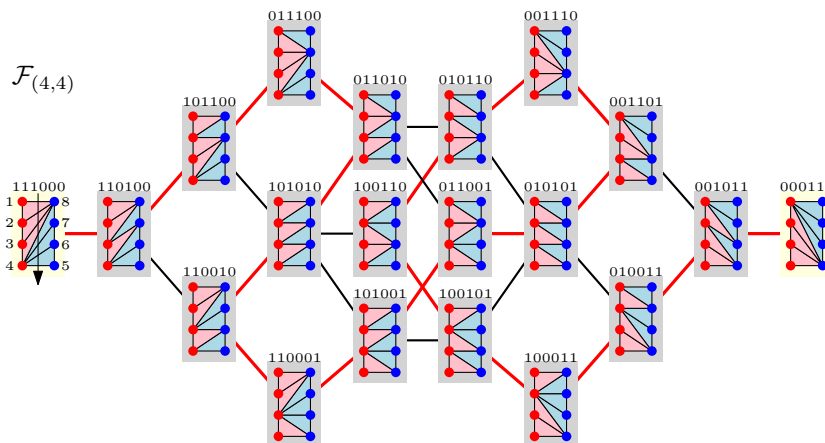
for the points $1, \dots, N$, where $N = \sum_{i=1}^{\ell} \alpha_i$, and \mathbf{r}^{α_i} and \mathbf{b}^{α_j} denote α_i -fold and α_j -fold repetition of red and blue, respectively. In words, the first α_1 many points are colored red, the next α_2 many points are colored blue, the next α_3 many points are colored red etc. Clearly, the special cases considered by Sagan are $\alpha_1 = \alpha_2 = \dots = \alpha_{\ell} = 1$ for even $N = \ell$, or $\alpha_1 = 2$ and $\alpha_2 = \dots = \alpha_{\ell} = 1$ for odd $N = \ell + 1$, respectively (in the second case, the two consecutive points of the same color are 1 and 2 instead of 1 and N as before, but this is only a cyclic shift of indices). We let \mathcal{F}_{α} be the induced subgraph of \mathcal{G}_N induced by the colorful triangulations with coloring sequence α .

We provide the following generalization of Theorem 1 before. Specifically, our next theorem applies to all coloring patterns with at least 10 changes of colors.

► **Theorem 2.** *For any coloring sequence $\alpha = (\alpha_1, \dots, \alpha_{\ell})$ of (even) length $\ell \geq 10$, the graph \mathcal{F}_{α} has a Hamilton cycle.*

Note that there are 2^{N-2} different coloring sequences satisfying the conditions of the theorem, i.e., there are exponentially many subgraphs of the associahedron to which Theorem 2 applies. This also shows that the associahedron has cycles of many different lengths.

In view of the last theorem, it remains to consider short coloring sequences, i.e., sequences of length $\ell \leq 8$. We offer three simple observations in this regime. We first consider the easiest case $\ell = 2$, i.e., the coloring sequence has the form $\alpha = (a, b)$. The resulting graph \mathcal{F}_{α} for $\alpha = (4, 4)$ is shown in Figure 4. Another way to think about such a triangulation is as a triangulation of the so-called double-chain, where each triangle has to touch both chains. We observe that the number of colorful triangulations in this case is $\binom{N-2}{a-1} = \binom{N-2}{b-1}$ where $N := a + b$. Moreover, these triangulations are in bijection with bitstrings of length $N - 2$ with $a - 1$ many 0s and $b - 1$ many 1s, so-called $(a - 1, b - 1)$ -combinations. This bijection is defined as follows; see Figure 4: Given a triangulation, we consider a ray separating the red from the blue points, and we record the types of triangles intersected by this ray one after the other, specifically we record a 1-bit or 0-bit if the majority color of the three triangle points is red and blue, respectively. We see that flips in the triangulations correspond to adjacent transpositions in the corresponding bitstrings. In the following, we use the generic



■ **Figure 4** Flip graph of colorful triangulations for the coloring sequence $\alpha = (4, 4)$ ($\mathbf{rrrrbbbb}$), which is isomorphic to the flip graph of $(3, 3)$ -combinations under adjacent transpositions. The black arrow in the leftmost triangulation is the ray that separates red from blue points, and the combination is obtained by reading the triangle types that intersect this ray from top to bottom (red=1, blue=0). The nodes of degree 1 and a Hamilton path in the flip graph are highlighted.

term *flip graph* for any graph that has as nodes a set of combinatorial objects, and an edge between any two objects that differ in a certain change operation. From what we said before, it follows that $\mathcal{F}_{(a,b)}$ is isomorphic to the flip graph of $(a - 1, b - 1)$ -combinations under adjacent transpositions. Applying known results from [4, 9] thus yields the following theorem.

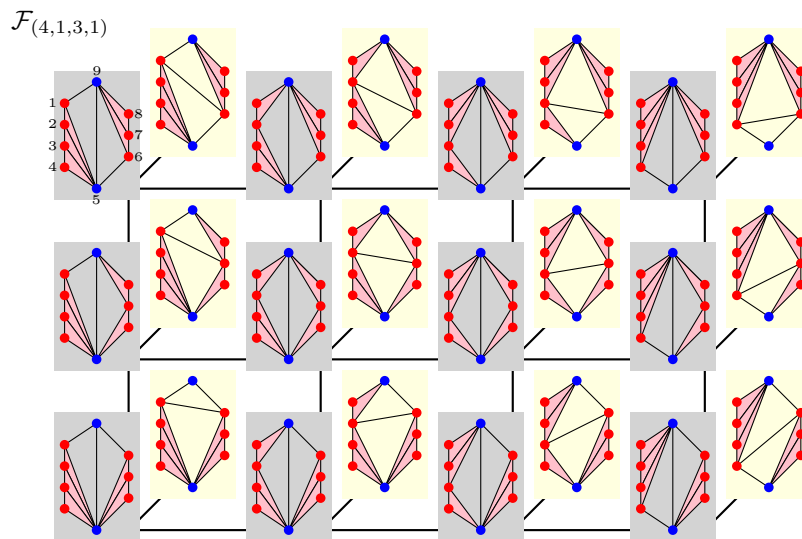
► **Theorem 3.** *For integers $a, b \geq 1$ with $a + b \geq 3$, the graph $\mathcal{F}_{(a,b)}$ is isomorphic to the flip graph of $(a - 1, b - 1)$ -combinations under adjacent transpositions. Consequently, $\mathcal{F}_{(a,b)}$ has a Hamilton path if and only if $a \in \{1, 2\}$, or $b \in \{1, 2\}$, or a and b are both even. Furthermore, if $a, b \geq 2$, then $\mathcal{F}_{(a,b)}$ has no Hamilton cycle.*

The reason for the non-existence of a Hamilton cycle is that $\mathcal{F}_{(a,b)}$ has two nodes of degree 1, corresponding to the combinations $1^{a-1}0^{b-1}$ and $0^{b-1}1^{a-1}$; see Figure 4.

The next result is a simple observation for the special case of coloring sequences of length $\ell = 4$ with exactly two non-consecutive blue points; see Figure 5.

► **Theorem 4.** *For integers $a, b \geq 1$, the graph $\mathcal{F}_{(a,1,b,1)}$ is isomorphic to an $a \times b$ rectangular grid with one pending edge attached to each node. Consequently, it does not have a Hamilton path unless $a \cdot b \leq 2$.*

The nodes of degree 1 are the triangulations in which the two blue points are not connected by an edge, in which case the only possible flip restores this edge between them.



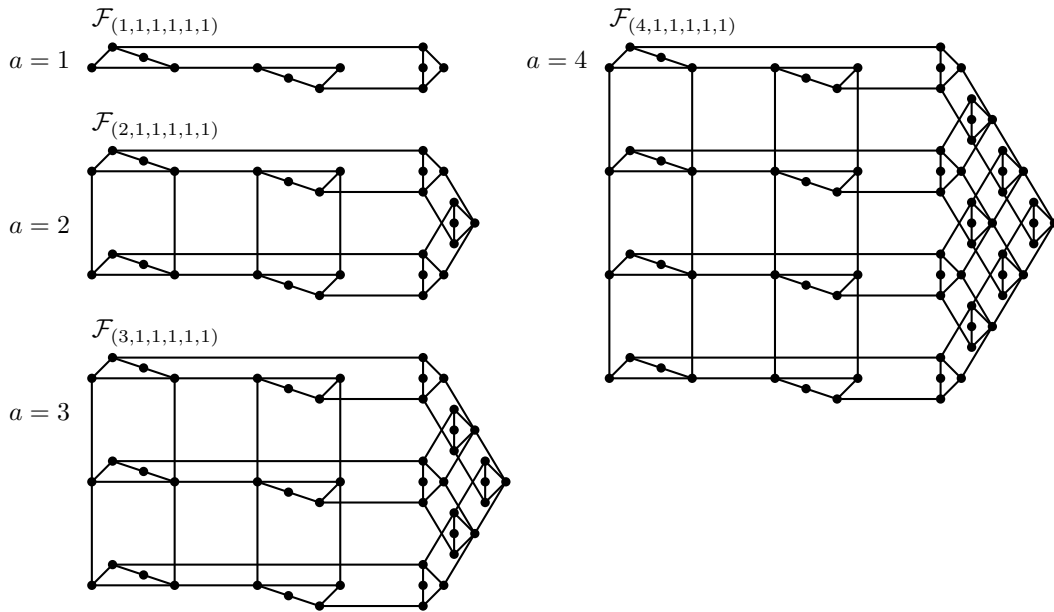
■ **Figure 5** Illustration of Theorem 4 for the coloring sequence $\alpha = (4, 1, 3, 1)$ (rrrrbrrrb). The nodes of degree 1 in the flip graph are highlighted.

The last result is for coloring sequences of length $\ell = 6$ and yields an infinite family of natural flip graphs that admit a Hamilton path but no Hamilton cycle, despite the fact that they have minimum degree 2; see Figure 6.

► **Theorem 5.** *For $\alpha = (a, 1, 1, 1, 1, 1)$, the graph \mathcal{F}_α has no Hamilton cycle if $a \geq 1$, but a Hamilton path unless $a \in \{1, 3\}$.*

1.2 Algorithmic questions and higher arity

We also provide an algorithmic version of Theorem 1.



■ **Figure 6** Illustration of the family of graphs $\mathcal{F}_{(a,1,1,1,1,1)}$.

► **Theorem 6.** *For any $N \geq 8$, a Hamilton path in the graph \mathcal{F}_N can be computed in time $\mathcal{O}(1)$ on average per node.*

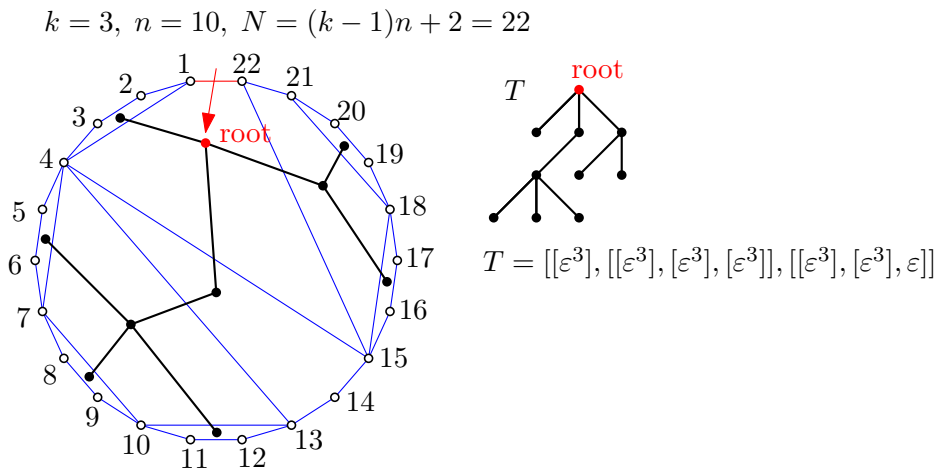
The initialization time and memory requirement for this algorithm are $\mathcal{O}(N)$.

Our construction of a Hamilton path/cycle in \mathcal{F}_N relies on a Gray code ordering of ternary trees by rotations. We first describe this setup, generalizing our earlier definitions about triangulations and binary trees; see Figures 7 and 8 for illustration. Let $k \geq 2$ and $n \geq 1$ be integers, and let $N := (k - 1)n + 2$. We consider a *dissection* of a convex N -gon into n many $(k + 1)$ -gons. A *flip* operation removes an edge shared by two $(k + 1)$ -gons and replaces it by one of the other $k - 1$ possible diagonals of the resulting $2k$ -gon. Dissections of an N -gon into $(k + 1)$ -gons are in bijection with k -ary trees with n vertices. Each k -ary trees arises as the geometric dual of a dissection into $(k + 1)$ -gons, with the root given by “looking through” the outer edge $1N$, and flips translate to tree rotations under this bijection.

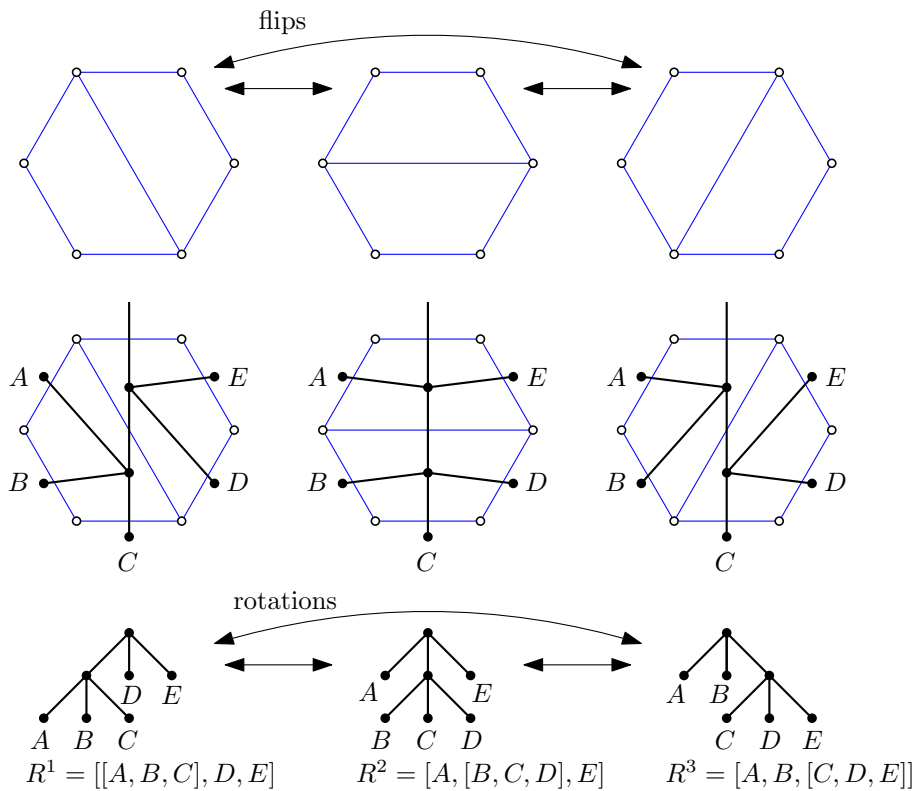
We denote the corresponding flip graph of dissections of an N -gon into $(k + 1)$ -gons by $\mathcal{G}_{N,k+1}$. The associahedron is the special case $k = 2$, i.e., the graph $\mathcal{G}_{N,3} = \mathcal{G}_N$. By what we said before, the graph $\mathcal{G}_{N,k+1}$ is isomorphic to the rotation graph of k -ary trees with n vertices, where $N = (k - 1)n + 2$. Huemer, Hurtado, and Pfeifle [15] first proved that $\mathcal{G}_{N,k+1}$ has a Hamilton cycle for all $k \geq 3$, which combined with the results of Hurtado and Noy [16] for the case $k = 2$ (binary trees) yields the following theorem.

► **Theorem 7** ([16] for $k = 2$; [15] for $k \geq 3$). *For any $k \geq 2$, $n \geq \max\{2, 5 - k\}$ and $N := (k - 1)n + 2$, the graph $\mathcal{G}_{N,k+1}$ has a Hamilton cycle.*

The proof from [15] for the case $k \geq 3$ does not generalize the simple inductive construction of a Hamilton path/cycle in the associahedron (the case $k = 2$) described in [16], and it imposes substantial difficulties when translating it to an efficient algorithm. Consequently, we provide a unified and simplified proof for Theorem 7, valid for all $k \geq 2$, which can be turned into an efficient algorithm. This result generalizes the efficient algorithm for computing a Hamilton path in the associahedron provided by Lucas, Roelants van Baronaigien, and Ruskey [19].



■ **Figure 7** Bijection between dissections of an N -gon into $(k + 1)$ -gons and k -ary trees, illustrated for the case $k = 3$.



■ **Figure 8** Correspondence between flips in quadrangulations (top) and rotations in ternary trees (bottom); cf. Figure 2.

► **Theorem 8.** For any $k \geq 2, n \geq \max\{2, 5 - k\}$ and $N := (k - 1)n + 2$, a Hamilton path in $\mathcal{G}_{N, k+1}$ can be computed in time $\mathcal{O}(k)$ on average per node.

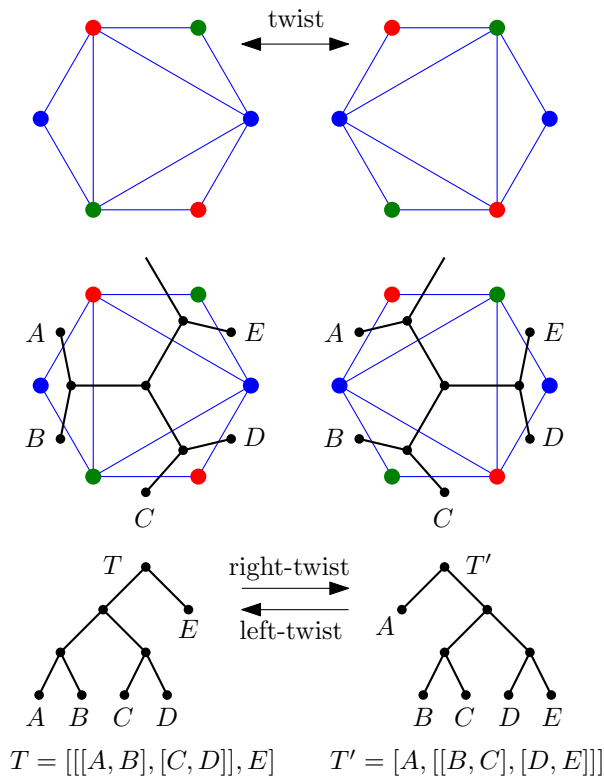
The initialization time and memory requirement for this algorithm are $\mathcal{O}(kn)$.

We implemented the algorithms mentioned in Theorems 6 and 8 in C++, and made the code available for download and experimentation on the Combinatorial Object Server [7].

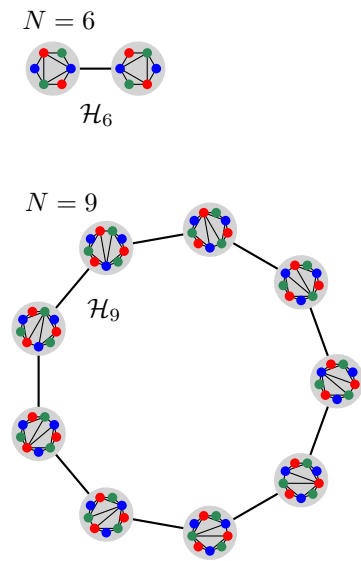
1.3 Three colors

We now consider colorings of the points $1, \dots, N$ with more than two colors. To start with, we color the points in counterclockwise order alternatingly red (\mathfrak{r}), blue (\mathfrak{b}) and green (\mathfrak{g}), and we consider triangulations in which every triangle has points of all three colors, i.e., one point of each color. This setting has also been considered by Sagan [23]. Note that flips of a single diagonal as before are not valid operations anymore (in the sense that the flip graph would not have any edges), so we consider a modified flip operation instead which consists of a particular sequence of 4 flips. Specifically, a *twist* “rotates” a triangle that is surrounded by three triangles, i.e., the inner triangle is removed, creating an empty 6-gon, and the triangle is inserted the other way; see Figure 9. We write \mathcal{H}_N for the flip graph of colorful triangulations under twists; see Figure 10.

► **Theorem 9.** *For any N that is a multiple of 3, the graph \mathcal{H}_N is connected.*



■ **Figure 9** Twist operation and the corresponding binary trees.



■ **Figure 10** Flip graphs \mathcal{H}_6 and \mathcal{H}_9 .

1.4 Outline of this paper

In this extended abstract we focus on proving Theorems 1 and 2. Before providing the proofs in Section 3, we collect a few definitions and auxiliary results in Section 2. The proofs of all other results can be found in the preprint [1]. We conclude with some open questions in Section 4.

2 Preliminaries

2.1 String operations

For any string x and any integer $k \geq 0$, we write x^k for the k -fold concatenation of x . Given any sequence $x = (x_1, \dots, x_\ell)$, we write $\text{rev}(x) := (x_\ell, x_{\ell-1}, \dots, x_1)$ for the reversed sequence.

2.2 Dissections and trees

For integers $k \geq 2$ and $n \geq 1$, let $N := (k-1)n + 2$. We write $\mathcal{D}_{N,k+1}$ for the set of all dissections of a convex N -gon into $(k+1)$ -gons. In particular, $\mathcal{D}_{N,3}$ are triangulations of a convex N -gon. We write $\mathcal{T}_{n,k}$ for the set of all k -ary trees with n vertices, and $t_{n,k} := |\mathcal{T}_{n,k}|$. Both objects are counted by the *k -Catalan numbers* (OEIS sequence A062993), i.e., we have

$$|\mathcal{D}_{N,k+1}| = |\mathcal{T}_{n,k}| = t_{n,k} = \frac{1}{(k-1)n+1} \binom{kn}{n}.$$

We also define $t'_{n,3} := \sum_{i=0}^n t_{i,3} \cdot t_{n-i,3}$ as the number of pairs of ternary trees with n vertices in total (OEIS A006013). We have the explicit formula

$$t'_{n,3} = \frac{1}{n+1} \binom{3n+1}{n}.$$

2.3 Colorful triangulations

For any coloring sequence α , we write \mathcal{C}_α for the set of colorful triangulations with coloring pattern defined in (1). By these definitions, \mathcal{F}_α is the subgraph of \mathcal{G}_N induced by the triangulations in \mathcal{C}_α . Sagan's question concerned the special case $\alpha := 1^N$ for even N and $\alpha := (2, 1^{N-2})$ for odd N , and for those particular coloring sequences α we simply write $\mathcal{C}_N = \mathcal{C}_\alpha$ and $\mathcal{F}_N = \mathcal{F}_\alpha$. Sagan proved the following.

► **Theorem 10** ([23, Thm. 2.1]). *For any $q \geq 1$ we have*

$$|\mathcal{C}_N| = \begin{cases} 2^q \cdot t_{q,3} = \frac{2^q}{2q+1} \binom{3q}{q} & \text{if } N = 2q + 2, \\ 2^q \cdot t'_{q,3} = \frac{2^q}{q+1} \binom{3q+1}{q} & \text{if } N = 2q + 3. \end{cases}$$

The two sequences in this theorem are OEIS A153231 and A369510, respectively.

2.4 Graphs

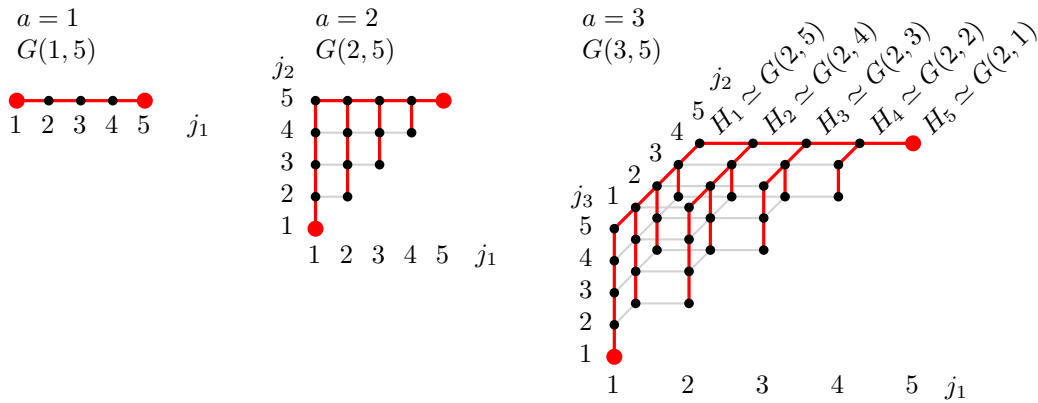
For a graph G , we write $\Delta(G)$ for its maximum degree. Also, we write $G \simeq H$ for two graphs G and H that are isomorphic.

For any integer $d \geq 1$, the *d -dimensional hypercube* Q_d is the graph that has as vertices all bitstrings of length d , and an edge between any two strings that differ in a single bit.

► **Lemma 11** ([8]). *For any $d \geq 2$ and any set E of at most $2d - 3$ edges in Q_d that together form vertex-disjoint paths, there is a Hamilton cycle that contains all edges of E .*

For integers $a \geq 1$ and $d \geq 1$ we define $S(a, d)$ as the set of all a -tuples of non-decreasing integers from the set $\{1, \dots, d\}$, i.e., $S(a, d) = \{(j_1, \dots, j_a) \mid 1 \leq j_1 \leq j_2 \leq \dots \leq j_a \leq d\}$. Furthermore, we let $G(a, d)$ be the graph with vertex set $S(a, d)$ and edges between any two a -tuples that differ in a single entry by ± 1 ; see Figure 11.

30:10 Flips in Colorful Triangulations



■ **Figure 11** Illustration of the graph $G(a, d)$ and Lemma 12. The spanning trees and the extremal vertices are highlighted.

► **Lemma 12.** For any $a \geq 1$ and $d \geq 1$, the graph $G(a, d)$ has a spanning tree T with $\Delta(T) \leq 3$.

We refer to the vertices 1^a and d^a as *extremal* vertices, and note that they have degree 1 in $G(a, d)$, unless $a = d = 1$, in which case the graph is a single vertex having degree 0.

Proof. We argue by induction on a and d . For $a = 1$ and any $d \geq 1$, the graph $G(a, d)$ is the path on d vertices, so the claim is trivially true. For the induction step let $a \geq 2$. We split $G(a, d)$ into subgraphs H_i for $i = 1, \dots, d$ where H_i contains all vertices in which the first coordinate equals i . Note that $H_i \simeq G(a - 1, d - (i - 1))$ for all $i = 1, \dots, d$, in particular $H_d \simeq G(a - 1, 1)$ is a single vertex. By induction, H_i has a spanning tree T_i with $\Delta(T_i) \leq 3$ for all $i = 1, \dots, d$. Furthermore, the two extremal vertices have degree 1 in T_i for $i = 1, \dots, d - 1$ and degree 0 in T_d . We join the trees T_i to a single spanning tree T of $G(a, d)$ by adding the edges $((i, d^{a-1}), (i + 1, d^{a-1}))$ for $i = 1, \dots, d - 1$ between their extremal vertices. ◀

3 Colorful triangulations

In this section we consider the setting of colorful triangulations introduced by Sagan, with the goal of proving Theorems 1 and 2. ri

3.1 Alternating colors

We first assume that the number N of points is even and the coloring sequence is $\alpha = 1^N$, i.e., the coloring pattern along the points $1, \dots, N$ is $\mathbf{rbrb} \cdots \mathbf{rb} = (\mathbf{rb})^{N/2}$. Recall that \mathcal{C}_N denotes the set of all colorful triangulations with this coloring sequence.

Let $T \in \mathcal{C}_N$ be a colorful triangulation. We say that an edge of T is *monochromatic* if both endpoints have the same color, and we say that it is *colorful* if both endpoints have distinct colors. We observe the following:

- (i) Every triangle of T has exactly one monochromatic edge.
- (ii) Every monochromatic edge of T is an inner edge.

Consequently, if we remove from T all monochromatic edges, keeping only the colorful ones, then the resulting dissection $r(T)$ is a quadrangulation on the point set. Indeed, by (i) every triangle is destroyed, and by (i)+(ii) destroying a triangle creates a quadrangle. While

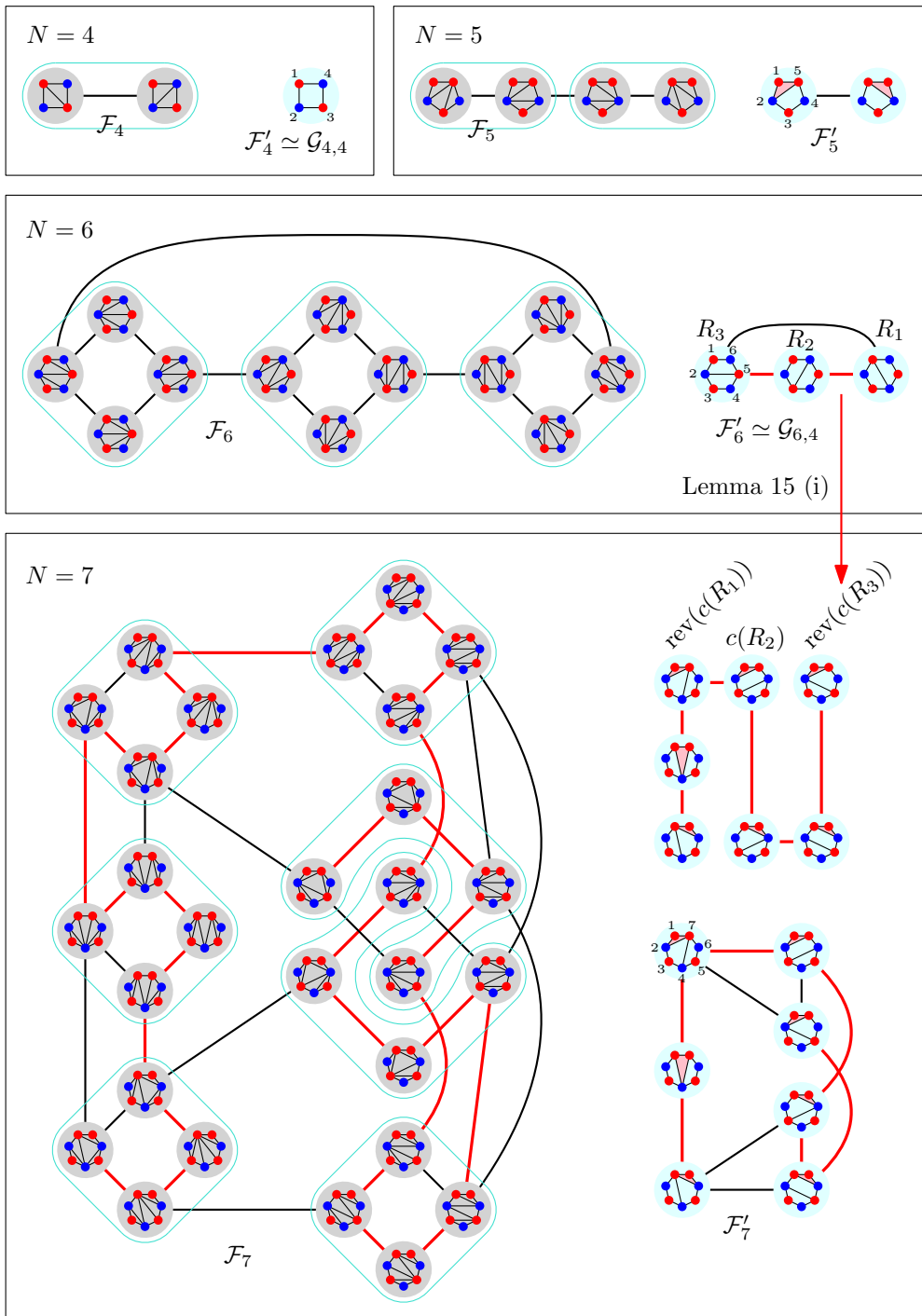
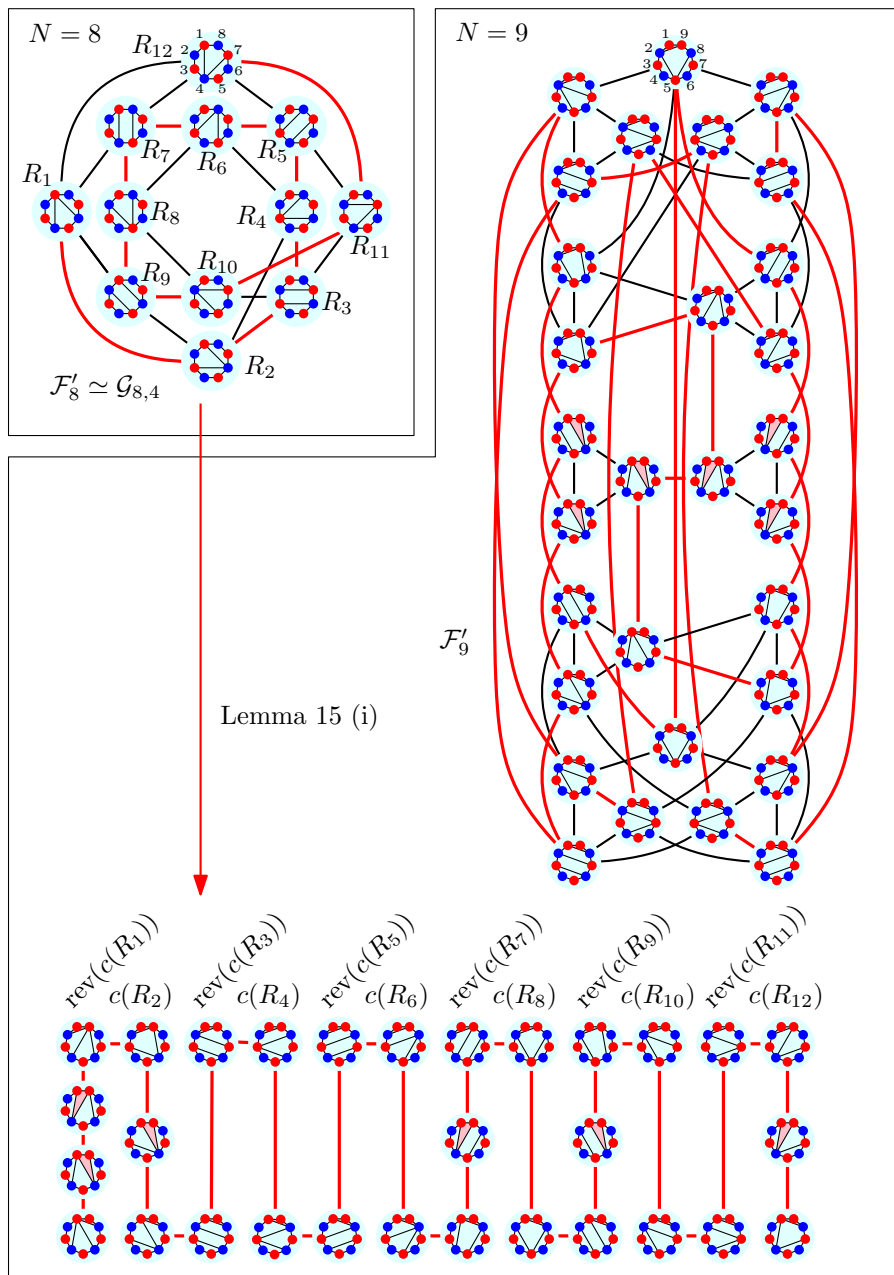


Figure 12 Flip graphs of colorful triangulations and reduced graphs for $N = 4, 5, 6, 7$.

T has $N - 2 = n$ triangles, $r(T)$ has $q := (N - 2)/2 = n/2$ quadrangles. Furthermore, there are 2^q many colorful triangulations that yield the same quadrangulation $r(T)$ by removing monochromatic edges. They are obtained from $r(T)$ by placing a diagonal in each of the q quadrangles in one of the two ways. Note that the subgraph of \mathcal{F}_N induced by

30:12 Flips in Colorful Triangulations



■ **Figure 13** Reduced graphs of colorful triangulations for $N = 8, 9$.

those 2^q triangulations is isomorphic to the q -dimensional hypercube Q_q , as each of the q monochromatic edges in T can be flipped independently from the others. We thus obtain a partition of \mathcal{F}_N into hypercubes Q_q , plus edges between them. These copies of hypercubes are highlighted by blue bubbles in Figure 12.

We also note that every quadrangulation R on N points equals $r(T)$ for some colorful triangulation $T \in \mathcal{C}_N$. Indeed, given R , then coloring the N points red and blue alternately will make all edges colorful. We define a reduced graph \mathcal{F}'_N , that has as nodes all quadrangulations on N points, and for any two colorful triangulations T and T' that differ

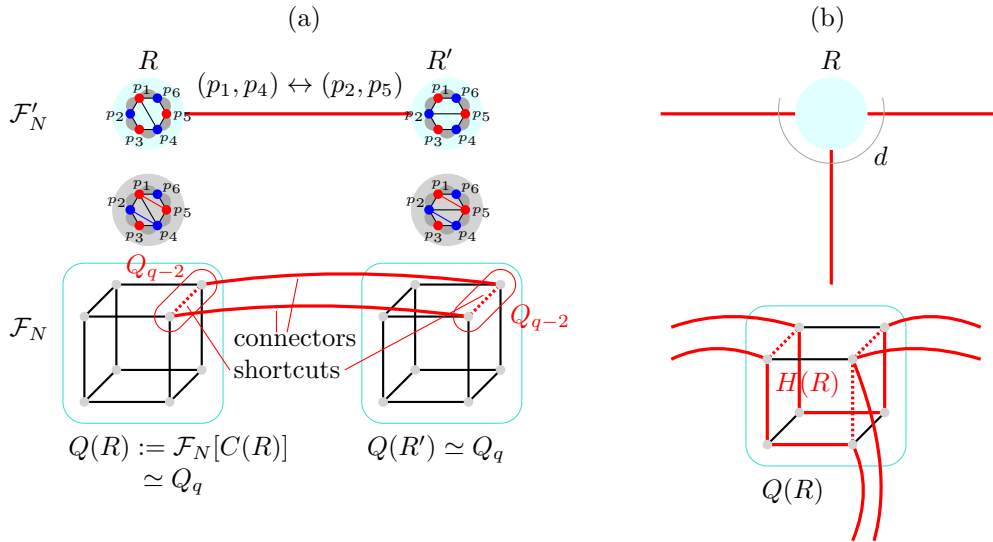
in a flip of a colorful edge, we add an edge between $r(T)$ and $r(T')$ in \mathcal{F}'_N ; see Figures 12 and 13. We observe that \mathcal{F}'_N is isomorphic to the flip graph of quadrangulations $\mathcal{G}_{N,4}$, i.e., we have $\mathcal{F}'_N \simeq \mathcal{G}_{N,4}$. These arguments yield a direct combinatorial proof for the first equality in Theorem 10.

- **Lemma 13.** *Let $q \geq 3$ be an integer and $N := 2q + 2$, and let \mathcal{S} be a spanning tree of \mathcal{F}'_N .*
- (i) *If $q = 3$ and $\Delta(\mathcal{S}) \leq 2$ (i.e., \mathcal{S} is a Hamilton path), then \mathcal{F}_N has a Hamilton cycle.*
 - (ii) *If $q \geq 4$ and $\Delta(\mathcal{S}) \leq 3$, then \mathcal{F}_N has a Hamilton cycle.*

The conclusions of the lemma do not hold when $q = 2$: Indeed, while \mathcal{F}'_6 has a Hamilton path (the graph is a triangle), there is no Hamilton path or cycle in \mathcal{F}_6 ; see Figure 12.

Proof. The idea is to “uncompress” the spanning tree \mathcal{S} in \mathcal{F}'_N to a Hamilton cycle in \mathcal{F}_N . Specifically, for any quadrangulation R we consider the 2^q colorful triangulations $C(R) := \{T \in \mathcal{C}_N \mid r(T) = R\}$, and we let $Q(R)$ denote the subgraph of \mathcal{F}_N spanned by the triangulations in $C(R)$. Recall that $Q(R) \simeq Q_q$, i.e., $Q(R)$ is isomorphic to the q -dimensional hypercube. In the first step of the uncompression, we replace each edge (R, R') of \mathcal{S} by two edges $(T_1, T'_1), (T_2, T'_2)$ with $T_1, T_2 \in C(R)$ and $T'_1, T'_2 \in C(R')$. We refer to the edges (T_1, T'_1) and (T_2, T'_2) as *connectors*, and to their end nodes T_1, T_2, T'_1, T'_2 as *terminals*. In the second step, each quadrangulation R with degree d in \mathcal{S} is replaced by d paths that together visit all nodes in $Q(R)$ and which join the connectors at their terminals to a single Hamilton cycle.

We now describe both steps in detail. For a given quadrangulation R , we associate each of the colorful triangulations $T \in C(R)$ by a bitstring $b(T) \in \{0, 1\}^q$ as follows: We label the q quadrangles of R arbitrarily by $j = 1, \dots, q$, and we define $b(T)_j := 0$ if the monochromatic edge of T that sits inside the j th quadrangle of R connects the two red points, and otherwise (if it connects two blue points) $b(T)_j := 1$.



■ **Figure 14** Illustration of the proof of Lemma 13.

Now consider an edge (R, R') of \mathcal{S} , which we aim to replace by two connectors $(T_1, T'_1), (T_2, T'_2)$ with $T_1, T_2 \in C(R)$ and $T'_1, T'_2 \in C(R')$. We denote the edge flipped in R by (p_1, p_4) , and we label the points of the adjacent 4-gons in circular order by p_1, p_2, p_3, p_4 and p_4, p_5, p_6, p_1 , respectively, such that the edge (p_1, p_4) is replaced by (p_2, p_5) ; see Figure 14 (a). It follows that T_1, T'_1, T_2, T'_2 must be triangulations that contain the two monochromatic edges (p_2, p_4)

and (p_5, p_1) (but neither (p_1, p_3) nor (p_4, p_6)). Consequently, the two corresponding bits of $b(T_1), b(T'_1), b(T_2), b(T'_2)$ must have a prescribed value, and therefore T_1, T_2 and T'_1, T'_2 can be chosen from a $(q-2)$ -dimensional subcube of $Q(R)$ and $Q(R')$, respectively. Also, we will choose the connectors so that the pairs of terminals (T_1, T_2) and (T'_1, T'_2) differ only in a single flip, i.e., we select the two pairs of terminals as edges in their respective cubes, and we call these edges in $Q(R)$ and $Q(R')$ *shortcuts*. Note that the two connectors with the two shortcuts form the 4-cycle (T_1, T'_1, T'_2, T_2) . By the assumption $q \geq 3$ we have $q-2 \geq 1$, i.e., there is at least one choice for each prescribed edge.

If the node R has degree d in \mathcal{S} , then we have to choose d distinct shortcut edges in the hypercube $Q(R)$, each selected from a distinct (but not necessarily disjoint) $(q-2)$ -dimensional subcube, and to find a Hamilton cycle $H(R)$ in $Q(R)$ that contains all of these edges; see Figure 14 (b). By Lemma 11, it is enough to ensure that the shortcut edges together form paths in $Q(R)$. If $d \leq 2$ (case (i) of the lemma), then this is clear, as one or two edges always form one or two paths. If $d = 3$ (case (ii) of the lemma), one has to avoid that all three shortcut edges are incident to the same node, which is easily possible under the stronger assumption $q \geq 4$.

Then the Hamilton cycle in \mathcal{F}_N is obtained by taking the symmetric difference of the edge sets of the cycles $H(R) \subseteq Q(R)$ for all quadrangulations R on N points with the 4-cycles formed by the connectors and shortcuts (i.e., the shortcuts are removed, and the connectors are added instead). This completes the proof. \blacktriangleleft

3.2 General coloring patterns

We now consider an arbitrary coloring sequence $\alpha = (\alpha_1, \dots, \alpha_\ell)$ and the corresponding coloring pattern defined in (1). Recall that \mathcal{C}_α denotes the set of all colorful triangulations with this coloring pattern, and that the corresponding flip graph is denoted by \mathcal{F}_α . The graph \mathcal{F}_α is an induced subgraph of the associahedron \mathcal{G}_N , where $N = \sum_{i=1}^{\ell} \alpha_i$. As in the previous section, a colorful triangulation $T \in \mathcal{C}_\alpha$ has two types of edges, namely monochromatic and colorful edges. We write E_α for the set of boundary edges that are monochromatic in T , i.e., these are the pairs of points $(i, i+1)$ for $i = 1, \dots, N$ (modulo N) where both endpoints receive the same color. Generalizing the discussion from the previous section, we observe the following:

- (i) Every triangle of T has exactly one monochromatic edge.
- (ii) Except the edges in E_α , every monochromatic edge of T is an inner edge.

Consequently, if we remove from T all monochromatic inner edges (the edges in E_α are boundary edges and hence not removed), keeping only the colorful ones, then the resulting dissection $r(T)$ has $t := N - \ell$ triangles that contain the edges in E_α and $q := (\ell - 2)/2$ quadrangles. Furthermore, there are 2^q many colorful triangulations that yield the same dissection $r(T)$ by removing monochromatic edges. They are obtained from $r(T)$ by placing a diagonal in each of the q quadrangles in one of the two ways. Note that the subgraph of \mathcal{F}_α induced by those 2^q triangulations is isomorphic to the q -dimensional hypercube Q_q . We thus obtain a partition of \mathcal{F}_α into hypercubes Q_q , plus edges between them.

We refer to a dissection of a convex N -gon into q quadrangles and t triangles that contain all the edges of E_α as an α -*angulation*, and we write \mathcal{D}_α for the set of all such dissections. We also note that every α -angulation R on N points equals $r(T)$ for some colorful triangulation $T \in \mathcal{C}_\alpha$. Indeed, given R , then coloring the N points according to the pattern in (1) will make all edges except the ones in E_α colorful. We define a reduced graph \mathcal{F}'_α that has as nodes all α -angulations on N points, and for any two colorful triangulations T and T' that differ in a flip of a colorful edge, we add an edge between $r(T)$ and $r(T')$ in \mathcal{F}'_α ; see Figures 12, 13 and 15.

The proof of Lemma 13 presented in the previous section generalizes straightforwardly, yielding the following statement. Note that the variable ℓ in Lemma 14 below plays the role of $N = 2q + 2$ in Lemma 13, and so the assumptions $\ell = 8$ and $\ell \geq 10$ translate to $q = (\ell - 2)/2 = 3$ and $q \geq 4$ used in the proof of Lemma 13, respectively.

► **Lemma 14.** *Let $\alpha = (\alpha_1, \dots, \alpha_\ell)$ be a coloring sequence of (even) length $\ell \geq 8$, and let \mathcal{S} be a spanning tree of \mathcal{F}'_α .*

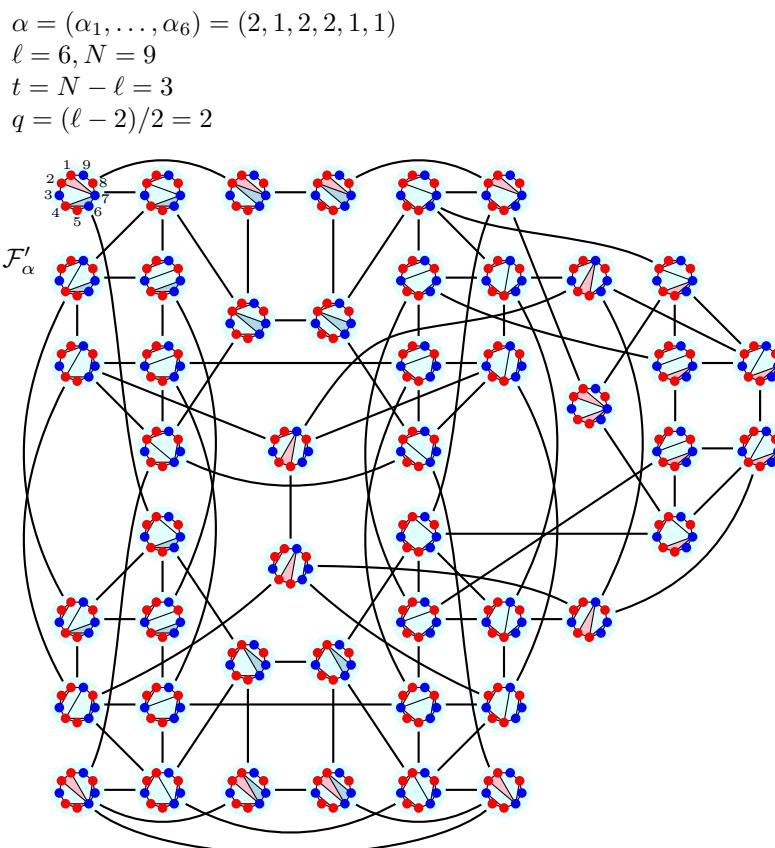
- (i) *If $\ell = 8$ and $\Delta(\mathcal{S}) \leq 2$ (i.e., \mathcal{S} is a Hamilton path), then \mathcal{F}_α has a Hamilton cycle.*
- (ii) *If $\ell \geq 10$ and $\Delta(\mathcal{S}) \leq 3$, then \mathcal{F}_α has a Hamilton cycle.*

The next lemma allows us to duplicate the occurrence of a color that appears only once (i.e., we change $\alpha_i = 1$ to some larger number $\alpha_i > 1$), while inductively maintaining spanning trees with small degrees in the corresponding reduced flip graphs.

► **Lemma 15.** *Let $\beta = (\beta_1, \dots, \beta_\ell)$ and $\alpha = (\alpha_1, \dots, \alpha_\ell)$ be coloring sequences of (even) length $\ell \geq 4$ that agree in all but the i th entry such that $\beta_i = 1$ and $\alpha_i > 1$.*

- (i) *If \mathcal{F}'_β has a Hamilton path and $\alpha_i = 2$, then \mathcal{F}'_α has a Hamilton path.*
- (ii) *If \mathcal{F}'_β has a spanning tree \mathcal{T} with $\Delta(\mathcal{T}) \leq 3$, then \mathcal{F}'_α has a spanning tree \mathcal{S} with $\Delta(\mathcal{S}) \leq 3$.*

In Figure 12, part (i) of this lemma is applied to construct a Hamilton path in \mathcal{F}'_7 from one in \mathcal{F}'_6 . Similarly, in Figure 13, a Hamilton path in \mathcal{F}'_8 is constructed from one in \mathcal{F}'_6 .

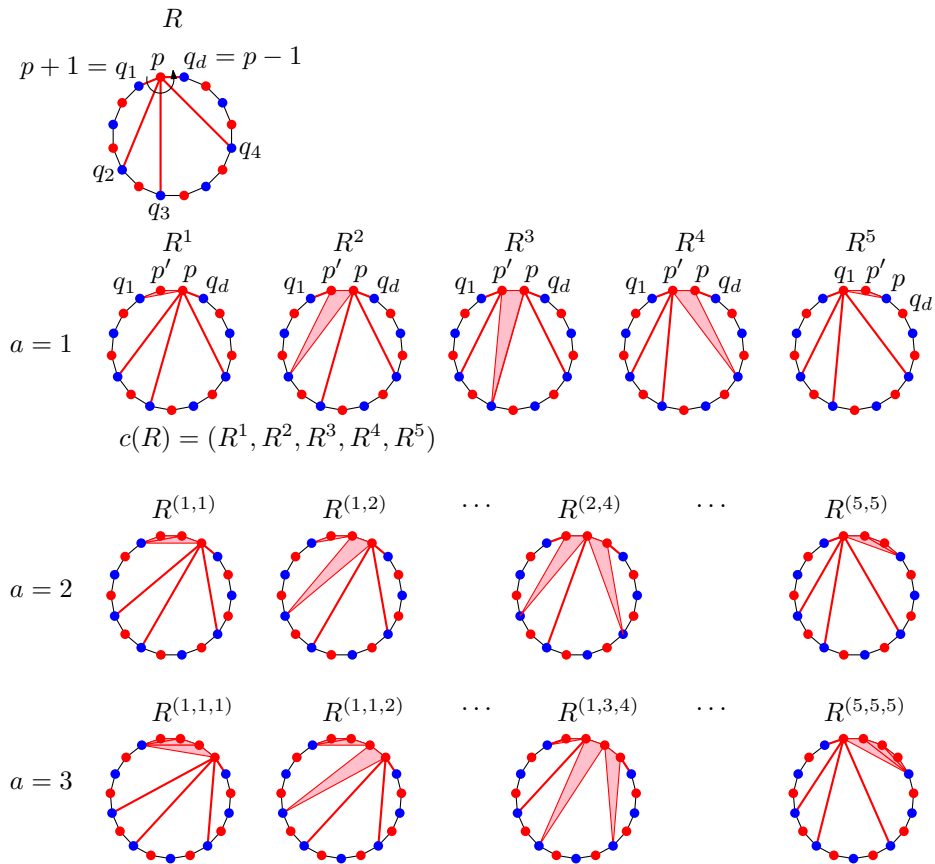


■ **Figure 15** Reduced graph of colorful triangulations for the coloring sequence $\alpha = (2, 1, 2, 2, 1, 1)$ (rrbrrbbrb).

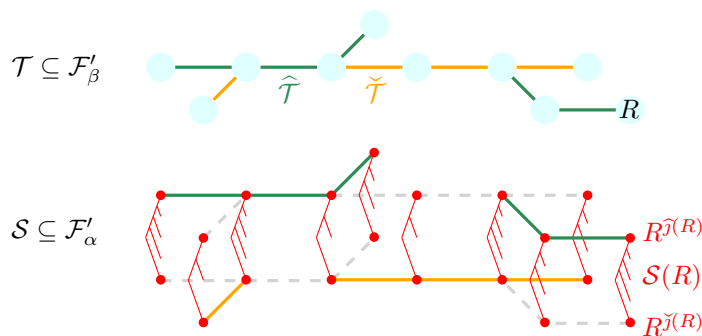
30:16 Flips in Colorful Triangulations

The idea for the proof of part (i) is the same as the one used by Hurtado and Noy [16].

Proof. We consider the point $p := \sum_{j=1}^i \beta_j$ on the boundary, which is neighbored by two points $p - 1$ and $p + 1$ (modulo $N = \sum_{j=1}^\ell \beta_j$) of the opposite color. We also define $a := \alpha_i - 1$, i.e., we want to add a points of the same color as p next to p . Let $R \in \mathcal{D}_\beta$ be a β -angulation, and let $(p, q_1), (p, q_2), \dots, (p, q_d)$ be the edges incident with the point p in R in counterclockwise order (all these edges are colorful), such that $q_1 = p + 1$ and $q_d = p - 1$; see Figure 16. If $a = 1$, then for $j = 1, \dots, d$ we let R^j be the α -angulation obtained from R by inflating the edge (p, q_j) to a triangle (p, p', q_j) . Specifically, the single point p is split into two consecutive points p and p' on the boundary joined by an edge, and q_1, \dots, q_j remain connected to p' , whereas q_j, q_{j+1}, \dots, q_d remain connected to p . More generally, we define $J(R) := \{(j_1, \dots, j_a) \mid 1 \leq j_1 \leq j_2 \leq \dots \leq j_a \leq d\}$, $\check{j}(R) := 1^a$ and $\hat{j}(R) := d^a$, and for any $(j_1, j_2, \dots, j_a) \in J(R)$ we let $R^{(j_1, \dots, j_a)}$ be the β -angulation obtained from R by inflating each of the edges $(p, q_{j_1}), \dots, (p, q_{j_a})$ to a triangle. Note that the same edge may be inflated multiple times; see the bottom rows with labels $a = 2$ and $a = 3$ in Figure 16. Specifically, if some value j_b , $b \in \{1, \dots, a\}$, appears c times in the list j_1, \dots, j_a , then the edge (p, q_{j_b}) is inflated to c many triangles. Furthermore, observe that $R^{(j_1, \dots, j_a)}$ differs from $R^{(j'_1, \dots, j'_a)}$ in a flip if and only if (j_1, \dots, j_a) and (j'_1, \dots, j'_a) differ in a single entry by ± 1 , i.e., the subgraph of \mathcal{F}'_α induced by the α -angulations $R^{(j_1, \dots, j_a)}$, $(j_1, \dots, j_a) \in J(R)$, is isomorphic



■ **Figure 16** Illustration of the proof of Lemma 15. Edges of the β -angulation R that are not incident to the point p are not shown for clarity.



■ **Figure 17** Illustration of the proof of part (ii) of Lemma 15. The fork-like structures are the spanning trees obtained from Lemma 12 (cf. Figure 11).

to the graph $G(a, d)$ defined in Section 2.4. By Lemma 12, it admits a spanning tree $\mathcal{S}(R)$ with $\Delta(\mathcal{S}(R)) \leq 3$, in which the nodes $R^{\check{r}(R)}$ and $R^{\hat{r}(R)}$ have degree 1. If $a = 1$, then this subgraph and spanning tree is simply a path, and we refer to it as children sequence $c(R) := (R^1, R^2, \dots, R^d)$. Also note that if (R, Q) is an edge in \mathcal{F}'_β , then $(R^{\check{r}(R)}, Q^{\check{r}(Q)})$ and $(R^{\hat{r}(R)}, Q^{\hat{r}(Q)})$ are both edges in \mathcal{F}'_α .

We now prove (i), using the assumption $\alpha_i = 2$, i.e., $a = 1$. Let $P = (R_1, \dots, R_L)$ be a Hamilton path in \mathcal{F}'_β . Then a Hamilton path in \mathcal{F}'_α is given by $P' := (\text{rev}(c(R_1)), c(R_2), \text{rev}(c(R_3)), c(R_4), \dots)$; see Figures 12 and 13.

For proving (ii), let \mathcal{T} be a spanning tree in \mathcal{F}'_β with $\Delta(\mathcal{T}) \leq 3$. We partition its edges into two disjoint forests of paths $\check{\mathcal{T}}$ and $\hat{\mathcal{T}}$, i.e., we have $\Delta(\check{\mathcal{T}}) \leq 2$ and $\Delta(\hat{\mathcal{T}}) \leq 2$; see Figure 17. We then define the spanning tree \mathcal{S} as the union of the trees $\mathcal{S}(R)$ for all β -angulations R plus the edges $\{(R^{\check{r}(R)}, Q^{\check{r}(Q)}) \mid (R, Q) \in \check{\mathcal{T}}\}$ and $\{(R^{\hat{r}(R)}, Q^{\hat{r}(Q)}) \mid (R, Q) \in \hat{\mathcal{T}}\}$. It is easy to check that \mathcal{S} is indeed a spanning tree of \mathcal{F}'_α with $\Delta(\mathcal{S}) \leq 3$. ◀

3.3 Proofs of Theorems 1 and 2

Proof of Theorem 2. For the given coloring sequence $\alpha = (\alpha_1, \dots, \alpha_\ell)$ of length $\ell \geq 10$, we consider the alternating coloring sequence $\beta = 1^\ell$ of length ℓ , i.e., all repetitions of colors are reduced to a single occurrence, and β corresponds to coloring ℓ points alternatingly red and blue. The corresponding reduced flip graph \mathcal{F}'_ℓ is isomorphic to the rotation graph of ternary trees, i.e., we have $\mathcal{F}'_\beta = \mathcal{F}'_\ell \simeq \mathcal{G}_{\ell,4}$. Theorem 7 yields a Hamilton path in the graph $\mathcal{F}'_\beta = \mathcal{F}'_\ell$. Applying Lemma 15 (ii) once for each α_i with $\alpha_i > 1$, we obtain that \mathcal{F}'_α has a spanning tree \mathcal{S} with $\Delta(\mathcal{S}) \leq 3$. Lastly, applying Lemma 14 (ii) yields that \mathcal{F}_α has a Hamilton cycle. ◀

Proof of Theorem 1. For $N \geq 10$ the result is a special case of Theorem 2, so it remains to cover the cases $N = 8$ and $N = 9$. A Hamilton path P in \mathcal{F}'_8 is guaranteed by Theorem 7; see Figure 13. Applying Lemma 13 (i) to P yields that \mathcal{F}_8 has a Hamilton cycle. Applying Lemma 15 (i) to P proves that \mathcal{F}'_9 has a Hamilton path P' . Applying Lemma 13 (i) to P' shows that \mathcal{F}_9 has a Hamilton cycle. ◀

4 Open questions

Tables 1 and 2 show all coloring sequences α on up to $N \leq 11$ points for which the graph \mathcal{F}_α has no Hamilton path or cycle. The sequences are shown up to rotational symmetry, reversal, and exchange of the two colors. In several cases, Theorems 3, 4 and 5 provide an explanation

30:18 Flips in Colorful Triangulations

■ **Table 1** Coloring sequences α for $N \leq 11$ for which \mathcal{F}_α has no Hamilton cycle but a Hamilton path.

N	α	Reason
6	(4, 2)	Thm. 3
	(2, 1, 1, 2)	
7	(5, 2)	Thm. 3
	(2, 1, 1, 1, 1, 1)	Thm. 5
8	(6, 2)	Thm. 3
	(4, 4)	Thm. 3
9	(7, 2)	Thm. 3
	(4, 1, 1, 1, 1, 1)	Thm. 5
10	(8, 2)	Thm. 3
	(6, 4)	Thm. 3
	(5, 1, 2, 2)	
	(5, 1, 1, 1, 1, 1)	Thm. 5
	(4, 1, 3, 2)	
11	(4, 1, 2, 3)	
	(9, 2)	Thm. 3
	(6, 1, 1, 1, 1, 1)	Thm. 5
	(5, 1, 2, 3)	
	(5, 1, 1, 4)	

■ **Table 2** Coloring sequences α for $N \leq 11$ for which \mathcal{F}_α has no Hamilton path.

N	α	Reason
6	(3, 1, 1, 1)	Thm. 4
	(3, 3)	Thm. 3
	(1, 1, 1, 1, 1, 1)	Thm. 5
7	(4, 3)	Thm. 3
	(3, 1, 2, 1)	Thm. 4
	(3, 1, 1, 2)	
8	(5, 1, 1, 1)	Thm. 4
	(5, 3)	Thm. 3
	(4, 1, 2, 1)	Thm. 4
	(4, 1, 1, 2)	
	(3, 1, 3, 1)	Thm. 4
9	(3, 1, 1, 1, 1, 1)	Thm. 5
	(3, 1, 1, 3)	
	(3, 2, 1, 2)	
	(6, 1, 1, 1)	Thm. 4
	(6, 3)	Thm. 3
10	(5, 1, 2, 1)	Thm. 4
	(5, 1, 1, 2)	
	(5, 4)	Thm. 3
	(4, 1, 3, 1)	Thm. 4
	(4, 1, 1, 3)	
	(3, 1, 3, 2)	
	(3, 1, 2, 3)	
	(7, 1, 1, 1)	Thm. 4
(7, 3)	Thm. 3	
11	(6, 1, 2, 1)	Thm. 4
	(6, 1, 1, 2)	
	(5, 1, 3, 1)	Thm. 4
	(5, 1, 1, 3)	
	(5, 5)	Thm. 3
	(4, 1, 4, 1)	Thm. 4
	(4, 1, 1, 4)	
(3, 1, 3, 3)		
11	(8, 1, 1, 1)	Thm. 4
	(8, 3)	Thm. 3
	(7, 1, 2, 1)	Thm. 4
	(7, 1, 1, 2)	
	(7, 4)	Thm. 3
	(6, 1, 3, 1)	Thm. 4
	(6, 1, 1, 3)	
	(6, 5)	Thm. 3
	(5, 1, 4, 1)	Thm. 4
	(5, 1, 3, 2)	
(4, 1, 3, 3)		

for the non-Hamiltonicity; see the third column in the tables. In the other cases, we are still missing such an explanation. Based on this data, we feel that Theorem 2 can be strengthened and the requirement $\ell \geq 10$ relaxed to $\ell \geq 8$. Furthermore, it seems that for $\ell = 6$ there is always a Hamilton path in \mathcal{F}_α unless $\alpha \in \{(1, 1, 1, 1, 1, 1), (3, 1, 1, 1, 1, 1)\}$.

Also, Theorem 9 on twists in 3-colored triangulations invites deeper investigation. We conjecture that the graph \mathcal{H}_N has a Hamilton cycle for all $N \geq 9$ that are divisible by 3. Furthermore, it seems that if $N \equiv 2 \pmod{3}$ the graph \mathcal{H}_N is not connected. What are the properties of the flip graphs for general coloring patterns with three or more colors?

Another interesting question concerns bijections between k -ary trees and classes of permutations. For $k = 2$ (binary trees), there is a natural bijection to 231-avoiding permutations. Are there similar correspondences between k -ary trees and pattern-avoiding permutations for $k \geq 3$? In particular, do tree rotations translate to nice operations on the permutations, specifically to so-called jumps heavily used in [12, 13, 20, 5]?

Going back to the uncolored setting and the associahedron \mathcal{G}_N , Theorem 2 shows that \mathcal{G}_N admits cycles of many different lengths. What is the *cycle spectrum* of \mathcal{G}_N , i.e., the set $S(\mathcal{G}_N)$ of all possible lengths of cycles in \mathcal{G}_N ? We conjecture that almost all lengths are possible.

► **Conjecture 16.** *We have $|S(\mathcal{G}_N)|/|\mathcal{D}_{N,3}| = 1 - o(1)$ as $N \rightarrow \infty$.*

Baur, Bergerova, Voon and Xu [2] recently introduced another family of flip graphs on triangulations in which the triangles are colored, not the vertices. The resulting graphs are disconnected in general, and their structure is still not very well understood (in [11] these graphs are related to the famous Four Color Theorem).

References

- 1 R. Acharya, T. Mütze, and F. Verciani. Flips in colorful triangulations. *arXiv:2406.03783*, June 2024. doi:10.48550/arXiv.2406.03783.
- 2 K. Baur, D. Bergerova, J. Voon, and L. Xu. Flip graphs of coloured triangulations of convex polygons. *arXiv:2402.06546*, February 2024.
- 3 L. A. Berry, B. Reed, A. Scott, and D. R. Wood. A logarithmic bound for the chromatic number of the associahedron. *arXiv:1811.08972*, November 2018.
- 4 M. Buck and D. Wiedemann. Gray codes with restricted density. *Discrete Math.*, 48(2–3):163–171, 1984. doi:10.1016/0012-365X(84)90179-1.
- 5 J. Cardinal, A. Merino, and T. Mütze. Combinatorial generation via permutation languages. IV. Elimination trees. To appear in *ACM Transactions on Algorithms*; preprint available on *arXiv:2106.16204*, 2024.
- 6 C. Ceballos, F. Santos, and G. M. Ziegler. Many non-equivalent realizations of the associahedron. *Combinatorica*, 35(5):513–551, 2015. doi:10.1007/s00493-014-2959-9.
- 7 The Combinatorial Object Server: Generate k -ary trees and dissections. <http://www.combos.org/kary>.
- 8 T. Dvořák. Hamiltonian cycles with prescribed edges in hypercubes. *SIAM J. Discrete Math.*, 19(1):135–144, 2005. doi:10.1137/S0895480103432805.
- 9 P. Eades, M. Hickey, and R. C. Read. Some Hamilton paths and a minimal change algorithm. *J. Assoc. Comput. Mach.*, 31(1):19–29, 1984. doi:10.1145/2422.322413.
- 10 R. Fabila-Monroy, D. Flores-Peñaloza, C. Huemer, F. Hurtado, J. Urrutia, and D. R. Wood. On the chromatic number of some flip graphs. *Discrete Math. Theor. Comput. Sci.*, 11(2):47–56, 2009. doi:10.46298/DMTCS.460.
- 11 S. Gravier and C. Payan. Flips signés et triangulations d’un polygone. *European J. Combin.*, 23(7):817–821, 2002. doi:10.1006/eujc.2002.0601.

- 12 E. Hartung, H. P. Hoang, T. Mütze, and A. Williams. Combinatorial generation via permutation languages. I. Fundamentals. *Trans. Amer. Math. Soc.*, 375(4):2255–2291, 2022. doi:10.1090/tran/8199.
- 13 H. P. Hoang and T. Mütze. Combinatorial generation via permutation languages. II. Lattice congruences. *Israel J. Math.*, 244(1):359–417, 2021. doi:10.1007/s11856-021-2186-1.
- 14 C. Hohlweg and C. E. M. C. Lange. Realizations of the associahedron and cyclohedron. *Discrete Comput. Geom.*, 37(4):517–543, 2007. doi:10.1007/s00454-007-1319-6.
- 15 C. Huemer, F. Hurtado, and J. Pfeifle. The rotation graph of k -ary trees is Hamiltonian. *Inform. Process. Lett.*, 109(2):124–129, 2008. doi:10.1016/j.ip1.2008.09.013.
- 16 F. Hurtado and M. Noy. Graph of triangulations of a convex polygon and tree of triangulations. *Comput. Geom.*, 13(3):179–188, 1999. doi:10.1016/S0925-7721(99)00016-4.
- 17 J.-L. Loday. Realization of the Stasheff polytope. *Arch. Math. (Basel)*, 83(3):267–278, 2004. doi:10.1007/s00013-004-1026-y.
- 18 J. M. Lucas. The rotation graph of binary trees is Hamiltonian. *J. Algorithms*, 8(4):503–535, 1987. doi:10.1016/0196-6774(87)90048-4.
- 19 J. M. Lucas, D. Roelants van Baronaigien, and F. Ruskey. On rotations and the generation of binary trees. *J. Algorithms*, 15(3):343–366, 1993. doi:10.1006/jagm.1993.1045.
- 20 A. Merino and T. Mütze. Combinatorial generation via permutation languages. III. Rectangulations. *Discrete Comput. Geom.*, 70(1):51–122, 2023. doi:10.1007/s00454-022-00393-w.
- 21 V. Pilaud, F. Santos, and G. M. Ziegler. Celebrating Loday’s associahedron. *Arch. Math. (Basel)*, 121(5-6):559–601, 2023. doi:10.1007/s00013-023-01895-6.
- 22 L. Pournin. The diameter of associahedra. *Adv. Math.*, 259:13–42, 2014. doi:10.1016/j.aim.2014.02.035.
- 23 B. E. Sagan. Proper partitions of a polygon and k -Catalan numbers. *Ars Combin.*, 88:109–124, 2008.
- 24 D. D. Sleator, R. E. Tarjan, and W. P. Thurston. Rotation distance, triangulations, and hyperbolic geometry. *J. Amer. Math. Soc.*, 1(3):647–681, 1988. doi:10.2307/1990951.

Revisiting ILP Models for Exact Crossing Minimization in Storyline Drawings

Alexander Dobler ✉ 

TU Wien, Austria

Michael Jünger

University of Cologne, Germany

Paul J. Jünger ✉

University of Bonn, Germany

Julian Meffert ✉ 

University of Bonn, Germany

Petra Mutzel ✉ 

University of Bonn, Germany

Martin Nöllenburg ✉ 

TU Wien, Austria

Abstract

Storyline drawings are a popular visualization of interactions of a set of characters over time, e.g., to show participants of scenes in a book or movie. Characters are represented as x -monotone curves that converge vertically for interactions and diverge otherwise. Combinatorially, the task of computing storyline drawings reduces to finding a sequence of permutations of the character curves for the different time points, with the primary objective being crossing minimization of the induced character trajectories. In this paper, we revisit exact integer linear programming (ILP) approaches for this NP-hard problem. By enriching previous formulations with additional problem-specific insights and new heuristics, we obtain exact solutions for an extended new benchmark set of larger and more complex instances than had been used before. Our experiments show that our enriched formulations lead to better performing algorithms when compared to state-of-the-art modelling techniques. In particular, our best algorithms are on average 2.6–3.2 times faster than the state-of-the-art and succeed in solving complex instances that could not be solved before within the given time limit. Further, we show in an ablation study that our enrichment components contribute considerably to the performance of the new ILP formulation.

2012 ACM Subject Classification Human-centered computing → Graph drawings; Mathematics of computing → Integer programming; Mathematics of computing → Permutations and combinations

Keywords and phrases Storyline drawing, crossing minimization, integer linear programming, algorithm engineering, computational experiments

Digital Object Identifier 10.4230/LIPIcs.GD.2024.31

Related Version *Full Version*: <https://arxiv.org/abs/2409.02858> [6]

Supplementary Material *Software*: <https://doi.org/10.17605/OSF.IO/3BUA2> [5]

Funding *Alexander Dobler*: Vienna Science and Technology Fund (WWTF) grant [10.47379/ICT19035].

Julian Meffert: Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 459420781.

Petra Mutzel: partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 459420781.

Martin Nöllenburg: Vienna Science and Technology Fund (WWTF) grant [10.47379/ICT19035].



© Alexander Dobler, Michael Jünger, Paul J. Jünger, Julian Meffert, Petra Mutzel, and Martin Nöllenburg;

licensed under Creative Commons License CC-BY 4.0

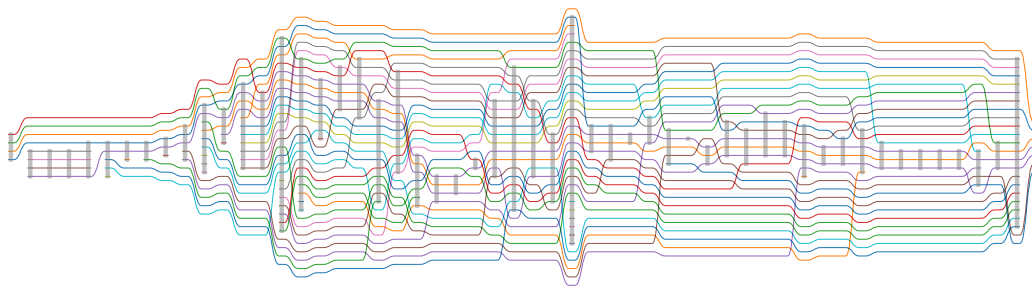
32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 31; pp. 31:1–31:19

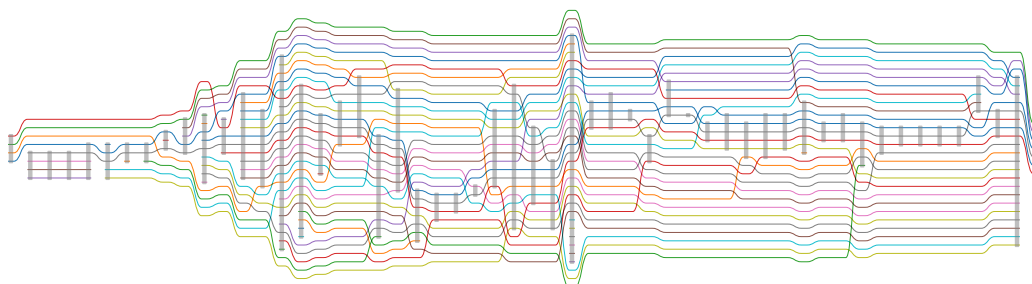


Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



(a) Solution with 374 crossings computed in 0.15 s by a greedy heuristic.



(b) Drawing with the minimum number of 236 crossings computed in 401.23 s by our new ILP formulation.

■ **Figure 1** Storylines for the first Harry Potter movie. Interactions are shown as vertical gray bars.

1 Introduction

Storyline drawings are a well-studied visualization style for complex event-based temporal interaction data and have been popularized by the xkcd comic “Movie Narrative Charts” in 2009 [19]. They show a set of characters, e.g., from the plot of a movie or book, and how they interact or co-occur in a sequence of events over time, e.g., by participating in the same scene or conversation of the evolving story. A storyline drawing is a two-dimensional diagram, where the x -axis represents time and the y -dimension is used for the vertical grouping of characters according to their interaction sequence. The exact temporal distance of interactions is usually not depicted, only their order. Each character is represented as an x -monotone curve, and interactions are represented by vertically grouping the curves of the participating characters at the x -coordinate corresponding to the interaction time. Characters that are not participating in an interaction at any specific point in time are vertically separated from each other. Figure 1 shows an example of a storyline drawing. Due to their popularity and the intuitive data encoding, they are well suited for visual storytelling and have since been used as visual metaphors for representing a variety of different event-based data sets beyond the original book and movie plots [19, 28], e.g., for software projects [20, 27], newspaper articles [1], political debates on social media [18], visual summaries of meeting contents [23], scientific collaborations [14], sports commentary [34], and gameplay data [33].

There is one main degree of freedom when computing and optimizing storyline drawings: the (vertical) linear order and positioning of the characters at each discrete time steps. The only hard constraint is that all characters participating in the same interaction must be consecutive as a group. This degree of freedom can thus be used to minimize the number of crossings between character curves, their wiggles (i.e. the amount of vertical movement of

character lines in the visualization), and any excessive white space in the diagram, which are the three major objectives that have been identified for computing storyline drawings [27,28]. While the number of crossings is determined purely combinatorially by the sequence of character permutations, wiggles and white space depend on the actual y -coordinates assigned to each character curve at each point in time.

In this paper, our interest is the combinatorial crossing minimization problem. It is the primary objective in practical storyline optimization pipelines [18], where it forms the input to the subsequent steps of reducing wiggles and white space while maintaining the character order. Additionally, crossing minimization is one of the most fundamental graph drawing problems [2,22] and it is well known that graph drawings with fewer crossings increase readability [21]. Unfortunately, crossing minimization in storyline drawings is an NP-hard problem [9,17] and hence practical approaches for storyline visualization usually resort to heuristics, even though they cannot guarantee optimal solutions.

We consider the crossing minimization problem from the opposite side and revisit exact integer linear programming (ILP) approaches [11] for computing provably optimal solutions. Such approaches often lead to practical exact algorithms. Our goal is to improve on the runtime performance of such exact methods by enriching the models with both new problem-specific insights and better heuristics. Faster exact algorithms for crossing minimization in storyline drawings are practically relevant for two reasons: firstly, solving moderately sized instances to optimality within a few seconds provides a strictly better alternative to commonly used suboptimal heuristics, and secondly, knowing optimal solutions for a large set of representative benchmark instances (even if their computations take several minutes or up to a few hours) is a prerequisite for any thorough experimental study on the performance of non-exact crossing minimization heuristics and for generating crossing-minimum stimuli in user experiments.

Related Work. Tanahashi and Ma [28] introduced storyline drawings as an information visualization problem, provided the first visual encoding model, and defined the above-mentioned optimization criteria (crossings, wiggles, white space). They suggested a genetic algorithm to compute storyline drawings. Ogawa and Ma [20] used a greedy algorithm to compute storylines to depict software evolution. Due to slow computation times of previous methods, Liu et al. [18] split the layout process into a pipeline of several subproblems ordered by importance, the first one being crossing minimization. They solved the character line ordering by an iterated application of a constrained barycenter algorithm, a classic heuristic for multi-layer crossing minimization [25]. Their results were obtained in less than a second and had fewer crossings than those computed by the genetic algorithm [28], which took more than a day to compute on some of the same instances. Tanahashi et al. [27] enhanced previous methods to take into account streaming data and apply a simple sequential left-to-right sorting heuristic. Recent practical works on storyline drawings focus on other aspects, such as an interactive editor [30] or a mixed-initiative system including a reinforcement learning AI component [29]; both these systems apply a two-layer crossing minimization heuristic [8].

Several authors focused on the combinatorial crossing minimization problem and its complexity. Kostitsyna et al. [17] observed that the NP-hardness of the problem follows from a similar bipartite crossing minimization problem [9] and proved fixed-parameter tractability when the number of characters is bounded by a parameter k . Gronemann et al. [11] were the first to model the problem as a special type of tree-constrained multi-layer crossing minimization problem. They implemented an exact branch-and-cut approach that exploits the equivalence of the quadratic unconstrained 0/1-optimization problem with the maximum

cut problem in a graph. They managed to solve many instances with up to 20 characters and 50 interactions optimally within a few seconds. Van Dijk et al. [31,32] proposed block-crossing minimization in storyline drawings, which counts grid-like blocks of crossings rather than individual crossings. They showed NP-hardness and proposed greedy heuristics, a fixed-parameter tractable algorithm, and an approximation algorithm. In a follow-up work, van Dijk et al. [32] implemented and experimentally evaluated an exact approach for the block crossing minimization problem using SAT solving. A different variation of storylines was studied by Di Giacomo et al. [10], who considered ubiquitous characters as x -monotone trees with multiple branches, enabling characters to participate in multiple simultaneous interactions; they solved the crossing minimization aspect using an adaptation of the previous SAT model [32]. Dobler et al. [7] consider time interval storylines, where additionally to the order of characters, the order of time steps in so-called time-intervals can be permuted.

The problem is also similar to crossing minimization in layered graph drawing, which was introduced by Sugiyama et al. [25]. The problem is to draw a graph with its vertices on multiple parallel lines while minimizing crossings. A notable difference to storyline crossing minimization is that vertices can have arbitrary degree and that edges can span more than one layer. For a survey of algorithms and techniques in layered graph drawing, we refer to Healy and Nikolov [13].

Contributions. The contributions of this paper are the following:

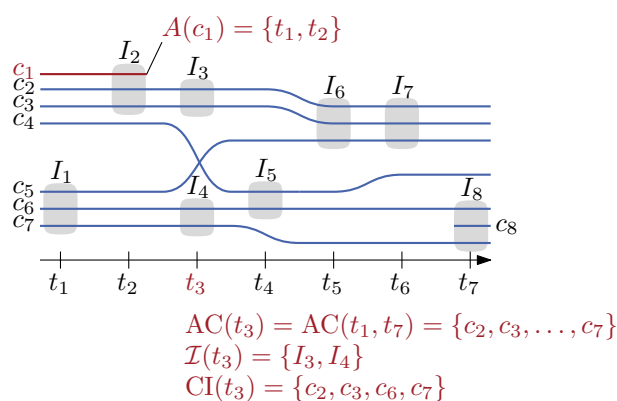
- We identify structural properties of storyline drawings and prove that there exist crossing-minimum drawings satisfying them, reducing the search space of feasible solutions.
- We propose a new ILP formulation exploiting these structural insights in order to (i) significantly reduce the number of required constraints and (ii) apply symmetry breaking constraints to strengthen the ILP model.
- We introduce several new heuristics that support the exact solver, either as initial heuristics to improve branch-and-bound pruning or for deriving integral solutions from fractional ones during the incremental ILP solving process.
- We have compiled a new benchmark set of storyline instances, including those of earlier studies, as well as several challenging new ones.
- We have conducted a detailed experimental evaluation of our new ILP model using the above benchmark set. We compare its ability to solve instances with state-of-the-art ILP models. Moreover, in an ablation study, we show that our further enhancements (e.g., adding symmetry breaking constraints and novel heuristics) contributes considerably to the performance of both the new and several state-of-the-art ILP formulations.
- We show that our ILP models are able to solve previously unsolved instances from the literature and obtain a speedup of 2.6–3.2 compared to the state of the art.

Data sets, source code, evaluation, and a visualization software are available on <https://osf.io/3bua2/>.

Due to space constraints, statements marked with (★) are proved in a full version of the paper [6].

2 Preliminaries

Permutations. Given a set $X = \{x_1, \dots, x_n\}$, a permutation π is a linear order of its elements, or equivalently, a bijective mapping from $\{1, 2, \dots, |X|\}$ to X . For $x, x' \in X$ we write $x \prec_\pi x'$ if x comes before x' in π . For $Y \subseteq X$, $\pi[Y]$ is the permutation π restricted to Y , formally, for $y, y' \in Y$, $y \prec_{\pi[Y]} y'$ if and only if $y \prec_\pi y'$. For two permutations π, ϕ



■ **Figure 2** Illustration of important notation throughout this paper with the aid of a storyline drawing depicting interactions I_1 – I_8 of the characters c_1 – c_8 over the time steps t_1 – t_7 .

of two sets X and Y with $X \cap Y = \emptyset$, we denote by $\pi \star \phi$ their concatenation. Given two permutations π, π' of the same set X , the *inversions* between π and π' is the number of pairs $x, x' \in X$ such that $\pi^{-1}(x) < \pi^{-1}(x')$ and $\pi'^{-1}(x) > \pi'^{-1}(x')$.

Problem input. A *storyline instance* consists of a 4-tuple $(T, \mathcal{C}, \mathcal{I}, A)$ where $T = \{t_1, t_2, \dots, t_\ell\}$ is a set of totally ordered *time steps* (or *layers*), $\mathcal{C} = \{c_1, c_2, \dots, c_n\}$ is a set of *characters*, and $\mathcal{I} = \{I_1, I_2, \dots, I_m\}$ is a set of interactions. Each interaction $I \in \mathcal{I}$ has a corresponding time step $\text{time}(I) \in T$ and consists of a set of characters $\text{char}(I) \subseteq \mathcal{C}$. Further, A maps each character $c \in \mathcal{C}$ to a consecutive set of time steps, i.e., $A(c) = \{t_i, t_{i+1}, \dots, t_j\}$ for $1 \leq i \leq j \leq \ell$. We say that character c is *active* at the time steps in $A(c)$, it *starts* at t_i and *ends* at t_j . We define $AC(t)$ for $t \in T$ as the set of all characters $c \in \mathcal{C}$ active at time t , i.e., $AC(t) = \{c \in \mathcal{C} \mid t \in A(c)\}$. Clearly, for each interaction $I \in \mathcal{I}$, $\text{char}(I) \subseteq AC(\text{time}(I))$ must hold. Next, we define the set of all characters active in the time interval $[t_i, t_j]$ ($1 \leq i \leq j \leq \ell$) as $AC(t_i, t_j) = AC(t_i) \cap AC(t_{i+1}) \cap \dots \cap AC(t_j)$. For a time step $t \in T$ we define the set of interactions at t as $\mathcal{I}(t) = \{I \in \mathcal{I} \mid \text{time}(I) = t\}$ and its corresponding set of characters as $CI(t) = \bigcup_{I \in \mathcal{I}(t)} \text{char}(I)$. Without loss of generality, for the interactions at time step t we assume that $|\mathcal{I}(t)| \neq 0$ and that the sets of characters of the interactions $\mathcal{I}(t)$ are pairwise disjoint. This is a reasonable assumption as characters usually participate in at most one interaction at any given time, e.g. in movies. Important notation is also illustrated in Figure 2.

Problem output. Solutions to storyline instances $(T, \mathcal{C}, \mathcal{I}, A)$ consist of a sequence of ℓ permutations $S = (\pi_1, \pi_2, \dots, \pi_\ell)$ such that π_i is a permutation of $AC(t_i)$ for all $i = 1, \dots, \ell$ satisfying the condition that the set of characters of each interaction $I \in \mathcal{I}(t_i)$ appears consecutively. We call S a *storyline solution* or *drawing*.

The number of crossings $\text{cr}(\pi_i, \pi_{i+1})$ between two consecutive permutations π_i and π_{i+1} is defined as the number of inversions of the two permutations $\pi_i[C]$ and $\pi_{i+1}[C]$, where $C = AC(t_i) \cap AC(t_{i+1})$. The number of crossings in a storyline solution is $\sum_{i=1}^{\ell-1} \text{cr}(\pi_i, \pi_{i+1})$. The problem addressed in this paper is the following.

► **Problem 1 (Storyline Problem).** *Given a storyline instance $(T, \mathcal{C}, \mathcal{I}, A)$, find a storyline drawing S with the minimum number of crossings.*

3 Standard Models for the Storyline Problem

The most natural ILP formulation to solve Problem 1 has a quadratic objective function and is based on the linear ordering model, which uses binary variables in order to encode the linear ordering at each time step. The number of crossings between two subsequent time steps is then given by the number of inversions of the two permutations.

From now on, we assume that characters c_u, c_v, c_w are pairwise different, even if we write for example $c_u, c_v \in C$ for some set C of characters.

Quadratic Model (QDR)

For each time step $t_i, i = 1, 2, \dots, \ell$ and each tuple of characters $c_u, c_v \in \text{AC}(t_i)$ we introduce a binary *ordering variable* $x_{i,u,v}$ which is equal to 1 if and only if $c_u \prec_{\pi_i} c_v$. The quadratic model QDR is given as follows:

$$\begin{aligned} \min \quad & \sum_{i=1}^{\ell-1} \sum_{c_u, c_v \in \text{AC}(t_i, t_{i+1})} x_{i,u,v} x_{i+1,v,u} & (\text{QDR}) \\ x_{i,u,v} = 1 - x_{i,v,u} \quad & \text{for all } i = 1, \dots, \ell; c_u, c_v \in \text{AC}(t_i) \text{ with } u < v & (\text{EQ}) \\ x_{i,u,v} + x_{i,v,w} + x_{i,w,u} \leq 2 \quad & \text{for all } i = 1, \dots, \ell; c_u, c_v, c_w \in \text{AC}(t_i) & (\text{LOP}) \\ x_{i,u,w} = x_{i,v,w} \quad & \text{for all } i = 1, \dots, \ell; I \in \mathcal{I}(t_i); & (\text{TREE}) \\ & c_u, c_v \in \text{char}(I), u < v; c_w \in \text{AC}(t_i) \setminus \text{char}(I) \\ x_{i,u,v} \in \{0, 1\} \quad & \text{for all } i = 1, \dots, \ell; c_u, c_v \in \text{AC}(t_i), & (\text{BIN}) \end{aligned}$$

The character curves for c_u and c_v cross between the two layers t_i and t_{i+1} if and only if one of the terms $x_{i,u,v}x_{i+1,v,u}$ and $x_{i,v,u}x_{i+1,u,v}$ equals 1. The (LOP) and (EQ) constraints ensure transitivity of the set of characters $\text{AC}(t_i)$ present at time step t_i and guarantee that they define a total order. For all interactions $I \in \mathcal{I}(t_i)$ the (TREE) constraints ensure that characters from I appear consecutively at the respective time step t_i .

Linearized Model (LIN)

The standard linearisation of quadratic integer programs introduces additional variables $y_{i,u,v}$ that substitute the quadratic terms $x_{i,u,v}x_{i+1,v,u}$ for all $t_i, i = 1, 2, \dots, \ell - 1$ and each tuple of characters $c_u, c_v \in \text{AC}(t_i, t_{i+1})$ in the objective function. In order to link the new variables with the ordering variables, we introduce the following constraints:

$$y_{i,u,v} \geq x_{i,u,v} - x_{i+1,u,v} \quad \text{for all } i = 1, \dots, \ell; c_u, c_v \in \text{AC}(t_i, t_{i+1}) \quad (\text{CR})$$

Obviously, the variable $y_{i,u,v}$ is forced to 1, if the character c_u is before c_v at time step t_i in the solution represented by the y -variables, and the order of both characters is reversed at time step t_{i+1} . The linearised model (LIN) is given as follows.

$$\begin{aligned} \min \quad & \sum_{i=1}^{\ell-1} \sum_{c_u, c_v \in \text{AC}(t_i, t_{i+1})} y_{i,u,v} & (\text{LIN}) \\ & y_{i,u,v}, x_{i,u,v} \text{ satisfy (BIN), (EQ), (LOP), (TREE), and (CR)} \end{aligned}$$

Max-Cut Model (CUT)

Gronemann et al. [11] have suggested a formulation based on the transformation of the problem into a quadratic unconstrained binary program with additional (TREE) constraints, which is then solved using a maximum cut approach. Here, we omit the detour via the quadratic binary program and directly provide the corresponding maximum cut formulation. Starting with a feasible storyline drawing $\hat{S} = (\hat{\pi}_1, \dots, \hat{\pi}_\ell)$, we define the graph $G_M = (V_M, E_M)$: The vertex set V_M is given by a vertex v^* and the union of the sets V_i ($i = 1, \dots, \ell$), where V_i has a vertex c_{uv}^i for each pair $c_u, c_v \in \text{AC}(t_i)$ with $c_u \prec_{\hat{\pi}_i} c_v$.

We introduce an edge between the vertices c_{uv}^i and c_{pq}^{i+1} if the corresponding characters coincide. In the case that $c_u = c_p$ and $c_v = c_q$, the (type-1) edge $e = e_{uv}^i$ gets a weight of $w_e = -1$, and in the case that $c_u = c_q$ and $c_v = c_p$, the (type-2) edge $e = e_{uv}^i$ gets a weight of $w_e = 1$. We define the constant K as the number of edges of type (2). The intention is the following: An edge of type (1) results in a crossing if and only if it is in the cut, and an edge of type (2) results in a crossing if and only if it is not in the cut. This construction allows for associating the maximum cut objective function values W to corresponding crossing numbers $K - W$. In particular, $W = 0$ for the empty cut corresponds to the number of crossings K in \hat{S} . In order to guarantee that the characters of an interaction appear consecutively, we introduce type-3 edges with weight 0 from the additional vertex v^* to every vertex in V_i for all $i = 1, \dots, \ell$, and add the additional constraints (TRC). We introduce a binary variable z_e for every edge $e \in E_M$ in the graph, which is 1 if and only if the edge is contained in the computed cut.

The following model guarantees that every optimal solution corresponds to a constrained maximum cut in the graph G_M that provides the optimal solution to the storyline problem. The constraints (CYC) capture the fact that any intersection of a cut and a cycle in a graph has even cardinality. The correctness is provided in [11], see also [3, 24].

$$\max \sum_{e \in E_M} w_e z_e \quad (\text{CUT})$$

$$\sum_{e \in F} z_e - \sum_{e \in C \setminus F} z_e \leq |F| - 1 \quad \text{for all cycles } C \subseteq E_M, F \subseteq C, |F| \text{ odd} \quad (\text{CYC})$$

$$0 \leq z_{(v^*, c_{uv}^i)} + z_{(v^*, c_{vw}^i)} - z_{(v^*, c_{uw}^i)} \leq 1 \quad \text{for all } i = 1, \dots, \ell; c_{uv}^i, c_{vw}^i, c_{uw}^i \in V_i \quad (\text{LOPC})$$

with $c_u \prec_{\hat{\pi}_i} c_v \prec_{\hat{\pi}_i} c_w$

$$\left. \begin{array}{l} z_{(v^*, c_{uw}^i)} = z_{(v^*, c_{vw}^i)} \text{ if } c_u, c_v \prec_{\hat{\pi}_i} c_w \\ z_{(v^*, c_{uw}^i)} = z_{(v^*, c_{vw}^i)} \text{ if } c_u, c_v \succ_{\hat{\pi}_i} c_w \end{array} \right\} \text{ for all } i = 1, \dots, \ell; I \in \mathcal{I}(t_i); \quad (\text{TRC})$$

$c_u, c_v \in \text{char}(I); c_w \in \text{AC}(t_i) \setminus \text{char}(I)$

$$z_e \in \{0, 1\} \quad \text{for all } e \in E_M \quad (\text{BIC})$$

4 Structural Properties of Storyline Solutions

In this section, we identify structural properties of storyline solutions that will help us to optimize the models proposed in Section 5, and that guide the exact optimization process. First, we define two properties of storyline drawings. Definition 2 captures that the relative order of characters in an interaction can be propagated backwards.

► **Definition 2** (Type-1 consistency). *Let $S = (\pi_1, \pi_2, \dots, \pi_\ell)$ be a solution to a storyline instance $(T, \mathcal{C}, \mathcal{I}, A)$. Let $I \in \mathcal{I}$, $t_i = \text{time}(I)$ and $C = \text{char}(I)$. Let $1 < j(I) \leq i$ be the index of the earliest time step $t_{j(I)}$ such that $C \subseteq \text{AC}(t_{j(I)}, t_i)$ and*

$$\forall k \in \{j(I) + 1, \dots, i\} : \text{CI}(t_k) \cap C = \emptyset \vee \exists I \in \mathcal{I}(t_k) : C \subseteq \text{char}(I).$$

We say that S is I -consistent if

$$\forall k \in \{j(I), j(I) + 1, \dots, i\} : \pi_k[C] = \pi_i[C].$$

Further, we say that S is type-1-consistent if it is I -consistent for all $I \in \mathcal{I}$.

Definition 3 defines the property that between suitable pairs of interactions with the same set of characters, these characters are kept together between the two time steps. Note that this is not implied by type-1 consistency.

► **Definition 3 (Type-2 consistency).** Let $S = (\pi_1, \pi_2, \dots, \pi_\ell)$ be a solution to a storyline instance $(T, \mathcal{C}, \mathcal{I}, A)$. Consider two interactions $I_1, I_2 \in \mathcal{I}$ such that

- $\text{char}(I_1) = \text{char}(I_2) = C$,
- $i = \text{time}(I_1) < \text{time}(I_2) = j$, and
- $\forall k \in \mathbb{N} : i < k < j \Rightarrow [\text{CI}(t_k) \cap C = \emptyset \vee \exists I_3 \in \mathcal{I}(t_k) : C \subseteq \text{char}(I_3)]$.

We say that S is (I_1, I_2) -consistent if

$$\forall i < k < j : \exists \pi^a, \pi^b : \pi_k = \pi^a \star \pi_i[C] \star \pi^b.$$

Further, we say that S is type-2-consistent if it is (I_1, I_2) -consistent for all such pairs (I_1, I_2) .

The following lemma shows that we can achieve type-1 consistency for storyline drawings without increasing the number of crossings. Essentially, if a storyline solution is not type-1 consistent for an interaction I , we can propagate the relative order of characters in that interaction forward from the time step $t_{j(I)}$ from Definition 2.

► **Lemma 4 (\star).** Let $(T, \mathcal{C}, \mathcal{I}, A)$ be an instance with a solution S . We can construct from S a type-1-consistent solution S' such that $\text{cr}(S') \leq \text{cr}(S)$. If S is type-2-consistent, so is S' .

A similar result with a related proof argument holds for type-2 consistency.

► **Lemma 5 (\star).** Let $(T, \mathcal{C}, \mathcal{I}, A)$ be an instance with a solution S . We can construct from S a type-2-consistent solution S' such that $\text{cr}(S') \leq \text{cr}(S)$. If S is type-1-consistent, so is S' .

The following is a direct consequence.

► **Corollary 6.** For each storyline instance $(T, \mathcal{C}, \mathcal{I}, A)$ there exists a crossing-minimum solution S that is type-1-consistent and type-2-consistent.

Theorem 7 is the main ingredient for a new ILP formulation given in Section 5. It shows that we can in specific cases assume the order of characters C_a above and C_b below an interaction at time step t_i to be equal to the relative order at t_{i-1} . This is similar to type-1-consistency, where the relative order of characters in an interaction sometimes can be kept.

► **Theorem 7 (\star).** Let $(T, \mathcal{C}, \mathcal{I}, A)$ be a storyline instance. There exists a crossing-minimum solution $S = (\pi_1, \pi_2, \dots, \pi_\ell)$ with the following property. For all $t_i \in \{t_2, t_3, \dots, t_\ell\}$ with $|\mathcal{I}(t_i)| = 1$, where $\mathcal{I}(t_i) = \{I\}$, the following holds.

- (1) $\exists C_a, C_b : \pi_i = \pi_i[C_a] \star \pi_i[\text{char}(I)] \star \pi_i[C_b]$,
- (2) if $C_a \subseteq \text{AC}(t_{i-1}, t_i)$, then $\pi_i[C_a] = \pi_{i-1}[C_a]$,
- (3) if $\text{char}(I) \subseteq \text{AC}(t_{i-1}, t_i)$, then $\pi_i[\text{char}(I)] = \pi_{i-1}[\text{char}(I)]$, and
- (4) if $C_b \subseteq \text{AC}(t_{i-1}, t_i)$, then $\pi_i[C_b] = \pi_{i-1}[C_b]$.

Proof sketch. For time steps t with $|\mathcal{I}(t)| = 1$ let C_a be the characters above the interaction in an optimal solution. Similarly, let C_b be the characters below. By imagining that C_a and C_b form two respective interactions, the result follows from Lemma 4. ◀

5 Refining the ILP models

We apply our structural insights from Section 4 to the models (besides the (CUT)-model) to obtain a new ILP formulation, including a reduction of the number of (LOP) constraints in Section 5.1 via Theorem 7 and the inclusion of additional symmetry breaking constraints in Section 5.2 via Corollary 6.

5.1 The Propagated Linear Ordering Model (PLO)

For our new formulation, we take the linearized model (LIN) as basis, but remove some of the (LOP)-constraints for time step t_i as we can make use of propagating the ordering at t_{i-1} by Theorem 7 as follows. If $\mathcal{I}(t_i)$ for $i > 1$ contains only one interaction I , and no characters outside the interaction start at t_i (i.e., $\text{AC}(t_i) \setminus \text{AC}(t_{i-1}) \subseteq \text{CI}(t_i)$), we only include a part of the (LOP)-constraints for time step t_i using a representative character $c_w \in \text{char}(I)$:

From the set of (LOP)-constraints containing at least one character in $\text{AC}(t_i) \setminus \text{char}(I)$, we keep only those that contain exactly two characters in $\text{AC}(t_i) \setminus \text{char}(I)$ and the representative character $c_w \in \text{char}(I)$. This is sufficient, because we can define the order of the active characters in t_i relative to the order of the characters in the interaction I based on Theorem 7. Hence, let c_w be a representative character from the set $\text{char}(I)$, and consider a pair of characters $c_u, c_v \in \text{AC}(t_i) \setminus \text{char}(I)$. By Theorem 7, if both c_u and c_v are above or below c_w , then their relative order can be fixed by their relative order at t_{i-1} . Otherwise, their relative order is already given by their relative order to c_w . That is, if, e.g., c_u is above c_w and c_v is below c_w , then we know that c_u is above c_v . To ensure the above, we add the following constraints in addition to the (LOP)-constraints for c_u, c_v , and c_w at time step t_i .

$$x_{i,u,v} \geq x_{i-1,u,v} + x_{i,u,w} + x_{i,v,w} - 2 \quad (\text{PROP-R1})$$

$$x_{i,u,v} \geq x_{i-1,u,v} + x_{i,w,u} + x_{i,w,v} - 2 \quad (\text{PROP-R2})$$

The two constraints ensure that c_u is above c_v if the requirements are met. By switching c_u and c_v , these constraints also ensure the case that c_u is below c_v .

If additionally $\text{char}(I) \subseteq \text{AC}(t_{i-1})$ we can apply Theorem 7 (3) to further reduce the number of those (LOP)-constraints, whose triples are taken from the set $\text{char}(I)$: In this case, we do not add any of the (LOP)-constraints for the characters in I , but instead for each pair $c_u, c_v \in \text{char}(I)$, we add the following constraint ensuring that the relative order of c_u and c_v is the same for t_i and t_{i-1} .

$$x_{i,u,v} = x_{i-1,u,v} \quad (\text{PROP-I})$$

If both reductions for (LOP) apply, we get a quadratic rather than cubic number of constraints for t_i . We call this formulation *propagated linear order* (PLO). Note that this idea of reducing the number of (LOP)-constraints also works for any of the other standard ILP models.

► **Theorem 8** (\star). *Every optimal solution to the formulation (PLO) corresponds to a crossing minimum storyline drawing.*

Proof sketch. Since we have only reduced some of the (LOP)-constraints, the correctness follows by induction on the time steps w.r.t. transitivity of the computed character order. ◀

5.2 Symmetry Breaking Constraints

We introduce the set (SBC) of *symmetry breaking constraints* that are based on Corollary 6 and might improve the solving process of the models, as they constitute equalities:

31:10 Revisiting ILP Models for Exact Crossing Minimization in Storyline Drawings

- We can assume that a crossing-minimum solution is type-1 consistent. Thus let $I \in \mathcal{I}$ with $t_i = \text{time}(I)$ and let $j(I)$ be defined as in Definition 2. For all pairs $c_u, c_v \in \text{char}(I)$ and all $j(I) \leq k < i$ we can add the following constraint enforcing type-1 consistency:

$$x_{k,u,v} = x_{i,u,v} \quad (\text{SBC-1})$$

- We can assume that a crossing-minimum solution is type-2 consistent. Thus, let $I_1, I_2 \in \mathcal{I}$ be two distinct interactions satisfying the properties of Definition 3. Let $i = \text{time}(I_1)$ and $j = \text{time}(I_2)$. For all $i < k < j$, all pairs $c_u, c_v \in \text{char}(I_1)$, and all $c_w \in \text{AC}(t_k) \setminus \text{char}(I_1)$ we add the following constraint, enforcing type-2 consistency:

$$x_{k,u,w} = x_{k,v,w} \quad (\text{SBC-2})$$

6 Implementation

In this section, we discuss relevant implementation details and new heuristic-based approaches to improve our algorithms.

6.1 Initial Heuristic

We propose a heuristic that is provided to the ILP solver as starting solution: Roughly, we solve intervals of the instance consisting of $\hat{\ell} < \ell$ consecutive time steps optimally using the (PLO) ILP formulation. The first $\hat{s} \leq \hat{\ell}$ layers of those are saved as the heuristic solution. Then, the last layer is fixed (i.e. layer \hat{s}) by fixing the ordering variables accordingly, in order to compute the solution for the next interval, and so on. Initial testing showed that $\hat{s} = 5$ and $\hat{\ell} = 30$ yields a good tradeoff between runtime and solution quality. A more detailed description is given in the full version [6].

6.2 Rounding and Local Improvement Heuristics

We propose a rounding heuristic that exploits fractional LP-solutions. Furthermore, we try to improve these solutions as well as incumbent solutions found by the solver software by proposing three local improvement heuristics **Rem-DC**, **Push-CR**, and **SL-Bary**.

Rounding Heuristic. We propose a strategy to round fractional solutions of the ordering variables to valid integer solutions corresponding to a drawing of the storyline instance. Roughly, the orders π_i are computed by considering for each ordering variable $x_{i,u,v}$ its rounded up or down value if the fractional value is different from 0.5 in the LP solution, and propagating the relative order of u and v in π_{i-1} if the fractional value is 0.5. Then, the characters are sorted by the sums of the rounded values, characters belonging to the same interaction are “treated as one character”. A detailed description is given in the full version [6].

Local Improvement Heuristics.

Rem-DC (remove double crossings) This heuristic finds pairs of characters that cross twice, and both crossings can be removed without increasing the total number of crossings. Formally, this is possible for a drawing S and two characters c, c' if there exist $1 \leq i < j \leq \ell$ with $j - i > 1$ such that

- c and c' cross between t_i and t_{i+1} , and t_{j-1} and t_j , and
- for all $k \in \mathbb{N}$ with $i < k < j$, c and c' either belong to the same interaction in t_k , or they both are in no interaction for t_k .

Then for all k as above we can exchange c and c' in π_k . This removes the double-crossing between c and c' and further does not introduce new crossings.

Push-CR This heuristic proceeds from $i = 2, \dots, \ell$ in this order and tries to push crossings between π_{i-1} and π_i forward by one time step: Let C be a maximal set of characters such that (1) all characters in C appear consecutively in π_i , (2) $C \subseteq AC(t_{i-1}, t_i)$, and (3) all characters in C either appear in the same interaction in t_i or no character in C is part of an interaction. For each such set of characters C we replace in π_i , $\pi_i[C]$ by $\pi_{i-1}[C]$. By similar arguments as in Section 4 this never increases the number of crossings.

Bary-SL Lastly, we describe a variant of the barycenter heuristic [25] for storylines that iteratively improves a storyline drawing by updating π_i for $1 \leq i \leq \ell$ based on π_{i-1} and π_{i+1} or one of them if not both exist. It is only applied to π_i if $|\mathcal{I}(t_i)| = 1$. Informally, we say that a pair of characters c, c' is comparable if c and c' have the same relative order in π_{i-1} and π_{i+1} . We compute an ordering π_i such that most comparable pairs have the same relative order in $\pi_{i-1}, \pi_i, \pi_{i+1}$ as follows. We compute the directed auxiliary graph G_C whose vertex set is a subset S of $AC(t_i)$ and which contains an arc from c to c' for each comparable pair c and c' such that c is before c' in π_{i-1} and π_{i+1} . Then, an order of S is built by iteratively selecting the vertex from G_C with the fewest incoming arcs. We also ensure that characters c that are not part of I are above or below the characters in I depending on which option leads to fewer crossings between c and $\text{char}(I)$ with respect to the considered time steps t_{i-1}, t_i, t_{i+1} . The algorithm computing the order based on the graph G_C is then applied to the characters in the interaction yielding π_I , and those not in the interaction yielding π_C , respectively. The ordering π_C is inserted into the maximum position of π_I such that all characters before π_C “prefer” being above the interaction with regard to crossings with $\text{char}(I)$. The new π_i is only accepted if it decreases the number of crossings.

Both **Bary-SL** and **Push-CR** are applied successively to layers $2, \dots, \ell$. This is repeated five times and applied to valid integer solutions found by the solver and the rounding heuristic described above. If enabled, the rounding heuristic is applied to every LP solution found by the solver. **Rem-DC** is applied five times to each pair of characters.

6.3 Max-Cut Implementation Details

Since the original implementation of Gronemann et al. [11] is not available, we provide our own implementation that was optimized beyond their algorithm. After reading the input, we first find an initial starting solution by applying adapted barycenter techniques as described in [11]. We start the root relaxation with the objective function and the tree constraints (TRC) as the only constraints, and start separating the odd cycle (CYC) (as suggested by Charfreitag et al. [4]) and the (LOPC) constraints. The (LOPC) constraints are separated by complete enumeration. Whenever a new LP solution is available, all nonbinding inequalities are eliminated, and we try to exploit the information in the (fractional) solution in order to obtain a better incumbent solution.

The root phase ends when no violated inequalities are found. Then the branch-and-cut phase is started by changing the variable types from continuous in the interval $[0, 1]$ to binary. In the Gurobi “MIPSOL” callbacks at branch-and-cut nodes with an integer solution, we check if the integral solution is the characteristic vector of a storyline drawing. If so and if the number of crossings is lower than the one of the incumbent solution, the latter is updated, otherwise, the exact (CYC) and (LOPC) separators are called to provide violated inequalities that are passed to Gurobi as *lazy constraints*. In the Gurobi “MIPNODE” callbacks it is tried to exploit the fractional solution for a possible update of the incumbent solution.

6.4 Implementation of the ILP Models

The models (QDR), (LIN), and (PLO) include many symmetries regarding ordering variables and crossing variables. For each $t_i \in T$ and pair of characters $c_u, c_v \in AC(t_i)$ we only keep the ordering variables $x_{i,u,v}$ and crossing variables $y_{i,u,v}$ with $u < v$. The constraints are adjusted with standard projections [11, 12].

We take the linearized model (LIN) as basis for our refined ILP model described in Section 5, because preliminary experiments showed no performance gain from refining (QDR) instead. Further, the linearized model (LIN) is competitive with the max-cut approach when implemented in Gurobi. Therefore, we decided on refining the linearized model that is simpler to implement and more accessible when compared with the max-cut approach. Furthermore, implementing any of the ILP models naively includes up to $\mathcal{O}(\ell n^3)$ (LOP)-constraints in the model. We have experimented with adding these constraints during a cutting-plane approach and also by including them into the Gurobi solver as *lazy constraints*, i.e., constraints that the solver can decide to include at later stages during the solving process. We decided to always add (LOP) as lazy constraints, as this leads to the best performance. Hence, we consider the following algorithms for our experimental evaluation.

- **MC**: the max-cut formulation (as a baseline) implemented as described in Section 6.3
- **LIN**: the linearized model (LIN) with (LOP)-constraints included as lazy constraints
- **QDR**: the quadratic model (QDR) with (LOP)-constraints included as lazy constraints
- **PLO**: the PLO formulation with (LOP)-constraints included as lazy constraints

The latter three algorithms are by default extended with the symmetry breaking constraints described in Section 5.2 (**SBC**), the initial heuristic from Section 6.1 (**INIT**), and the rounding and local improvement heuristics from Section 6.2 (**RND**). This is not done for **MC**, as it should serve as a state-of-the-art baseline and allow comparison with Gronemann et al. [11].

7 Experiments and Evaluation

In our experimental evaluation, we are interested in the following research questions.

- Q1**: Does the algorithm **PLO** based on our new ILP model dominate the state-of-the-art model **MC**? Will we be able to solve hard instances that have not been solved to optimality before? How do the various algorithms compare to each other?
- Q2**: What effect do the structural insights have when applied to the **LIN**-formulation?
- Q3**: What is the effect of the newly introduced components **SBC**, **INIT**, and **RND**?

In the following we describe our experimental setup, our benchmark instances, and the results of our study. We also provide our results and analysis on <https://osf.io/3bua2/>.

7.1 Setup

Systems employed for all experiments have AMD EPYC 7402, 2.80GHz 24-core CPUs and 1024GB of RAM, running Ubuntu 18.04.6 LTS; experiments were run using a single thread.

MC is implemented in C and compiled with gcc 7.5.0, GNU make 4.1, and flag `-O3`, all remaining code is written in C++17, compiled with cmake 3.10.2 and g++ 11.4.0 in `Release` mode. To solve the ILPs we used Gurobi 11.0.1. The time limit is 3600s (same as Gronemann et al. [11]) and the memory limit is 16GB for all experiments. We do not know the memory limit for Gronemann, however memory was certainly not our limiting factor. The time for the initial heuristic is negligible ($< 1\%$ of the overall runtime), so it is not counted towards the solving time.

To mitigate performance variability, we ran each instance-setting combination with five different seeds provided to Gurobi; data displayed below corresponds to the seed with the median runtime. With this, an instance counts as “solved in the time limit”, if the majority of the five runs does not time out. Source code for the new formulations is available on <https://osf.io/3bua2/>.

7.2 Test Data

The instances used in our computational study are taken from the literature [7, 10, 11, 26]. However, we also present a new data set, including existing instances, in a specifically designed storyline data format, together with tools for transformation and visualization of the storyline layouts on <https://osf.io/3bua2/>. We also provide data on best known crossing numbers.

The existing instances from Gronemann et al. [11] consist of three book instances from the Stanford GraphBase database [16], i.e., *Anna Karenina* (anna), *Les Misérables* (jean) and *Adventures of Huckleberry Finn* (huck), and the movie instances TheMatrix, Inception, and StarWars. The instances gdea10, gdea20 from Dobler et al. [7] consist of publication data from 10 (resp. 20) authors from the GD conference. The publication instances ubiq1, ubiq2 are from Di Giacomo et al. [10]. Furthermore, anna and jean are split up into slices of 1-4 consecutive chapters as was done by Gronemann et al. [11]¹. The new instances consist of scenes from nine blockbuster movies, namely Avatar, Back to The Future, Barbie, Forrest Gump, Harry Potter 1, Jurassic Park, Oceans 11, Oppenheimer, and Titanic.

In all 59 resulting instances, characters are active from their first interaction to their last interaction, and most instances have one interaction per time step.

7.3 Evaluation

Several instances could be solved within 30s by all algorithms, others could not be solved within the time limit by any of the algorithms. The three instances TheMatrix, Inception, and StarWars used commonly in heuristic storyline visualization were all solved within 450ms. We exclude all these instances and focus on the 23 challenging instances that remain. Out of these, the maximum number of characters is 88, and the maximum number of layers is 234. An extended experimental evaluation and more detailed statistics are given in the full version [6].

Answering Q1 and Q2. Figure 3 displays the number of instances solved over time for each algorithm. We observe that the algorithms differ in their ability to solve challenging instances: **PLO** solves the most, followed by **LIN** and **MC**, with **QDR** last. In fact, **PLO** solves one instance that cannot be solved by any other algorithm, and additionally solves six instances that could not be solved by Gronemann et al. [11] and three more than **MC** within the same time limit. Hence, we answer the first part in **Q1** positively. For further illustration, the exact runtimes per instance are also shown in Figure 4.

Answering **Q2**, the structural insights as applied in **PLO** reduce the number of constraints by a factor of five on average, comparing **LIN** and **PLO**. More so, they enhance Gurobi’s capabilities of strengthening the LP relaxation, as the two instances not solved by **LIN** are

¹ We could not replicate this process fully equivalently, as sometimes our optimal crossing numbers are different to those of Gronemann et al. [11].

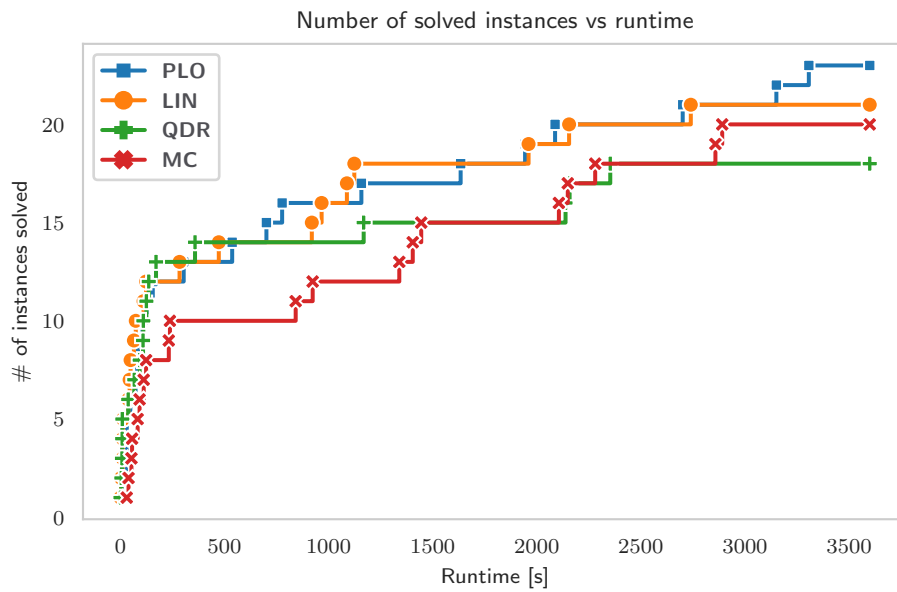


Figure 3 Number of solved instances per time limit given.

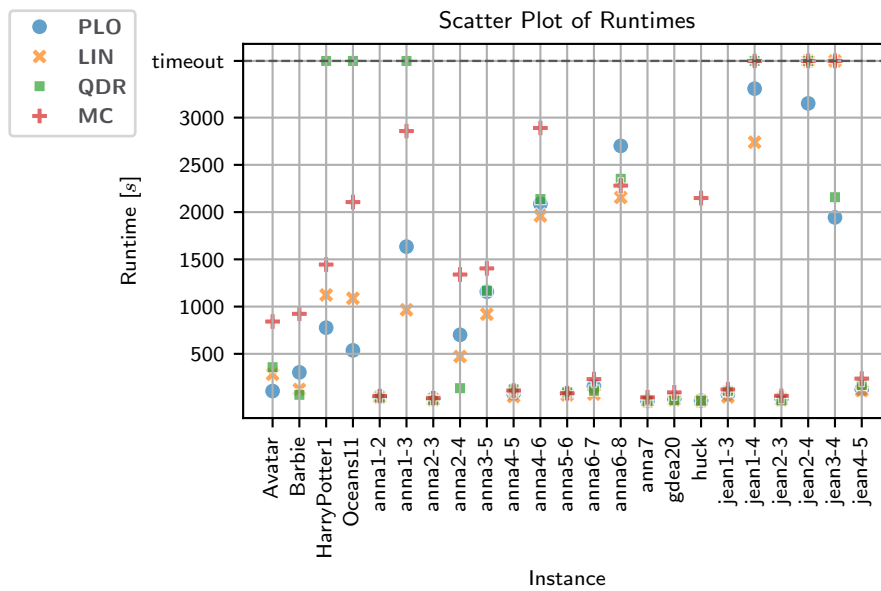
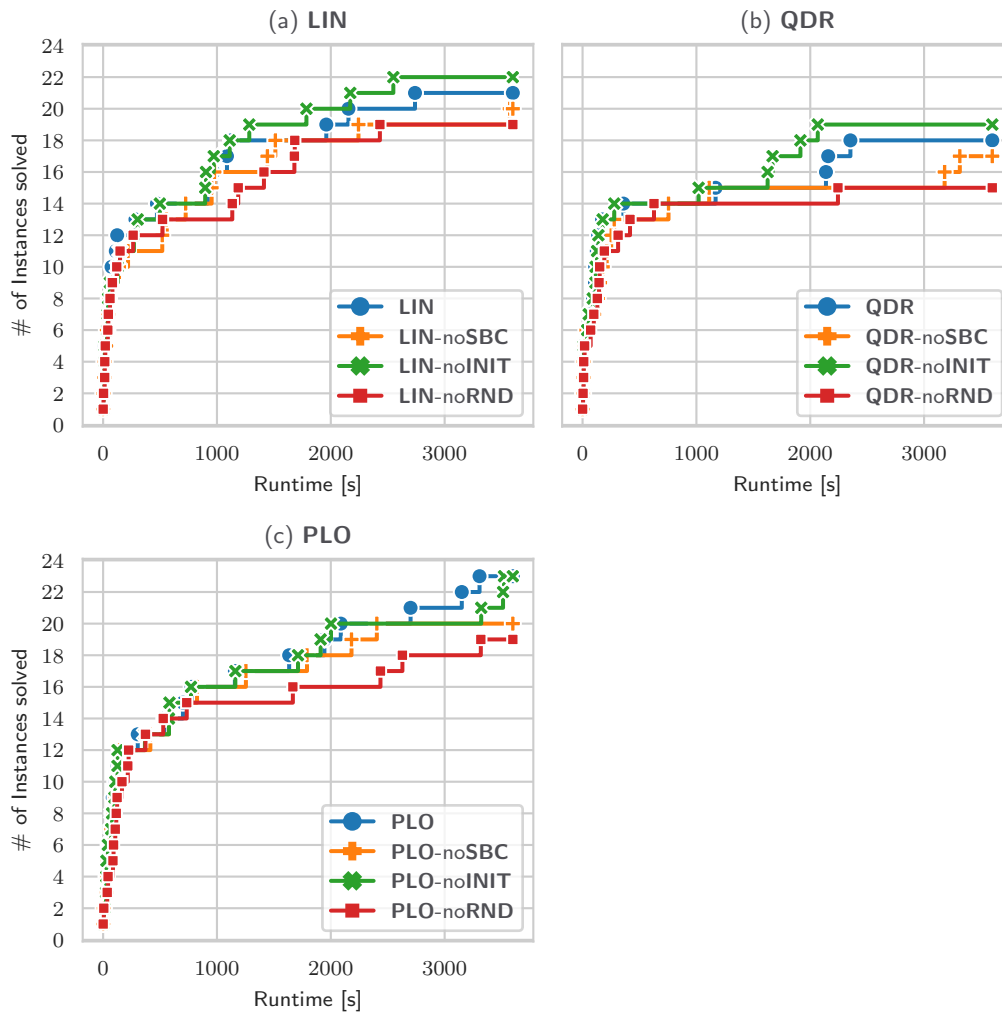


Figure 4 Runtimes of algorithms per instance.



■ **Figure 5** Number of instances solved per time limit given, broken down by algorithm. Algorithms are compared with their counterparts where exactly one component is disabled.

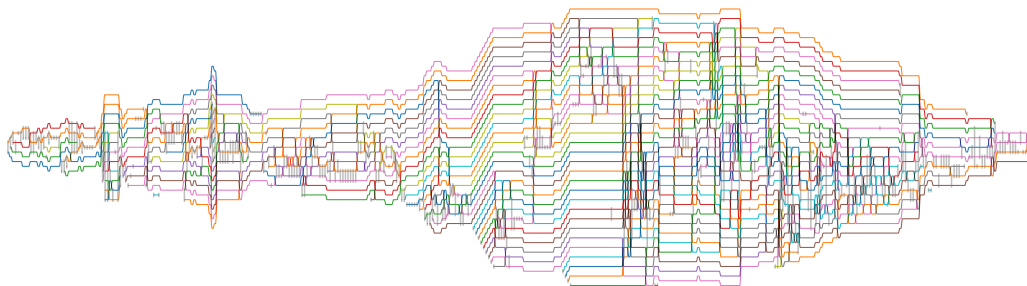
solved by **PLO** in the root, while **LIN** starts branching early and times out. **QDR** enters branching in all 23 instances, **PLO** in two, **MC** in three, **LIN** in seven instances. **PLO** solves 21 out of 23 instances in the root, the remaining two with branching.

Furthermore, we computed the *speedup factor* of **PLO**, **LIN**, and **QDR**, when compared with **MC** on instances where both respective algorithms did not time out. This factor is the runtime of **MC** divided by the runtime of, e.g., **PLO**. The geometric means of these values are 2.6 for **PLO**, 3.2 for **LIN**, and 2.7 for **QDR**. Hence, our new algorithms are 2.6–3.2 times faster than the state-of-the-art algorithm **MC**.

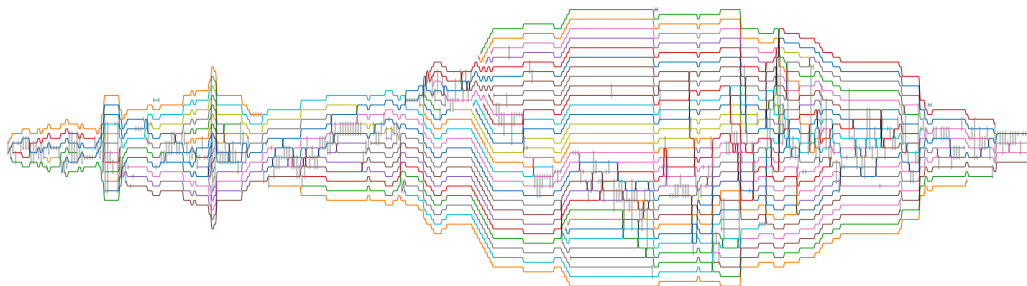
Ablation study to answer Q3. We conduct an ablation study to discern the impact each of the methods proposed in Sections 5 and 6 has on the algorithms' performance. To this end, we enable all the proposed methods as the baseline configuration for **PLO**, **LIN**, and **QDR**, namely **SBC**, **INIT** and **RND**. Then, each component is disabled one at a time to measure the component's impact on overall performance. In Table 1 we present the speedup factors of

■ **Table 1** Geometric means of the speedup factor of each baseline algorithm vs. its counterpart where the respective component is disabled.

speedup factor	PLO	LIN	QDR
SBC	1.12	1.35	1.42
INIT	1.05	1.09	0.97
RND	1.50	1.37	1.52



(a) Solution with 765 crossings computed in 0.57 s by a greedy heuristic.



(b) With the minimum number 244 of crossings computed in ≈ 7 hours.

■ **Figure 6** Storyline of Les Misérables (*jean*) with 80 characters and 402 layers.

the algorithms vs. their counterparts with the specific component disabled. From this table, we conclude that **SBC** and the **RND** are beneficial for all algorithms, while **INIT** has a small to no noticeable impact. This is further supported by Figure 5, which shows that disabling **SBC** or **RND**, results in all the formulations solving fewer instances (curves with **noSBC** and **noRND** are below the baseline). This is because **SBC** introduces equalities between two variables, and hence improve presolving capabilities and reduce the search space that solvers have to explore. The heuristics of **RND** help the solver find optimal solutions early in the process. This answers **Q3**.

Large Instances. Finally, we demonstrate that our implementations are capable of solving even very large instances to proven optimality: Figure 6(a) shows the raw drawing of the data for Victor Hugo’s Les Misérables [15] as provided in the data file *jean.dat* of the Stanford GraphBase [16]. After roughly 7 hours of single thread computation, we obtained the proven crossing minimum layout shown in Figure 6(b).

8 Conclusion and Future Work

As shown in our experimental study, our new methods and algorithms dominate the state-of-the-art algorithms and are able to solve large instances to optimality, while the newly introduced improvements are beneficial towards all considered formulations. We observe two directions for future work.

- Our new components for improvement could be implemented into the max-cut formulation. However, initial experiments have shown that the simple linearized formulation (LIN) performs comparably to the more complex max-cut formulation, hence we expect that this will result in a negligible or no improvement over our proposed formulations.
- Out of our 59 instances (see <https://osf.io/3bua2/>) we were able to solve 55 when increasing the time limit. The remaining four unsolved instances should pose a challenge to engineer new exact methods for crossing minimization in storylines.




References

- 1 Vanessa Peña Araya, Tong Xue, Emmanuel Pietriga, Laurent Amsaleg, and Anastasia Bezerianos. Hyperstorylines: Interactively untangling dynamic hypergraphs. *Inf. Vis.*, 21(1):38–62, 2022. doi:10.1177/14738716211045007.
- 2 Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, 1999.
- 3 Christoph Buchheim, Angelika Wiegele, and Lanbo Zheng. Exact algorithms for the quadratic linear ordering problem. *INFORMS J. Comput.*, 22(1):168–177, 2010. doi:10.1287/IJOC.1090.0318.
- 4 Jonas Charfreitag, Michael Jünger, Sven Mallach, and Petra Mutzel. *McSparse: Exact Solutions of Sparse Maximum Cut and Sparse Unconstrained Binary Quadratic Optimization Problems*, pages 54–66. Proc. Symposium on Algorithm Engineering and Experiments (ALENEX'22), 2022. doi:10.1137/1.9781611977042.5.
- 5 Alexander Dobler, Michael Jünger, Paul J. Jünger, Julian Meffert, Petra Mutzel, and Martin Nöllenburg. Revisiting ILP Models for Exact Crossing Minimization in Storyline Drawings: Supplementary Material. Software (visited on 2024-10-14). URL: <https://doi.org/10.17605/OSF.IO/3BUA2>.
- 6 Alexander Dobler, Michael Jünger, Paul Jünger, Julian Meffert, Petra Mutzel, and Martin Nöllenburg. Revisiting ILP models for exact crossing minimization in storyline drawings. *CoRR*, abs/2409.02858, 2024. doi:10.48550/arXiv.2409.02858.
- 7 Alexander Dobler, Martin Nöllenburg, Daniel Stojanovic, Anaïs Villedieu, and Jules Wulms. Crossing minimization in time interval storylines. *CoRR*, abs/2302.14213, 2023. doi:10.48550/arXiv.2302.14213.
- 8 Michael Forster. A fast and simple heuristic for constrained two-level crossing reduction. In János Pach, editor, *Proc. Graph Drawing and Network Visualization (GD'04)*, volume 3383 of *LNCS*, pages 206–216. Springer, 2004. doi:10.1007/978-3-540-31843-9_22.
- 9 Michael R. Garey and David S. Johnson. Crossing number is NP-complete. *SIAM J. on Algebraic and Discrete Methods*, 4(3):312–316, 1983. doi:10.1137/0604033.
- 10 Emilio Di Giacomo, Walter Didimo, Giuseppe Liotta, Fabrizio Montecchiani, and Alessandra Tappini. Storyline visualizations with ubiquitous actors. In David Auber and Pavel Valtr, editors, *Proc. Graph Drawing and Network Visualization (GD'20)*, volume 12590 of *LNCS*, pages 324–332. Springer, 2020. doi:10.1007/978-3-030-68766-3_25.
- 11 Martin Gronemann, Michael Jünger, Frauke Liers, and Francesco Mambelli. Crossing minimization in storyline visualization. In Yifan Hu and Martin Nöllenburg, editors, *Proc. Graph Drawing and Network Visualization (GD'16)*, volume 9801 of *LNCS*, pages 367–381. Springer, 2016. doi:10.1007/978-3-319-50106-2_29.

- 12 Martin Grötschel, Michael Jünger, and Gerhard Reinelt. Facets of the linear ordering polytope. *Math. Program.*, 33(1):43–60, 1985. doi:10.1007/BF01582010.
- 13 Patrick Healy and Nikola S. Nikolov. Hierarchical drawing algorithms. In Roberto Tamassia, editor, *Handbook on Graph Drawing and Visualization*, chapter 13, pages 409–453. Chapman and Hall/CRC, 2013.
- 14 Tim Herrmann. Storyline-Visualisierungen für wissenschaftliche Kollaborationsgraphen. Master’s thesis, Universität Würzburg, 2022. URL: <https://www1.pub.informatik.uni-wuerzburg.de/pub/theses/2022-herrmann-masterarbeit.pdf>.
- 15 Victor Hugo. *Les Misérables*. Jules Rouff et Cie editeurs, Paris, 1862.
- 16 Donald E. Knuth. *The Stanford GraphBase – A platform for combinatorial computing*. ACM, 1993.
- 17 Irina Kostitsyna, Martin Nöllenburg, Valentin Polishchuk, André Schulz, and Darren Strash. On minimizing crossings in storyline visualizations. In Emilio Di Giacomo and Anna Lubiw, editors, *Proc. Graph Drawing and Network Visualization (GD’15)*, volume 9411 of *LNCS*, pages 192–198. Springer, 2015. doi:10.1007/978-3-319-27261-0_16.
- 18 Shixia Liu, Yingcai Wu, Enxun Wei, Mengchen Liu, and Yang Liu. Storyflow: Tracking the evolution of stories. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2436–2445, 2013. doi:10.1109/TVCG.2013.196.
- 19 Randall Munroe. Movie narrative charts, 2009. URL: <https://xkcd.com/657/>.
- 20 Michael Ogawa and Kwan-Liu Ma. Software evolution storylines. In Alexandru C. Telea, Carsten Görg, and Steven P. Reiss, editors, *Software Visualization (SoftVis’10)*, pages 35–42. ACM, 2010. doi:10.1145/1879211.1879219.
- 21 Helen Purchase. Which aesthetic has the greatest effect on human understanding? In *Proc. Graph Drawing and Network Visualization (GD’97)*, volume 1353 of *LNCS*, pages 248–261. Springer, 1997. doi:10.1007/3-540-63938-1_67.
- 22 Marcus Schaefer. *Crossing Numbers of Graphs*. CRC Press, 2018.
- 23 Yang Shi, Chris Bryan, Sridatt Bhamidipati, Ying Zhao, Yaoyue Zhang, and Kwan-Liu Ma. Meetingvis: Visual narratives to assist in recalling meeting context and content. *IEEE Trans. Vis. Comput. Graph.*, 24(6):1918–1929, 2018. doi:10.1109/TVCG.2018.2816203.
- 24 Caterina De Simone. The cut polytope and the boolean quadric polytope. *Discret. Math.*, 79(1):71–75, 1990. doi:10.1016/0012-365X(90)90056-N.
- 25 Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiro Toda. Methods for visual understanding of hierarchical system structures. *IEEE Trans. Syst. Man Cybern.*, 11(2):109–125, 1981. doi:10.1109/TSMC.1981.4308636.
- 26 Y. Tanahashi. Movie data set, 2013. URL: http://vis.cs.ucdavis.edu/~tanahashi/datadownloads/storylinevisualizations/story_data.tar.
- 27 Yuzuru Tanahashi, Chien-Hsin Hsueh, and Kwan-Liu Ma. An efficient framework for generating storyline visualizations from streaming data. *IEEE Trans. Vis. Comput. Graph.*, 21(6):730–742, 2015. doi:10.1109/TVCG.2015.2392771.
- 28 Yuzuru Tanahashi and Kwan-Liu Ma. Design considerations for optimizing storyline visualizations. *IEEE Trans. Vis. Comput. Graph.*, 18(12):2679–2688, 2012. doi:10.1109/TVCG.2012.212.
- 29 Tan Tang, Renzhong Li, Xinke Wu, Shuhan Liu, Johannes Knittel, Steffen Koch, Lingyun Yu, Peiran Ren, Thomas Ertl, and Yingcai Wu. Plotthread: Creating expressive storyline visualizations using reinforcement learning. *IEEE Trans. Vis. Comput. Graph.*, 27(2):294–303, 2021. doi:10.1109/TVCG.2020.3030467.
- 30 Tan Tang, Sadia Rubab, Jiewen Lai, Weiwei Cui, Lingyun Yu, and Yingcai Wu. iStoryline: Effective convergence to hand-drawn storylines. *IEEE Trans. Vis. Comput. Graph.*, 25(1):769–778, 2019. doi:10.1109/TVCG.2018.2864899.
- 31 Thomas C. van Dijk, Martin Fink, Norbert Fischer, Fabian Lipp, Peter Markfelder, Alexander Ravsky, Subhash Suri, and Alexander Wolff. Block crossings in storyline visualizations. *J. Graph Algorithms Appl.*, 21(5):873–913, 2017. doi:10.7155/JGAA.00443.

- 32 Thomas C. van Dijk, Fabian Lipp, Peter Markfelder, and Alexander Wolff. Computing storyline visualizations with few block crossings. In Fabrizio Frati and Kwan-Liu Ma, editors, *Proc. Graph Drawing and Network Visualization (GD'17)*, volume 10692 of *LNCS*, pages 365–378. Springer, 2017. doi:10.1007/978-3-319-73915-1_29.
- 33 Günter Wallner, Letian Wang, and Claire Dormann. Visualizing the spatio-temporal evolution of gameplay using storyline visualization: A study with league of legends. *Proc. ACM Hum. Comput. Interact.*, 7(CHI):1002–1024, 2023. doi:10.1145/3611058.
- 34 Yunchao Wang, Guodao Sun, Zihao Zhu, Tong Li, Ling Chen, and Ronghua Liang. *E*²Storyline: Visualizing the relationship with triplet entities and event discovery. *ACM Trans. Intell. Syst. Technol.*, 15(1):16:1–16:26, 2024. doi:10.1145/3633519.


On the Complexity of Recognizing k^+ -Real Face Graphs

Michael A. Bekos   




University of Ioannina, Greece

Giuseppe Di Battista   




University of Roma Tre, Italy

Emilio Di Giacomo   

University of Perugia, Italy

Walter Didimo   

University of Perugia, Italy

Michael Kaufmann   

University of Tübingen, Germany

Fabrizio Montecchiani   

University of Perugia, Italy

Abstract

A nonplanar drawing Γ of a graph G divides the plane into topologically connected regions, called *faces* (or *cells*). The boundary of each face is formed by vertices, crossings, and edge segments. Given a positive integer k , we say that Γ is a k^+ -real face drawing of G if the boundary of each face of Γ contains at least k vertices of G . The study of k^+ -real face drawings started in a paper by Binucci et al. (WG 2023), where edge density bounds and relationships with other beyond-planar graph classes are proved. In this paper, we investigate the complexity of recognizing k^+ -real face graphs, i.e., graphs that admit a k^+ -real face drawing. We study both the general unconstrained scenario and the 2-layer scenario in which the graph is bipartite, the vertices of the two partition sets lie on two distinct horizontal layers, and the edges are straight-line segments. We give NP-completeness results for the unconstrained scenario and efficient recognition algorithms for the 2-layer setting.

2012 ACM Subject Classification Mathematics of computing \rightarrow Graph algorithms; Mathematics of computing \rightarrow Graph theory; Theory of computation \rightarrow Graph algorithms analysis

Keywords and phrases Beyond planarity, k^+ -real face drawings, 2-layer drawings, recognition algorithm, NP-hardness

Digital Object Identifier 10.4230/LIPIcs.GD.2024.32

Funding *Giuseppe Di Battista*: MUR PRIN Proj. 2022TS4Y3N – EXPAND: scalable algorithms for EXPLoratory Analyses of heterogeneous and dynamic Networked Data.

Walter Didimo: AIDMIX – Artificial Intelligence for Decision Making: Methods for Interpretability and eXplainability – Ricerca di Base 2021. RASTA: Realtà Aumentata e Story-Telling Automatizzato per la valorizzazione di Beni Culturali ed Itinerari; Italian MUR PON Proj. ARS01_00540.

Fabrizio Montecchiani: MUR PRIN Proj. 2022ME9Z78 – NextGRAAL: Next-generation algorithms for constrained GRAph visuALization.

Acknowledgements Research started at the Dagstuhl Seminar: Beyond-Planar Graphs: Models, Structures and Geometric Representations; seminar number: 24062, February 4-9, 2024.

1 Introduction

Drawing nonplanar graphs in the plane while avoiding forbidden crossing configurations is a prominent line of research in graph drawing. Over the past twenty years, numerous papers have addressed this topic, commonly recognized as *beyond-planar graph drawing*. For



© Michael A. Bekos, Giuseppe Di Battista, Emilio Di Giacomo, Walter Didimo, Michael Kaufmann, and Fabrizio Montecchiani;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 32; pp. 32:1–32:22



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

example, one of the most studied graph hierarchies in this context is the one of *k-planar graphs*, i.e., graphs that admit a drawing in which each edge is crossed at most k times (see, e.g., [24, 26]). Another classical example is the class of *right-angle-crossing graphs (RAC graphs)*, which admit a straight-line drawing where any two crossing edges form 90° angles at their crossing point (see, e.g., [11, 12]). We refer the reader to surveys, books, and seminar reports for a comprehensive overview on beyond-planar graph drawing [13, 18, 19, 21].

In this paper, we study *k^+ -real face graphs*, a beyond-planar graph hierarchy recently introduced in [5, 6] and further studied in [23]. Namely, a nonplanar graph Γ of a graph G divides the plane into topologically connected regions, called *faces* (or *cells*). The boundary of each face is formed by vertices, crossings, and edge segments. Given a positive integer k , the drawing Γ is a *k^+ -real face drawing* of G if the boundary of each face of Γ contains at least k vertices of G . In this case, G is a *k^+ -real face graph*. The research in [5, 6, 23] concentrates on providing tight bounds on the edge density of k^+ -real face graphs, and on establishing relationships between k^+ -real face graphs and other prominent beyond-planar graph classes. Conversely, testing which graphs admit a k^+ -real face drawing and computing such a drawing when the test is positive are almost unexplored problems. Trivial recognition algorithms exist only for complete graphs and complete bipartite graphs, exploiting results about edge density and crossing numbers for these families [5].

Contribution. Following a consolidated line of research in beyond-planar graph drawing, and addressing a problem mentioned in [5], we investigate the complexity of recognizing k^+ -real face graphs. We study both the general unconstrained scenario and the classical 2-layer scenario for bipartite graphs, in which the vertices of the two partition sets are placed on two distinct horizontal lines and the edges are drawn as straight-line segments. We remark that the 2-layer scenario has a long tradition in graph drawing (see, e.g., [1, 15, 16, 22, 27, 28]) and in beyond-planar graph drawing (see, e.g., [3, 4, 7, 9, 10]). Our results are as follows:

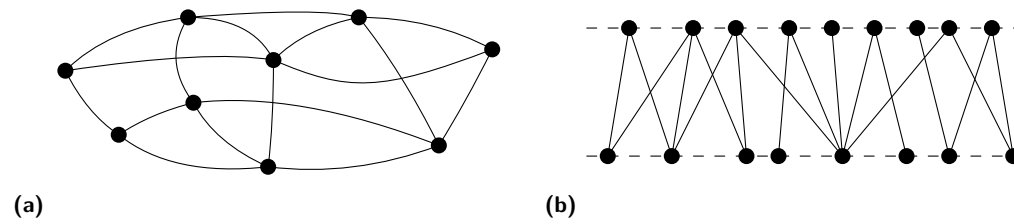
- We prove that, for the set of instances $\langle G, k \rangle$, where G is a graph and k is a positive integer, testing whether G admits a k^+ -real face drawing is NP-complete. More specifically, we prove that the problem is already NP-complete for $k \in \{1, 2\}$ and even if G is biconnected (Section 3). This excludes that recognizing k^+ -real face graphs is fixed-parameter tractable (FPT) or even slice-wise polynomial (XP) when parameterized by k , unless $P=NP$.
- We provide tight upper bounds on the edge density of 2-layer k^+ -real face graphs for any positive integer k (Section 4.1). Then, we describe linear-time algorithms for recognizing 2-layer k^+ -real face graphs for any $k \geq 2$, and for recognizing *optimal* 2-layer k^+ -real face graphs for any given $k \geq 1$ (Section 4.2). The optimal graphs are those that match the maximum possible edge density. Recognizing optimal graphs for specific beyond-planar graph families is also a classical problem in graph drawing [2, 8, 20].

For space reasons, some proofs are sketched or omitted.

2 Basic Terminology and Tools

We consider connected, *simple* graphs, i.e., without parallel edges and self-loops. Given a graph $G = (V, E)$ and a set $E' \subseteq E$, let $V' \subseteq V$ be the set of the end-vertices of the edges in E' . The graph $G' = (V', E')$ is the subgraph of G *induced by E'* . A *block* B of G is a biconnected component of G . If B is an edge, it is a *trivial block*, otherwise B is *non-trivial*.

In a *drawing* Γ of a graph G the vertices are represented as points of the plane and the edges are simple Jordan arcs. We only consider *simple* drawings, that is: (i) adjacent edges do not intersect, except at their common endpoint; (ii) two independent (i.e., non-adjacent)



■ **Figure 1** (a) A 2^+ -real face drawing. (b) A 2-layer 2^+ -real face drawing.

edges intersect at most in one of their interior points, called a *crossing point*; and (iii) no three edges intersect at a common crossing point. A *vertex* of Γ is either a point corresponding to a vertex of G , called a *real-vertex*, or a crossing point, called a *crossing-vertex*. An *edge* of Γ is a curve connecting two vertices of Γ ; an edge of Γ whose endpoints are both real-vertices coincides with an edge of G ; otherwise, it is just a proper portion of an edge of G . A drawing Γ subdivides the plane into topologically connected regions, called *faces* (or *cells*). The boundary of each face is a circular sequence of vertices and edges of Γ . The face corresponding to the the infinite plane region is the *external face* of Γ ; the other faces are the *internal faces*.

For an integer $k \geq 0$, we say that f is a k -*real face* (resp. k^+ -*real face*) if it contains exactly (resp. at least) k real vertices. For an integer $k \geq 1$, a k^+ -*real face drawing* of a graph G is a drawing such that each face is a k^+ -real face (see Figure 1a for an example with $k = 2$). If G admits such a drawing, it is a k^+ -*real face graph*. A k^+ -real face graph whose number of edges is the maximum possible over all its number of vertices is said to be *optimal*.

2-layer drawings. In a *2-layer drawing* Γ of a bipartite graph $G = (V_1 \cup V_2, E)$, the vertices in V_1 and in V_2 lie on two distinct horizontal lines L_1 and L_2 , called *layers*, and the edges are straight-line segments. If Γ is also k^+ -real face then it is a *2-layer k^+ -real face drawing*, and G is a *2-layer k^+ -real face graph* (see Figure 1b for an example with $k = 2$). Again, a 2-layer k^+ -real face graph is *optimal* if it matches the maximum possible edge density.

Given a 2-layer drawing Γ of a graph G , we say that there is a *fan crossing* in Γ if two adjacent edges of G are crossed by a third one in Γ ; we also say that these three edges *form a fan crossing*. Further, Γ is a *2-layer RAC drawing* if any two crossing edges only cross at right angles. A *2-layer RAC graph* is a graph admitting a 2-layer RAC drawing.

Given a bipartite graph $G = (V_1 \cup V_2, E)$, let π_1 and π_2 be two linear (left-to-right) orderings of the vertices in V_1 and in V_2 , respectively. A *2-layer embedding* $\gamma = (\pi_1, \pi_2)$ of G is the equivalence class of 2-layer drawings of G that induce the orderings π_1 and π_2 . In other words, γ is an abstraction of a 2-layer drawing where only the vertex orderings on the layers matter, independent of the vertex coordinates. A *drawing of γ* is any 2-layer drawing of G in the class γ . If γ has a 2-layer RAC drawing, it is a *2-layer RAC embedding*. Analogously, if γ has a k^+ -real face drawing (for some $k \geq 1$), then γ is a *2-layer k^+ -real face embedding*. Note that, in fact, if γ is a 2-layer k^+ -real face embedding, *every* drawing of γ is a 2-layer k^+ -real face drawing. Indeed, it is not difficult to see that any two distinct drawings of the same 2-layer k^+ -real face embedding γ have the same set of faces, which is uniquely determined by the linear orderings of the vertices on the two layers¹ Hence, for a 2-layer k^+ -real face embedding γ , we will refer to the faces of γ to indicate the faces of any 2-layer drawing of γ . Similarly, the edge crossings of a graph G in a 2-layer drawing Γ of G only

¹ If a 2-layer drawing Γ has a face without real-vertices, its 2-layer embedding γ does not uniquely determine the set of faces, i.e., another drawing of γ may have a set of faces different from that of Γ .

depend on the 2-layer embedding γ of Γ . A crossing in γ between two edges of G refers to the crossing formed by these edges in any drawing of γ ; real- and crossing-vertices of γ refer to the real- and crossing-vertices of any drawing of Γ . Since two real-vertices of the same layer cannot belong to the same internal face of a 2-layer drawing, the next property holds.

► **Property 1.** *Let γ be a 2-layer embedding of a connected bipartite graph G . In every drawing of γ , each internal face has at most two real-vertices.*

A *caterpillar* is a tree such that removing all its leaves yields a path. A *ladder* is a bipartite outerplanar graph consisting of two paths of the same length $\langle u_1, \dots, u_{\frac{n}{2}} \rangle$ and $\langle w_1, \dots, w_{\frac{n}{2}} \rangle$ plus the edges (u_i, w_i) for $i = 1, \dots, \frac{n}{2}$; the edges (u_1, w_1) and $(u_{\frac{n}{2}}, w_{\frac{n}{2}})$ are the *extremal edges* of the ladder (see Figure 6). The next result will be used in the following.

► **Theorem 2** (Di Giacomo et al. [9]). *A 2-layer embedding γ of a bipartite graph is a 2-layer RAC embedding if and only if there is no fan crossings in γ .*

3 Recognizing Unconstrained k^+ -real face graphs

For optimal graphs, the results in [5, 6] imply that for $k \geq 3$, recognizing k^+ -real face graphs corresponds to recognizing planar graphs that have an embedding with all faces of degree k . This last problem is tractable for $k \leq 6$ whereas it is NP-complete for odd $k \geq 7$ and for even $k \geq 10$ [25]. Moreover, recognizing optimal 2^+ -real face graphs is equivalent to recognizing optimal 1-planar graphs [6], which is linear-time solvable [8]. In this section we prove that recognizing k^+ -real face graphs is NP-complete; in particular, the next theorem shows that the problem is NP-complete for $k = 2$ and even for biconnected graphs.

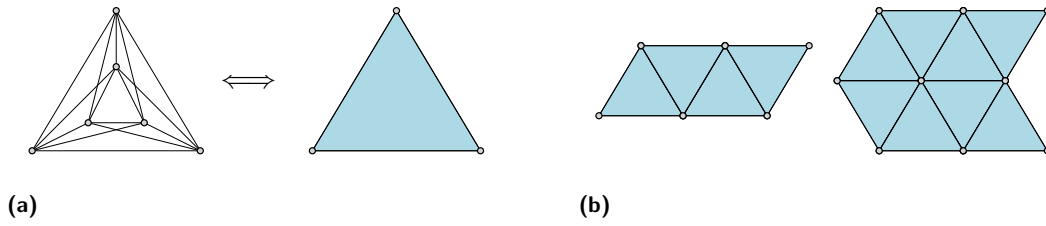
► **Theorem 3.** *Deciding if a graph G is 2^+ -real face is NP-complete, even if G is biconnected.*

Membership of the problem in NP can be easily verified using standard arguments. We reduce from the 3-PARTITION problem, which is known to be strongly NP-hard [17]. Recall that an instance of 3-PARTITION consists of a set $A = \{a_1, a_2, \dots, a_{3m}\}$ of $3m$ integers, each of which is strictly between $B/4$ and $B/2$, where $B = \frac{1}{m} \sum_{i=1}^{3m} a_i$. Then, the problem asks whether A can be partitioned into m subsets A_1, A_2, \dots, A_m , each of cardinality 3, such that the sum of the integers in each subset is B .

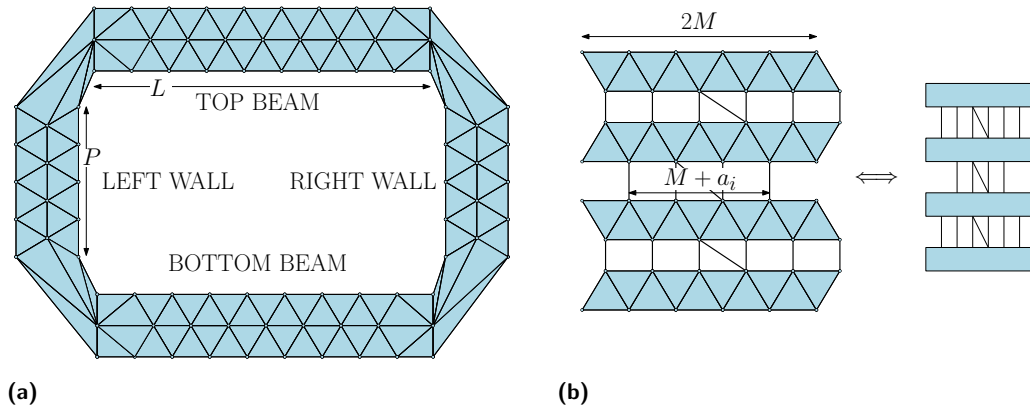
Proof overview. The idea is to construct a rigid frame which admits a unique 2^+ -real face drawing (up to a homeomorphism of the plane) and contains a large face. Inside this face, we arrange, in a grid-like fashion, $3m$ vertical gadgets and m horizontal paths. The former, called *columns*, encode the integers of an instance A of 3-PARTITION; see Figure 5a. If a 2^+ -real face drawing exists, one can read a solution for A by looking at how the paths intersect the columns; see Figure 5b. A crucial ingredient is an intertwined design of the columns and of the path gadgets such that the latter must cross the former in a controlled manner.

Construction. Let A be an instance of 3-PARTITION. We will construct in polynomial time a graph G , such that A admits a partition if and only if G admits a 2^+ -real face drawing. In our construction, we will leverage K_6 as a building block, since any 2^+ -real face drawing of it is 1-planar as we prove in the following lemma; refer to Figure 2a for an illustration.

► **Lemma 4.** *Any 2^+ -real face drawing of K_6 is 1-planar.*



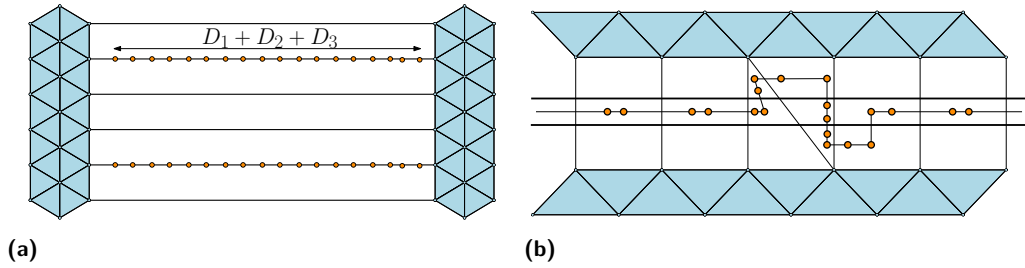
■ **Figure 2** (a) A 1-planar and 2^+ -real face drawing of K_6 , and its schematic representation. (b) Schematic representations of a belt of length three, and of a 2-belt.



■ **Figure 3** (a) A ring barrier. (b) Three channels of a column, two are dense and one is sparse, along with their schematic representation.

Proof. By Lemma 1 of [6] it follows that, for a 2^+ -real face drawing, it holds $\chi \leq n - 2$, where χ is the number of crossings in the drawing and n is the number of real vertices. For K_6 , this implies $\chi \leq 4$. Suppose by contradiction that K_6 admits a 2^+ -real face drawing Γ with an edge crossed at least twice. Thus, Γ contains a face of degree at least four (namely a face with at least two real vertices and two crossing-vertices). Since each other face of Γ has at least degree three and since the sum of the degrees of the faces of Γ equals twice the number μ of its edges, we have $2\mu \geq 3(\phi - 1) + 4 = 3\phi + 1$, where ϕ denotes the number of faces of Γ . Since $\mu = m + 2\chi$ and $\phi = m + \chi + 2 - n$, the last inequality implies that $2(m + 2\chi) \geq 3(m + \chi + 2 - n) + 1$ or equivalently $\chi \geq m - 3n + 7$ holds. For K_6 , we know $n = 6$ and $m = 15$, thus $\chi \geq 10$, which contradicts the bound $\chi \leq 4$. ◀

A *belt of length k* is a chain of k copies of K_6 that are glued such that two consecutive copies share one edge; see also Figure 2b. A *b-belt of length k* is obtained by merging together $b > 1$ belts of length k , as shown in Figure 2b. To construct graph G , we first create a quite rigid structure, called *ring barrier R* , consisting of four components: the *top beam*, the *right wall*, the *bottom beam* and the *left wall*. Each of the top and bottom beams consists of a *T-belt of length L* , while each of the left and right walls consists of a *T-belt of length P* , with the following choice of parameters: (i) $M = \lceil B/2 \rceil + 1$; (ii) $X = 2M$; (iii) $L = 3mX$; (iv) $P = 3m + 2$; (v) $T = L^2$. Note that $M > a_i$ for each $i = 1, \dots, 3m$. Also, L and P are chosen to accommodate $3m$ *column gadgets* and m *transversal gadgets*, to be defined later. T makes the ring barrier thick enough; it is formed by gluing in a circular arrangement the endpoints of the top beam, right wall, bottom beam and left wall; see Figure 3a.

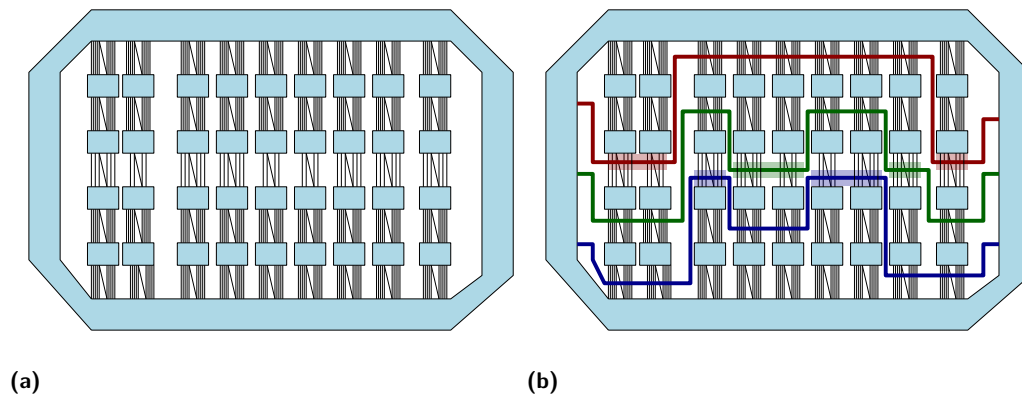


■ **Figure 4** (a) Two transversal gadgets (only a part of the ring barrier is shown and columns are omitted). (b) A transversal gadget routed through a channel.

The top and bottom beams are connected by a set of $3m$ columns; see Figure 3b. Each column contains a stack of $2m - 2$ copies of the *barrier-gadget*, which is formed by gluing X copies of K_6 . Within each column, consecutive copies of the barrier-gadget are connected by an even set of pairwise disjoint edges whose size will be defined later, interleaved by an edge in the middle of the sequence forming two triangles, as shown in Figure 3b. The topmost and the bottommost copy of the barrier-gadget of each column is connected to the top and bottom beam, respectively, also in the same fashion. The edges that realize these connections are called *vertical edges* and form the so-called *channels*. In particular, there are $m - 1$ *topmost channels*, one *central channel* and $m - 1$ *bottommost channels*. The central channel of the i -th column is *sparse* containing only $M + a_i$ vertical edges (note that, by construction, $X = 2M > M + a_i$); the remaining ones are *dense* containing X edges each.

We conclude the construction of graph G by introducing m pairwise disjoint gadgets, $\pi_1, \pi_2, \dots, \pi_m$, called the *transversal gadgets* of G . Each transversal gadget consists of two edges, called *guide edges*, and one path of $D_1 + D_2 + D_3$ vertices, where the three parameters are specified below. The intuition is that, in order to realize a 2^+ -real face drawing, each path must cross all columns through one of their channels, and each path will be able to do so if and only if it can be routed through exactly three sparse channels whose number of vertical edges is equal to B ; therefore, the length of a single path is crucial: (i) $D_1 = (3m - 3)(2X + 8)$, for the path to be able to cross $3m - 3$ dense channels; (ii) $D_2 = 2B + 6M + 24$, for the path to be able to cross 3 sparse channels; (iii) $D_3 = 2(3m - 1)$, for the path to be able to cross the faces between consecutive columns. These gadgets are attached at independent consecutive vertices along the left and right walls, as shown in Figure 4a (each of the m gadgets takes three vertices on the left wall and on the right wall, which are both made by T -belts of length $3m + 2$). Note that G does not contain any cut-vertex.

Proof sketch for Theorem 3. To prove that A admits a partition if and only if G admits a 2^+ -real face drawing, we need a few definitions. A *canonical drawing* of G is one such that, if two edges cross, then one of the following cases applies: (i) both edges are part of a K_6 , or (ii) one edge is part of a transversal gadget and the other is a vertical edge of a channel, or (iii) one is a guide edge of transversal gadget and the other is an edge of the path of the same transversal gadget. Consider a column C and a channel c of C . We say that a transversal gadget π is *routed through* c , if c is the only channel of C whose edges are crossed by some edges of π , see Figure 4b. If A_1, \dots, A_m is a solution of A , then a 2^+ -real face drawing of G can be obtained by routing each transversal gadget π_i through $m - 3$ dense channels and 3 sparse channels corresponding to the elements of A_i as shown in Figure 5. Proving the other direction requires a more involved argument. We first prove that a crossing-minimal 2^+ -real face drawing of G , if any, is a canonical drawing. Next we



■ **Figure 5** (a) A canonical drawing (schematic representation) where the transversal gadgets are omitted. (b) A complete canonical drawing in which the three transversal gadgets are schematized as thick polylines of different colors and their intersections with sparse cells are highlighted.

can show that each transversal gadget is routed through $m - 3$ dense channels and 3 sparse channels such that no two gadgets traverse the same channel. This allows the construction of a solution A_1, \dots, A_m of A as follows: if π_j crosses the κ -th, λ -th and μ -th columns of G through sparse channels, where $1 \leq \kappa, \lambda, \mu \leq 3m$, then $A_j = \{a_\kappa, a_\lambda, a_\mu\}$. ◀

► **Remark 5.** The proof of Theorem 3 can be adapted to show that recognizing 1^+ -real face graphs is also NP-complete. At high level, we substitute the copies of K_6 with copies of K_7 , slightly modify the channel structure by introducing two edges that cross (to ensure that a single guide edge cannot be routed through a channel without an associated path) and halve the length of the paths. In the interest of space, we defer the details to the journal version.

4 2-Layer k^+ graphs

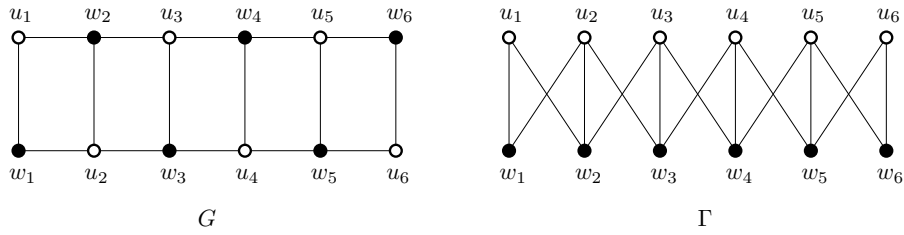
In this section we focus on 2-layer k^+ -real face drawings. We start giving edge-density results for each positive integer k (Section 4.1); they represent a preliminary step for the recognition problem. Then, we describe algorithms to recognize 2-layer k^+ -real face graphs for $k \geq 2$, and algorithms to recognize optimal 2-layer k^+ -real face graphs for $k \geq 1$ (Section 4.2).

4.1 Edge Density

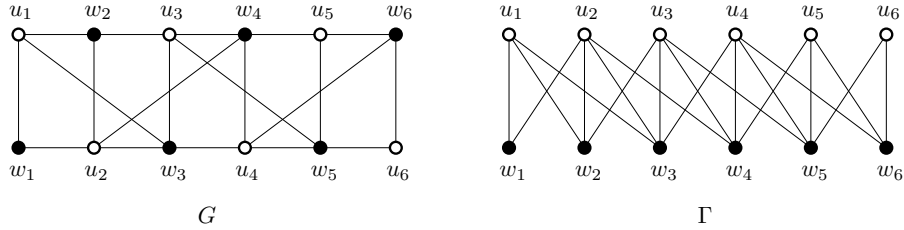
We give tight upper bounds on the edge density of n -vertex 2-layer k^+ -real face graphs, for any $k \in [1, n]$. For $k \in [3, n]$ the next theorem establishes that the n -vertex connected 2-layer k^+ -real face graphs are caterpillars, thus they have $n - 1$ edges.

► **Theorem 6.** *An n -vertex connected graph is 2-layer k^+ -real face for any $k \in [3, n]$ if and only if it is a caterpillar.*

Proof. A connected graph has a 2-layer planar embedding if and only if it is a caterpillar [14]. Suppose first that γ is a 2-layer k^+ -real face embedding of a connected graph G for a given $k \in [3, n]$. By Property 1, γ cannot have internal faces, i.e., all real-vertices of γ belong to the external face. In particular, there cannot be any two edges of G that cross in γ , otherwise, since G is connected, there would be an internal face in γ . Hence G is 2-layer



■ **Figure 6** Illustration for Theorem 7. An optimal 2-layer 2^+ -real face graph G with $n = 12$ vertices (left) and a 2-layer 2^+ -real face drawing Γ of G (right).



■ **Figure 7** Illustration for Theorem 8. An optimal 2-layer 1^+ -real face graph G with $n = 12$ vertices (left) and a 2-layer 1^+ -real face drawing Γ of G (right).

planar, which implies that it is a caterpillar. Conversely, if G is a caterpillar it has a 2-layer planar embedding. This embedding has a unique face (the external face), which contains all the vertices of the graph, thus it is a 2-layer k^+ -real face embedding for every $k \in [3, n]$. ◀

► **Theorem 7.** Any n -vertex 2-layer 2^+ -real face graph has at most $1.5n - 2$ edges, and this bound is tight.

Proof. Let $G = (V_1 \cup V_2, E)$ be an n -vertex bipartite graph that admits a 2-layer 2^+ -real face drawing Γ , and let m be the number of edges of G . Augment Γ (and G) with $n - 2$ (non-crossing) straight-line edges that connect all the vertices in each vertex set V_i ($i = 1, 2$), in their linear ordering along the corresponding layer. The resulting drawing Γ' is an outer 2^+ -real face drawing of a graph G' with $n' = n$ vertices, i.e., a 2^+ -real face drawing with all vertices on the external face. In [6] it is proved that such a graph G' has at most $2.5n - 4$ edges. Since G' has $m' = m + n - 2$ edges, we have $m \leq 1.5n - 2$. About the tightness of the bound, the ladders on n vertices are optimal 2-layer 2^+ -real face graphs (Figure 6). ◀

► **Theorem 8.** Any n -vertex 2-layer 1^+ -real face graph has at most $2n - 4$ edges, and this bound is tight.

Sketch. The proof is analogous to that of Theorem 7, but exploits the fact that an outer 1^+ -real face drawing has at most $3n - 6$ edges [6]. A 2-layer 1^+ -real face graph that matches this bound is a ladder with some extra edges (see Figure 7 when $n = 12$). Namely, let $V_1 = \{u_1, \dots, u_{\frac{n}{2}}\}$ and $V_2 = \{w_1, \dots, w_{\frac{n}{2}}\}$. For $i = 1, \dots, \frac{n}{2}$ there is an edge (u_i, w_i) . For $i = 1, \dots, \frac{n}{2} - 1$ there are the two edges (u_i, w_{i+1}) and (u_{i+1}, w_i) . For each $i \in \{1, \dots, \frac{n}{2} - 2\}$ there is an edge (u_i, w_{i+2}) . This graph has in total $m = \frac{n}{2} + n - 2 + \frac{n}{2} - 2 = 2n - 4$ edges. ◀

4.2 Recognition

For $k \in [3, n]$, Theorem 6 implies that testing whether an n -vertex graph G is 2-layer k^+ -real face is equivalent to testing whether G is a caterpillar. Hence, the following holds.

► **Theorem 9.** *Let G be any connected graph with n vertices and let $k \geq 3$. There exists an $O(n)$ -time algorithm that tests whether G is 2-layer k^+ -real face and that computes a 2-layer k^+ -real face drawing if one exists.*

Theorem 9 extends trivially to the recognition of optimal 2-layer k^+ -real face graphs for $k \geq 3$, since by Theorem 6 any connected 2-layer k^+ -real face graph with $k \geq 3$ is optimal. Recognizing 2-layer k^+ -real face graphs for $k \in \{1, 2\}$ is more involved. The following definition will be used in the next subsections. Let G be a bipartite graph, P be a simple path in G , and γ be a 2-layer embedding of G . We say that P is a *zig-zag path* in γ if the restriction of γ to P is not self-crossing (see, e.g., the path from w to w' in Figure 8). If P is a zig-zag path, there exists a drawing of γ in which P is x -monotone; thus, with a slight abuse of terminology, we also refer to the left-to-right order of the vertices of P in γ .

4.2.1 2-layer 2^+ -real face graphs

We start with the following inclusion relationship.

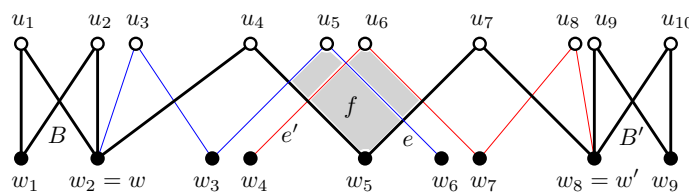
► **Lemma 10.** *Any 2-layer 2^+ -real face embedding is a 2-layer RAC embedding.*

Proof. Suppose that γ is a 2-layer 2^+ -real face embedding of a graph. There cannot be a fan crossing in γ , as otherwise γ would have a triangular 1-real face, contradicting the hypothesis that γ is a 2^+ -real face embedding. Hence, by Theorem 2, γ is a 2-layer RAC graph. ◀

Lemma 10 implies that the family of 2-layer 2^+ -real face graphs is included in the family of 2-layer RAC graphs. The reverse does not hold, as the next lemma proves.

► **Lemma 11.** *There exist infinitely many graphs that are 2-layer RAC but not 2-layer 2^+ -real face.*

Sketch. For any even positive integer k , consider a bipartite graph $G = (V_1 \cup V_2, E)$ consisting of: (i) two 4-cycles B and B' ; (ii) a path \bar{P} of length k between a vertex $w \in V_2$ of B and a vertex $w' \in V_2$ of B' ; (iii) two paths P and P' of length $\frac{k}{2} + 2$, where P is attached to w and P' is attached to w' . See Figure 8 for an illustration where: $k = 4$, $w = w_2$, $w' = w_8$; C , C' , and \bar{P} are in bold; P is in blue and P' is in red. In any 2-layer RAC embedding of G , P and P' are zig-zag paths that cross each other, thus forming a 1-real face f (see Figure 8). ◀



■ **Figure 8** Illustration for Lemma 11. A 2-layer RAC graph that is not 2-layer 2^+ -real face.

Note however that the graphs of Lemma 11 are not biconnected. If we restrict to biconnected graphs, we are able to prove that 2-layer RAC graphs and 2-layer 2^+ -real face graphs are in fact the same family. More precisely, the following result is known (see [9]).

► **Theorem 12** (Di Giacomo et al. [9]). *An n -vertex biconnected graph G is 2-layer RAC if and only if it is a spanning subgraph of a ladder. Also, there exists an $O(n)$ -time algorithm that tests whether G is 2-layer RAC, and computes a 2-layer RAC drawing of G if one exists.*

32:10 On the Complexity of Recognizing k^+ -Real Face Graphs

From the considerations above, we derive the following characterization.

► **Lemma 13.** *A biconnected graph is 2-layer 2^+ -real face if and only if it is a spanning subgraph of a ladder.*

Proof. As shown, every ladder is a 2-layer 2^+ -real face graph, and hence every spanning subgraph of a ladder is a 2-layer 2^+ -real face graph. Vice versa, let G be a biconnected 2-layer 2^+ -real face graph and let γ be a 2-layer 2^+ -real face embedding of G . By Lemma 10, γ is 2-layer RAC, and hence, G is a spanning subgraph of a ladder by Theorem 12. ◀

The next theorem follows by combining Lemma 13 and Theorem 12.

► **Theorem 14.** *An n -vertex biconnected G is 2-layer 2^+ -real face if and only if it is 2-layer RAC. Also, there exists an $O(n)$ -time algorithm that tests whether G is 2-layer 2^+ -real face and that computes a 2-layer 2^+ -real face drawing if one exists.*

Moreover, by Lemma 13 and since any n -vertex ladder is optimal 2-layer 2^+ -real face (Theorem 7), whereas any non-biconnected n -vertex 2-layer 2^+ -real face graph has less than $1.5n - 2$ edges, we get the following result for optimal 2-layer 2^+ -real face graphs.

► **Corollary 15.** *An n -vertex graph G is optimal 2-layer 2^+ -real face if and only if it is a ladder. Optimal 2-layer 2^+ -real face graphs can be recognized in $O(n)$ time.*

We now focus on the recognition of connected 2-layer 2^+ -real face graphs that are not biconnected. By Lemma 10, every 2-layer 2^+ -real face embedding of a bipartite graph G (if any) must be searched in the space of 2-layer RAC embeddings of G . Hence, we first recall in some details what is the structure of any 2-layer RAC embedding γ of a connected graph G ; then we establish an extra property that γ must fulfill to be a 2-layer 2^+ -real face embedding.

Structure of 2-layer RAC embeddings. Let G be a 1-connected graph and let γ be a 2-layer RAC embedding of G . As showed in [9], the embedding γ consists of two parts:

- **Skeleton.** The first part, called *skeleton*, is a (possibly empty) left-to-right sequence of non-trivial blocks; any two consecutive blocks either share a cut-vertex of G or are connected by a zig-zag path, placed between them, which we call an *in-between path*. The first (last) block of the sequence may be preceded (followed) by a *maximal zig-zag path* attached to it, called an *extremal path*. If G is a tree (without non-trivial blocks), the skeleton is just a single zig-zag path; by convention, the extremal paths coincide with such a zig-zag path. We denote by $\text{skel}_\gamma(G)$ the subgraph of G induced by the edges of G in the skeleton of γ .
- **Dangling Paths.** The second part is a set of zig-zag paths, each path P sharing exactly one vertex w with the skeleton. We call P a *dangling path* and w the *attaching vertex* of P . Two dangling paths are either edge-disjoint or they have exactly one edge in common, which is the one containing the attaching vertex of the paths. We denote by $\text{dang}_\gamma(G)$ the subgraph of G induced by the edges that belong to the dangling paths.

Clearly, the edges of $\text{skel}_\gamma(G)$ and of $\text{dang}_\gamma(G)$ partition the edge set of G . When G is not a tree, we also denote by $\text{skel}_\gamma^-(G)$ the subgraph of $\text{skel}_\gamma(G)$ consisting only of the non-trivial blocks and their in-between paths. Figure 9 shows an example of 2-layer RAC embedding and its parts. Note that, by definition, if G is not a tree, each zig-zag path of $\text{skel}_\gamma(G)$ is attached to two non-trivial blocks (if this path is an in-between path) or to one non-trivial block (if this path is an extremal path). Further, by Lemma 13, each non-trivial

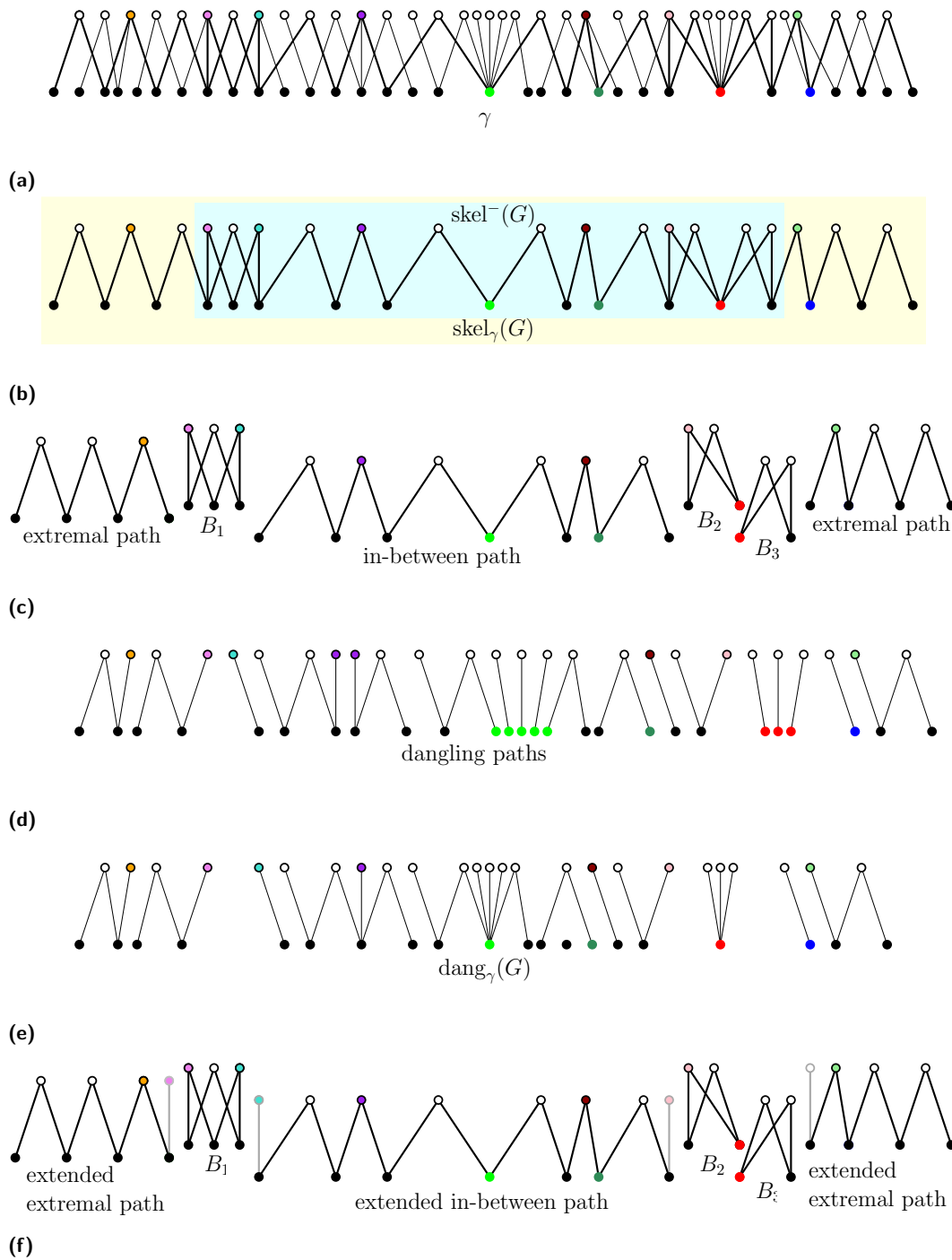


Figure 9 (a) A 2-layer RAC embedding γ of a graph G ; the colored vertices are attaching vertices; (b) $\text{skel}_\gamma(G)$ and $\text{skel}_\gamma^-(G)$; (c) the components of $\text{skel}_\gamma(G)$; (d) the dangling paths of γ ; (e) $\text{dang}_\gamma(G)$. (f) The extended zig-zag paths of the 2-layer RAC embedding γ .

block B of G is a spanning subgraph of a ladder, and the leftmost and rightmost edges of B , which we call the *extremal edges of B* in γ , coincide with the extremal edges of such a ladder. A skeleton zig-zag path extended with the extremal edges of the non-trivial blocks to which it is attached will be called an *extended zig-zag path* of $\text{skel}_\gamma(G)$. For example, Figure 9f shows the extended zig-zag paths of the 2-layer RAC embedding of Figure 9a.

The following properties hold for any 2-layer RAC embedding γ , otherwise it is easy to see that γ would contain a fan-crossing (see [9] for details).

- **Property 16.** *A dangling path cannot cross any non-trivial block in γ .*
- **Property 17.** *Let P and P' be two edge-disjoint dangling paths of γ that cross each other, and let w and w' be their attaching vertices. Then: (a) w and w' are distinct vertices of the same extended zig-zag path of the skeleton; (b) if (w.l.o.g.) w is to the left of w' , then all the vertices of P (P') distinct from w (w') are to the right of w (to the left of w').*
- **Property 18.** *Each edge of a dangling path in γ crosses at most one edge of the skeleton. Also, if P and P' are two edge-disjoint dangling paths that cross each other in γ , then each edge of P and of P' , with the possible exception of the edges incident to their attaching vertices, crosses exactly one edge of the skeleton.*
- **Property 19.** *Let w be any vertex of an extended zig-zag path of the skeleton. Then, there are at most two dangling paths of length larger than one attached to w .*
- **Property 20.** *If G is not a tree and if γ' is a 2-layer RAC embedding of G distinct from γ , then $\text{skel}_\gamma^-(G) = \text{skel}_{\gamma'}^-(G)$. Also the restriction of γ to $\text{skel}_\gamma^-(G)$ coincides with the restriction of γ' to $\text{skel}_{\gamma'}^-(G)$ (up to mirroring).*

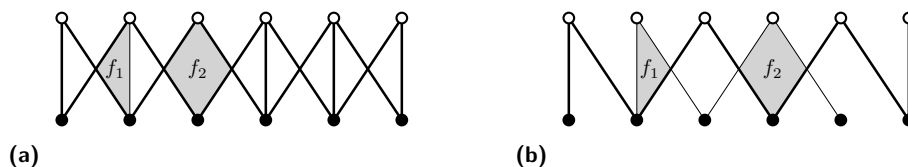
By Property 20, we can denote $\text{skel}_\gamma^-(G)$ by $\text{skel}^-(G)$, as it does not depend on the specific embedding γ . We can get $\text{skel}^-(G)$ by recursively removing from G the degree-1 vertices.

Structure of 2-layer 2^+ -real face graphs. A 2-layer RAC embedding γ of a graph G is *dangling-crossing free* if it does not have two dangling paths that cross each other. The next lemma is a key ingredient to efficiently recognize 2-layer 2^+ -real face graphs.

- **Lemma 21.** *Let γ be a 2-layer RAC embedding of a connected graph G . Then, γ is a 2-layer 2^+ -real face embedding if and only if it is dangling-crossing free.*

Sketch. Let $n \geq 2$ denote the number of vertices of G . Suppose first that γ is dangling-crossing free. The external face of γ contains all the vertices of G , thus it is an n -real face, and hence a 2^+ -real face. Consider now any internal face f of γ . By Property 16, a dangling path cannot cross any edge of a non-trivial block of G in γ . Hence, if f is formed only by skeleton edges of γ then it is a face internal to the embedding of a non-trivial block B . Since by Lemma 13, B is a spanning subgraph of a ladder (and it is drawn RAC), the face f is either a triangle with two real-vertices or a quadrilateral with two real-vertices (see Figure 10a). Finally, assume that f is formed by skeleton edges and by edges of a single dangling path P (f cannot contain edges of two distinct dangling paths, because we are assuming that the dangling paths are pairwise non-crossing). In this case, the skeleton edges that cross P to form f belong to a zig-zag path (either an extremal path or an in-between path of γ). Since by Property 18 each edge of P crosses at most one skeleton edge and since P is also a zig-zag path, f is either a 2-real triangle or a 2-real quadrilateral (see Figure 10b).

Suppose vice versa that γ is a 2-layer 2^+ -real face embedding. If there were in γ two dangling paths that cross each other, they would form a 1-real face, like f in Figure 8. ◀



■ **Figure 10** (a) Internal faces created by skeleton edges (bold); (b) Internal faces created by skeleton edges (bold) and dangling path edges (thin).

Before stating the main result of this section, we give an additional auxiliary lemma.

► **Lemma 22.** *If G is a 2-layer 2^+ -real face graph, then G admits a 2-layer 2^+ -real face embedding such that the attaching vertex of each dangling path has degree at least three.*

► **Theorem 23.** *Let G be an n -vertex bipartite graph. There exists an $O(n)$ -time algorithm that tests whether G is 2-layer 2^+ -real face, and that computes a 2-layer 2^+ -real face drawing of G if one exists.*

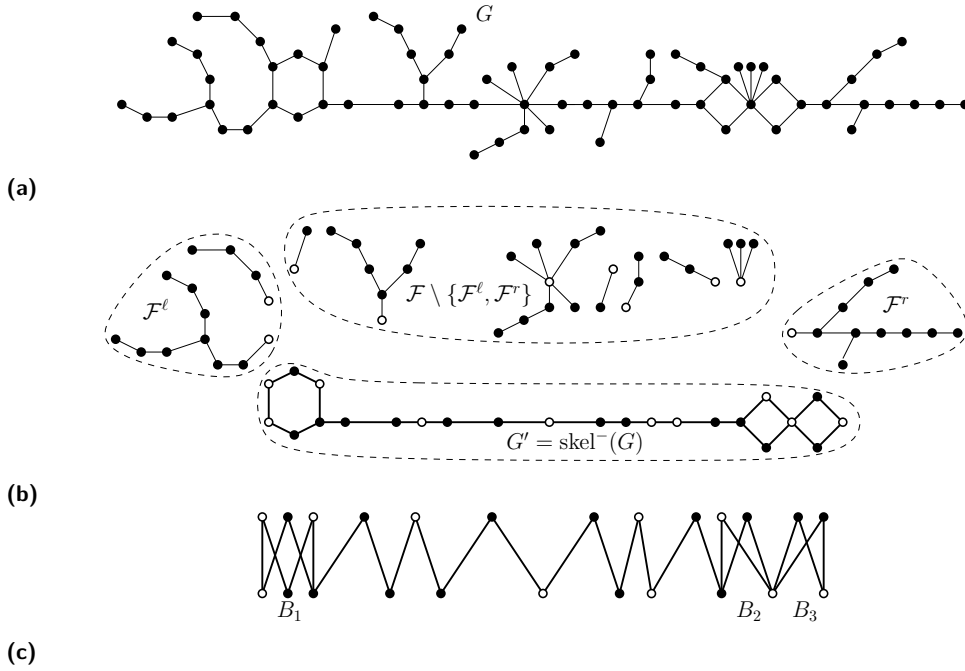
Proof. Based on Lemma 21, we describe an algorithm that attempts to construct a dangling-crossing free 2-layer RAC embedding γ of G , if one exists. We distinguish two main cases: Case 1 – G is not a tree (G contains at least one non-trivial block); Case 2 – G is a tree.

Case 1 – G is not a tree. The algorithm executes the following steps:

Step 1. (See Figure 11.) It tests whether there exists a subgraph of G that is a valid $\text{skel}^-(G)$. To this aim, it recursively removes from G all the vertices of degree one and then applies the algorithm of Theorem 12 to check whether the resulting graph G' admits a 2-layer RAC embedding, and to compute one if any. If such an embedding does not exist, the algorithm stops and rejects the instance. Otherwise, G' coincides with $\text{skel}^-(G)$ and, by Property 20, its 2-layer RAC embedding γ' is unique (up to mirroring); the algorithm goes to the next step.

Step 2. Let B_1, \dots, B_h ($h \geq 1$) be the non-trivial blocks in the left-to-right order defined by γ' . The edges of $G \setminus \text{skel}^-(G)$ form a forest \mathcal{F} of trees, each tree sharing exactly one vertex with $\text{skel}^-(G)$. At most two of these trees share a vertex with the leftmost extremal edge of B_1 in γ' ; let \mathcal{F}^ℓ be the subset of \mathcal{F} that contains these (at most two) trees. Analogously, at most two trees share a vertex with the rightmost extremal edge of B_h in γ' ; let \mathcal{F}^r be the subset of \mathcal{F} that contains these trees. In this step, the algorithm tests if the trees in $\mathcal{F} \setminus \{\mathcal{F}^\ell, \mathcal{F}^r\}$ form a valid set of dangling paths that can be attached to γ' to get a dangling-crossing free 2-layer RAC embedding γ'' . This is done by executing the following substeps.

- **Step 2.1.** (See Figure 12a.) First, the algorithm considers the paths attached to every cut-vertex shared by two non-trivial blocks. Specifically, for each such cut-vertex w , all the paths attached to w can be successfully embedded (between the two non-trivial blocks sharing w) if and only if each of them consists of a single edge; otherwise one of these paths would cross a non-trivial block, thus violating Property 16.
- **Step 2.2.** (See Figure 12b.) For each in-between path \bar{P} , delimited by two blocks B_i and B_{i+1} , the algorithm checks whether there are some paths attached to the vertex u of the rightmost extremal edge of B_i not in \bar{P} , or attached to the vertex v of the leftmost extremal edge of B_{i+1} not in \bar{P} . In particular, there can be at most one such a path, call it P (resp. P'), attached to u (resp. to v), because two paths attached to u (or to v) cannot be embedded without causing a fan crossing with \bar{P} . Further, P and P' must



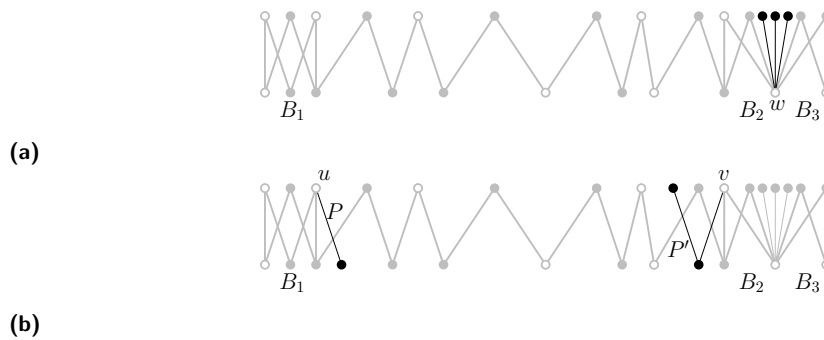
■ **Figure 11** Step 1: (a) an input bipartite graph G ; (b) decomposition of G into $\text{skel}^-(G)$ and a forest \mathcal{F} ; (c) a 2-layer RAC embedding of $\text{skel}^-(G)$. The attaching vertices are in white.

necessarily be embedded to the right of u and to the left of v , respectively. Hence, if both P and P' exist in the graph, the algorithm checks that they do not cross each other in their unique embedding. If only one among P and P' exists, say P , the algorithm checks that it does not cross B_{i+1} in its unique embedding. If any of the above checks fails, then the algorithm rejects the instance, otherwise it continues with the next substep.

- **Step 2.3.** (See Figures 13a and 13b.) For each in-between path \bar{P} of γ' , the algorithm considers the paths that in G are attached to \bar{P} , and checks if they can be embedded so to be pairwise non-crossing. Formally, let B_i and B_{i+1} be the two non-trivial blocks that delimit \bar{P} ($1 \leq i \leq h-1$). Also, let $W = \langle w_1, w_2, \dots, w_p \rangle$ be the left-to-right sequence of attaching vertices of \bar{P} , i.e., the vertices to which some trees of $\mathcal{F} \setminus \{\mathcal{F}^\ell, \mathcal{F}^r\}$ are attached. By Property 16, all the paths attached to \bar{P} must be embedded between B_i and B_{i+1} . Hence, we can process the vertices of W in their left-to-right order and test, for each vertex $w_j \in W$, if all paths attached to w_j can be suitably embedded as zig-zag paths so that: (i) they do not cross with any previously embedded dangling paths attached to w_g , with $g < j$, or with B_i , or with the dangling path P attached to B_i embedded in Step 2.2; (ii) they leave the maximum degree of freedom for embedding the dangling paths attached to w_{j+1} , subject to condition (i). Conditions (i) and (ii) together guarantee the correctness of the testing algorithm. However, to satisfy these conditions, we sometimes need to process contemporary all the vertices in specific subsequences of W , as done in [9] for testing 2-layer RAC embeddability (a necessary condition in our case).

More in details as proved in [9], there are only three possible types of graph structures, called *feasible structures*, that could be attached to the vertices of W without necessarily creating fan crossings (see Figures 14a and 14b):

- **star-tree:** it is a subdivision of a star rooted at a vertex $w \in W$. By Property 19, at most two paths attached to w have length larger than one. Call them the *long paths*.
- **y -tree:** it is a tree attached to a vertex $w \in W$ and consisting of two paths sharing



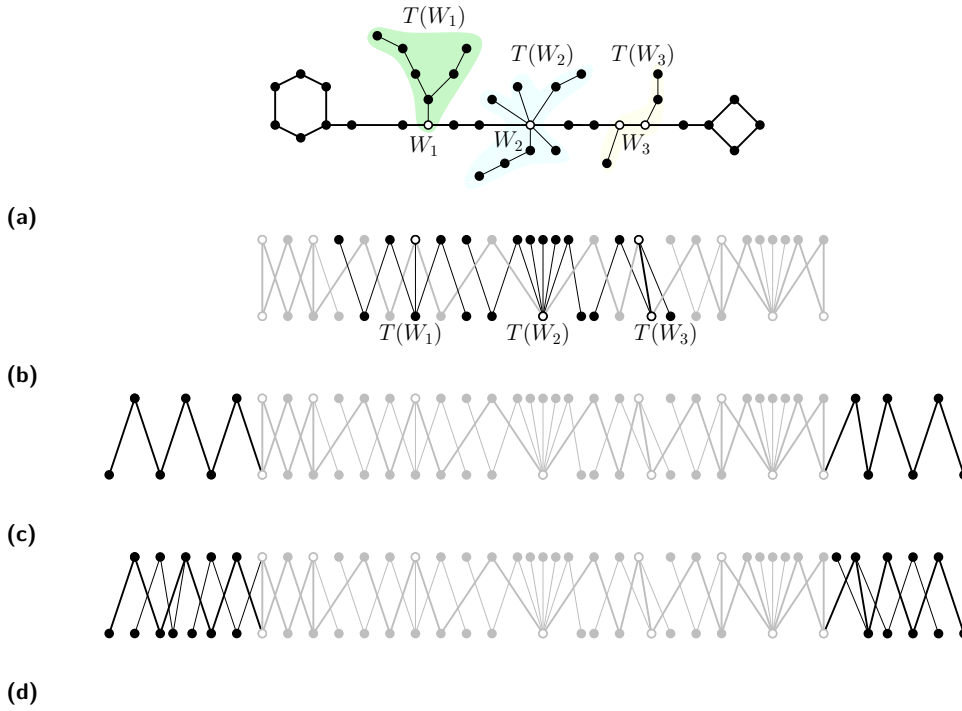
■ **Figure 12** (a) Step 2.1: Addition of paths attached to a cut-vertex w shared by two non-trivial blocks; (b) Step 2.2: Addition of the paths P and P' .

only the edge that contains w .

- k -fence: it is a set of vertex-disjoint paths attached to a maximal subsequence W' of vertices of W that are consecutive in \bar{P} . Each $w \in W'$ has exactly one path attached to it. Also, if $k \geq 5$, all the paths attached to the vertices of W' , except possibly those attached to the first two and to the last two vertices of W' , have length at most two.

Hence, the testing algorithm first checks whether all the vertices in W can be partitioned into maximal subsequences W_1, W_2, \dots, W_q ($q \geq 1$), such that each subsequence W_i contains the attaching vertices of a feasible structure, which we denote by $T(W_j)$ ($|W_j| = 1$ if $T(W_j)$ is a star-tree or a y -tree). If this is not possible, the algorithm rejects the instance. Otherwise, it searches for an embedding of each $T(W_j)$ such that all these embeddings, along with the embeddings of B_i, B_{i+1} , and \bar{P} , result in a 2-layer RAC embedding that is dangling-crossing free. Namely, in [9] it is shown that, for the 2-layer RAC embeddability, the number of candidate embeddings for each structure is bounded by a small constant. More precisely (see Figures 14c–14f): (i) For a star-tree it must be decided which of the (at most) two long paths can go to the left and which to the right (the paths of length one can always be embedded without crossing the skeleton); (ii) for a y -tree it must be decided which of the two paths goes to the left and which to the right; (iii) for a k -fence, the set of candidate embeddings is at most 2 if $k = 2$, or at most 3 if $k = 3$, or at most 4 if $k \geq 4$. In particular, in a valid embedding of each k -fence, one of its paths will be embedded to the left of the k -fence, one to the right, and the others in-between the leftmost and the rightmost vertices of the k -fence (see [9] for more details).

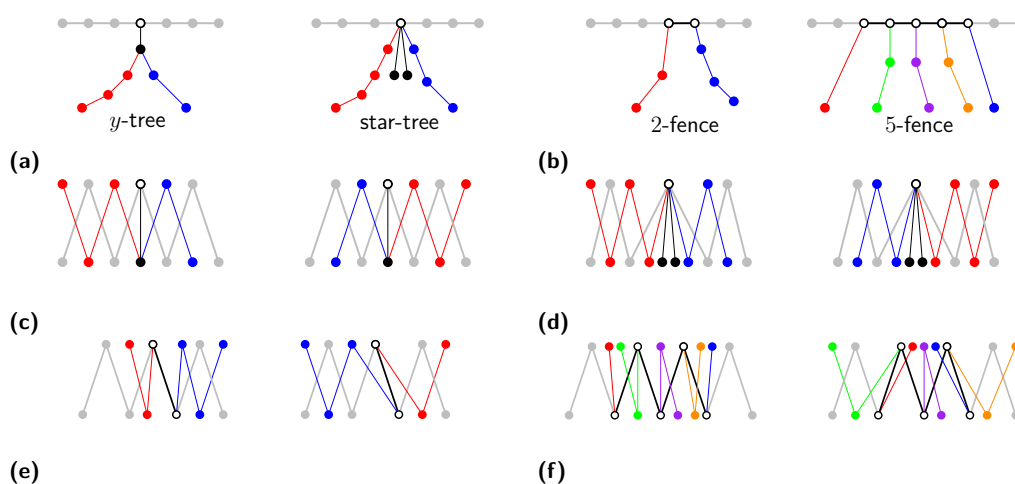
However, differently from [9], we can only accept the embeddings of $T(W_1), \dots, T(W_q)$ where no two dangling paths cross each other. To this aim, our testing algorithm processes all W_j from left to right in a greedy fashion. Each time a subsequence W_j is considered, the algorithm checks if the feasible structure $T(W_j)$ has some candidate RAC embeddings that do not cause crossings with dangling paths already embedded to the left of W_j . If not, the instance is rejected; otherwise, among the candidate embeddings, the algorithm selects one for which the dangling path of $T(W_j)$ that goes to the right is as short as possible, which maximizes the degrees of freedom for dangling paths that will be processed in the future. This guarantees that the test is positive if and only if a 2-layer 2^+ -real face embedding exists. At the end, the algorithm checks whether the embedding of $T(W_q)$ causes a crossing of a dangling path of $T(W_q)$ with B_{i+1} (or with a dangling path attached to B_{i+1} in Step 2.2). If so, it rejects the instance, otherwise goes to the next step.



■ **Figure 13** Step 2.3: (a) feasible structures attached to the in-between path of the graph of Figure 11a; (b) addition of $T(W_1)$, $T(W_2)$, and $T(W_3)$. Step 3: (c) addition of the extremal paths; (d) completion of the 2-layer RAC embedding.

Step 3. (See Figures 13c and 13d.) The algorithm tests if \mathcal{F}^ℓ and \mathcal{F}^r can be added to the embedding γ'' of Step 2, to form the final dangling-crossing free 2-layer RAC embedding γ . If so, \mathcal{F}^ℓ and \mathcal{F}^r coincide with the graph formed by the two extremal paths of $\text{skel}_\gamma(G)$ and by the dangling paths attached to (the extended version of) these extremal paths. To perform this test for \mathcal{F}^r , the algorithm has to determine the path corresponding to the extremal path \overline{P} in \mathcal{F}^r . Then, it will apply the same procedures as in Steps 2.2 and 2.3 to test whether the remaining part of \mathcal{F}^r consists of dangling paths attached to (the extended version of) \overline{P} , and whether they can be embedded without crossing each other and without creating fan crossings. The test for \mathcal{F}^ℓ is the same, but the procedures of Steps 2.2 and 2.3 are applied going from right to left. We now explain how the algorithm can test for the existence of a valid \overline{P} in \mathcal{F}^r (for \mathcal{F}^ℓ the algorithm is symmetric).

Let $e = (u, v)$ be the rightmost extremal edge of the last non-trivial block B_h , and let T_u and T_v be the two trees of \mathcal{F}^r attached to u and v , respectively. If both T_u and T_v are non-empty and contain vertices of degree larger than two (other than u and v), then \mathcal{F}^r cannot be the union of an extended extremal path with dangling paths attached to it. Hence, the algorithm can reject the instance in this case. On the other hand, if both T_u and T_v are paths, the algorithm can arbitrarily choose one of them as the desired extremal path and the other as a dangling path; these two paths can always be embedded as zig-zag paths going from left to right, without creating fan crossings. Finally, assume without loss of generality, that T_u is the only tree (among T_u and T_v) that contains vertices of degree larger than two (in addition to u). Denote by W the set of vertices of degree at least three in T_u (including u). The desired extremal path (if any) must contain all vertices in W . If this is not the case, the algorithm rejects the instance. Otherwise, let \overline{P}' be the path in T_u starting from u and



■ **Figure 14** (a) Examples of a y -tree and a star-tree; (b) Examples of a 2-fence and of a 5-fence; (b)-(e) Two possible 2-layer RAC embeddings for each of the structures in (a) and (b).

containing all the vertices of W ; also, let w be the last vertex of W encountered along $\overline{P'}$. The desired extremal path \overline{P} must be obtained by extending $\overline{P'}$ with a path attached to w . Let \mathcal{P} denote the set of paths attached to w and extending $\overline{P'}$. Note that, there can be at most three paths in \mathcal{P} with more than one edge, because, by Property 19, there can be at most two dangling paths of length larger than one attached to w . We can get \overline{P} by extending $\overline{P'}$ with an arbitrarily chosen path $P \in \mathcal{P}$ of maximum length. Indeed, suppose that there exists a 2-layer 2^+ -real face embedding γ where \overline{P} is extended with a path $P' \in \mathcal{P}$ shorter than P . We can always obtain from γ a new 2-layer embedding by exchanging P' with P , possibly after a horizontal flip of these two paths. Since P' is shorter than P , such a flip can only reduce the number of crossings in γ and does not create two dangling paths that cross each other; thus the new embedding remains a 2^+ -real face embedding.

Case 2 – G is a tree. In this case, if γ is a 2-layer 2^+ -real face embedding of G , the skeleton $\text{skel}_\gamma(G)$ of γ is simply a zig-zag path, and the graph structures formed by the dangling paths attached to $\text{skel}_\gamma(G)$ must be star-trees, or y -trees, or k -fences. Since, by Lemma 10, γ is also a 2-layer RAC embedding, the algorithm can just apply the procedure of Step 2.3 on every path of G that is a candidate skeleton for a 2-layer RAC embedding of G . In [9], it is proved that the number of such candidate skeletons is bounded by a small constant (precisely, there are at most 49 candidate skeletons).

Time Complexity. About the time complexity of the described algorithm, consider first the case in which G is not a tree. The recursive removal of degree-1 vertices in Step 1 is executed in $O(n)$ time, and testing if the resulting graph admits a 2-layer RAC embedding is done in $O(n)$ time by the algorithm in [9]. Hence, Step 1 takes overall $O(n)$ time.

About Step 2, we have that: (i) Steps 2.1 and 2.2 can be easily executed in $O(n)$ time by visiting the subgraphs in the set $\mathcal{F} \setminus \{\mathcal{F}^r, \mathcal{F}^\ell\}$. (ii) In Step 2.3, we apply the $O(n)$ -time algorithm in [9] to test whether all subgraphs attached to an in-between path are feasible structures and to partition W into maximal subsequences W_1, W_2, \dots, W_q . Then, the subsequent greedy procedure that processes W_1, W_2, \dots, W_q from left to right can be executed in linear time, because for each W_j the algorithm evaluates a constant number of candidate embeddings. Hence, Steps 2 takes $O(n)$ time.

About Step 3, the procedure described to find a candidate extremal path \overline{P} is executed in linear time by simply visiting \mathcal{F}^r and \mathcal{F}^ℓ . The remaining part of this step uses the same strategy as Steps 2.2 and 2.3, thus it takes $O(n)$ time.

Finally, if G is a tree, the algorithm computes all candidate skeletons, which are bounded by a constant number. This is done through the $O(n)$ -time procedure in [9]. For each candidate skeleton, the algorithm uses the strategy of Steps 2.2, which takes $O(n)$ time. ◀

4.2.2 Optimal 2-layer 1^+ -real face graphs

To derive a linear-time recognition algorithm for optimal 2-layer 1^+ -real face graphs, we prove several structural properties of these graphs. We denote by G an optimal 2-layer 1^+ -real face graph and by Γ a 2-layer 1^+ -real face drawing of G . Since the removal of vertices or edges from Γ yields a 2-layer 1^+ -real face drawing, any subgraph of G is also 2-layer 1^+ -real face.

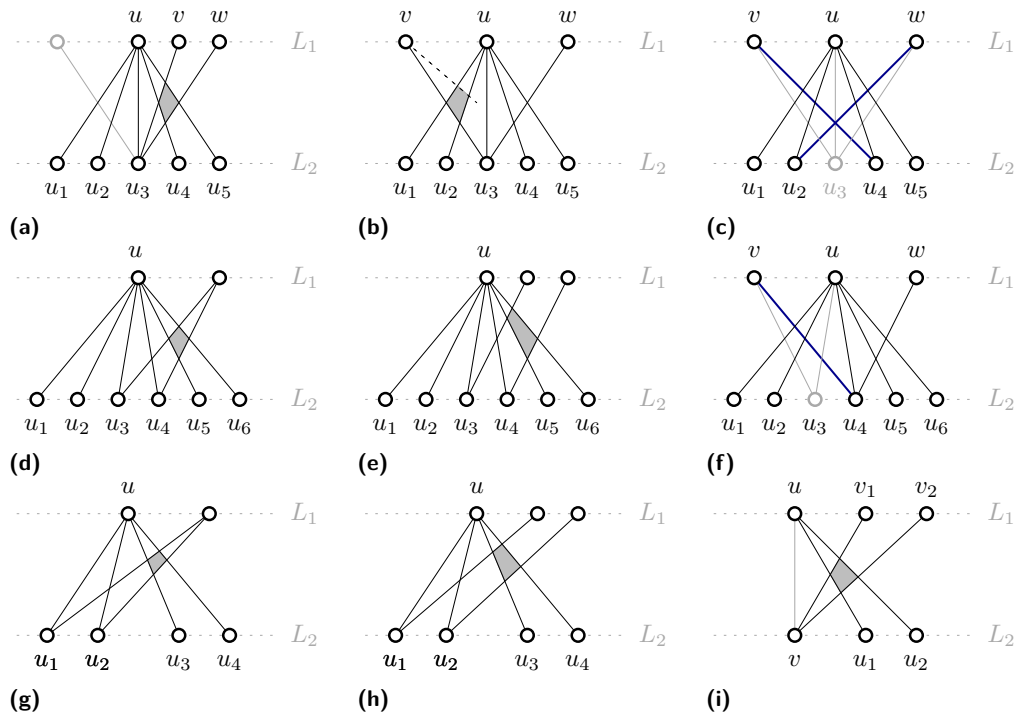
► **Property 24.** *The minimum vertex-degree of G is 2.*

Proof. Assume for a contradiction that G has a vertex u with $\deg(u) = 1$. Let H be the graph obtained from G by removing vertex u , that is, $H = G \setminus \{u\}$. Since G has n vertices and $2n - 4$ edges, graph H has $n - 1$ vertices and $2n - 3$ edges. By Theorem 8, this is a contradiction, since graph H is a 2-layer 1^+ -real face graph (as a subgraph of G). ◀

► **Property 25.** *The maximum vertex-degree of G is 4.*

Proof. Assume to the contrary that G has a vertex u with $\deg(u) \geq 5$. To derive a contradiction, we first consider the case in which $\deg(u) = 5$. Assume without loss of generality that u belongs to layer L_1 of Γ and let u_1, \dots, u_5 be the neighbors of u in L_2 in this left-to-right order. We observe that vertex u_3 cannot have two neighbors, say v and w , that are both either to the left or to the right of u in L_1 , as otherwise the edges (u_3, v) and (u_3, w) together with the edges (u, u_1) and (u, u_2) in the former case or with the edges (u, u_4) and (u, u_5) in the latter case would form a face in Γ that does not contain a real-vertex on its boundary (see the gray colored face in Figure 15a); a contradiction. This implies that $\deg(u_3) \leq 3$, namely, u_3 can be adjacent to u , to a vertex v to the left of u , and to a vertex w to the right of u in L_1 ; see Figure 15b. In particular, if (u_3, v) belongs to G , then v cannot be connected to a vertex to the right of u_3 in L_2 , as otherwise this connection together with the edges (u_3, v) , (u, u_1) , and (u, u_2) would form a face in Γ without real-vertices on its boundary (see the gray colored face in Figure 15b); a contradiction. Symmetrically, if (u_3, w) belongs to G , then w cannot be connected to a vertex to the left of u_3 in L_2 . Let H be the graph obtained by (i) removing vertex u_3 from G , (ii) adding the edge (u_4, v) , if the edge (u_3, v) belongs to G , and (iii) adding the edge (u_2, w) , if the edge (u_3, w) belongs to G . Since graph G is a 2-layer 1^+ -real face graph, graph H is a 2-layer 1^+ -real face graph, as well. Since G has n vertices and $2n - 4$ edges, graph H has $n - 1$ vertices and $2n - 3$ edges. By Theorem 8, this is a contradiction, since graph H is a 2-layer 1^+ -real face graph.

To complete the proof, consider the case in which $\deg(u) \geq 6$. Let $u_1, \dots, u_{\deg(u)}$ be the neighbors of u in L_2 in this left-to-right order. Vertices u_3 and u_4 can have neither a common neighbor nor two distinct neighbors that are both to the left or to the right of u in L_1 , as otherwise this would imply a face in Γ with no real-vertex on its boundary; see Figures 15d and 15e, respectively. Further, as in the case in which $\deg(u) = 5$, we can prove that neither u_3 nor u_4 can have two neighbors that are both to the left or both to the right of u in L_1 ; see Figure 15a. Hence, by Property 24, each of u_3 and u_4 has degree exactly 2 in G . In particular, there exist two vertices v and w on opposite sides of u along L_1 , such that u_3 is adjacent to v and u_4 is adjacent to w . Let H be the graph obtained by removing vertex



■ **Figure 15** Illustrations for the proof of Theorem 31.

u_3 from G and by adding the edge (u_4, v) ; see Figure 15f. Since G is a 2-layer 1^+ -real face graph, graph H is a 2-layer 1^+ -real face graph, as well. Since G has n vertices and $2n - 4$ edges, graph H has $n - 1$ vertices and $2n - 3$ edges, which again contradicts Theorem 8. ◀

► **Property 26.** *Neither the leftmost nor the rightmost vertex of each layer in Γ has degree 4.*

Proof. Assume to the contrary that the leftmost vertex, say u , of L_1 has degree 4 and let u_1, \dots, u_4 be its neighbors in L_2 in this left-to-right order. By Property 24, $\deg(u_1) \geq 2$ and $\deg(u_2) \geq 2$; hence, u_1 and u_2 either have a common neighbor to the right of u in L_1 or have two distinct neighbors that are both to the right of u in L_1 . Both cases, however, cause a face without real-vertices in Γ ; see Figures 15g and 15h, respectively; a contradiction. ◀

► **Property 27.** *Either the leftmost (rightmost) vertex of L_1 or the leftmost (rightmost) vertex of L_2 in Γ has degree 2.*

Proof. Assume to the contrary that neither the leftmost vertex u in L_1 nor the leftmost vertex v in L_2 has degree 2. By Properties 24–26, $\deg(u) = \deg(v) = 3$. Let u_1 and u_2 be two neighbors of u in L_2 , and let v_1 and v_2 be two neighbors of v in L_1 , such that $\{u, v\} \cap \{u_1, u_2, v_1, v_2\} = \emptyset$. The edges (u, u_1) , (u, u_2) , (v, v_1) , and (v, v_2) form a face in Γ without real-vertices (see Figure 15h); a contradiction. ◀

► **Property 28.** *The leftmost (rightmost) vertex of L_1 and the leftmost (rightmost) vertex of L_2 in Γ are adjacent.*

Proof. If the leftmost (resp. rightmost) vertices of L_1 and L_2 are not adjacent, one can connect them without introducing any crossing, which contradicts that G is optimal. ◀

32:20 On the Complexity of Recognizing k^+ -Real Face Graphs

► **Property 29.** *The leftmost (rightmost) vertex of L_1 and the leftmost (rightmost) vertex of L_2 in Γ cannot be both of degree 2.*

Proof. Assume to the contrary that the leftmost vertex u in L_1 and the leftmost vertex v in L_2 are both of degree 2. By Property 28, u and v are adjacent. Let H be the subgraph of G obtained by removing u and v from G . Since G has n vertices and $2n - 4$ edges, it follows that H has $n - 2$ vertices and $2n - 7$ edges, which contradicts Theorem 8. ◀

► **Property 30.** *If G has at least seven vertices, the vertex to the right (left) of the leftmost (rightmost) degree-2 vertex in Γ has degree 3.*

Proof. By Properties 26–28, the leftmost two vertices of G are of degree 2 and 3, and the same holds for the rightmost two vertices of G . Since by Property 25 the maximum degree of G is 4 and since G has n vertices and $2n - 4$ edges, by the hand-shaking lemma G has two more vertices of degree 3 (besides the two aforementioned extreme ones), while each of the remaining $n - 6$ vertices has degree 4. Assume by contradiction that the vertex to the right of the leftmost degree-2 vertex has degree 4. Let H be the graph obtained by removing the leftmost degree-2 vertex from G . The obtained subgraph H has $n - 1$ vertices and $2n - 6 = 2(n - 1) - 4$ edges, thus it is still optimal. Hence, it satisfies Properties 26 and 28. In particular, none of its two leftmost vertices can be of degree 4, a contradiction. ◀

► **Theorem 31.** *Let G be an n -vertex bipartite graph. There exists an $O(n)$ -time algorithm that tests whether G is an optimal 2-layer 1^+ -real face graph, and that computes a 2-layer 1^+ -real face drawing of G if one exists.*

Proof. Let $G = (V_1 \cup V_2, E)$ be an n -vertex optimal 2-layer 1^+ -real face graph and let Γ be a 2-layer 1^+ -real face drawing of G . The vertex-degree of G ranges between 2 and 4 (Properties 24 and 25). However, neither the leftmost nor the rightmost vertex of each of the layers L_1 and L_2 of Γ has degree 4 (Property 26). In particular, assuming that the graph has at least seven vertices, the leftmost (rightmost) two vertices of L_1 and L_2 are adjacent, such that one of them has degree 2 and the other has degree 3 (Properties 27, 28, and 30).

From these properties, we can derive our linear-time recognition algorithm. If $n \leq 6$, then we can check whether G is an optimal 2-layer 1^+ -real face graph by generating all its 2-layer embeddings and checking whether at least one of them is a 2-layer 1^+ -real face. If $n \geq 7$, then we identify one of its vertices of degree 2, say v . If there is no such vertex, then the instance is rejected (by Property 27). Otherwise, we additionally check whether v is neighboring a degree-3 vertex, say w . If no such vertex exists, then the instance is rejected (by Properties 24, 26, and 29). Otherwise, we remove v from G and recursively check whether the obtained instance is a 2-layer 1^+ -real face graph starting now from w (which has degree 2). The implementation is straightforward, and the algorithm works in $O(n)$ time. The correctness follows from a direct application of Properties 24–30. ◀

5 Open Problems


A question that directly stems from our research is whether 2-layer 1^+ -real face graphs can be recognized efficiently. In the unconstrained scenario, are there subfamilies of k^+ -real face graphs that can be recognized efficiently? Also, are there meaningful parameterizations that make the recognition problem tractable?

References

- 1 Abu Reyan Ahmed, Patrizio Angelini, Michael A. Bekos, Giuseppe Di Battista, Michael Kaufmann, Philipp Kindermann, Stephen G. Kobourov, Martin Nöllenburg, Antonios Symvonis, Anaïs Villedieu, and Markus Wallinger. Splitting vertices in 2-layer graph drawings. *IEEE Computer Graphics and Applications*, 43(3):24–35, 2023. doi:10.1109/MCG.2023.3264244.
- 2 Michael A. Bekos, Sabine Cornelsen, Luca Grilli, Seok-Hee Hong, and Michael Kaufmann. On the recognition of fan-planar and maximal outer-fan-planar graphs. In Christian A. Duncan and Antonios Symvonis, editors, *Graph Drawing - 22nd International Symposium, GD 2014, Würzburg, Germany, September 24-26, 2014, Revised Selected Papers*, volume 8871 of *Lecture Notes in Computer Science*, pages 198–209. Springer, 2014. doi:10.1007/978-3-662-45803-7_17.
- 3 Carla Binucci, Markus Chimani, Walter Didimo, Martin Gronemann, Karsten Klein, Jan Kratochvíl, Fabrizio Montecchiani, and Ioannis G. Tollis. 2-layer fan-planarity: From caterpillar to stegosaurus. In *GD*, volume 9411 of *Lecture Notes in Computer Science*, pages 281–294. Springer, 2015. doi:10.1007/978-3-319-27261-0_24.
- 4 Carla Binucci, Markus Chimani, Walter Didimo, Martin Gronemann, Karsten Klein, Jan Kratochvíl, Fabrizio Montecchiani, and Ioannis G. Tollis. Algorithms and characterizations for 2-layer fan-planarity: From caterpillar to stegosaurus. *J. Graph Algorithms Appl.*, 21(1):81–102, 2017. doi:10.7155/jgaa.00398.
- 5 Carla Binucci, Giuseppe Di Battista, Walter Didimo, Vida Dujmović, Seok-Hee Hong, Michael Kaufmann, Giuseppe Liotta, Pat Morin, and Alessandra Tappini. Graphs drawn with some vertices per face: Density and relationships. *IEEE Access*, 12:68828–68846, 2024. doi:10.1109/ACCESS.2024.3401078.
- 6 Carla Binucci, Giuseppe Di Battista, Walter Didimo, Seok-Hee Hong, Michael Kaufmann, Giuseppe Liotta, Pat Morin, and Alessandra Tappini. Nonplanar graph drawings with k vertices per face. In Daniël Paulusma and Bernard Ries, editors, *Graph-Theoretic Concepts in Computer Science - 49th International Workshop, WG 2023, Fribourg, Switzerland, June 28-30, 2023, Revised Selected Papers*, volume 14093 of *Lecture Notes in Computer Science*, pages 86–100. Springer, 2023. doi:10.1007/978-3-031-43380-1_7.
- 7 Carla Binucci, Emilio Di Giacomo, Walter Didimo, Fabrizio Montecchiani, Maurizio Patrignani, Antonios Symvonis, and Ioannis G. Tollis. Fan-planarity: Properties and complexity. *Theor. Comput. Sci.*, 589:76–86, 2015. doi:10.1016/j.tcs.2015.04.020.
- 8 Franz J. Brandenburg. Recognizing optimal 1-planar graphs in linear time. *Algorithmica*, 80(1):1–28, 2018. doi:10.1007/S00453-016-0226-8.
- 9 Emilio Di Giacomo, Walter Didimo, Peter Eades, and Giuseppe Liotta. 2-layer right angle crossing drawings. *Algorithmica*, 68(4):954–997, 2014. doi:10.1007/S00453-012-9706-7.
- 10 Walter Didimo. Density of straight-line 1-planar graph drawings. *Inf. Process. Lett.*, 113(7):236–240, 2013. doi:10.1016/j.ipl.2013.01.013.
- 11 Walter Didimo. Right angle crossing drawings of graphs. In *Beyond Planar Graphs*, pages 149–169. Springer, 2020. doi:10.1007/978-981-15-6533-5_9.
- 12 Walter Didimo, Peter Eades, and Giuseppe Liotta. Drawing graphs with right angle crossings. *Theor. Comput. Sci.*, 412(39):5156–5166, 2011. doi:10.1016/j.tcs.2011.05.025.
- 13 Walter Didimo, Giuseppe Liotta, and Fabrizio Montecchiani. A survey on graph drawing beyond planarity. *ACM Comput. Surv.*, 52(1):4:1–4:37, 2019. doi:10.1145/3301281.
- 14 Peter Eades, Brendan D. McKay, and Nicholas C. Wormald. On an edge crossing problem. In *9th Australian Computer Science Conference*, pages 327–334, 1986. URL: <https://api.semanticscholar.org/CorpusID:116397155>.
- 15 Peter Eades and Sue Whitesides. Drawing graphs in two layers. *Theor. Comput. Sci.*, 131(2):361–374, 1994. doi:10.1016/0304-3975(94)90179-1.
- 16 Peter Eades and Nicholas C. Wormald. Edge crossings in drawings of bipartite graphs. *Algorithmica*, 11(4):379–403, 1994. doi:10.1007/BF01187020.

- 17 M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- 18 Seok-Hee Hong. Beyond planar graphs: Introduction. In *Beyond Planar Graphs*, pages 1–9. Springer, 2020. doi:10.1007/978-981-15-6533-5_1.
- 19 Seok-Hee Hong, Michael Kaufmann, Stephen G. Kobourov, and János Pach. Beyond-planar graphs: Algorithmics and combinatorics (Dagstuhl Seminar 16452). *Dagstuhl Reports*, 6(11):35–62, 2016. doi:10.4230/DagRep.6.11.35.
- 20 Seok-Hee Hong and Hiroshi Nagamochi. A linear-time algorithm for testing full outer-2-planarity. *Discret. Appl. Math.*, 255:234–257, 2019. doi:10.1016/J.DAM.2018.08.018.
- 21 Seok-Hee Hong and Takeshi Tokuyama, editors. *Beyond Planar Graphs, Communications of NII Shonan Meetings*. Springer, 2020. doi:10.1007/978-981-15-6533-5.
- 22 Michael Jünger and Petra Mutzel. 2-layer straightline crossing minimization: Performance of exact and heuristic algorithms. *J. Graph Algorithms Appl.*, 1(1):1–25, 1997. doi:10.7155/JGAA.00001.
- 23 Michael Kaufmann, Boris Klemz, Kristin Knorr, Meghana M. Reddy, Felix Schröder, and Torsten Ueckerdt. The density formula: One lemma to bound them all. *CoRR*, abs/2311.06193, 2023. doi:10.48550/arXiv.2311.06193.
- 24 Stephen G. Kobourov, Giuseppe Liotta, and Fabrizio Montecchiani. An annotated bibliography on 1-planarity. *Comput. Sci. Rev.*, 25:49–67, 2017. doi:10.1016/j.cosrev.2017.06.002.
- 25 Giordano Da Lozzo, Vít Jelínek, Jan Kratochvíl, and Ignaz Rutter. Planar embeddings with small and uniform faces. In Hee-Kap Ahn and Chan-Su Shin, editors, *Algorithms and Computation - 25th International Symposium, ISAAC 2014, Jeonju, Korea, December 15-17, 2014, Proceedings*, volume 8889 of *Lecture Notes in Computer Science*, pages 633–645. Springer, 2014. doi:10.1007/978-3-319-13075-0_50.
- 26 János Pach and Géza Tóth. Graphs drawn with few crossings per edge. *Combinatorica*, 17(3):427–439, 1997. doi:10.1007/BF01215922.
- 27 Vicente Valls, Rafael Martí, and Pilar Lino. A branch and bound algorithm for minimizing the number of crossing arcs in bipartite graphs. *European Journal of Operational Research*, 90(2):303–319, 1996. doi:10.1016/0377-2217(95)00356-8.
- 28 David R. Wood. 2-layer graph drawings with bounded pathwidth. *J. Graph Algorithms Appl.*, 27(9):843–851, 2023. doi:10.7155/JGAA.00647.

Crossing Numbers of Beyond Planar Graphs Re-Visited: A Framework Approach

Markus Chimani ✉ 

Theoretical Computer Science, Osnabrück University, Germany

Torben Donzelmann ✉

Theoretical Computer Science, Osnabrück University, Germany

Nick Kloster ✉

Theoretical Computer Science, Osnabrück University, Germany

Melissa Koch ✉

Theoretical Computer Science, Osnabrück University, Germany

Jan-Jakob Völlering ✉

Theoretical Computer Science, Osnabrück University, Germany

Mirko H. Wagner ✉ 

Theoretical Computer Science, Osnabrück University, Germany

Abstract

Beyond planarity concepts (prominent examples include k -planarity or fan-planarity) apply certain restrictions on the allowed patterns of crossings in drawings. It is natural to ask, how much the number of crossings may increase over the traditional (unrestricted) crossing number. Previous approaches to bound such ratios, e.g. [13, 28], require very specialized constructions and arguments for each considered beyond planarity concept, and mostly only yield asymptotically non-tight bounds. We propose a very general proof framework that allows us to obtain asymptotically tight bounds, and where the concept-specific parts of the proof typically boil down to a couple of lines. We show the strength of our approach by giving improved or first bounds for several beyond planarity concepts.

2012 ACM Subject Classification Mathematics of computing → Graph theory

Keywords and phrases Beyond planarity, crossing number, crossing ratio, proof framework

Digital Object Identifier 10.4230/LIPIcs.GD.2024.33

Related Version *Extended Version*: <https://arxiv.org/abs/2407.05057>

1 Introduction

Throughout this paper, we only consider *simple* graphs, i.e., no self-loops or multi-edges. Given a graph G , let n and m denote its number of vertices $V(G)$ and edges $E(G)$, respectively. A drawing \mathcal{D} of G is a mapping of $V(G)$ to distinct points in \mathbb{R}^2 , and $E(G)$ to curves connecting the respective end points. Such a curve must not contain any vertex point other than its end points. When two edge curves intersect on an internal point x , this is a *crossing* of these two edges at x . The *crossing number* $cr(G)$ is the smallest number of crossings over all drawings. To achieve it, we can safely assume *simple drawings*, i.e., no three edges cross at a common point, no edge crosses itself or adjacent edges, and no pair of edges crosses multiple times. A graph is *planar* if it allows a crossing-free drawing. In the last decades, several *beyond planarity* concepts have been established that generalize planar graphs in that certain special crossing patterns are allowed or forbidden, see, e.g., [15, 17] for overviews. A prominent example is k -planarity, where any edge may be crossed at most k times.



© Markus Chimani, Torben Donzelmann, Nick Kloster, Melissa Koch, Jan-Jakob Völlering, and Mirko H. Wagner;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 33; pp. 33:1–33:17



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

We are interested in the minimum number of crossings within any beyond planarity restricted drawing. Intuitively speaking, a beyond planarity concept is a way to (try to) formalize our understanding of what aspects constitute a readable drawing. Still, within these restrictions it is natural to ask for a drawing with the least number of crossings. Alternatively, we can ask how much it “costs” in terms of crossings to follow a certain drawing paradigm.

Formally, let $\mathfrak{B}(k)$ denote any beyond planarity concept, where we omit the parameter k for parameterless beyond planarity concepts. Then, $\mathcal{G}_{\mathfrak{B}(k)}$ ($\mathcal{G}_{\mathfrak{B}(k)}(n)$) denotes the set of graphs (on n vertices, respectively) allowing a $\mathfrak{B}(k)$ -drawing. The $\mathfrak{B}(k)$ -crossing number $\text{cr}_{\mathfrak{B}(k)}(G)$ for a graph $G \in \mathcal{G}_{\mathfrak{B}(k)}$ is the least number of crossings over all $\mathfrak{B}(k)$ -drawings of G . The crossing ratio $\rho_{\mathfrak{B}(k)}(n)$ of $\mathfrak{B}(k)$ is the largest attainable ratio between the $\mathfrak{B}(k)$ -crossing number and the (normal) crossing number, over all n -vertex $\mathfrak{B}(k)$ -graphs:

$$\rho_{\mathfrak{B}(k)}(n) := \sup_{G \in \mathcal{G}_{\mathfrak{B}(k)}(n)} \frac{\text{cr}_{\mathfrak{B}(k)}(G)}{\text{cr}(G)}.$$

Crossing ratios of beyond planarity concepts have for the first time been explicitly considered in [12, 13], where linear lower bounds $\Omega(n)$ were established for 1-planar, k -quasi-planar, and (weakly) fan-planar graphs. In all but the first case, the upper bounds were (at least) quadratic in n , and it was conjectured in [13] that the real lower bounds should rather be $\Omega(n^2)$. Later, [28] generalized the results to k -planarity and seven more beyond planarity concepts. However, none of the provided bounds were tight except for planarly-connected and straight-line RAC. In both publications, each planarity measure needs a very specific construction and intricate specialized arguments to prove the lower bounds (which tend to always be the most complicated part in crossing number proofs). Furthermore, both only consider simple drawings, when establishing upper bounds on the crossing ratio. Simple and non-simple crossing ratios may differ, if there are $\mathfrak{B}(k)$ -graphs that only allow non-simple $\mathfrak{B}(k)$ -drawings. Our bounds below work with and without the simplicity assumption.

Our contribution. In this paper, we propose a general framework that, when applicable, simplifies crossing ratio proofs down to only a couple of $\mathfrak{B}(k)$ -specific lines (10–20 lines for the lower bound). In contrast to previous schemes, this framework is able to prove asymptotically tight bounds for all considered beyond planarity concepts, and our proofs further show that the crossing ratio is already achieved between any two subsequent parameterizations k and $k + 1$, in all considered parametrized concepts. Table 1 summarizes our results. A key idea lies in the simplification of the necessary lower bound arguments, by turning them into counting arguments over a set of (sufficiently) disjoint Kuratowski-subdivisions (see Section 2). In particular, this allows us to avoid any intricate discussions about alternative drawing possibilities, the simplicity of drawings, etc. We present the proof framework in Section 3, and showcase its versatility and strength for various beyond planarity concepts in Section 4.

2 Preliminaries

Kuratowski subdivisions. A graph G is planar if and only if it does not contain a K_5 or $K_{3,3}$ subdivision (summarily called *Kuratowski subdivisions*) [24]. Let K be a $K_{3,3}$ subdivision in G . The six degree-3 vertices of K are the *Kuratowski nodes* of K , and the paths between them (not containing other Kuratowski nodes) its *Kuratowski paths*; by replacing each Kuratowski path by a single edge, one obtains $K_{3,3}$. Two Kuratowski paths are *adjacent*, if they share a common Kuratowski node. Clearly, we have $\text{cr}(G) \geq 1$, and any drawing \mathcal{D} of G (even

■ **Table 1** Overview on our results on the crossing ratios of beyond planarity concepts for n -vertex graphs. Formally, [13] considers weakly fan-planar, but their proof works for all 4 variants. The column “ $\bar{\varrho}$?” denotes whether we also show the (same) rectilinear crossing ratios in Section 4.10. The *-marked previous upper bounds need no explicit discussion of non-simple drawings by the nature of the beyond planarity concept. All our bounds, except the upper bounds for adjacency-crossing and fan-crossing, work with and without the simplicity assumption.

beyond planarity concept	previous best	our results	Section	$\bar{\varrho}$?
k -planar	$\Omega(n/k) \cap \mathcal{O}(k\sqrt{kn})$ * [28]	$\Theta(n)$	4.1	✓
k -vertex-planar	—	$\Theta(n)$	4.2	✓
IC-planar	—	$\Theta(n)$	4.3	✓
NIC-planar	—	$\Theta(n)$	4.4	✓
NNIC-planar	—	$\Theta(n^2)$	4.5	✓
k -fan-crossing-free	$\Omega(n^2/k^3) \cap \mathcal{O}(k^2n^2)$ * [28]	$\Theta(n^2/k)$	4.5	✓
straight-line RAC	$\Theta(n^2)$ * [28]	(direct corollary)	4.5	—
adjacency-crossing	$\Omega(n) \cap \mathcal{O}(n^2)$ [13]	$\Theta(n^2)$	4.6	×
fan-crossing				
weakly fan-planar				
strongly fan-planar				
k -edge-crossing	—	$\Theta(k)$	4.7	✓
k -gap-planar	$\Omega(n/k^3) \cap \mathcal{O}(k\sqrt{kn})$ * [28]	$\Theta(n/k)$	4.8	✓
k -apex-planar	$\Omega(n/k) \cap \mathcal{O}(k^2n^2)$ [28]	$\Theta(n^2/k)$	4.9	✓
skewness- k	$\Omega(n/k) \cap \mathcal{O}(kn)$ [28]	$\Theta(n)$	4.9	✓

a non-simple one) contains at least one crossing x between two edges from *non-adjacent* Kuratowski paths. We say that x *covers* K . Every Kuratowski subdivision in G has to be covered in \mathcal{D} ; a single crossing may cover several such subdivisions.

Upper bounds. The *crossing lemma* [2] states that any graph G with n vertices and $m > 4n$ edges has $\text{cr}(G) \in \Omega(\frac{m^3}{n^2})$. We may split $\mathcal{G}_{\mathfrak{B}(k)}(n)$ into graphs $\mathcal{G}_{\mathfrak{B}(k)}^{\text{dense}}(n) := \{G \in \mathcal{G}_{\mathfrak{B}(k)}(n) \mid m > 4n\}$ that are sufficiently dense for the crossing lemma, and $\mathcal{G}_{\mathfrak{B}(k)}^{\text{sparse}}(n) := \mathcal{G}_{\mathfrak{B}(k)}(n) \setminus \mathcal{G}_{\mathfrak{B}(k)}^{\text{dense}}(n)$. It will later turn out that we attain the largest ratios on graphs of the latter subset. Let $\varrho_{\mathfrak{B}(k)}^{\text{sparse}}(n) := \sup_{G \in \mathcal{G}_{\mathfrak{B}(k)}^{\text{sparse}}(n)} \frac{\text{cr}_{\mathfrak{B}(k)}(G)}{\text{cr}(G)}$. Then

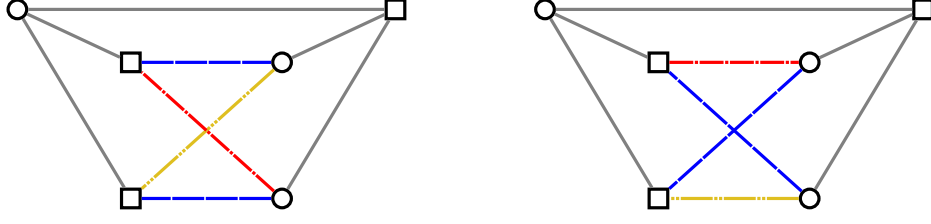
$$\varrho_{\mathfrak{B}(k)}(n) = \max \left\{ \varrho_{\mathfrak{B}(k)}^{\text{sparse}}(n), \sup_{G \in \mathcal{G}_{\mathfrak{B}(k)}^{\text{dense}}(n)} \frac{\text{cr}_{\mathfrak{B}(k)}(G)}{\text{cr}(G)} \right\},$$

where we want to use the crossing lemma to bound the second term in the maximum.

► **Observation 1.** *We have:*

- (a) If $\text{cr}_{\mathfrak{B}(k)}(G) \in \mathcal{O}(m^2)$, then $\varrho_{\mathfrak{B}(k)}(n) \in \mathcal{O}(\varrho_{\mathfrak{B}(k)}^{\text{sparse}}(n) + m^2 \cdot \frac{n^2}{m^3}) \subseteq \mathcal{O}(\varrho_{\mathfrak{B}(k)}^{\text{sparse}}(n) + n)$.
- (b) If $\text{cr}_{\mathfrak{B}(k)}(G) \in \mathcal{O}(mk)$, then $\varrho_{\mathfrak{B}(k)}(n) \in \mathcal{O}(\varrho_{\mathfrak{B}(k)}^{\text{sparse}}(n) + mk \cdot \frac{n^2}{m^3}) \subseteq \mathcal{O}(\varrho_{\mathfrak{B}(k)}^{\text{sparse}}(n) + k)$.

All simple drawings satisfy the prerequisite of Observation 1(a). Furthermore, in the context of connected sparse graphs in $\mathcal{G}_{\mathfrak{B}(k)}^{\text{sparse}}(n)$ (and thus for $\varrho_{\mathfrak{B}(k)}^{\text{sparse}}(n)$), we know $m \in \Theta(n)$.



■ **Figure 1** Red-yellow (**left**) and blue-blue (**right**) drawing of F used for standard drawings of G_ℓ .

1-jumping. The crossing ratio $\varrho_{\mathfrak{B}(k)}(n)$ relates the $\mathfrak{B}(k)$ -restricted crossing number to the normal one. It is natural to ask, how small a parameter $\Delta \geq 1$ can be chosen such that $\varrho_{\mathfrak{B}(k)}^\Delta(n) := \sup_{G \in \mathcal{G}_{\mathfrak{B}(k)}(n)} \frac{\text{cr}_{\mathfrak{B}(k)}(G)}{\text{cr}_{\mathfrak{B}(k+\Delta)}(G)}$ attains the same asymptotic bound. We say $\varrho_{\mathfrak{B}(k)}(n)$ is Δ -jumping if $\varrho_{\mathfrak{B}(k)}^\Delta(n) \in \Omega(\varrho_{\mathfrak{B}(k)}(n))$. For all concepts in Section 4 except k -gap-planarity (Section 4.8), we will observe 1-jumping as our upper bounds on the normal crossing numbers are already attained by $\mathfrak{B}(k+1)$ -drawings.

3 Framework for Proving Lower Bounds on Crossing Ratios

Let $\mathfrak{B}(k)$ be a beyond planarity concept, and $\psi: \mathbb{N}^2 \rightarrow \mathbb{N}$ a function dependent on k and the graph's number of vertices n . To show that the crossing ratio $\varrho_{\mathfrak{B}(k)} \geq \psi(k, n)$, we construct an infinite family $\{G_\ell\}_{\ell \geq \ell_0}$, for some $\ell_0 \in \mathbb{N}$, such that for each G_ℓ we have

$$\frac{\text{cr}_{\mathfrak{B}(k)}(G_\ell)}{\text{cr}(G_\ell)} \geq \psi(k, |V(G_\ell)|).$$

To construct this family, we always start with a *frame* $F = (N, C, \text{col})$, an edge-colored $K_{3,3}$. To avoid ambiguity with the graphs constructed based on F , we call elements of $N := \{v_1, v_2, v_3, w_1, w_2, w_3\}$ *nodes* and elements of $C := \{\{v_i, w_j\} \mid i, j \in \{1, 2, 3\}\}$ *connections*. Each connection $c \in C$ has a color $\text{col}(c)$ that is either *red*, *blue*, *yellow*, or *gray*.

Based on this frame F and some parameter ℓ , we construct *framework graphs* G_ℓ . To this end, we define for each of the four colors *col* a graph S_{col} (which we will call *con-graph* in the following) with two pole vertices s, t . The size of S_{col} often depends on ℓ, k , or both. Intuitively, we may talk about, e.g., a *red* con-graph if it is associated with the color red. We then replace each connection $\{a, b\} \in C$ with a new copy $S_{\{a,b\}}$ of its color's con-graph $S_{\text{col}(\{a,b\})}$, identifying this con-graphs' poles s and t with a and b , respectively.

In most of our proofs, the con-graphs are rather simple: an (i, j) -*bundle* is a con-graph consisting of i internally-vertex-disjoint s - t -paths, each of length j (i.e., j edges in each path); an $(i, j)^+$ -*bundle* furthermore contains the edge $\{s, t\}$. In all our constructions, the yellow con-graph is the single edge $\{s, t\}$, which can equivalently be seen as a $(1, 1)$ -bundle.

Standard Drawings of G_ℓ . In our framework, we need to show (i) that G_ℓ is indeed in $\mathcal{G}_{\mathfrak{B}(k)}(n)$, (ii) an upper bound on $\text{cr}(G_\ell)$, and (iii) a lower bound on $\text{cr}_{\mathfrak{B}(k)}(G_\ell)$. We show (i) and (ii) by constructing *standard drawings* of G_ℓ : These drawings are constructed by first drawing the frame F ; in particular this maps connections to curves in the plane. We obtain a drawing of G_ℓ by drawing each con-graph in a small neighborhood of its corresponding connection's curve (instead of drawing the connection itself). Whenever possible, we draw con-graphs planarly with their poles on their local outer face; otherwise we explicitly specify their drawing. Apart from this, we only need to specify how to draw crossing con-graphs.

Unless specified otherwise, our frame F contains a 4-cycle of a blue, a yellow, another blue, and a red connection in this order; its remaining five connections are gray. Thus, to cover the $K_{3,3}$, there has to be a red-yellow crossing, a blue-blue crossing, or a crossing involving a gray connection. We can classify drawings of F with a single crossing by the colors of the crossing connections; we are in particular interested in a red-yellow and a blue-blue drawing of F , cf. Figure 1. Such drawings yield corresponding standard drawings of G_ℓ . We use a red-yellow standard drawing to show an upper bound on $\text{cr}(G_\ell)$. As the yellow con-graph is only a single edge, the number of crossings is kept low. However, the red con-graph is chosen such that this drawing is not a $\mathfrak{B}(k)$ -drawing. We use a blue-blue standard drawing to show that G_ℓ is in $\mathcal{G}_{\mathfrak{B}(k)}(n)$. However, the blue con-graphs are chosen such that many crossings arise. Lastly, the gray con-graphs are chosen such that a standard drawing with a crossed gray connection is not a $\mathfrak{B}(k)$ -drawing and also requires many crossings.

Lower bounding $\text{cr}_{\mathfrak{B}(k)}(G_\ell)$. To prove a lower bound on $\text{cr}_{\mathfrak{B}(k)}(G_\ell)$, we aim to show that the number of crossings in the drawing establishing $G_\ell \in \mathcal{G}_{\mathfrak{B}(k)}(n)$ is asymptotically optimal. Until now, we only considered standard drawings of G_ℓ , where each con-graph was treated as a unit. For the proof of the lower bound on $\text{cr}_{\mathfrak{B}(k)}(G_\ell)$, we consider an *arbitrary* $\mathfrak{B}(k)$ -drawing \mathcal{D} of G_ℓ . In particular, therein con-graphs may, for example, partially cross themselves or one another. For every connection $\mathfrak{c} \in C$, let $P_\mathfrak{c}$ be a (not necessarily maximal) set of s - t -paths in the con-graph $S_\mathfrak{c}$. If a con-graph $S_\mathfrak{c}$ is a bundle, then we use the set of all its edge-disjoint paths as $P_\mathfrak{c}$; otherwise we will define it specifically. Let $w(\mathfrak{c}) := w(S_\mathfrak{c}) := |P_\mathfrak{c}|$ be the *width* of $S_\mathfrak{c}$; further, let the *height* $h(\mathfrak{c}) := h(S_\mathfrak{c}) := \max_{p \in P_\mathfrak{c}} |p|$ be the number of edges in the longest path in $P_\mathfrak{c}$. For an edge $e \in S_\mathfrak{c}$, let $P_\mathfrak{c}[e]$ be those paths in $P_\mathfrak{c}$ that contain e . We define $\mathcal{K} := \mathcal{K}(G_\ell, N)$ as the set of all $K_{3,3}$ subdivisions of G_ℓ that have N as their Kuratowski nodes and where each Kuratowski path is from $\bigcup_{\mathfrak{c} \in C} P_\mathfrak{c}$. There is a mapping between the Kuratowski paths of \mathcal{K} and the connections C of F , such that each Kuratowski path consists of an s - t -path in the respective connection's con-graph.

Consider some Kuratowski subdivision $K \in \mathcal{K}$ that is covered in \mathcal{D} by a crossing x between $e_1, e_2 \in E(G)$. For $i \in \{1, 2\}$, let $\mathfrak{c}_i \in C$ be the connection with $e_i \in S_{\mathfrak{c}_i}$. The connections \mathfrak{c}_1 and \mathfrak{c}_2 are non-adjacent in F . The crossing x may cover all Kuratowski subdivisions that have a Kuratowski path from $P_{\mathfrak{c}_1}[e_1]$ and one from $P_{\mathfrak{c}_2}[e_2]$, but no other of \mathcal{K} . By definition of \mathcal{K} , the crossing x covers at most $\frac{|P_{\mathfrak{c}_1}[e_1]| \cdot |P_{\mathfrak{c}_2}[e_2]|}{|P_{\mathfrak{c}_1}| \cdot |P_{\mathfrak{c}_2}|}$ of the Kuratowski subdivisions in \mathcal{K} . In most cases (in particular whenever $S_{\mathfrak{c}_1}, S_{\mathfrak{c}_2}$ are both bundles) we have $|P_{\mathfrak{c}_1}[e_1]| = |P_{\mathfrak{c}_2}[e_2]| = 1$, and then the crossing x covers $\frac{1}{|P_{\mathfrak{c}_1}| \cdot |P_{\mathfrak{c}_2}|}$ of the \mathcal{K} -subdivisions.

In our proofs, we sometimes consider a specific subdrawing of \mathcal{D} . Let $\mathfrak{c} \in C$ be a connection with con-graph $S_\mathfrak{c}$ and let R be an s - t -path in $S_\mathfrak{c}$. Subdrawing \mathcal{D}' is the R -restriction of \mathcal{D} where all of $S_\mathfrak{c}$ except for R (a bundle of width 1) is removed. Within \mathcal{D}' , R is considered to be the con-graph of \mathfrak{c} , and we are interested in the minimum number of crossings covering the Kuratowski subdivisions $\mathcal{K}_{\mathcal{D}'} \subseteq \mathcal{K}$ in \mathcal{D}' . We may consider an (R_1, R_2) -restriction, where we perform this operation twice, for distinct connections.

4 Proving Crossing Ratios

We now use our framework to prove the crossing ratio for several beyond planarity concepts. Generally, whenever we define such a concept $\mathfrak{B}(k)$, we do so via $\mathfrak{B}(k)$ -drawings. This is *always* to be understood to implicitly define the corresponding graph class as those graphs that allow such a $\mathfrak{B}(k)$ -drawing: e.g., a graph is k -planar if it allows a k -planar drawing. See Table 1 for previously established crossing ratios (if any).

4.1 k -Planar: $\varrho_{k\text{-pl}}(n)$

In a k -planar ($k \geq 1$, shorthand “ k -pl”) drawing, no edge is crossed more than k times [25, 26].

► **Theorem 2.** *For every $\ell \geq 41$, there exists a k -planar graph G_ℓ with $n \in \Theta(\ell^2 k)$ vertices such that $\text{cr}_{k\text{-pl}}(G_\ell) \in \Omega(nk)$ and $\text{cr}(G_\ell) \in \mathcal{O}(k)$. Thus $\varrho_{k\text{-pl}}(n) \in \Omega(n)$.*

Proof. Let G_ℓ be a framework graph, where the red con-graph is a $(k+1, 2)$ -bundle, the blue con-graphs are $(\ell k, \ell)$ -bundles, and the gray con-graphs are $(\ell k, 2)$ -bundles, so $n \in \Theta(\ell^2 k)$. A red-yellow standard drawing shows $\text{cr}(G_\ell) \in \mathcal{O}(k)$. A blue-blue standard drawing, see Figure 2a for how the blue con-graphs cross, shows that G_ℓ is k -planar.

Let \mathcal{D} be a k -planar drawing of G_ℓ . At least one s - t -path R in the red con-graph is not crossed by the unique yellow edge, as the latter has at most k crossings. From now on, we thus only consider the R -restriction \mathcal{D}' of \mathcal{D} , which does not have any red-yellow crossings. Let $\mathfrak{c}, \tilde{\mathfrak{c}}$ be two connections. Any crossing between edges from \mathfrak{c} and $\tilde{\mathfrak{c}}$ can cover at most $\frac{1}{w(\mathfrak{c}) \cdot w(\tilde{\mathfrak{c}})}$ of the $\mathcal{K}_{\mathcal{D}'}$ -subdivisions (recall that the width of the red con-graph here in \mathcal{D}' is 1). By k -planarity, there can be at most $k \cdot \min\{|E(\mathfrak{c})|, |E(\tilde{\mathfrak{c}})|\}$ crossings between edges from \mathfrak{c} and $\tilde{\mathfrak{c}}$. As all con-graphs are bundles, the number of edges in each con-graph is the product of its height and width. Thus, such crossings can overall cover at most $k \cdot \frac{\min\{w(\mathfrak{c}) \cdot h(\mathfrak{c}), w(\tilde{\mathfrak{c}}) \cdot h(\tilde{\mathfrak{c}})\}}{w(\mathfrak{c}) \cdot w(\tilde{\mathfrak{c}})}$ of the $\mathcal{K}_{\mathcal{D}'}$ -subdivisions. Let \mathfrak{c} be a gray connection, then crossings between \mathfrak{c} and another connection $\tilde{\mathfrak{c}}$ can cover at most

$$k \cdot \frac{\min\{\ell k \cdot 2, w(\tilde{\mathfrak{c}}) \cdot h(\tilde{\mathfrak{c}})\}}{\ell k \cdot w(\tilde{\mathfrak{c}})} = \left. \begin{array}{l} \frac{1 \cdot 1}{\ell \cdot 1} \quad \text{if } \tilde{\mathfrak{c}} \text{ is yellow} \\ \frac{1 \cdot 2}{\ell \cdot 1} \quad \text{if } \tilde{\mathfrak{c}} \text{ is red} \\ \frac{\ell k \cdot 2}{\ell \cdot \ell k} \quad \text{if } \tilde{\mathfrak{c}} \text{ is blue} \\ \frac{\ell k \cdot 2}{\ell \cdot \ell k} \quad \text{if } \tilde{\mathfrak{c}} \text{ is gray} \end{array} \right\} \leq \frac{2}{\ell}$$

of the $\mathcal{K}_{\mathcal{D}'}$ -subdivisions. Five con-graphs are gray and four connections are non-adjacent to a given connection, so crossings involving a gray edge can cover at most $5 \cdot 4 \cdot \frac{2}{\ell} \leq \frac{40}{41}$ of the $\mathcal{K}_{\mathcal{D}'}$ -subdivisions. Thus, blue-blue crossings cover at least $\frac{1}{41}$ of the $\mathcal{K}_{\mathcal{D}'}$ -subdivisions. Since each such crossing covers at most $\frac{1}{(\ell k)^2}$ of the $\mathcal{K}_{\mathcal{D}'}$ -subdivisions, this yields a total of $\Omega((\ell k)^2)$ crossings in \mathcal{D} and so $\text{cr}_{k\text{-pl}}(G) \in \Omega(nk)$ and $\varrho_{k\text{-pl}}(n) \in \Omega(n)$. ◀

A non- k -planar drawing has an edge that is crossed at least $k+1$ times, and $\text{cr}_{k\text{-pl}}(G_\ell) \leq mk$, as in a k -planar drawing each edge is crossed at most k times. Thus, Theorem 2 and Observation 1(b) with $\varrho_{k\text{-pl}}^{\text{sparse}}(n) \leq \frac{mk}{k+1} \in \mathcal{O}(n)$ yield:

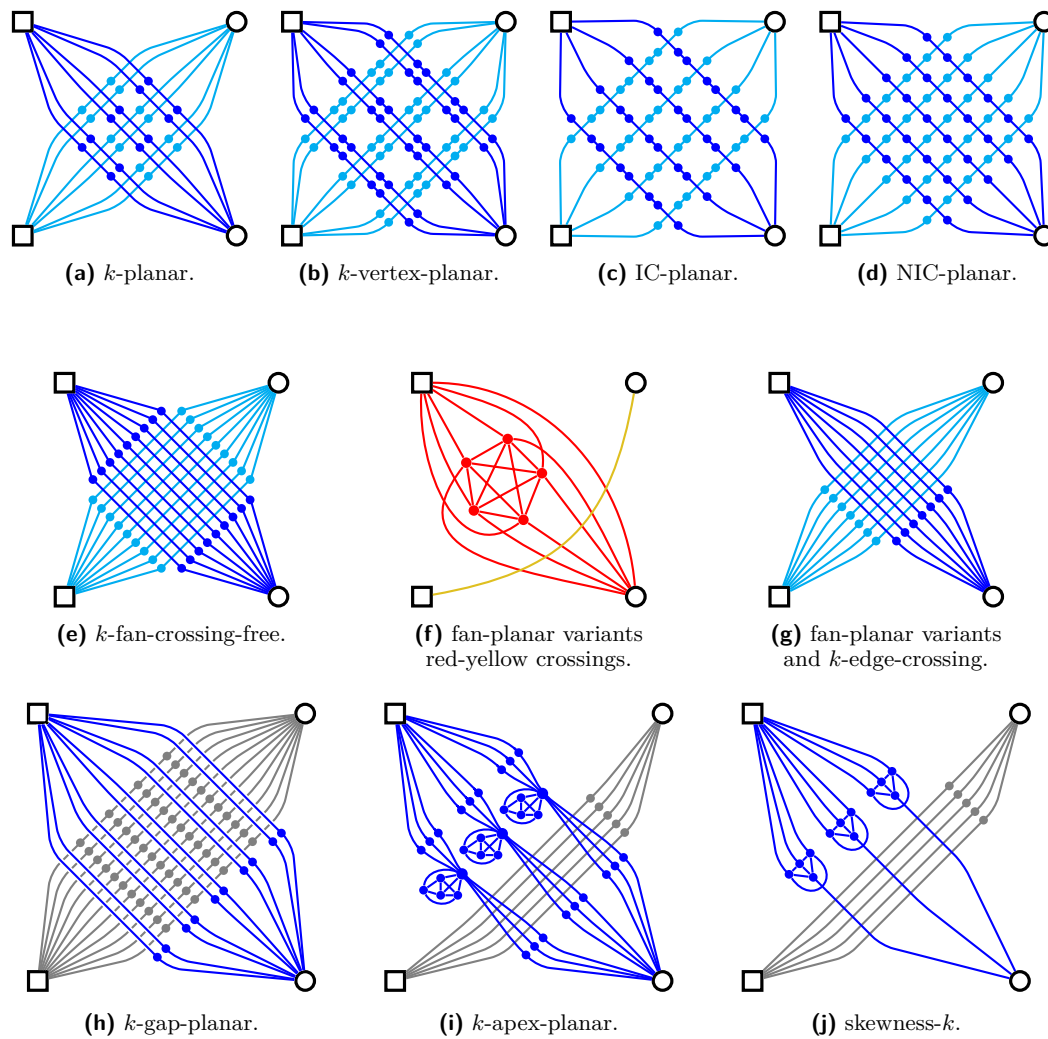
► **Corollary 3.** *The k -planar crossing ratio $\varrho_{k\text{-pl}}(n)$ is in $\Theta(n)$ and 1-jumping.*

4.2 k -Vertex-Planar: $\varrho_{k\text{-vp}}(n)$

A vertex v is *adjacent* to a crossing x , if x is on an edge incident to v . In a k -vertex-planar (shorthand “ k -vp”) drawing, no vertex may be adjacent to more than k crossings.

► **Theorem 4.** *For every $\ell \geq 11$, there exists a k -vertex-planar graph G_ℓ with $n \in \Theta(\ell^2 k)$ vertices such that $\text{cr}_{k\text{-vp}}(G_\ell) \in \Omega(nk)$ and $\text{cr}(G_\ell) \in \mathcal{O}(k)$. Thus $\varrho_{k\text{-vp}}(n) \in \Omega(n)$.*

Proof. Let G_ℓ be a framework graph, where the red con-graph is a $(k+1, 2)$ -bundles, the blue con-graphs are $(\ell k, 2\ell+1)$ -bundles, and the gray con-graphs are $(\ell k, 2)$ -bundles, so $n \in \Theta(\ell^2 k)$. A red-yellow standard drawing shows $\text{cr}(G_\ell) \in \mathcal{O}(k)$. A blue-blue standard drawing shows that G_ℓ is k -vertex-planar, cf. Figure 2b.



■ **Figure 2** Visualizations of standard drawings of crossing con-graphs, for different beyond planarity concepts. The con-graphs are colored consistently to their color type (blue, gray, red, yellow); we use two shades of blue when showing crossing blue con-graphs. The depicted values for ℓ are chosen for easy comprehension of the drawing paradigm (e.g., (a) shows $\ell = 3$, $k = 2$); they are smaller than required in the proofs.

Let \mathcal{D} be a k -vertex-planar drawing of G_ℓ . Since the yellow edge e has at most k crossings, there exists an s - t -path R in the red con-graph not crossed by e . Let \mathcal{D}' be the R -restriction of \mathcal{D} ; it does not have any red-yellow crossings. Let c be a gray connection with con-graph S_c ; it has width ℓk . Since any crossing on S_c is adjacent to one of the poles of S_c , there are at most $2k$ crossings on S_c and thus crossings involving at least one edge from S_c can cover at most $\frac{2k}{\ell k} = \frac{2}{\ell}$ of the $\mathcal{K}_{\mathcal{D}'}$ -subdivisions. Five con-graphs are gray, so crossings involving a gray edge can cover at most $5 \cdot \frac{2}{\ell} \leq \frac{10}{11}$ of the $\mathcal{K}_{\mathcal{D}'}$ -subdivisions. Thus, blue-blue crossings cover at least $\frac{1}{11}$ of the $\mathcal{K}_{\mathcal{D}'}$ -subdivisions. Since each such crossing covers at most $\frac{1}{(\ell k)^2}$ of the $\mathcal{K}_{\mathcal{D}'}$ -subdivisions, this yields $\text{cr}_{k\text{-vp}}(G) \in \Omega((\ell k)^2) = \Omega(nk)$. ◀

A non- k -vertex-planar drawing has at least $k + 1$ crossings, and $\text{cr}_{k\text{-vp}}(G) \leq nk$, as every vertex may be adjacent to at most k crossings. Thus, Theorem 4 and Observation 1(b) with $\varrho_{k\text{-vp}}^{\text{sparse}}(n) \leq \frac{nk}{k+1} \in \mathcal{O}(n)$ yield:

► **Corollary 5.** *The k -vertex-planar crossing ratio $\varrho_{k\text{-vp}}(n)$ is in $\Theta(n)$ and 1-jumping.*

4.3 IC-Planar: $\varrho_{\text{IC}}(n)$

Two crossings *share a vertex* v if they are both on edges incident to v . A drawing is *IC-planar* (*independent crossings planar*, shorthand “IC”), if no two crossings share a vertex [3, 23, 30]. Note that cr_{IC} must not be mistaken for the *independent crossing number*, where there are no restrictions on the crossings, but one only counts crossings between non-adjacent edges [27]. In fact, IC-planarity is equivalent to 1-vertex-planarity. We can obtain near-tight non-asymptotic bounds for this special case.

► **Theorem 6.** *For every $\ell \geq 2$, there exists an IC-planar graph G_ℓ with $n = 4\ell^2 + 12$ vertices such that $\text{cr}_{\text{IC}}(G_\ell) = \frac{n}{4} - 3$ and $\text{cr}(G) \leq 2$. Thus $\varrho_{\text{IC}}(n) \geq \frac{n}{8} - \frac{3}{2}$.*

Proof. Let G_ℓ be a framework graph, where the red and gray con-graphs are $(1, 2)^+$ -bundles (i.e., triangles), and the blue con-graphs are $(\ell, 2\ell + 1)$ -bundles, so $n = 4\ell^2 + 12$. The graph has $n = 6 + 6 + 2 \cdot 2\ell \cdot \ell = 4\ell^2 + 12$ vertices. A red-yellow standard drawing shows that $\text{cr}(G_\ell) \leq 2$. A blue-blue standard drawing shows that the graph is IC-planar, cf. Figure 2c.

Consider an IC-planar drawing \mathcal{D} of G_ℓ . For each red or gray connection $\mathfrak{c} = \{s, t\}$, at least one of the two s - t -paths in its con-graph has to be uncrossed, as otherwise there would be non-independent crossings. Since the yellow connection is adjacent to the blue connections, all Kuratowski subdivisions have to be covered by blue-blue crossings. Since blue con-graphs have width ℓ , any blue-blue crossing can only cover at most $\frac{1}{\ell^2}$ of the \mathcal{K} -subdivisions, yielding $\text{cr}(G_\ell) \geq \ell^2 = \frac{n}{4} - 3$. ◀

A non-IC-planar drawing has at least 2 crossings, and $\text{cr}_{\text{IC}}(G) \leq n/4$, as no crossings share a vertex, thus $\varrho_{\text{IC}}(n) \leq \frac{n/4}{2} = \frac{n}{8}$. Together with Theorem 6 we have:

► **Corollary 7.** *For the IC-planar crossing ratio it holds that $\varrho_{\text{IC}}(n) \in [\frac{n}{8} - \frac{3}{2}, \frac{n}{8}]$.*

We note that by shrinking the length of a single path in each of the two blue con-graphs by 2, our construction still works and (since then $n = 4\ell^2 + 8$) we would get the slightly tighter lower bound of $\frac{n}{8} - 1$, at the expense of a slightly longer proof.

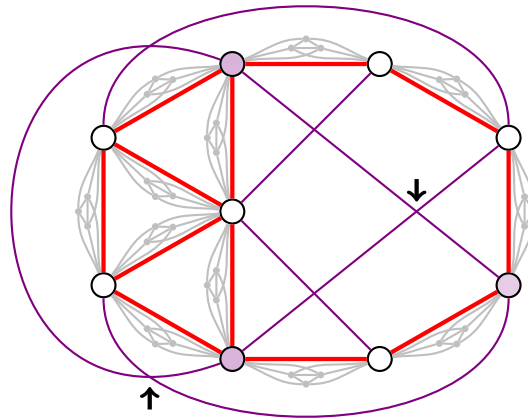
4.4 NIC-Planar: $\varrho_{\text{NIC}}(n)$

In an *NIC-planar* (*nearly independent crossings*, shorthand “NIC”) drawing, any two crossings may share at most one vertex [29, 30]. Clearly, each edge is involved in at most one crossing.

► **Theorem 8.** *For every $\ell \geq 4$, there exists an NIC-planar graph G_ℓ with $n \in \Theta(\ell^2)$ vertices such that $\text{cr}_{\text{NIC}}(G_\ell) \in \Omega(n)$ and $\text{cr}(G_\ell) \leq 2$. Thus $\varrho_{\text{IC}}(n) \in \Omega(n)$.*

Proof. Let G_ℓ be a framework graph where the red con-graph is a $(1, 2)^+$ -bundle, the blue con-graphs are $(\ell, \ell + 2)$ -bundles, and the gray con-graphs are $(\ell, 2)$ -bundles, so $n \in \Theta(\ell^2)$. A red-yellow standard drawing shows $\text{cr}(G_\ell) \leq 2$. A blue-blue standard drawing, shows that G_ℓ is NIC-planar, cf. Figure 2d.

Consider a NIC-planar drawing \mathcal{D} of G_ℓ . Since the yellow edge can be crossed at most once and all other con-graphs have width at least 2, a crossing on the yellow edge covers at most $\frac{1}{2}$ of the \mathcal{K} -subdivisions. Any red-gray crossing can cover at most $\frac{1}{2\ell}$ of the \mathcal{K} -subdivisions. As



■ **Figure 3** A fan-crossing-free but not NNIC-planar graph. The red edges cannot be crossed in any FCF drawing. The depicted fan-crossing-free subembedding of the non-gray edges is unique (up to mirroring), but not NNIC-planar: arrows point to two crossings that share three (shaded) vertices.

each of the three red edges is involved in at most one crossing, such crossings cover at most $\frac{3}{2\ell}$ of the \mathcal{K} -subdivisions in total. Therefore, at least $1 - \frac{1}{2} - \frac{3}{2\ell} \geq \frac{1}{8}$ of the \mathcal{K} -subdivisions have to be covered by crossings involving only gray and blue edges. Any such crossing covers at most $\frac{1}{2}$ of the \mathcal{K} -subdivisions, yielding $\text{cr}_{\text{NIC}}(G_\ell) \in \Omega(\ell^2) = \Omega(n)$. ◀

A non-NIC-planar drawing has at least 2 crossings, and $\text{cr}_{\text{IC}}(G) \leq m/2$, as each crossing requires two edges not involved in any other crossing. Since NIC-planar graphs have at most $\frac{18}{5}n$ edges [4], we have $\varrho_{\text{NIC}}(n) \leq \frac{9}{10}n$ and together with Theorem 8:

► **Corollary 9.** *The NIC-planar crossing ratio $\varrho_{\text{NIC}}(n)$ is in $\Theta(n)$.*

4.5 NNIC-Planar and k -Fan-Crossing-Free: $\varrho_{\text{NNIC}}(n)$ and $\varrho_{k\text{-fcf}}(n)$

A simple drawing is *NNIC-planar* (*nearly nearly independent crossings*, shorthand “NNIC”), if any two crossings share at most two vertices. This beyond planarity concept has seemingly not been investigated before, but it is closely related to *fan-crossing-free* (FCF) drawings [10]: simple drawings where we disallow a single edge to cross over a fan, i.e., multiple edges that share a common incident vertex. Observe that if an edge $\{x, y\}$ would cross two edges incident to a common vertex z , these two crossings would share the three vertices x, y, z . Thus, NNIC may at first sight seem identical to FCF. However, while every NNIC-planar graph is clearly also FCF, the inverse is not necessarily the case:

► **Theorem 10.** *There exist fan-crossing-free graphs that are not NNIC-planar.*

Proof sketch. Figure 3 shows an FCF graph that is not NNIC-planar. We refer to the arXiv-version [11] for a detailed proof of that fact. A central idea is that copies of K_5 can be used as “blocking walls” as they cannot be crossed in an FCF drawing. ◀

More generally, we can consider the generalization of FCF graphs, the *k -fan-crossing-free* (shorthand “ k -fcf”) graphs, $k \geq 2$, where any edge may cross at most $k - 1$ edges of a fan in a simple drawing [10]. Standard FCF is thus identical to 2-fcf.

► **Theorem 11.** *For every $\ell \geq 109$, there exists a k -fan-crossing-free graph G_ℓ with $n \in \Theta(\ell k)$ vertices such that $\text{cr}_{k\text{-fcf}}(G_\ell) \in \Omega(n^2)$ and $\text{cr}(G_\ell) \in \mathcal{O}(k)$. Thus $\varrho_{k\text{-fcf}}(n) \in \Omega(\frac{n^2}{k})$.*

Proof. Let G_ℓ be a framework graph, where the red con-graph is a $(2k, 2)$ -bundle, blue con-graphs are $(\ell k, 3)$ -bundles, and gray con-graphs are $(\ell k, 2)$ -bundles, so $n \in \Theta(\ell k)$. A red-yellow standard drawing shows $\text{cr}(G_\ell) \in \mathcal{O}(k)$. Observe that in a blue con-graph, the “middle” edges of all the ℓk paths are pairwise non-adjacent. We can thus obtain a fan-crossing-free drawing by considering a blue-blue standard drawing where all middle edges of the paths of one blue con-graph cross all middle edges of the paths of the other blue con-graph, cf. Figure 2e.

Let \mathcal{D} be a k -fan-crossing-free drawing of G_ℓ . Each red or gray edge is adjacent to one of the six frame nodes. Therefore, the yellow edge e can cross at most $2(k-1)$ of the red edges, so there exists an s - t -path R in S_{red} not crossed by e . Let \mathcal{D}' be the R -restriction of \mathcal{D} ; it does not have any red-yellow crossings. Consider some edge e in \mathcal{D}' . Let $\mathfrak{c} = \{s, t\}$ be the connection with $e \in E(\mathfrak{c})$. For each of the four frame nodes $a \in N \setminus \{s, t\}$, edge e can have at most $k-1$ crossings with gray edges incident to a . The crossings of e with any gray edge cover at most $4(k-1) \cdot \frac{1}{w(S_{\text{gray}}) \cdot w(\mathfrak{c})} = \frac{4(k-1)}{\ell k \cdot w(\mathfrak{c})}$ of the $\mathcal{K}_{\mathcal{D}'}$ -subdivisions (recall that the width of the red con-graph here in \mathcal{D}' is 1). As all con-graphs are bundles, there are $\sum_{\tilde{\mathfrak{c}} \in C} w(\tilde{\mathfrak{c}}) \cdot h(\tilde{\mathfrak{c}})$ choices for e , so crossings involving a gray edge can cover at most $\sum_{\tilde{\mathfrak{c}} \in C} \frac{4(k-1) \cdot w'(\tilde{\mathfrak{c}}) \cdot h(\tilde{\mathfrak{c}})}{\ell k \cdot w'(\tilde{\mathfrak{c}})}$ of the $\mathcal{K}_{\mathcal{D}'}$ -subdivisions. Since $|C| = 9$, $\max_{\tilde{\mathfrak{c}} \in C} \{h(\tilde{\mathfrak{c}})\} = 3$, and $\ell \geq 109$, this simplifies to at most $9 \cdot \frac{4(k-1) \cdot 3}{\ell k} \leq \frac{108(k-1)}{109k} \leq \frac{108}{109}$. Thus, blue-blue crossings have to cover at least $\frac{1}{109}$ of the $\mathcal{K}_{\mathcal{D}'}$ -subdivisions. As each such crossing covers at most $\mathcal{O}(\frac{1}{(\ell k)^2})$ of the $\mathcal{K}_{\mathcal{D}'}$ -subdivisions, we have $\text{cr}_{k\text{-fcf}}(G_\ell) \in \Omega((\ell k)^2) = \Omega(n^2)$. ◀

In a non- k -fan-crossing-free drawing there has to be an edge that crosses $k \in \mathcal{O}(n)$ edges of a fan, so there are at least k crossings. The k -fan-crossing-free drawing may require $\mathcal{O}(m^2)$ crossings, as by definition k -fan-crossing-free drawings are simple. Therefore, Theorem 11 together with $\varrho_{k\text{-fcf}}^{\text{sparse}}(n) \in \mathcal{O}(\frac{n^2}{k})$ in the context of Observation 1(a), yields:

► **Corollary 12.** *The k -fan-crossing-free crossing ratio $\varrho_{k\text{-fcf}}(n)$ is in $\Theta(\frac{n^2}{k})$ and 1-jumping.*

Since NNIC-planarity is more restrictive than fan-crossing-free (which is k -FCF for $k = 2$), but on the other hand NNIC-planar drawings are also necessarily simple, we also have:

► **Corollary 13.** *The NNIC crossing ratio $\varrho_{\text{NNIC}}(n)$ is in $\Theta(n^2)$.*

A *straight-line RAC* (right-angle crossings) drawing [14] is a drawing where edges are represented by straight lines and all crossings have to be at 90 degrees. Consider any G_ℓ from Theorem 11. In a straight-line RAC drawing, there can be no edge crossing over two fan edges, so $\text{cr}_{\text{sl-RAC}}(G_\ell) \geq \text{cr}_{2\text{-fcf}}(G_\ell)$. However, as witnessed by the blue-blue standard drawing (Figure 2e), each G_ℓ is straight-line RAC drawable. A straight-line RAC graph can have at most $4n - 10$ edges [14], and thus no more than $\mathcal{O}(n^2)$ crossings in any simple drawing. Therefore, our proof automatically also confirms [28, Corollary 15], without the need of an individual construction or proof:

► **Corollary 14.** *The straight-line RAC crossing ratio $\varrho_{\text{sl-RAC}}(n)$ is in $\Theta(n^2)$.*

4.6 Fan-Planar and Variants: $\varrho_{\text{ac}}(n)$, $\varrho_{\text{fc}}(n)$, $\varrho_{\text{wfp}}(n)$, $\varrho_{\text{sfp}}(n)$

A drawing is *adjacency-crossing* [8], if no edge crosses two independent edges. In other words, whenever an edge e crosses some edges f_1, f_2 , the latter two have to have a common incident vertex. A drawing is *fan-crossing* [8], a slight further restriction of the former, if all edges

$F = \{f_1, f_2, \dots\}$ that cross e are incident to a common vertex; observe that the only difference to before is whether e may cross all edges of a triangle. A drawing is *weakly fan-planar*, if all edges from F furthermore cross e “from the same side”; it is *strongly fan-planar*, if also furthermore no endpoint of e is enclosed by e and the edges from F , see [9, 19] for details. In our notations, we use the shorthands “ac”, “fc”, “wfp”, and “sfp” for the above concepts, where $\text{cr}_{\text{ac}}(G) \leq \text{cr}_{\text{fc}}(G) \leq \text{cr}_{\text{wfp}}(G) \leq \text{cr}_{\text{sfp}}(G)$ for any graph G by definition. When considering straight-line drawings, all four concepts are identical.

► **Theorem 15.** *For every $\ell \geq 1$, there exists a strongly fan-planar graph G_ℓ with $n \in \Theta(\ell)$ vertices such that $\text{cr}_{\text{ac}}(G_\ell) \in \Omega(n^2)$ and $\text{cr}(G_\ell) \in \mathcal{O}(1)$. Thus $\varrho_{\text{ac}}(n), \varrho_{\text{fc}}(n), \varrho_{\text{wfp}}(n), \varrho_{\text{sfp}}(n) \in \Omega(n^2)$.*

Proof. Let G_ℓ be a framework graph, in which the red and gray con-graphs are complete graphs on seven vertices (K_7) and the blue con-graphs are $(\ell, 2)$ -bundles, so $n \in \Theta(\ell)$. We need to describe how a K_7 con-graph is drawn in a standard drawing, see Figure 2f: Let $\mathbf{c} = \{s, t\}$ be the connection corresponding to a K_7 con-graph; as per definition, we draw the K_7 in a small neighborhood of \mathbf{c} 's curve such that s, t are on the local outer face. Observe that K_7 can then still be drawn strongly fan-planarly. However, any adjacency-crossing drawing of a K_7 contains at least one s - t -path P where every edge in P is crossed by another edge of the K_7 [7]¹. We may call this property $\langle P \rangle$. A red-yellow standard drawing shows that $\text{cr}(G_\ell) \in \mathcal{O}(1)$. A blue-blue standard drawing, where all blue-blue crossings are adjacent to the same two frame nodes shows that the graph is strongly fan-planar, cf. Figure 2g.

Consider an adjacency-crossing drawing \mathcal{D} of G_ℓ . By property $\langle P \rangle$, we know that each red or gray K_7 con-graph gives rise to a pole-connecting path that cannot be crossed by any edge outside of its con-graph. Since the yellow edge is adjacent to both blue con-graphs, all \mathcal{K} -subdivisions need to be covered by blue-blue crossings. Since each such crossing covers at most $\frac{1}{\ell^2}$ of the \mathcal{K} -subdivisions, we have $\text{cr}_{\text{ac}}(G_\ell) \in \Omega(\ell^2) = \Omega(n^2)$. ◀

For $G \in \mathcal{G}_{\text{wfp}}(n)$, the upper bound $\text{cr}_{\text{wfp}}(G) \in \mathcal{O}(m^2)$, is attained by a simple drawing [20, 21]. In strongly fan-planar drawings, edges cannot cross multiple times (and there can be at most $\mathcal{O}(m^2)$ crossings between adjacent edges, if they exist at all), so the same asymptotic upper bound holds. Therefore, Theorem 15 together with Observation 1(a) yields:

► **Corollary 16.** *The weakly and strongly fan-planar crossing ratios $\varrho_{\text{wfp}}(n), \varrho_{\text{sfp}}(n)$ are in $\Theta(n^2)$.*

There are no results for the number of crossings a non-simple adjacency-crossing or fan-crossing drawing may have. Since a simple drawing has at most $\mathcal{O}(m^2)$ crossings, Theorem 15 together with Observation 1(a) yields:

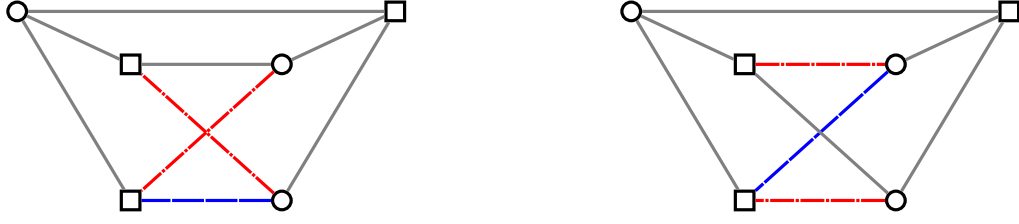
► **Corollary 17.** *The adjacency-crossing and fan-crossing crossing ratios $\varrho_{\text{ac}}^{\text{simple}}(n), \varrho_{\text{fc}}^{\text{simple}}(n)$ are in $\Theta(n^2)$ for simple drawings.*

4.7 k -Edge-Crossing: $\varrho_{k\text{-ec}}(n)$

A drawing is *k -edge-crossing* (shorthand “ k -ec”) if at most k edges have crossings [16, 18].

► **Theorem 18.** *For every $\ell \geq 1$, there exists a k -edge-crossing graph G_ℓ with $n \in \Theta(\ell k)$ vertices such that $\text{cr}_{k\text{-ec}}(G_\ell) \in \Omega(k^2)$ and $\text{cr}(G_\ell) \in \mathcal{O}(k)$. Thus $\varrho_{k\text{-ec}}(n) \in \Omega(k)$.*

¹ We note that [7, Lemma 7] is stated in terms of (weak) fan-planarity, but it is simple to see that the same argument even holds for fan- and adjacency-crossing drawings.



■ **Figure 4** Drawings of the differently colored frame used in Theorems 20, 23, and 25 for the k -gap-planar, k -apex-planar, and skewness- k crossing ratios, respectively.

Proof. Let G_ℓ be a framework graph, where the red con-graph is a $(k, 2)$ -bundle, the gray con-graphs are $(\ell k, 2)$ -bundles, and blue con-graphs are $(\frac{k}{2}, 2)$ -bundles, so $n \in \Theta(\ell k)$. A red-yellow standard drawing shows that $\text{cr}_{k\text{-ec}}(G_\ell) \in \mathcal{O}(k)$. A blue-blue standard drawing shows that G_ℓ is k -edge-crossing, cf. Figure 2g.

Consider a drawing \mathcal{D} of G_ℓ with at most k crossed edges. A con-graph is *cut* if all paths between its poles are crossed. There are at least two cut con-graphs in \mathcal{D} , as otherwise, it would contain non-covered \mathcal{K} -subdivisions. Each cut con-graph contains at least its width many crossed edges. Only yellow and blue con-graphs have widths less than k , so only they can be cut without exceeding k crossed edges. Therefore, at least $\frac{k}{2} + 1$ crossed edges are blue or yellow. Consequently, at most $\frac{k}{2} - 1$ crossed edges are red or gray. Any crossing of these red or gray edges with the unique yellow edge can cover at most $\frac{1}{k}$ of the \mathcal{K} -subdivisions, so they together cover at most $(\frac{k}{2} - 1)\frac{1}{k} \leq \frac{1}{2}$ of the \mathcal{K} -subdivisions. The remaining at least $\frac{1}{2}$ of the \mathcal{K} -subdivisions have to be covered by other crossings, but they cover at most $\mathcal{O}(\frac{1}{k^2})$ of the \mathcal{K} -subdivisions each. This yields $\text{cr}_{k\text{-ec}}(G_\ell) \in \Omega(k^2)$. ◀

In a non- k -ec drawing there are at least $\frac{k}{2}$ crossings as there are at least $k + 1$ crossing edges. In a k -ec drawing there are less than $\frac{k^2}{2} \in \mathcal{O}(mk)$ crossings as each of the at most k crossing edges can cross each other crossing edge, leading to $\varrho_{k\text{-ec}}^{\text{sparse}}(n) \in \mathcal{O}(\frac{k^2}{k}) = \mathcal{O}(k)$ in the context of Observation 1(b); together with Theorem 18 we yield:

► **Corollary 19.** *The k -edge crossing crossing ratio $\varrho_{k\text{-ec}}(n)$ is in $\Theta(k)$ and 1-jumping.*

4.8 k -Gap-Planar: $\varrho_{k\text{-gap-pl}}(n)$

Let $E(x)$ denote the two edges involved in a crossing x . A drawing is *k -gap-planar* (shorthand “ k -gap-pl”) if there exists a mapping from each crossing x to one of $E(x)$, such that any edge $e \in E(G)$ is mapped to at most k times overall [5, 6]. The intuition (and reason for the name) is that we may want to visualize crossings by drawing a “gap” on one of the two involved edge curves, but each edge curve may not attain too many gaps.

► **Theorem 20.** *For every $\ell \geq 5$, there exists a k -gap-planar graph G_ℓ with $n \in \Theta(\ell k)$ vertices such that $\text{cr}_{k\text{-gap-pl}}(G_\ell) \in \Omega(nk)$ and $\text{cr}(G_\ell) \in \mathcal{O}(k^2)$. Thus $\varrho_{k\text{-gap-pl}}(n) \in \Omega(\frac{n}{k})$.*

Proof. Consider a coloring of the frame F where a blue connection is adjacent to two independent red connections, and all other connections are gray, see Figure 4. Let G_ℓ be a framework graph, where the blue and red con-graphs are $(5k, 2)$ -bundles, and the gray con-graphs are $(\ell k, 5)$ -bundles, so $n \in \Theta(\ell k)$. A red-red standard drawing shows $\text{cr}(G_\ell) \in \mathcal{O}(k^2)$. A blue-gray standard drawing shows that G_ℓ is k -gap-planar, cf. Figure 2h.

Let \mathcal{D} be a k -gap-planar drawing of G_ℓ . A red-red crossing covers at most $\frac{1}{(5k)^2} = \frac{1}{25k^2}$ of the \mathcal{K} -subdivisions, but the $2 \cdot 2 \cdot 5k$ red edges can have at most $20k \cdot k = 20k^2$ crossings between each other. As the blue connection is adjacent to the red connections, at least $1 - \frac{20k^2}{25k^2} = \frac{1}{5}$ of the \mathcal{K} -subdivisions have to be resolved by crossings involving a gray edge. Since such a crossing can cover at most $\mathcal{O}(\frac{1}{\ell k \cdot k})$ of the \mathcal{K} -subdivisions, we have $\text{cr}_{k\text{-gap-pl}}(G_\ell) \in \Omega(\ell k^2) = \Omega(nk)$. ◀

A non- k -gap-planar drawing has over k^2 crossings, since there is an edge that has at least $k+1$ crossings mapped to it, and all the corresponding crossing edges have at least k crossings mapped to them. In a k -gap-planar drawing, there can be at most mk crossings as at most k crossings can be mapped to each of the m edges, leading to $\varrho_{k\text{-gap-pl}}^{\text{sparse}}(n) \in \mathcal{O}(\frac{mk}{k^2}) = \mathcal{O}(\frac{n}{k})$ in the context of Observation 1(b); together with Theorem 20 we yield:

► **Corollary 21.** *The k -gap-planar crossing ratio $\varrho_{k\text{-gap-pl}}(n)$ is in $\Theta(\frac{n}{k})$.*

4.9 k -Apex-Planar and Skewness- k : $\varrho_{k\text{-apex}}(n)$ and $\varrho_{\text{skew-}k}(n)$

A drawing is k -apex-planar (*skewness- k*) – with shorthand “ k -apex” (“skew- k ”) – if there are k vertices, called *apex vertices* (edges, called *skewness edges*, respectively), whose removal yields a planar subdrawing. Let $\mathfrak{B}(k)$ be either of the two concepts. To prove crossing ratio upper bounds, [28] furthermore needs to assume that a crossing minimal $\mathfrak{B}(k)$ -drawing is simple (which is, unfortunately, not true in general). We do not want to impose the simplicity requirement, and for our bounds it suffices:

► **Lemma 22.** *Let $\mathfrak{B}(k) \in \{k\text{-apex}, \text{skew-}k\}$. All $\mathfrak{B}(k)$ -graphs allow simple $\mathfrak{B}(k)$ -drawings. Thus, the (non-simple) $\mathfrak{B}(k)$ -crossing number is always upper bounded by a well-defined simple $\mathfrak{B}(k)$ -crossing number.*

Proof. Let R be the edges whose removal leaves a planar subgraph G' (we retain apex vertices, all of which now have degree 0). Draw G' planar with straight lines. Now add all edges R as straight lines. The resulting straight-line drawing (possibly after ε -perturbations to avoid collinearities) is a simple k -apex-planar (or skewness- k) drawing by definition. ◀

► **Theorem 23.** *For every $\ell \geq 1$, there exists a k -apex-planar graph G_ℓ with $n \in \Theta(\ell k)$ vertices such that $\text{cr}_{k\text{-apex}}(G_\ell) \in \Omega(n^2)$ and $\text{cr}(G_\ell) \in \mathcal{O}(k)$. Thus $\varrho_{k\text{-apex}}(n) \in \Omega(\frac{n^2}{k})$.*

Proof. Consider a coloring of the frame F where a blue connection is adjacent to two independent red connections, and all other connections are gray, see Figure 4. Let G_ℓ be a framework graph, where the red con-graphs are single edges, the gray con-graphs are $(\ell k, 2)$ -bundles, but the blue con-graph is more involved: Start with a $(k, 2)$ -bundle B and let A be its k degree-2 vertices; replace every edge of B with a $(\ell, 2)$ -bundle; finally, for each $v \in A$, add four new vertices that form a K_5 together with v . This yields $n \in \Theta(\ell k)$. For standard drawings, we draw the blue con-graph without the K_5 s planarly and then, for each $v \in A$, we draw its K_5 in a small neighborhood of v with one crossing that is adjacent to v . In such a drawing, there are k crossings and we have to remove k vertices (for example A) to obtain a planar subdrawing. A red-red standard drawing shows that $\text{cr}(G_\ell) \leq k+1$. A blue-gray standard drawing with all crossings incident to a vertex of A shows that G_ℓ is k -apex-planar, cf. Figure 2i.

Consider a k -apex-planar drawing \mathcal{D} of G_ℓ . Each of the k disjoint K_5 subgraphs in S_{blue} needs to be covered by a crossing. Thus, all k apex nodes are in S_{blue} and so all of the \mathcal{K} -subdivisions are covered by blue-gray crossings. As each such crossing can cover at most $\frac{1}{(\ell k)^2}$ of the \mathcal{K} -subdivisions, we have $\text{cr}_{k\text{-apex}}(G_\ell) \in \Omega((\ell k)^2) = \Omega(n^2)$. ◀

Any non- k -apex-planar drawing has at least $k + 1$ crossings; by Lemma 22 we have $\varrho_{k\text{-apex}}^{\text{sparse}}(n) \in \mathcal{O}(\frac{m^2}{k}) = \mathcal{O}(\frac{n^2}{k})$ in the context of Observation 1(a). With Theorem 23 we yield:

► **Corollary 24.** *The k -apex-planar crossing ratio $\varrho_{k\text{-apex}}(n)$ is in $\Theta(\frac{n^2}{k})$ and 1-jumping.*

► **Theorem 25.** *For every $\ell > k$, there exists a skewness- k graph G_ℓ with $n \in \Theta(\ell k)$ vertices such that $\text{cr}_{\text{skew-}k}(G_\ell) \in \Omega(nk)$ and $\text{cr}(G_\ell) \in \mathcal{O}(k)$. Thus $\varrho_{\text{skew-}k}(n) \in \Omega(n)$.*

Proof. Consider the same coloring of the frame F as in Theorem 23, see Figure 4. Let G_ℓ be a framework graph, where the red con-graphs are single edges, the gray con-graphs are $(\ell k, 2)$ -bundles, but the blue con-graph is more involved: Start with a $(k, 2)$ -bundle B and let A be its k degree-2 vertices and s and t its poles; for each $v \in A$, add three new vertices W_v that form a K_5 together with v and s , but remove the edge $\{s, v\}$. This yields $n \in \Theta(\ell k)$. Let Q be the ℓk edge-disjoint s - t -paths of gray edges in G_ℓ . We observe k edge-disjoint K_5 subdivisions, one for each v , induced by the vertices $\{s, v\} \cup W_v$ together with the edge $\{v, t\}$ and a path of Q to establish the v - s -Kuratowski path. In standard drawings, we draw the blue con-graph with k crossings, one for each $v \in A$, such that $\{v, t\}$ is crossed by an edge between vertices of W_v . We can remove the k blue edges incident to t to obtain a planar blue subdrawing. A red-red standard drawing shows that $\text{cr}(G_\ell) \leq k + 1$. A blue-gray standard drawing, where each crossing involves a blue edge incident to t , shows that G_ℓ is skewness- k , cf. Figure 2j.

Consider a skewness- k drawing \mathcal{D} of G_ℓ . For each $v \in A$, consider the ℓk K_5 subdivisions that differ only by the chosen path of Q . Since there cannot be ℓk (gray) skewness edges, we have a blue skewness edge per v , which is not in any considered K_5 subdivision for any other vertex in A . Thus, all k skewness edges are blue and all of the \mathcal{K} -subdivisions have to be covered by blue-gray crossings. As each such crossing can cover at most $\frac{1}{\ell k^2}$ of the \mathcal{K} -subdivisions, we have $\text{cr}_{k\text{-apex}}(G_\ell) \in \Omega(\ell k^2) = \Omega(nk)$. ◀

A non-skewness- k drawing has at least $k + 1$ crossings. A simple skewness- k drawing has at most mk crossings, attained when each of the k edges we want to remove crosses all other edges. Via Lemma 22, this leads to $\varrho_{\text{skew-}k}^{\text{sparse}}(n) \in \mathcal{O}(\frac{mk}{k}) = \mathcal{O}(n)$ for Observation 1(b). Thus

► **Corollary 26.** *The skewness- k crossing ratio $\varrho_{\text{skew-}k}(n)$ is in $\Theta(n)$ and 1-jumping.*

4.10 Rectilinear Crossing Ratios: $\bar{\varrho}_{\mathfrak{B}(k)}(n)$

The rectilinear crossing number $\bar{\text{cr}}(G) \geq \text{cr}(G)$ is the minimum number of crossings in a straight-line drawing of G [1]. Analogously, we can define $\bar{\text{cr}}_{\mathfrak{B}(k)}(G) \geq \text{cr}_{\mathfrak{B}(k)}(G)$ as the minimum number of crossings over all straight-line drawings of G subject to the beyond planarity concept $\mathfrak{B}(k)$. We naturally yield the rectilinear crossing ratio $\bar{\varrho}_{\mathfrak{B}(k)}(n)$, defined analogously to $\varrho_{\mathfrak{B}(k)}(n)$ but w.r.t. the rectilinear crossing numbers.

All previous crossing number *lower* bounds still hold in the straight-line setting. Further, we establish the crossing number *upper* bounds via straight-line drawings in nearly all above theorems – only for the fan-planar variants, its K_7 s require non-straight edges. Thus:

► **Corollary 27.** *The rectilinear crossing ratio bounds for k -planar, k -vertex-planar, IC-planar, NIC-planar, NNIC-planar, k -fan-crossing-free, k -gap-planar, k -edge-crossing, skewness- k , and k -apex-planar graphs coincide with those of the above crossing ratios.*

5 Conclusion

We presented a framework to yield (short) proofs for the crossing ratios of many different beyond-planarity concepts, summarized in Table 1. We also included straight-line and non-simple variants of these concepts. Most importantly, and in contrast to most previous approaches, this allows us to attain asymptotically tight bounds. The key idea is to consider graphs arising from a single $K_{3,3}$ by “thickening” edges via subgraphs, and then count how many of the arising $\bar{K}_{3,3}$ -subdivisions can be resolved by certain crossings, yielding a lower bound on the total number of crossings in beyond-planarity-restricted drawings.

There are only three beyond-planarity concepts which we did not discuss although their crossing ratios (on simple drawings) have previously been studied: For *planarly connected* graphs, the tight bound $\Theta(n^2)$ is already known [28]. For *k-quasi-planar* graphs, our Observation 1(a) easily yields the improved upper bound $\varrho_{k\text{-qp}}(n) \in \mathcal{O}(n^2/k^2)$ instead of $\mathcal{O}(f(k) \cdot n^2(\log n)^2)$ [13] for some growing function f . A *(k, l)-grid-free* drawing, $l \leq k$, can have at most $\mathcal{O}(k^{1/l} \cdot m^{2-1/l})$ crossings [22]. Thus, for *(k, l)-grid-free* graphs, our Observation 1(a) yields the upper bound $\varrho_{(k,l)\text{-grid-free}}(n) \in \mathcal{O}(n^{2-1/l}/(lk^{1-1/l})) \subset \mathcal{O}(n^2/l)$ instead of $\mathcal{O}(g(k, l) \cdot n^2)$ [28], where g is exponential in l . The idea of counting crossings that resolve certain percentages of Kuratowski subdivisions seems to allow us to also improve, at least slightly, the known crossing ratio lower bounds for both. However, this cannot easily be done purely within our framework and the lower bounds remain linear in n and therefore far from matching the upper bounds.

Interestingly, for every but one considered beyond-planarity concept $\mathfrak{B}(k)$, we were able to show that the ratio is *1-jumping*, i.e., we can observe the worst-case crossing ratio already for graphs that would attain the optimal crossing number if we were to increase k by just 1. Only for the *k-gap-planar* crossing ratio, our construction does not yield 1-jumping; it is unclear whether this is a limitation of our framework or if the *k-gap-planar* crossing ratio is indeed not 1-jumping.

References

- 1 Bernardo M. Ábrego and Silvia Fernández-Merchant. A lower bound for the rectilinear crossing number. *Graphs Comb.*, 21(3):293–300, September 2005. doi:10.1007/s00373-005-0612-5.
- 2 Miklós Ajtai, Vašek Chvátal, Monroe Monty Newborn, and Endre Szemerédi. Crossing-free subgraphs. In *Theory and Practice of Combinatorics*, volume 60 of *North-Holland Mathematics Studies*, pages 9–12. North-Holland, 1982.
- 3 Michael O. Albertson. Chromatic number, independence ratio, and crossing number. *Ars Math. Contemp.*, 1(1):1–6, 2008. doi:10.26493/1855-3974.10.2d0.
- 4 Christian Bachmaier, Franz J. Brandenburg, Kathrin Hanauer, Daniel Neuwirth, and Josef Reislhuber. Nic-planar graphs. *Discret. Appl. Math.*, 232:23–40, 2017. doi:10.1016/j.dam.2017.08.015.
- 5 Sang Won Bae, Jean-Francois Baffier, Jinhee Chun, Peter Eades, Kord Eickmeyer, Luca Grilli, Seok-Hee Hong, Matias Korman, Fabrizio Montecchiani, Ignaz Rutter, et al. Gap-planar graphs. *Theoretical Computer Science*, 745:36–52, 2018.
- 6 Sang Won Bae, Jean-François Baffier, Jinhee Chun, Peter Eades, Kord Eickmeyer, Luca Grilli, Seok-Hee Hong, Matias Korman, Fabrizio Montecchiani, Ignaz Rutter, and Csaba D. Tóth. Gap-planar graphs. In Fabrizio Frati and Kwan-Liu Ma, editors, *Graph Drawing and Network Visualization - 25th International Symposium, GD 2017, Boston, MA, USA, September 25-27, 2017, Revised Selected Papers*, volume 10692 of *Lecture Notes in Computer Science*, pages 531–545. Springer, 2017. doi:10.1007/978-3-319-73915-1_41.

- 7 Carla Binucci, Emilio Di Giacomo, Walter Didimo, Fabrizio Montecchiani, Maurizio Patrignani, Antonios Symvonis, and Ioannis G. Tollis. Fan-planarity: Properties and complexity. *Theor. Comput. Sci.*, 589:76–86, 2015. doi:10.1016/j.tcs.2015.04.020.
- 8 Franz J. Brandenburg. On fan-crossing graphs. *Theor. Comput. Sci.*, 841:39–49, 2020. doi:10.1016/j.tcs.2020.07.002.
- 9 Otfried Cheong, Henry Förster, Julia Katheder, Maximilian Pfister, and Lena Schlipf. Weakly and strongly fan-planar graphs. In Michael A. Bekos and Markus Chimani, editors, *Graph Drawing and Network Visualization - 31st International Symposium, GD 2023, Isola delle Femmine, Palermo, Italy, September 20-22, 2023, Revised Selected Papers, Part I*, volume 14465 of *Lecture Notes in Computer Science*, pages 53–68. Springer, 2023. doi:10.1007/978-3-031-49272-3_4.
- 10 Otfried Cheong, Sarel Har-Peled, Heuna Kim, and Hyo-Sil Kim. On the number of edges of fan-crossing free graphs. *Algorithmica*, 73(4):673–695, 2015. doi:10.1007/s00453-014-9935-z.
- 11 Markus Chimani, Torben Donzelmann, Nick Kloster, Melissa Koch, Jan-Jakob Völlering, and Mirko H. Wagner. Crossing numbers of beyond planar graphs re-visited: A framework approach, 2024. arXiv:2407.05057.
- 12 Markus Chimani, Philipp Kindermann, Fabrizio Montecchiani, and Pavel Valtr. Crossing numbers of beyond-planar graphs. In Daniel Archambault and Csaba D. Tóth, editors, *Graph Drawing and Network Visualization - 27th International Symposium, GD 2019, Prague, Czech Republic, September 17-20, 2019, Proceedings*, volume 11904 of *Lecture Notes in Computer Science*, pages 78–86. Springer, 2019. doi:10.1007/978-3-030-35802-0_6.
- 13 Markus Chimani, Philipp Kindermann, Fabrizio Montecchiani, and Pavel Valtr. Crossing numbers of beyond-planar graphs. *Theor. Comput. Sci.*, 898:44–49, 2022. doi:10.1016/j.tcs.2021.10.016.
- 14 Walter Didimo, Peter Eades, and Giuseppe Liotta. Drawing graphs with right angle crossings. *Theor. Comput. Sci.*, 412(39):5156–5166, 2011. doi:10.1016/j.tcs.2011.05.025.
- 15 Walter Didimo, Giuseppe Liotta, and Fabrizio Montecchiani. A survey on graph drawing beyond planarity. *ACM Comput. Surv.*, 52(1):4:1–4:37, 2019. doi:10.1145/3301281.
- 16 Graeme Gange, Peter J. Stuckey, and Kim Marriott. Optimal k -level planarization and crossing minimization. In Ulrik Brandes and Sabine Cornelsen, editors, *Graph Drawing - 18th International Symposium, GD 2010, Konstanz, Germany, September 21-24, 2010. Revised Selected Papers*, volume 6502 of *Lecture Notes in Computer Science*, pages 238–249. Springer, 2010. doi:10.1007/978-3-642-18469-7_22.
- 17 Seok-Hee Hong and Takeshi Tokuyama, editors. *Beyond Planar Graphs*. Communications of NII Shonan Meetings. Springer, 2020.
- 18 Kazumasa Ishiguro. The minimum non-crossing edge number of graphs. Talk at 20th Workshop on Topological Graph Theory, Yokohama National University, Japan, 2008.
- 19 Michael Kaufmann and Torsten Ueckerdt. The density of fan-planar graphs. *Electron. J. Comb.*, 29(1):#1.29, 2022. doi:10.37236/10521.
- 20 Boris Klemz, Kristin Knorr, Meghana M. Reddy, and Felix Schröder. Simplifying non-simple fan-planar drawings. In Helen C. Purchase and Ignaz Rutter, editors, *Graph Drawing and Network Visualization - 29th International Symposium, GD 2021, Tübingen, Germany, September 14-17, 2021, Revised Selected Papers*, volume 12868 of *Lecture Notes in Computer Science*, pages 57–71. Springer, 2021. doi:10.1007/978-3-030-92931-2_4.
- 21 Boris Klemz, Kristin Knorr, Meghana M. Reddy, and Felix Schröder. Simplifying non-simple fan-planar drawings. *J. Graph Algorithms Appl.*, 27(2):147–172, 2023. doi:10.7155/jgaa.00618.
- 22 Tamás Kóvári, Vera T. Sós, and Pál Turán. On a problem of K. Zarankiewicz. In *Colloq. Math*, volume 3, pages 50–57, 1954.
- 23 Daniel Král' and Ladislav Stacho. Coloring plane graphs with independent crossings. *J. Graph Theory*, 64(3):184–205, 2010. doi:10.1002/jgt.20448.

- 24 Kazimierz Kuratowski. Sur le probleme des courbes gauches en topologie. *Fundamenta mathematicae*, 15(1):271–283, 1930.
- 25 János Pach and Géza Tóth. Graphs drawn with few crossings per edge. *Comb.*, 17(3):427–439, 1997. doi:10.1007/BF01215922.
- 26 Gerhard Ringel. Ein Sechsfarbenproblem auf der Kugel. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 29(1–2):107–117, 1965.
- 27 Marcus Schaefer. The graph crossing number and its variants: A survey. *The electronic journal of combinatorics*, DS21(Version 7, Apr 8, 2022), 2013–.
- 28 Nathan van Beusekom, Irene Parada, and Bettina Speckmann. Crossing numbers of beyond-planar graphs revisited. *J. Graph Algorithms Appl.*, 26(1):149–170, 2022. doi:10.7155/jgaa.00586.
- 29 Xin Zhang. Drawing complete multipartite graphs on the plane with restrictions on crossings. *Acta Mathematica Sinica, English Series*, 30(12):2045–2053, 2014.
- 30 Xin Zhang and Guizhen Liu. The structure of plane graphs with independent crossings and its applications to coloring problems. *Central European Journal of Mathematics*, 11:308–321, 2013.

Separable Drawings: Extendability and Crossing-Free Hamiltonian Cycles

Oswin Aichholzer  

Institute of Software Technology, Graz University of Technology, Austria

Joachim Orthaber  

Institute of Software Technology, Graz University of Technology, Austria

Birgit Vogtenhuber  

Institute of Software Technology, Graz University of Technology, Austria

Abstract

Generalizing pseudospherical drawings, we introduce a new class of simple drawings, which we call *separable drawings*. In a separable drawing, every edge can be closed to a simple curve that intersects each other edge at most once. Curves of different edges might interact arbitrarily.

Most notably, we show that (1) every separable drawing of any graph on n vertices in the plane can be extended to a simple drawing of the complete graph K_n , (2) every separable drawing of K_n contains a crossing-free Hamiltonian cycle and is plane Hamiltonian connected, and (3) every generalized convex drawing and every 2-page book drawing is separable. Further, the class of separable drawings is a proper superclass of the union of generalized convex and 2-page book drawings. Hence, our results on plane Hamiltonicity extend recent work on generalized convex drawings by Bergold et al. (SoCG 2024).

2012 ACM Subject Classification Mathematics of computing → Combinatorics; Mathematics of computing → Graph theory; Theory of computation → Computational geometry

Keywords and phrases Simple drawings, Pseudospherical drawings, Generalized convex drawings, Plane Hamiltonicity, Extendability of drawings, Recognition of drawing classes

Digital Object Identifier 10.4230/LIPIcs.GD.2024.34

Related Version *Full Version*: <http://arxiv.org/abs/2410.09922>

Funding *Oswin Aichholzer*: Partially supported by the Austrian Science Fund (FWF) grant W1230.

Joachim Orthaber: Supported by the Austrian Science Fund (FWF) grant W1230.

Birgit Vogtenhuber: Partially supported by Austrian Science Fund within the collaborative DACH project *Arrangements and Drawings* as FWF project I 3340-N35.

Acknowledgements We thank Alexandra Weinberger for fruitful discussions during the early stages of our research and we thank the anonymous referees for helpful comments on the paper.

1 Introduction

A *simple drawing* of a graph G is a representation of G in the plane (or on the sphere) such that the vertices of G are mapped to distinct points and the edges of G are mapped to Jordan arcs connecting their respective end-vertices. Furthermore, every pair of edges is allowed to have at most one point in common, which is either a common end-vertex or a proper crossing. Simple drawings of graphs are widely studied combinatorial objects that have received considerable attention in different areas of graph drawing, for example, every crossing-minimizing drawing of a graph is simple.

Several classes of simple drawings have been considered, including *straight-line drawings* (where the edges are straight-line segments), *x -monotone drawings* (where the edges are x -monotone curves), *2-page book drawings*¹ (where all vertices lie on a straight line and the

¹ 2-page book drawings are also known as linear layouts.



edges are drawn as half-circles), *pseudolinear drawings* (for which there exists an arrangement of pseudolines such that every edge lies on one pseudoline), and *pseudocircular drawings* (for which there exists an arrangement of pseudocircles such that every edge lies on one pseudocircle). For details on and relations between these and several more classes (some of which are mentioned later) see for example [3].

A drawing class that was introduced by Arroyo, Richter, and Sunohara [7] and that is of special interest for this work is the class of *pseudospherical drawings*. These are pseudocircular drawings with the additional property that every edge of the drawing intersects every pseudocircle of the underlying arrangement at most once. Stated differently, a pseudospherical drawing \mathcal{D} of a graph G is a simple drawing in which every edge e is contained in a simple closed curve γ_e such that

1. the only two vertices of \mathcal{D} on γ_e are the end-vertices of e ,
2. for any two edges $e \neq f$ the curves γ_e and γ_f intersect in exactly two crossing points, and
3. γ_e intersects every edge $f \neq e$ of \mathcal{D} at most once, either in a crossing or in an end-vertex.

In this work we introduce a new class of simple drawings, which we call *separable drawings*. These are all simple drawings that fulfill Properties 1 and 3 of pseudospherical drawings (but not necessarily Property 2). Separable drawings can also be seen as “locally pseudospherical” because locally for every edge, they look like pseudospherical drawings, but the curves γ_e and γ_f of different edges e and f of \mathcal{D} may interact arbitrarily. This additional freedom gives the advantage that for recognizing separable drawings, it is sufficient to independently find a curve for each edge of the drawing. That is, we do not have to consider the set of potential such curves for all edges simultaneously, which can be relevant from a computational point of view. Moreover, we show that it is a real additional freedom in the sense that the class of separable drawings is strictly larger than the one of pseudospherical drawings.

Note that pseudocircular drawings are all simple drawings that fulfill Properties 1 and 2 of the definition of pseudospherical drawings. Hence the class of pseudospherical drawings is the intersection of the classes of separable and pseudocircular drawings.

Our motivation for studying separable drawings stems from the quest of solving two classic graph drawing questions for simple drawings, namely, the extendability to simple drawings of complete graphs and the existence of crossing-free Hamiltonian cycles in drawings of complete graphs. In this work, we answer both questions for the class of separable drawings and elucidate the relation of separable drawings to further classes of simple drawings.

Edge extension of simple drawings. It is easy to see that every straight-line drawing in the plane on n vertices in general position can be extended to a straight-line drawing of the complete graph K_n . As a consequence of Levi’s Enlargement Lemma [17], an analogous statement is true for pseudolinear drawings. For simple drawings the situation is very different. Kynčl showed that extendability to complete graphs is not always possible [14]. Further, there exist simple drawings of graphs with only a linear number of edges, which cannot be extended by any of the missing edges without violating simplicity [11]. The decision problem of whether a given drawing can be extended by some given edges is NP-complete [4], even for a single edge and if the drawing is pseudocircular [5]. To the positive, the edge extension problem is fixed-parameter tractable (FPT) when parameterized by the number of edges to insert and an upper bound on newly created crossings [9]. The complexity of deciding whether a simple drawing (of some class) can be extended to a simple drawing of the complete graph is still an open problem. Recently, Kynčl and Soukup [16] showed that every x -monotone drawing admits an extension to an x -monotone drawing of the complete graph.

Crossing-free Hamiltonian cycles and paths. It is well known that every straight-line drawing of K_n contains a crossing-free Hamiltonian cycle, and that this property does not hold for straight-line drawings of general graphs (it already breaks for K_n minus one edge). In 1988, Rafla [20] conjectured that the same is true for simple drawings of K_n .

► **Conjecture 1.** *Every simple drawing of K_n with $n \geq 3$ vertices contains a crossing-free Hamiltonian cycle.*

If Conjecture 1 is true, then every simple drawing of K_n also contains at least n crossing-free Hamiltonian paths and 2 crossing-free perfect matchings. Pach, Solymosi, and Tóth [18] made the study of crossing-free subdrawings popular. For simple drawings, a lot of effort went into the search for crossing-free matchings, with the current best lower bound for their size being $\Omega(\sqrt{n})$ [2]. With regard to special drawing classes, the existence of a crossing-free Hamiltonian cycle is an easy exercise for 2-page book drawings and x -monotone drawings. Further, Conjecture 1 was proven to hold for generalized twisted drawings on an odd number of vertices [2]. In a previous work, we also confirmed it for cylindrical drawings as well as strongly c -monotone drawings [3]. In that work, we further stated the following conjecture, which we showed to be a strengthening of Conjecture 1 in the sense of an affirmative answer for all simple drawings of K_n (but not necessarily for a restricted class of simple drawings). Further, we showed that the implication holds for cylindrical and strongly c -monotone drawings and confirmed both conjectures for them.

► **Conjecture 2.** *Every simple drawing \mathcal{D} of K_n on $n \geq 2$ vertices contains, for each pair of vertices $v \neq w$ in \mathcal{D} , a crossing-free Hamiltonian path with end-vertices v and w .*

Very recently, both conjectures have been verified for the large class of g -convex² (short for generalized convex) drawings [8], where the authors also coined the term *plane Hamiltonian connected* for drawings fulfilling Conjecture 2.

A simple drawing \mathcal{D} of K_n is called *g -convex* if every *triangle* in \mathcal{D} has a *convex side*. A *triangle* in \mathcal{D} is the simple closed curve formed by an induced subdrawing on three vertices in \mathcal{D} . Every triangle splits the plane (or sphere) into two connected components, their closures are the *sides* of the triangle. A side S of a triangle is called *convex* if the subdrawing of \mathcal{D} that is induced by all vertices in S is completely contained in S (that is, no edge between two such vertices crosses the triangle).

G -convex drawings have been introduced by Arroyo, McQuillan, Richter, and Salazar [6] as the largest class of a hierarchy of classes of simple drawings of K_n , all of which are combinatorial generalizations of straight-line drawings. Hence the results of [8] on plane Hamiltonicity are quite strong.

Our contribution. In Section 2 we introduce some more notation and show first properties of separable drawings, also explaining why we chose the name “separable”. We further observe that every 2-page book drawing is separable (Observation 6) and show that for simple drawings of K_n being separable is a property of the rotation system (Lemma 5).

In Section 3 we consider the extension problem. We prove that for every graph G on n vertices, every separable drawing of G can be completed to a simple drawing of K_n and that the same holds for crossing-minimizing drawings of G (Theorems 8 and 9). We further discuss that extension to simple drawings is the best we can hope for by presenting an example of a separable drawing that cannot be extended to any separable drawing of K_n (Figure 4).

² G -convex drawings are just called convex drawings in [6, 8]. However, we prefer the term generalized convex or g -convex to avoid confusion, since the term convex drawing classically refers to a straight-line drawing with vertices in convex position.

In Section 4, we turn our attention to the plane Hamiltonicity problem. We show that all separable drawings of K_n fulfill both Conjecture 1 (Theorem 11) and Conjecture 2 (Theorem 10). Further, we prove that separable drawings are a proper superclass of g -convex drawings (Theorem 14). Thus our results on plane Hamiltonicity constitute a strengthening of the according results on g -convex drawings in [8].

Finally, we consider the question of recognizing separable drawings in Section 5. We show that the recognition problem is solvable in polynomial time for simple drawings of K_n (Theorem 15) and NP-complete for simple drawings of general graphs (Theorem 16).

We conclude with some open problems in Section 6.

2 Preliminaries

Before we get to first properties of separable drawings, we introduce some more notation to facilitate argumentation. We call an edge e of a simple drawing \mathcal{D} a *separator edge* if there exists a simple closed curve γ_e containing e such that the only vertices of \mathcal{D} on γ_e are the end-vertices of e and such that, for each edge $f \neq e$ of \mathcal{D} , γ_e has at most one point in common with f . We call the curve γ_e a *witness* for e . With this definition, a simple drawing \mathcal{D} is separable if and only if every edge of \mathcal{D} is a separator edge.

Note that a simple closed curve γ partitions the plane into two connected components. We call the closures of these components the *sides* of γ . To ease reasoning, we sometimes refer to the bounded side of γ in the plane as the *inside* and the other side as the *outside*.

The following lemma motivates why we call separable drawings “separable”.

► **Lemma 3.** *Let γ_e be a witness of a separator edge e in a simple drawing \mathcal{D} . Then every edge f of \mathcal{D} that connects two vertices on the same side of γ_e is fully contained in that side.*

Proof. The statement is clear for e itself. Further, by the definition of a separator edge, each edge $f \neq e$ of \mathcal{D} has at most one point in common with γ_e . Every edge f incident to e already has an end-vertex in common with γ_e and, therefore, is contained in one side of γ_e . Finally, every edge f with both end-vertices on the same side of γ_e and not incident to e crosses γ_e an even number of times. Since f crosses γ_e at most once, it does not cross γ_e at all, which implies that f is contained in one side of γ_e . ◀

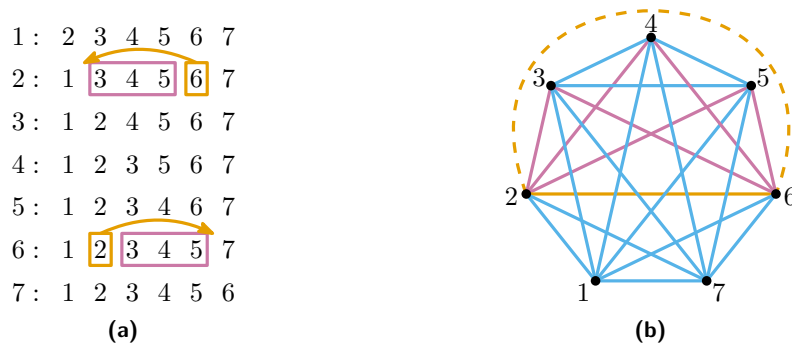
Lemma 3 tells us that, for every edge e in a separable drawing \mathcal{D} , each witness γ_e of e separates \mathcal{D} into two induced subdrawings that together cover all vertices of \mathcal{D} , and that do not interact with each other except for sharing the common edge e . In Lemma 5 (Item 2) we show that, for simple drawings of K_n , the existence of two such induced subdrawings is an equivalent characterization of separability. This implies that, for complete graphs, we do not need to check edges between the two sides of γ_e for multiple intersections with γ_e .

A special case of a separator edge is an uncrossed edge e . Indeed, we can close e to a simple curve γ_e in a small neighborhood of e itself. Then γ_e has one point in common with every edge incident to e and no point in common with any other edge. With respect to the separation into two subdrawings, this means that one of them only consists of the edge e .

► **Observation 4.** *Every uncrossed edge is a separator edge.*

A classic combinatorial abstraction of a simple drawing \mathcal{D} of K_n is its rotation system. The *rotation* of a vertex in a simple drawing is the (clockwise) cyclic order of its incident edges, which is classically given by an accordingly sorted list of its adjacent vertices. The *rotation system* of a simple drawing is the collection of the rotations of all of its vertices. Gioan [10] and Kynčl [13] independently showed that two simple drawings of K_n have the

same set of crossings if and only if they have the same rotation system. An *abstract rotation system* of K_n gives, for each vertex, a (potentially arbitrary) cyclic order of its incident edges. An abstract rotation system of K_n is called *realizable* if there exists a simple drawing of K_n with this rotation system. As shown by Kynčl [15] in combination with computational results from [1], an abstract rotation system is realizable if and only if all its subrotation systems on five vertices are. This implies that deciding whether an abstract rotation system of K_n is realizable can be done in $O(n^5)$ time.



■ **Figure 1** (a) A rotation system corresponding to a convex straight-line drawing of K_7 . The only possible flip of the edge $e = \{2, 6\}$ is marked. (b) As we implicitly show in the proof of Lemma 5, Item 3 \Rightarrow Item 1, a flip of e in the rotation system corresponds to redrawing e (the dashed version is after the flip, the solid version before) in any simple drawing realizing the rotation system.

We next discuss that for a simple drawing \mathcal{D} of K_n , separability only depends on the rotation system of \mathcal{D} . To this end, we first introduce local changes in rotation systems, which we call flips. A *flip* in the rotation system of a simple drawing \mathcal{D} of K_n , see Figure 1 for an example illustration, is the operation of removing an edge $e = \{v, w\}$ in the rotations of its two incident vertices and adding it again in a different position such that

1. in the counter-clockwise rotation of v and the clockwise rotation of w , the sets of vertices between the position of e before and after the operation coincide and are non-empty, and
2. the resulting (abstract) rotation system is realizable.

To relate separator edges to flips in rotation systems (Item 3 of Lemma 5) we will make use of a result by Schaefer [21]. It states that every pair of drawings of K_n minus a non-perfect matching having the same set of crossings can be transformed into each other via triangle mutations (the operation of moving an edge over the crossing between two other edges; see also Figure 2), plus a homeomorphism of the plane.

► **Lemma 5.** *Let \mathcal{D} be a simple drawing of K_n and let $e = \{v, w\}$ be an edge of \mathcal{D} . Then the following are equivalent:*

1. *The edge e is a separator edge.*
2. *The edge e can be closed to a simple curve γ'_e such that every edge f of \mathcal{D} that connects two vertices on the same side of γ'_e is fully contained in that side.*
3. *The edge e is either uncrossed or it can be flipped to an edge e' such that e and e' cross disjoint sets of edges.*

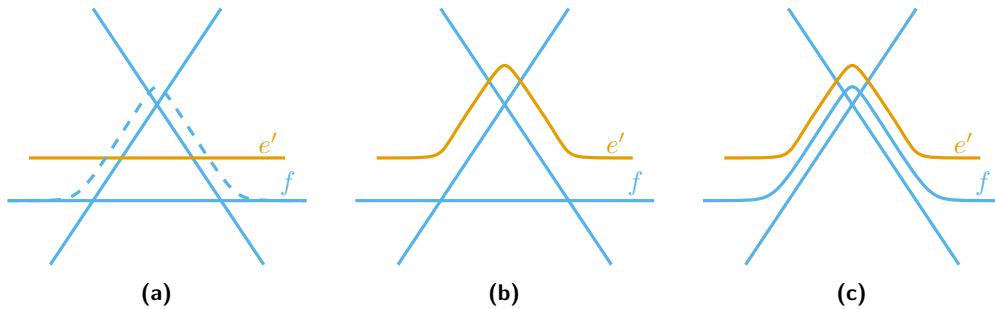
Proof. In the following we show that Item 1 is equivalent to Item 2 and that Item 1 is equivalent to Item 3. The implication Item 1 \Rightarrow Item 2 is given by Lemma 3. Hence, there are three implications left to show.

Item 2 \Rightarrow Item 1 Let \mathcal{D}_1 be the subdrawing of \mathcal{D} induced by all vertices of the inside of γ'_e and similarly \mathcal{D}_2 for all vertices of the outside of γ'_e . Since γ'_e separates \mathcal{D}_1 and \mathcal{D}_2 , any edge in \mathcal{D}_1 or \mathcal{D}_2 has at most one point in common with γ'_e . It remains to consider edges $f = \{v_1, v_2\}$ with $v_1 \in \mathcal{D}_1$ and $v_2 \in \mathcal{D}_2$. If f crosses e , which it can cross at most once, then f lies in the inside of the crossing K_4 on the vertices $\{v, v_1, w, v_2\}$, or, in other words, it is separated from $\gamma'_e \setminus e$ by the 4-cycle v, v_1, w, v_2 . Hence it cannot cross γ'_e a second time. The remaining case is that f crosses $\gamma'_e \setminus e$. Let \mathcal{B} be the boundary of the unbounded cell of \mathcal{D}_1 . We show that f crosses \mathcal{B} exactly once.

Assume for a contradiction that f crosses \mathcal{B} more than once. Let x_1 and x_2 be two consecutive such crossings along f such that the part f' between x_1 and x_2 lies inside \mathcal{B} . Then f' crosses \mathcal{D}_1 and separates the inside of \mathcal{B} into two connected components F_1 and F_2 . Let F_1 be the component that contains e . If no vertex of \mathcal{D}_1 lies in F_2 , then every edge in \mathcal{D}_1 that is crossed by f' would have to be crossed at least twice, a contradiction to \mathcal{D} being simple. Hence, there is some vertex z of \mathcal{D}_1 in F_2 . If v_1 lies in F_1 , then $\{v_1, z\}$ crosses its incident edge f . If v_1 lies in F_2 , then $\{v_1, v\}$ and $\{v_1, w\}$ cross their incident edge f . Since we have a contradiction in both cases, f crosses \mathcal{B} exactly once.

We reroute $\gamma'_e \setminus e$ arbitrarily close to \mathcal{B} along the outside of \mathcal{B} . This does not change any crossings with \mathcal{D}_1 or \mathcal{D}_2 and, by the arguments above, every edge f between the two subdrawings \mathcal{D}_1 or \mathcal{D}_2 is crossed exactly once by the adapted curve γ'_e . Consequently, e is a separator edge.

Item 1 \Rightarrow Item 3 Recall that γ_e has at most one point in common with every edge $f \neq e$ in \mathcal{D} . Hence, replacing e by $e' = \gamma_e \setminus e$ gives a simple drawing \mathcal{D}' . If \mathcal{D} and \mathcal{D}' have the same crossings, then both e and e' are uncrossed. Otherwise the rotation system must have changed and, by Lemma 3, the change is exactly as defined for a flip.



■ **Figure 2** (a) If after a triangle mutation the redrawn edge f (dashed) would cross e' twice, then (b) we first move e' over the respective crossing and then (c) redraw f as planned.

Item 3 \Rightarrow Item 1 If e is uncrossed, then e is a separator edge by Observation 4. So assume that e can be flipped to e' such that no edge is crossed by both e and e' , and let \mathcal{D}' be a simple drawing realizing the rotation system with e' instead of e .

We first show that $\mathcal{D} - e$ and $\mathcal{D}' - e'$ have the same crossing edge pairs. Recall that for a rotation system of K_n , $n \geq 4$, and any pair f, g of independent edges, the subrotation system on the 4-tuple of the four end-vertices of f and g determines whether or not f and g cross. When flipping e to e' , the subrotation system of any 4-tuple that contains at most one end-vertex of e remains unchanged. Hence it remains to consider 4-tuples v, w, v_1, v_2 that involve both end-vertices of e . If in \mathcal{D} , such a 4-tuple forms a crossing that does not involve e , then the order of v_1 and v_2 in the counter-clockwise rotation around v and the clockwise rotation around w (when starting with $\{v, w\}$) is inverse, implying that the subrotation system after the flip is the same as the one before the flip and hence that

the same edge pair also crosses in \mathcal{D}' . If in \mathcal{D} , such a 4-tuple is non-crossing or contains the crossing between e and $\{v_1, v_2\}$, then the order of v_1 and v_2 in the counter-clockwise rotation around v and the clockwise rotation around w (when starting with $\{v, w\}$) is the same. Hence, after the flip, the subrotation system again is either non-crossing or forms the crossing between e' and $\{v_1, v_2\}$. Altogether, this implies that all crossing edge pairs that exist in exactly one of \mathcal{D} and \mathcal{D}' involve e or e' .

Since $\mathcal{D} - e$ and $\mathcal{D}' - e'$ have the same crossing edge pairs, they at most differ in the order of crossings along edges. Hence we can apply Schaefer's generalization of Gioan's theorem [21] to transform $\mathcal{D}' - e'$ to $\mathcal{D} - e$ via triangle mutations, potentially also transforming e' . More exactly, whenever we want to move an edge f over a crossing and e' lies between the edge and the crossing, we first move e' over that crossing and then make the originally planned move with f ; see Figure 2 for an illustration. This process may change the order of crossings along e' , but changes neither the crossing edge pairs nor the rotation at any vertex. Hence, once we have transformed $\mathcal{D}' - e'$ to $\mathcal{D} - e$, we have obtained a transformed edge e' such that $\mathcal{D} - e + e'$ is a realization of the flipped rotation system. Since no edge of $\mathcal{D} - e$ is crossed by both e and e' and no edge incident to e crosses any of e and e' , the union of e and e' fulfills the properties of a witness γ_e , which shows that e is a separator edge. ◀

Note that in the proof of Item 2 \Rightarrow Item 1 the subdrawings \mathcal{D}_1 and \mathcal{D}_2 are interchangeable. That is, we could also reroute γ_e close to the boundary of \mathcal{D}_2 . Furthermore, in the proof of Item 3 \Rightarrow Item 1 the drawing \mathcal{D} can be an arbitrary realization of the respective rotation system. Therefore, every simple drawing that corresponds to a rotation system that has at least one realization as a separable drawing is itself a separable drawing. This makes being separable a property of the rotation system for simple drawings of K_n .

Before we come to the main results of this paper, let us mention 2-page book drawings \mathcal{D} . There the vertices lie on a common line and all edges e are drawn as half-circles. Hence, by closing e to a circle, we get a simple closed curve that has at most one point in common with any edge $f \neq e$ of \mathcal{D} .

► **Observation 6.** *Every 2-page book drawing is separable.*

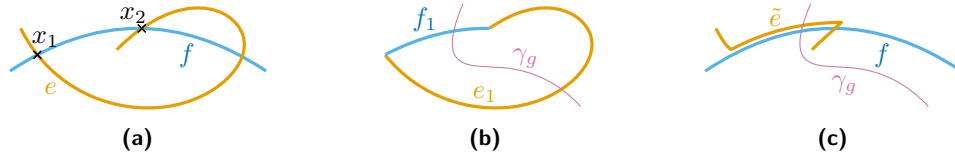
3 Extendability

In the following we prove that every separable drawing \mathcal{D} of a graph G on n vertices can be completed to a simple drawing of K_n . As a first step we show how to add one edge to \mathcal{D} . To do so, we impose a minimality condition regarding the witnesses of all edges in \mathcal{D} . In particular, we call a collection \mathcal{D}° of witnesses, one for every edge in \mathcal{D} , a *witness set* for \mathcal{D} . Further, for an edge $\{u, v\}$ not in G , we call a continuous curve that connects the drawn end-vertices $\mathcal{D}(u)$ and $\mathcal{D}(v)$ in \mathcal{D} a *realization* of $\{u, v\}$ in \mathcal{D} .

► **Lemma 7.** *Let \mathcal{D} be a separable drawing of a non-complete graph G and let \mathcal{D}° be a witness set for \mathcal{D} . For a fixed edge $\{u, v\}$ not in G , let e be a realization of $\{u, v\}$ in \mathcal{D} that, over all possible realizations, minimizes the number of crossings with the witness set \mathcal{D}° . Then the drawing $\mathcal{D}' = \mathcal{D} + e$ is simple.*

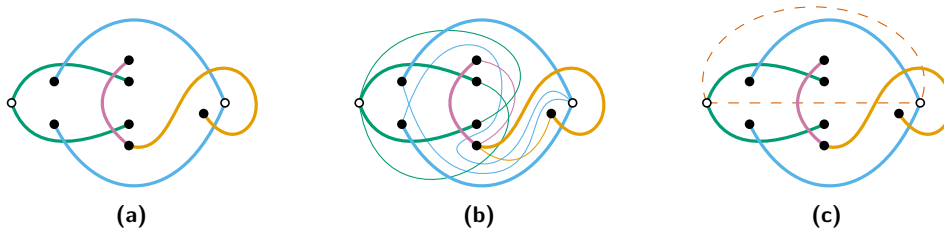
Proof. Let e be as described and assume, to the contrary, that \mathcal{D}' is not simple. The minimality condition implies that e is self-avoiding, hence, the assumption implies that e has more than one point in common with an edge f of \mathcal{D} ; see Figure 3 for an example illustration. Let x_1 and x_2 be two of those common points that are consecutive along f . Then the parts

e_1 and f_1 of e and f , respectively, between x_1 and x_2 each, form a simple closed curve. Since every witness γ_g in \mathcal{D}° for an edge g in \mathcal{D} has at most one point in common with f it follows that, if γ_g crosses f_1 , then γ_g also has to cross e_1 . Therefore, rerouting e along f between x_1 and x_2 reduces the number of crossings of e with \mathcal{D}° by at least one; a contradiction to the minimality condition on e . ◀



■ **Figure 3** (a) The edges e and f have more than one point in common, with x_1 and x_2 being consecutive common points on f . (b) Every witness γ_g that crosses f_1 also has to cross e_1 . (c) The result of rerouting e along f between x_1 and x_2 .

A natural way to get to a simple drawing of K_n would be to iterate the argument of Lemma 7. However, we would need the drawing in each step to be separable, which might not be the case. In particular, Figure 4(a) shows an example of a separable drawing \mathcal{D} on 9 vertices that cannot be completed to a *separable* drawing of K_n . Figure 4(b) shows a witness set for \mathcal{D} , and Figure 4(c) indicates that, with respect to crossings, there are only two different ways to add the edge e between the leftmost and rightmost vertex in \mathcal{D} . Hence the witness of e would have to be the union of these two options. However, both cross the rightmost edge in \mathcal{D} (orange), which is not allowed for a witness.



■ **Figure 4** A separable drawing of a non-complete graph that cannot be extended to any *separable* drawing of a complete graph.

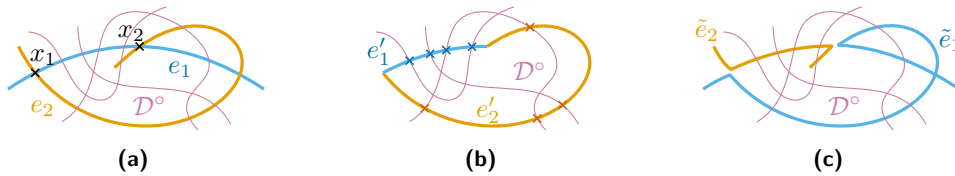
By imposing a second minimality condition, however, we can still extend to a *simple* drawing of K_n .

► **Theorem 8.** *Let \mathcal{D} be a separable drawing of a non-complete graph on n vertices. Then \mathcal{D} can be extended to a simple drawing of K_n .*

Proof. Let \mathcal{D}° be a witness set for \mathcal{D} . We extend \mathcal{D} to a drawing \mathcal{D}' of K_n such that (1) each added edge e creates a minimum number of additional crossings when being added to \mathcal{D}° and such that under this condition (2) \mathcal{D}' has the least total number of crossings. Then, by Lemma 7, $\mathcal{D} + e$ is simple for each of those added edges.

Hence, an obstruction to simplicity can only occur between two added edges e_1 and e_2 in \mathcal{D}' ; see Figure 5 for an example illustration. Let x_1 and x_2 be two consecutive common points on e_1 , and let e'_1 and e'_2 be the respective parts between x_1 and x_2 . By the first minimality condition, e'_1 and e'_2 must have the same number of crossings with \mathcal{D}° , otherwise

we could reroute one of e'_1 or e'_2 along (a part of) the other to get fewer crossings. But then exchanging e'_1 and e'_2 produces a drawing \mathcal{D}' fulfilling the first minimality condition but with fewer crossings than \mathcal{D}' ; a contradiction to the second minimality condition on \mathcal{D}' . ◀



■ **Figure 5** (a) The edges e_1 and e_2 have more than one point in common, with x_1 and x_2 being consecutive common points on e_1 . (b) The parts e'_1 and e'_2 must have the same number of common points (4 each in this example) with the witness set \mathcal{D}° . (c) The result of exchanging e'_1 and e'_2 .

Inspired by the question whether every crossing-minimizing drawing of K_n is pseudospherical, we also investigate the extendability of crossing-minimizing drawings of non-complete graphs. Interestingly, as we detail in the full version of this paper, the proof works rather similar to that for separable drawings, we only need to replace the arguments regarding the witness set with arguments using that the initial drawing is crossing-minimizing.

► **Theorem 9.** *Let \mathcal{D} be a crossing-minimizing drawing of a non-complete graph on n vertices. Then \mathcal{D} can be extended to a simple drawing of K_n .*

4 Crossing-free Hamiltonian cycles and paths

This section is about separable drawings of the complete graph K_n . We first show that they are plane Hamiltonian connected, that is, there exists a crossing-free Hamiltonian path between each pair of vertices, which proves Conjecture 2 for this class.

► **Theorem 10.** *Every separable drawing \mathcal{D} of K_n contains, for each pair of vertices $v \neq w$ in \mathcal{D} , a crossing-free Hamiltonian path with end-vertices v and w .*

Proof. The proof is by induction on n . For $n \leq 2$ the statement is trivially true. For the induction step, let $n \geq 3$, let $v \neq w$ be two arbitrary vertices in \mathcal{D} , and consider some edge $e = \{v, v'\}$ with $v' \neq w$ and witness γ_e . Further, let \mathcal{D}_1 be the subdrawing of \mathcal{D} induced by the set of vertices on the side of γ_e not containing w and let \mathcal{D}_2 be the subdrawing of \mathcal{D} induced by the set of vertices on the other side of γ_e but without vertex v .

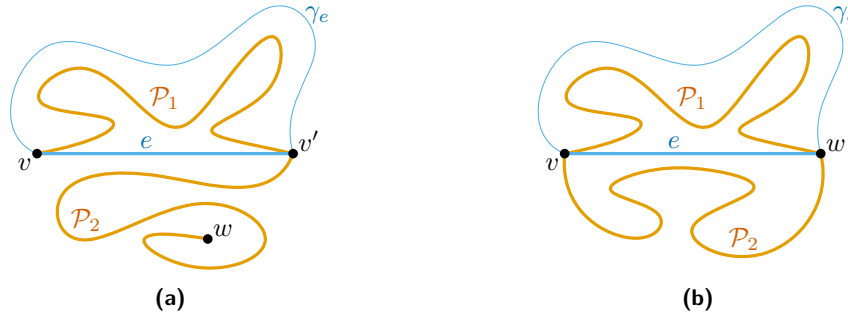
Then \mathcal{D}_1 and \mathcal{D}_2 are both proper subdrawings of \mathcal{D} . Hence, by the induction hypothesis, there exists a crossing-free Hamiltonian path \mathcal{P}_1 in \mathcal{D}_1 with end-vertices v and v' and there exists a crossing-free Hamiltonian path \mathcal{P}_2 in \mathcal{D}_2 with end-vertices v' and w . By Lemma 3, no edge of the path \mathcal{P}_1 crosses any edge of the path \mathcal{P}_2 . Consequently, the union of \mathcal{P}_1 and \mathcal{P}_2 forms a crossing-free Hamiltonian path in \mathcal{D} with end-vertices v and w . ◀

Figure 6(a) gives an illustration of the proof. With a similar approach we obtain that separable drawings of K_n also contain a crossing-free Hamiltonian cycle, by this proving Conjecture 1 for them. Figure 6(b) shows an illustration of how to get the Hamiltonian cycle.

► **Theorem 11.** *Every separable drawing \mathcal{D} of K_n with $n \geq 3$ vertices contains a crossing-free Hamiltonian cycle.*

34:10 Separable Drawings: Extendability and Crossing-Free Hamiltonian Cycles

Proof. Let $e = \{v, w\}$ be an arbitrary edge in \mathcal{D} with witness γ_e and let \mathcal{D}_1 and \mathcal{D}_2 be the subdrawings of \mathcal{D} induced by the vertices on the two sides of γ_e , respectively. By Theorem 10, there exists a crossing-free Hamiltonian path \mathcal{P}_i in \mathcal{D}_i with end-vertices v and w , for $i \in \{1, 2\}$. By Lemma 3, no edge of \mathcal{P}_1 crosses any edge of \mathcal{P}_2 . Hence, the union of \mathcal{P}_1 and \mathcal{P}_2 forms a crossing-free Hamiltonian cycle in \mathcal{D} . ◀



■ **Figure 6** (a) Finding a crossing-free Hamiltonian path between two given vertices and (b) finding a crossing-free Hamiltonian cycle in a separable drawing of K_n .

For the proofs of Theorems 10 and 11 it is actually sufficient that for every pair of vertices v and w , one of them is incident to a separator edge that is not $\{v, w\}$. In particular, this is the case when every vertex is incident to at least 2 separator edges. We call this property *degree-2-separable*. In the proof we further rely on induction. Therefore, we call a class \mathcal{S} of simple drawings *subset-closed* if every subdrawing of a drawing in \mathcal{S} is itself in \mathcal{S} . With this we get the following observation, which might be helpful to show Conjecture 1 for even larger classes of simple drawings.

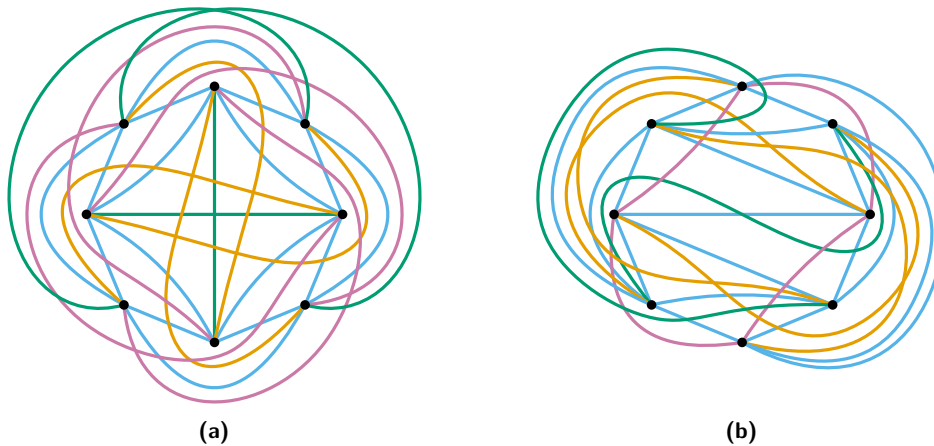
► **Observation 12.** *Let \mathcal{S} be a subset-closed class of simple drawings of complete graphs such that every drawing in \mathcal{S} is degree-2-separable. Then every drawing in \mathcal{S} contains a crossing-free Hamiltonian cycle.*

Let us further mention that a single separator edge is enough to find a crossing-free matching of linear size; let us call this property *1-separable* for a subset-closed class of simple drawings. Indeed, we can add the separator edge e to the matching and then recurse on the subdrawings in the two sides of the witness γ_e . In the worst case, for each edge that we add, we get two subdrawings with only one vertex each that cannot be matched anymore.

► **Observation 13.** *Let \mathcal{S} be a subset-closed class of simple drawings of complete graphs such that every drawing in \mathcal{S} is 1-separable. Then every drawing in \mathcal{S} contains a crossing-free matching of linear size in n .*

Unfortunately there exist simple drawings of K_n without a single separator edge. Figure 7 shows the (up to weak isomorphism³) only two simple drawings of K_8 with this property; the different edge colors are just for better visibility. This result has been obtained by applying the algorithm of Theorem 15 to all different rotation systems of K_8 as given in [1]. Note that Harborth and Mengersen [12] proved that simple drawings of K_n for $n \leq 7$ always have uncrossed edges, and therefore, they have separator edges by Observation 4. Hence, the drawings depicted in Figure 7 are the smallest examples without any separator edge.

³ Two simple drawings of a graph are *weakly isomorphic* if they have the same crossing edge pairs. For simple drawings of K_n , weak isomorphism is equivalent to having the same rotation system [10, 13].



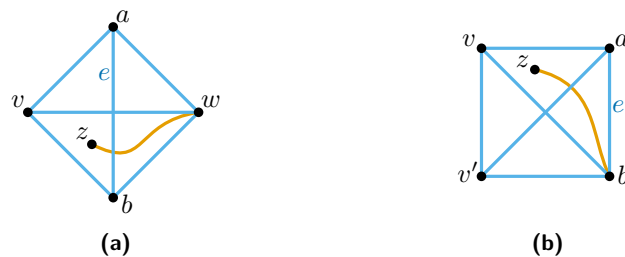
■ **Figure 7** The two simple drawings of K_8 that do not have a single separator edge.

We conclude this section by proving that all g -convex drawings are separable, therefore showing that our results on plane Hamiltonicity improve upon the work of Bergold, Felsner, M. Reddy, Orthaber, and Scheucher [8]. Our proof is inspired by the proof of Arroyo, Richter, and Sunohara [7] that all so-called hereditarily convex drawings (of K_n) are pseudospherical.

► **Theorem 14.** *Every g -convex drawing (of K_n) is separable.*

Proof. We show that every edge $e = \{a, b\}$ in a g -convex drawing \mathcal{D} is a separator edge. If e is uncrossed, then it is a separator edge by Observation 4. Hence, we can assume that e is crossed by at least one edge. In the following we find a simple closed curve γ_e fulfilling Item 2 of Lemma 5, thereby showing that e is a separator edge. In particular, we find vertex sets V_L and V_R that will correspond to the vertices on the two sides of γ_e , respectively.

We fix an orientation of e and say that a vertex v of \mathcal{D} lies on *the left* or on *the right* of e if the convex side of the triangle spanned by e and v lies to the left or right of the oriented edge e , respectively. Recall that both sides of such a triangle can be convex and that the convex side is unique if and only if it is part of a crossing $K_4 = \{a, b, v, w\}$. In this K_4 , e can either be a diagonal or a boundary edge. In the first case v and w lie on different sides of e and in the second case they lie on the same side of e ; see Figure 8 for an illustration.



■ **Figure 8** In a crossing K_4 , e is either (a) a diagonal edge or (b) a boundary edge.

We start with $V_L = V_R = \{a, b\}$. In a first step, we consider crossing K_4 's where e is a diagonal and we add the respective vertices v that are to the left of e to V_L . Since \mathcal{D} is g -convex, the respective vertices w that are to the right of e will never be added to V_L . In a second step, we successively add vertices v to V_L if there exists a crossing $K_4 = \{a, b, v', v\}$

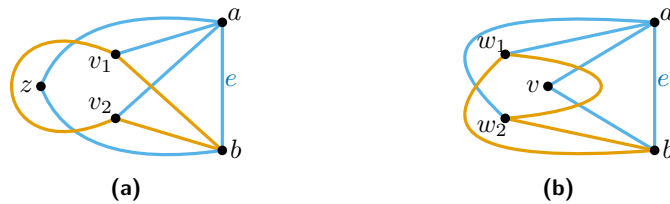
such that e is a boundary edge and v' was already added to V_L before. Once we cannot add anymore vertices to V_L in this manner, we add all remaining vertices to V_R . Note that for all vertices v in V_L the unique convex side of the triangle spanned by v and e is to the left of e .

Let \mathcal{D}_L and \mathcal{D}_R be the subdrawings of \mathcal{D} induced by the vertices V_L and V_R , respectively. Note that e is uncrossed in both those subdrawings. Consider the cell F_∞ in \mathcal{D}_L that is incident to e and to its right. We show that all vertices of V_R lie in F_∞ .

Assume first that a vertex z of V_R lies in the convex side of a triangle spanned by e and a vertex v added to V_L in the first step, that is, there is an edge $\{v, w\}$ that crosses e . Then, by convexity, the edge $\{z, w\}$ must lie in the crossing side of the K_4 spanned by $\{v, a, w, b\}$ and hence $\{z, w\}$ also crosses e ; see Figure 8(a) for an example. But this implies that z lies to the left of e and was added to V_L in the first step; a contradiction to $z \in V_R$.

Assume next that z lies in the unique convex side of a triangle added to V_L in the second step but not in one added in the first step. Then either $\{z, a\}$ or $\{z, b\}$ has to cross a triangle spanned by e and a vertex v' in V_L ; see Figure 8(b) for an example. Therefore, z is added to V_L in the second step; again a contradiction.

Assume last that z lies neither in F_∞ nor in any of the triangles spanned by e and V_L . Then the edges $\{z, a\}$ and $\{z, b\}$ cannot cross any of those triangle edges as argued before. Further, no vertex of V_L can lie in the triangle spanned by z and e to the right of e because we would not have added it to V_L then. Since z does not lie in F_∞ , some part of \mathcal{D}_L separates it from there. Hence, there is an edge $f = \{v_1, v_2\}$ in \mathcal{D}_L crossing $\{z, a\}$ or $\{z, b\}$. Since v_1 and v_2 are on the same side of the triangle $\{z, a, b\}$ and f does not cross e , f crosses both $\{z, a\}$ and $\{z, b\}$; see Figure 9(a) for an illustration. Consequently, the triangles $\{v_1, v_2, a\}$ and $\{v_1, v_2, b\}$ have no convex side; a contradiction to \mathcal{D} being g-convex.



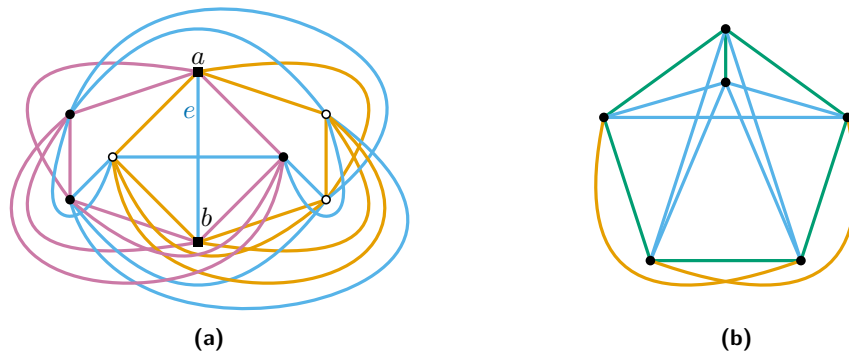
■ **Figure 9** Two situations leading to a triangle (marked orange) not having a convex side.

It remains to show that no edge $f = \{w_1, w_2\}$ of \mathcal{D}_R can cross any edge of \mathcal{D}_L . We first show that f cannot cross any edge incident to e . If f itself is incident to e , this cannot happen similar to before. Otherwise, since both w_1 and w_2 are in F_∞ , f has to pass through a triangle spanned by e and a vertex v of V_L ; see Figure 9(b) for an illustration. This results in the triangles $\{w_1, w_2, a\}$ and $\{w_1, w_2, b\}$ not having a convex side. Consequently, f can only cross some edge in \mathcal{D}_L that is independent to e . But then f has to cross that edge more than once. This is a contradiction in both cases.

Hence, we can close the edge e in F_∞ close to the boundary of \mathcal{D}_L to a simple curve γ_e that fulfills all properties of a witness, that is, e is a separator edge. ◀

Note that we could not just add *all* vertices to the left or right of e to V_L or V_R , respectively. Figure 10(a) shows an example where this would not result in two separated subdrawings.

Further, to see that separable drawings are not only the union of g-convex and 2-page book drawings, for example, consider a straight-line drawing with at least 5 vertices $\{v_1, \dots, v_k\}$ on the convex hull and reroute the edges $\{v_1, v_3\}$ and $\{v_2, v_4\}$ outside of the convex hull. This is always separable, not g-convex, and in most cases also not weakly isomorphic to any 2-page book drawing. Figure 10(b) shows the smallest such example.



■ **Figure 10** (a) A g-convex drawing where a complete left-right splitting via convex sides is not possible. (b) A separable drawing that is neither g-convex nor a 2-page book drawing.

5 Recognition

We have shown in Section 2 that for simple drawings \mathcal{D} of the complete graph K_n being separable is a property of the rotation system (Item 3 of Lemma 5). From this we obtain a polynomial time recognition algorithm for separable drawings (and rotation systems) of K_n .

► **Theorem 15.** *It can be decided in $\mathcal{O}(n^6)$ time whether a given simple drawing \mathcal{D} of K_n is separable.*

Proof. We check, for each edge $e = \{v, w\}$ in \mathcal{D} , whether it is a separator edge. If e is uncrossed, then it is a separator edge by Observation 4 and we are done. Otherwise we use the relation between separator edges and flips in rotation systems given by Item 3 of Lemma 5 to determine all possible flips of e in the rotation system. Recall that by Lemma 5, separability of a simple drawing \mathcal{D} of K_n is a property of the rotation system of \mathcal{D} . Given the drawing \mathcal{D} , its rotation system can be computed in $\mathcal{O}(n^2)$ time in a straight forward way. In the following, we use this rotation system to check separability.

By the definition of a flip of e in the rotation system, the subsets in the counter-clockwise rotation of v and the clockwise rotation of w between the position of e before and after the flip must coincide. We get all such possibilities of potential flips for e in $\mathcal{O}(n)$ time, by going through the rotations of v and w in parallel (starting with $\{v, w\}$) and keeping a parity list of all vertices how often they appeared in the subsets. A counter is used to see how many of the vertices appeared an odd number of times, that is, showed up in only one of the two subsets so far. Every time this counter is zero we have a potential flip.

Checking whether the new rotation system after a potential flip is realizable takes $\mathcal{O}(n^3)$ time because we only need to test all 5-tuples that contain the flipped edge [15]. Further, checking whether the flipped edge has all different crossings from the original edge takes $\mathcal{O}(n^2)$ time by testing for all $\mathcal{O}(n^2)$ new crossings whether they also existed before.

In total there are $\mathcal{O}(n^2)$ many edges e , each of them has $\mathcal{O}(n)$ potential flips, and testing whether such a flip yields a witness for e takes $\mathcal{O}(n^3)$ time as argued. Hence, we can decide in $\mathcal{O}(n^6)$ time whether a simple drawing of K_n is separable. ◀

Unfortunately, the situation is very different for simple drawings of arbitrary graphs. In particular, we construct simple drawings of matchings in the following where it is NP-hard to decide whether they are separable. For this we use a reduction from linked planar 3-SAT with negated edges on one side, which was shown to be NP-hard by Pilz [19, Theorem 10].

34:14 Separable Drawings: Extendability and Crossing-Free Hamiltonian Cycles

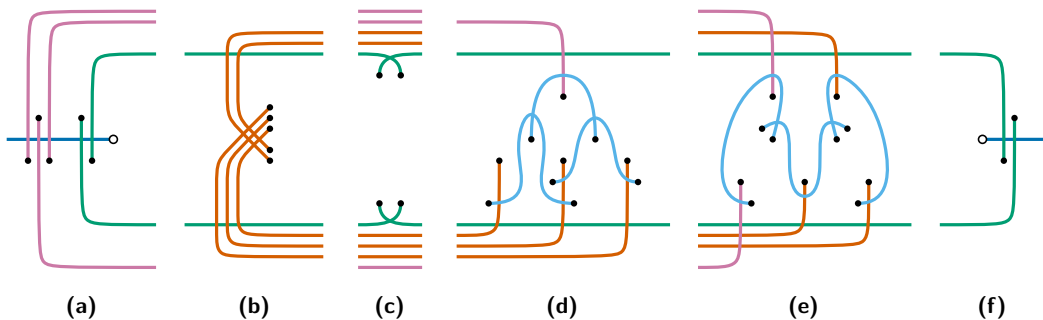
The incidence graph G_ϕ of a 3-SAT formula ϕ has one vertex for each variable and each clause in ϕ and an edge between a variable vertex and a clause vertex if the variable occurs in the clause (as a positive or negative literal). If G_ϕ is a planar graph, then ϕ is a *planar* 3-SAT instance. For *linked* planar 3-SAT there is a Hamiltonian cycle \mathcal{C} that first visits all variable vertices and then all clause vertices such that the union of G_ϕ and \mathcal{C} is still a planar graph. Further, in the restriction “with negated edges on one side”, there exists an embedding of $G_\phi \cup \mathcal{C}$ such that all edges in G_ϕ corresponding to positive literals are drawn inside of \mathcal{C} and all edges corresponding to negative literals are drawn outside of \mathcal{C} .

► **Theorem 16.** *It is NP-complete to decide whether a given simple drawing of an arbitrary graph is separable.*

Proof. Given a 3-SAT formula ϕ that is an instance of linked planar 3-SAT with negated edges on one side, we construct a simple drawing \mathcal{D} containing a special edge e such that e is a separator edge if and only if ϕ is satisfiable. Therefore, it is NP-hard to decide whether e is a separator edge. Moreover, we show that all other edges in \mathcal{D} are definitely separator edges. Consequently, it is NP-hard to decide whether \mathcal{D} is separable. In Figure 11 we illustrate the individual gadgets of the following construction and in Figure 12 we show an example of the complete drawing \mathcal{D} corresponding to a small 3-SAT formula.

Given an embedding of the union of the incidence graph G_ϕ and the Hamiltonian cycle \mathcal{C} , we let e be the part of \mathcal{C} between the clause vertices and the variable vertices. We then add four *boundary* edges, close to \mathcal{C} and on both sides next to the variable and the clause part each, crossing e and crossing each other in the middle; see Figures 11(a), 11(c), and 11(f). Thereby we restrict the potential witness γ_e of e to be drawn within a strip close to \mathcal{C} .

For each edge of G_ϕ we also add an edge to \mathcal{D} . We call these edges *literal* edges. Instead of the variable vertices of G_ϕ we let the incident literal edges in \mathcal{D} cross in a grid such that edges for positive literals are drawn in one direction and those for negative literals in the other direction; see Figure 11(b). This is possible because \mathcal{C} splits those edges into inside and outside, respectively. In that way we force γ_e to either cross all positive or all negative literal edges of the corresponding variable. In the following, crossing the positive side encodes the variable being set to FALSE and vice versa.



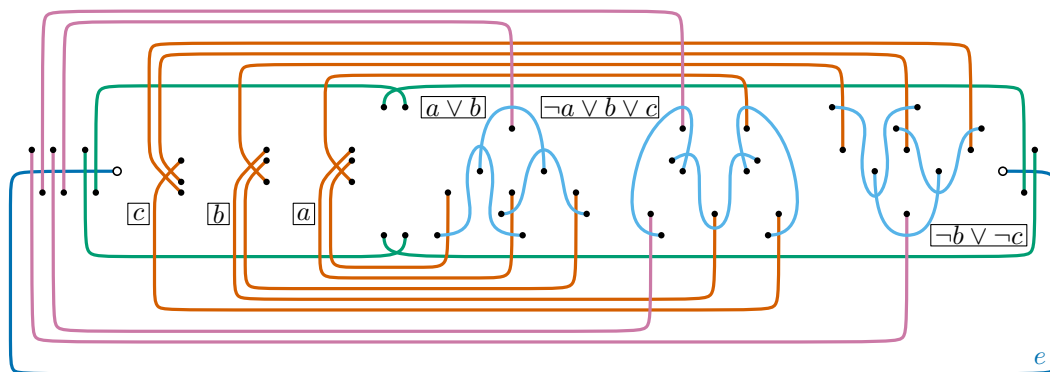
■ **Figure 11** The variable gadget (b) and the two clause gadgets (d) and (e). Boundary edges are drawn green, literal edges darkorange, auxiliary edges purple, and local edges lightblue.

For the clause variables of G_ϕ we construct special clause gadgets depending on how many positive/negative literals are in the clause. We can assume, for simplicity and without loss of generality, that all clauses contain exactly three literals (duplicating one literal if necessary). Hence we have two cases, either all literals are of the same type (negated or not) or two are of one type and one of the other. See Figures 11(d) and 11(e) for the two

constructions of clause gadgets. In addition to the literal edges, we need some *auxiliary* edges that cross e and a boundary edge, and some *local* edges in the gadgets. Since auxiliary edges cross e , they cannot be crossed by γ_e again. Further, literal edges can only be crossed if they were not yet crossed in the variable gadget, that is, if they have the value TRUE. Finally, γ_e can pass through a clause gadget without crossing any local edge twice if and only if it can cross at least one literal edge.

So far we have shown that e is a separator edge if and only if ϕ is satisfiable. It remains to show that all other edges in the construction are separator edges in any case. The local edges can be closed locally within the gadget. The boundary edges can be closed next to the boundary edge on the other side of the strip. Further, we let the auxiliary edges in the construction cross e in reverse order (nested) to how they enter the strip within the boundary edges. Therefore they pairwise do not cross and can be closed outside of the other side of the strip. For the literal edges we go back next to the boundary on the other side of the strip. Since G_ϕ is planar, we can cross all other literal edges except for those corresponding to the same variable. In each clause gadget we can cross the auxiliary edge or at least one of the three literal edges. Between the clause and variable gadgets (Figure 11(c)) we change sides and then cross all literal edges on the same side.

This finishes the proof for NP-hardness. For NP-completeness observe that a witness set for \mathcal{D} can be encoded and checked in polynomial space and time. ◀



■ **Figure 12** The simple drawing \mathcal{D} corresponding to the 3-SAT formula $(a \vee b) \wedge (\neg a \vee b \vee c) \wedge (\neg b \vee \neg c)$ as an instance to decide whether the edge e (darkblue) is a separator edge. The first and third clause use the same gadget, just upside down, and the literals a and $\neg c$, respectively, are duplicated to have exactly 3 literals in all clauses.

6 Future Work

Extending the question whether all crossing-minimizing drawings of K_n are g-convex [6] we ask the following.

► **Question 17.** *Is every crossing-minimizing drawing of an arbitrary graph separable?*

Further, we showed NP-hardness for recognizing separable drawings. The corresponding question for pseudospherical drawings [7] is still open to the best of our knowledge.


► **Question 18.** *Is it NP-hard to decide whether a simple drawing of an arbitrary graph is pseudospherical?*

References

- 1 Bernardo M. Ábrego, Oswin Aichholzer, Silvia Fernández-Merchant, Thomas Hackl, Jürgen Pammer, Alexander Pilz, Pedro Ramos, Gelasio Salazar, and Birgit Vogtenhuber. All good drawings of small complete graphs. In *Proceedings of the 31st European Workshop on Computational Geometry (EuroCG 2015)*, pages 57–60, 2015. URL: <http://eurocg15.fri.uni-lj.si/pub/eurocg15-book-of-abstracts.pdf>.
- 2 Oswin Aichholzer, Alfredo García, Javier Tejel, Birgit Vogtenhuber, and Alexandra Weinberger. Twisted ways to find plane structures in simple drawings of complete graphs. *Discrete & Computational Geometry*, 30:40–66, 2024. doi:10.1007/s00454-023-00610-0.
- 3 Oswin Aichholzer, Joachim Orthaber, and Birgit Vogtenhuber. Towards crossing-free Hamiltonian cycles in simple drawings of complete graphs. *Computing in Geometry and Topology*, 3(2):5:1–5:30, 2024. doi:10.57717/cgt.v3i2.47.
- 4 Alan Arroyo, Martin Derka, and Irene Parada. Extending simple drawings. In *Proceedings of the 27th International Symposium on Graph Drawing and Network Visualization (GD 2019)*, pages 230–243, 2019. doi:10.1007/978-3-030-35802-0_18.
- 5 Alan Arroyo, Fabian Klute, Irene Parada, Raimund Seidel, Birgit Vogtenhuber, and Tilo Wiedera. Inserting one edge into a simple drawing is hard. *Discrete & Computational Geometry*, 69(3):745–770, 2023. doi:10.1007/s00454-022-00394-9.
- 6 Alan Arroyo, Dan McQuillan, R. Bruce Richter, and Gelasio Salazar. Convex drawings of the complete graph: topology meets geometry. *Ars Mathematica Contemporanea*, 22(3):27, 2022. doi:10.26493/1855-3974.2134.ac9.
- 7 Alan Arroyo, R. Bruce Richter, and Matthew Sunohara. Extending drawings of complete graphs into arrangements of pseudocircles. *SIAM Journal on Discrete Mathematics*, 35(2):1050–1076, 2021. doi:10.1137/20M1313234.
- 8 Helena Bergold, Stefan Felsner, Meghana M. Reddy, Joachim Orthaber, and Manfred Scheucher. Plane Hamiltonian cycles in convex drawings. In *Proceedings of the 40th International Symposium on Computational Geometry (SoCG 2024)*, pages 18:1–18:16, 2024. doi:10.4230/LIPIcs.SoCG.2024.18.
- 9 Robert Ganian, Thekla Hamm, Fabian Klute, Irene Parada, and Birgit Vogtenhuber. Crossing-optimal extension of simple drawings. In *Proceedings of the 48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, pages 72:1–72:17, 2021. doi:10.4230/LIPICS.ICALP.2021.72.
- 10 Emeric Gioan. Complete graph drawings up to triangle mutations. *Discrete & Computational Geometry*, 67(4):985–1022, 2022. doi:10.1007/s00454-021-00339-8.
- 11 Péter Hajnal, Alexander Igamberdiev, Günter Rote, and André Schulz. Saturated simple and 2-simple topological graphs with few edges. *Journal of Graph Algorithms and Applications*, 22(1):117–138, 2018. doi:10.7155/jgaa.00460.
- 12 Heiko Harborth and Ingrid Mengersen. Edges without crossings in drawings of complete graphs. *Journal of Combinatorial Theory, Series B*, 17(3):299–311, 1974. doi:10.1016/0095-8956(74)90035-5.
- 13 Jan Kynčl. Simple realizability of complete abstract topological graphs in P. *Discrete & Computational Geometry*, 45(3):383–399, 2011. doi:10.1007/s00454-010-9320-x.
- 14 Jan Kynčl. Improved enumeration of simple topological graphs. *Discrete & Computational Geometry*, 50(3):727–770, 2013. doi:10.1007/s00454-013-9535-8.
- 15 Jan Kynčl. Simple realizability of complete abstract topological graphs simplified. *Discrete & Computational Geometry*, 64(1):1–27, 2020. doi:10.1007/s00454-020-00204-0.
- 16 Jan Kynčl and Jan Soukup. Extending simple monotone drawings. In *Proceedings of the 40th European Workshop on Computational Geometry (EuroCG 2024)*, pages 48:1–48:6, 2024. URL: https://eurocg2024.math.uoi.gr/data/uploads/paper_48.pdf, arXiv:2312.17675.
- 17 Friedrich Levi. Die Teilung der projektiven Ebene durch Gerade oder Pseudogerade. *Berichte über die Verhandlungen der Sächsischen Akademie der Wissenschaften zu Leipzig, Mathematisch-Physische Klasse*, 78:256–267, 1926. In German.

- 18 János Pach, József Solymosi, and Géza Tóth. Unavoidable configurations in complete topological graphs. *Discrete & Computational Geometry*, 30(2):311–320, 2003. doi:10.1007/s00454-003-0012-9.
- 19 Alexander Pilz. Planar 3-SAT with a clause/variable cycle. *Discrete Mathematics & Theoretical Computer Science*, 21(3):18:1–18:20, 2019. doi:10.23638/DMTCS-21-3-18.
- 20 Nabil H. Rafla. *The Good Drawings D_n of the Complete Graph K_n* . PhD thesis, McGill University, Montreal, 1988. URL: <https://escholarship.mcgill.ca/concern/theses/x346d4920>.
- 21 Marcus Schaefer. Taking a detour; or, Gioan’s theorem, and pseudolinear drawings of complete graphs. *Discrete & Computational Geometry*, 66(1):12–31, 2021. doi:10.1007/s00454-021-00296-2.

On k -Plane Insertion into Plane Drawings

Julia Katheder ✉ 

Universität Tübingen, Germany

Philipp Kindermann ✉ 

Universität Trier, Germany

Fabian Klute ✉ 

Universitat Politècnica de Catalunya, Barcelona, Spain

Irene Parada ✉ 

Department of Mathematics, Universitat Politècnica de Catalunya, Barcelona, Spain

Ignaz Rutter ✉ 

Universität Passau, Germany

Abstract

We introduce the k -Plane Insertion into Plane drawing (k -PIP) problem: given a plane drawing of a planar graph G and a set F of edges, insert the edges in F into the drawing such that the resulting drawing is k -plane. In this paper, we show that the problem is NP-complete for every $k \geq 1$, even when G is biconnected and the set F of edges forms a matching or a path. On the positive side, we present a linear-time algorithm for the case that $k = 1$ and G is a triangulation.

2012 ACM Subject Classification Mathematics of computing → Combinatoric problems

Keywords and phrases Graph drawing, edge insertion, k -planarity

Digital Object Identifier 10.4230/LIPIcs.GD.2024.35

Related Version *Full Version:* <https://doi.org/10.48550/arXiv.2402.14552> [19]

Funding *Julia Katheder:* Funded by the Deutsche Forschungsgemeinschaft (DFG) – 364468267.

Fabian Klute: F. K. is supported by a “María Zambrano grant for attracting international talent” and by grant PID2019-104129GB-I00 funded by MICIU/AEI/10.13039/501100011033.

Irene Parada: I. P. is a Serra Hünter Fellow. Partially supported by grant 2021UPC-MS-67392 funded by the Spanish Ministry of Universities and the European Union (NextGenerationEU) and by grant PID2019-104129GB-I00 funded by MICIU/AEI/10.13039/501100011033.

Ignaz Rutter: Funded by the Deutsche Forschungsgemeinschaft (DFG) – 541433306.

1 Introduction

Inserting edges into planar graphs is a long-studied problem in graph drawing. Most commonly, the goal is to find a way to insert the edges while minimizing the number of crossings and maintaining the planarity of the prescribed subgraph. This problem is a core step in the planarization method to find graph drawings with few crossings [23]. Gutwenger et al. [16] have studied the case of a single edge. For multiple edges, the picture is more complicated. In case the edges are all incident to one vertex previously not present in the graph, the problem can be solved in polynomial time [8]. However, the general problem is NP-hard even when the given drawing is fixed and the underlying graph is biconnected [24, 26]. Assuming a fixed drawing, Hamm and Hliněný presented an FPT-algorithm parameterized by the number of crossings [17]. Finally, Chimani and Hliněný [9] gave an FPT-algorithm for the fixed and variable embedding settings with the number of inserted edges as a parameter.

In this paper, we take a slightly different viewpoint and do not restrict the overall number of created crossings, but instead their structure. Moreover, we focus on the case when the drawing of the given planar graph G is fixed. Then our goal is, given a plane drawing Γ



© Julia Katheder, Philipp Kindermann, Fabian Klute, Irene Parada, and Ignaz Rutter;
licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 35; pp. 35:1–35:11

Leibniz International Proceedings in Informatics



LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

of G and a set F of edges not present in G , to find a k -plane drawing containing Γ as a subdrawing plus the edges of F . Here, a k -plane drawing of a graph is one in which no edge is crossed more than k times. The class of k -planar graphs, which are those admitting a k -plane drawing, is widely studied in graph drawing [11, 18].

► **Problem 1** (k -Plane Insertion into Plane drawing (k -PIP)). *Let Γ be a plane drawing of a graph $G = (V, E)$ and let $H = (V, E')$ be its complement. Given G, Γ and a set $F \subseteq E'$ of edges, find a k -plane drawing of the graph $(V, E \cup F)$ that contains Γ as a subdrawing.*

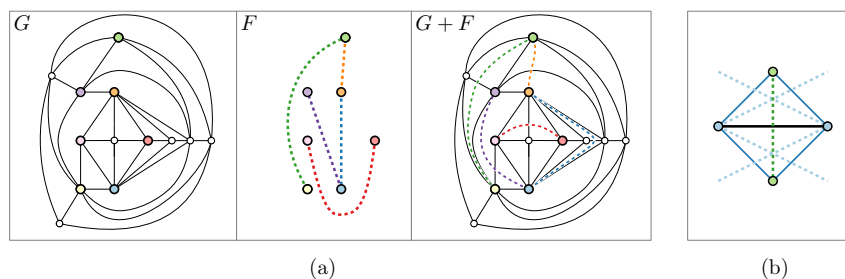
For any fixed $k \in \mathbb{N}$, an *instance* (G, Γ, F) of k -PIP consists of a graph G , a plane drawing Γ of G , and a set of edges F from the complement of G .

Our contribution. In addition to introducing this problem, we give two results. In Section 2, we present an $O(|V|)$ algorithm for 1-PIP for the case that G is a triangulation. To accomplish this, we first reduce the number of possible ways one edge can be inserted into the given drawing to at most two per edge in F and then use a 2-SAT formulation to compute a solution if possible. In Section 3, we show that k -PIP is NP-complete for every $k \geq 1$ even if G is biconnected and the edges in F form a path or a matching.

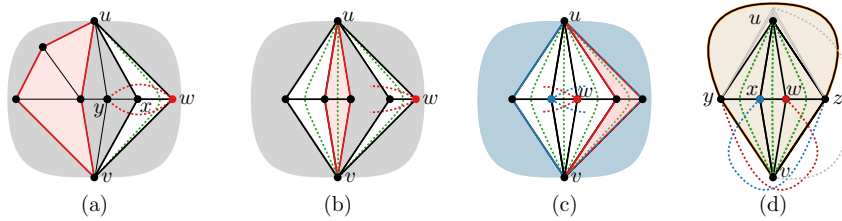
Related work. k -PIP is related to the problem of extending a partial drawing of a graph to a drawing of the full graph. Usually, the goal in such problems is to maintain certain properties of the given drawing. For example, in works by Angelini et al. [1], Eiben et al. [13, 14], Ganian et al. [15], or Arroyo et al. [3, 4] the input is a plane, 1-plane, k -plane, or simple drawing, respectively, and the desired extension must maintain the property of being plane, 1-plane, k -plane, or simple. Restrictions of the drawing such as it being straight-line [25], level-planar [5], upward [21], or orthogonal [2] have been explored. Other results consider the number of bends [7] or assume that the partially drawn subgraph is a cycle [6, 22].

2 1-PIP: Efficiently inserting edges into a triangulation

We assume standard notation and concepts from graph theory; compare, e.g., [12]. Given an instance (G, Γ, F) of 1-PIP, often times an edge $e \in F$ can be inserted into Γ in different ways. Note that e cannot be inserted without crossings in a triangulation. An *option* for e is an edge γ of G such that e can be inserted into Γ crossing only γ . Note that in a triangulation, a pair of adjacent faces uniquely defines an edge γ that must be crossed if e is inserted into said pair of faces. Thus, we also use the term option to refer to such a pair of



■ **Figure 1** (a) The 1-PIP problem: a plane graph G , a set F of edges, and a 1-plane drawing of $G + F$. (b) In a triangulation, an edge in G (bold) can only be an option for a single edge in F (green) and clashes with at most four other options (blue).



■ **Figure 2** Cases with three or more options in a triangulation.

faces. An option for e is *safe* if, in case the instance admits a solution, there is a solution in which e is inserted according to this option. Two options for two edges e and e' of F *clash* if inserting both e and e' according to these options violates 1-planarity. Examples of safe options are those of edges with a single option and an option without clashes. An immediate solution can be found if each edge in F has a non-clashing option. However, it is not sufficient for each edge in F to have a safe option in order to find a solution, e.g., in the case that two single options are clashing. Observe that in a triangulation, each edge of Γ can only be an option for one edge of F and clashes with at most four other options; see Figure 1(b). Further, for a triangulation, we have the following property where a *blocking cycle* in the drawing forces an edge to have only clashing options; see Figure 2(a).

► **Property 2.** Let $e = (u, v)$ be an edge in F and let $\sigma_i = (x, w)$ be one of its options. For an edge $e' = (w, y) \in F$ having at least one clashing option with σ_i , there is other non-clashing option with σ_i , if there is a cycle C in G such that $u, v \in C$ and $x, w, y \notin C$.

Proof. The cycle $C_{\sigma_i} = (u, x, v, w, u)$ and C share only the vertices u and v and since Γ is plane, C and C_{σ_i} partition Γ into four regions, where the edges in C and C_{σ_i} constitute the borders of said regions. The edge e' has an option that clashes with σ_i , i.e., this option is an edge in C_{σ_i} . Then the endpoint y of e' lies in the region bordered by edges in C and C_{σ_i} . By 1-planarity, e' cannot have an option not clashing with σ_i , as this would require crossing C twice. ◀

► **Theorem 3.** 1-PIP can be solved in linear time for instances (G, Γ, F) where G is a triangulation.

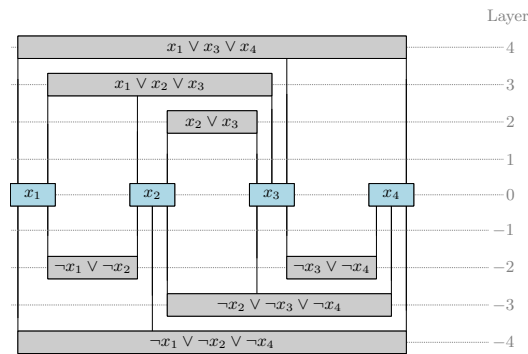
Proof. The idea is to preprocess the instance until we are left with a set $F' \subseteq F$ of edges with two options each. The resulting instance can then be solved using a 2SAT formula. We begin by computing all options for every $e \in F$, resulting in $O(|V|)$ options, since each option is an edge in the plane drawing Γ , crossed by a unique edge in F . Since Γ is plane, we can get the triangles incident to each $v \in V$ in cyclic order and also the options for edges in F incident to v . Hence, we get the overall $O(|V|)$ options for edges in F in $O(|V|)$ time. For an edge $(u, v) \in F$, $u, v \in V$, with two or more options we say that two options are *consecutive* if the corresponding faces are consecutive in the cyclic order around u (or v); see the options for (u, v) in Figure 2(c) for an illustration. We say a set of options is *cyclically consecutive* if the corresponding edges induce a cycle in G ; see the options for (u, v) in Figure 2(d). Whenever an edge e has no option left, we stop and output **no** and if e has exactly one option left, we insert it into Γ . Every time we insert an edge, we need to remove at most four options of other edges plus all the options of the just inserted edge. Consider an edge $e = (u, v) \in F$, $u, v \in V$, that has three or more options. We consider three cases.

- (a) There are at least three options for e , and at least one of them, σ_i , is not consecutive to any of the other two; see rightmost option in Figure 2(b). We claim that σ_i is either safe or never possible in a solution. If σ_i is not clashing with any other option, it is safe and we add it. Otherwise, let w and x be the two endpoints of σ_i . Option σ_i can only be clashing with two options for edges in F incident to w and two options for edges in F incident to x . Moreover, any option for those edges clashes with σ_i . To see this, consider the cycle C formed by u, v , and the endpoints of another option for e other than σ_i (illustrated in red in Figure 2(b)). By σ_i being a non-consecutive option, C fulfills Property 2 for clashing options of edges in F incident to x and w .
- (b) There are at least four consecutive non-cyclic options for e ; see Figure 2(c). Let σ_i be one of the inner options. Then, similar to the previous case, we can find a blocking cycle as follows. If the option clashes with the rightmost (leftmost) option, we can find a blocking cycle formed by u, v and the endpoints of the leftmost (rightmost) option. Otherwise, the cycle formed by u, v and the first and last endvertex in the path formed by the consecutive options of e forms a cycle fulfilling Property 2 for the endvertices of σ_i .
- (c) There are three consecutive or four cyclically consecutive options for e ; see Figure 2(d). Consider the middle option σ_i (or any option if there were four). If it is safe, we just add it. Else, let w and x be the endpoints of σ_i and y, z the other endpoints of options for e . Assume, w.l.o.g., that σ_i clashes with an option of an edge e_w incident to w and to vertex y . For σ_i to be a possible option in a solution, e_w must have an option that does not clash with it. There is only one possibility, and it implies that v, y, z or u, y, z form a triangle. Assume, w.l.o.g., the former, so (y, z) is an edge in Γ . Let V_\diamond be the set of vertices $\{u, v, w, x, y, z\}$ and G_\diamond the octahedron subgraph of G induced by V_\diamond . Edges in F with exactly one endpoint in $V_\diamond \setminus \{u\}$ have at most one option. Thus, we can insert them first and see whether we are still in Case (d). Edges incident to u and to a vertex not in V_\diamond cannot clash with any option of an edge between vertices in V_\diamond . Thus, we can solve the constant-size subinstance consisting of inserting such edges into G_\diamond independently, taking into account the single-option edges that we might have inserted.

Once each edge has exactly two options we create a 2SAT formula containing one variable per option and clauses that ensure exactly one option per edge in F' and exclude clashes. This formula has size $O(|V|)$ and is satisfiable iff the original instance has a solution. ◀

3 $k_{\geq 1}$ -PIP: Inserting a path or a matching is NP-complete

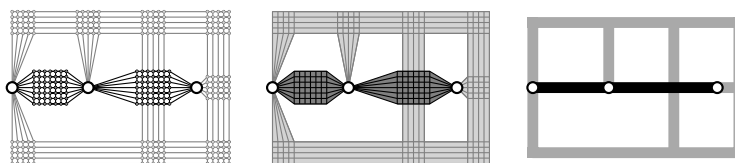
The membership of $k_{\geq 1}$ -PIP in NP is straightforward; we prove NP-hardness by reduction from PLANAR MONOTONE 3-SAT. Let ϕ be a Boolean formula in CNF with variables $V = \{x_1, \dots, x_n\}$ and clauses $C = \{c_1, \dots, c_m\}$. Each clause has at most three literals and is either *positive* (all literals are positive) or *negative* (all literals are negative). Furthermore, there is a rectilinear representation Γ_ϕ of the variable-clause incidence graph of ϕ such that all variables and clauses are depicted as axis-aligned rectangles or *bars* connected via vertical segments and all variables are positioned on the x -axis, all positive clauses lie above, and all negative clauses lie below the x -axis; see Figure 3 for an example. This problem is known to be NP-complete [10, 20]. The bars in Γ_ϕ can be layered decreasingly from top to bottom. We set the layer of the variables as layer zero, and insert two empty layers, one directly above and one below the variable layer. We denote by $L(c)$ the layer of clause c ; see Figure 3.



■ **Figure 3** Rectilinear representation of the variable-clause incidence graph of a PLANAR MONOTONE 3-SAT instance.

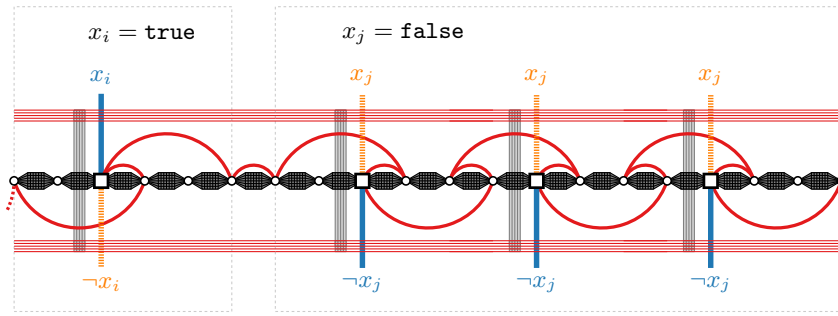
In the following, starting from Γ_ϕ , we construct a graph $G = (V, E)$, its plane drawing Γ , and the edge set F , which will be inserted into Γ in a specific way. We start with the case of F forming a path (see Theorem 7) and describe the changes to our construction for F being a matching afterwards (see Corollary 8).

We denote by H^+ the graph consisting of an axis-aligned $(k + 1) \times (k + 1)$ -vertex grid, where all the $k + 1$ vertices on the left side of the grid are connected to a vertex u , while all vertices on the right are connected to a vertex v . We create chains of copies of H^+ , that are connected via the vertices u, v . Further, we denote by H^- the axis-aligned grid graph consisting of $(k - 1) \times (k - 1)$ vertices. In our construction of G , we create grids of H^- graphs, by connecting two opposing vertical or horizontal sides of their respective vertex grid via $k - 1$ non-crossing edges. The grid construction can also be connected to copies of H^+ via $k - 1$ non-crossing edges, leaving out the corner vertices of the H^+ vertex grid. If it is necessary to connect a single vertex v to an H^- , we connect v via a fan of $k - 1$ edges to one side of the vertex grid. Note that for the case of $k = 1$, structures parameterized by $k - 1$ such as H^- are meant to disappear from the construction. Figure 4(a) shows a structure consisting of multiple copies of H^+ and H^- and their schematic representation used in more complex figures. We say that an edge $e \in F$ is ℓ -spanning if there are ℓ different copies of H^+ in the chain between its endpoints.

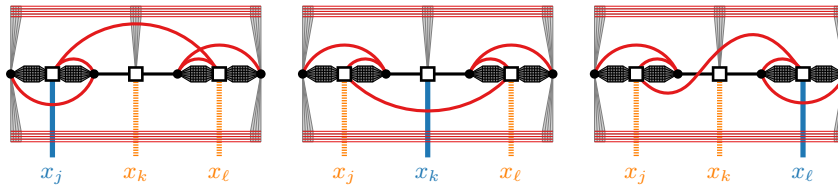


■ **Figure 4** Different representations used in the drawings of our construction. (Left) every vertex and every edge, (middle) a simplification, and (right) a highly abstracted representation.

The variable gadget. We replace each bar of a variable x in Γ_ϕ by a variable gadget which consists of an H^+ -chain of $4a + 1$ copies, where a is the maximum over the number of positive and negative occurrences of x in ϕ . Let u_1, \dots, u_{4a+2} be the vertices that join the copies of H^+ as well as the two unjoined vertices of the first and last H^+ copy in the chain, from left to right. Moreover, we mark for $i \in \{0, \dots, a - 1\}$ the vertices u_{4i+3} as *variable endpoints* (squares in Figure 5). Each such vertex is incident to two *literal edges*, which are connecting the variable gadgets to adjacent layers and encode the truth value of the respective variables. We call a literal edge exiting its variable endpoint upwards (downwards) *positive* (*negative*).



■ **Figure 5** Drawing of the variable gadget illustrating Lemma 4.



■ **Figure 6** Drawing of the clause gadget illustrating Lemma 5.

For every copy of H^+ with position $4i + 2$, $0 \leq i < a$, we connect the top and the bottom side of its vertex grid to one copy of H^- each. For each side – top and bottom – of the H^+ -chain, the copies of H^- will be connected by $k - 1$ non-crossing edges in F as schematically shown in Figure 5. For an illustration showing all vertices and edges, see Figure 4(a).

For each variable gadget, its edges in F (bold red in Figure 5) then consist of alternating 3- and 1-spanning edges. Formally, for each $i \in \{0, \dots, a - 1\}$, the path F passes through the vertices $u_{4i+1}, u_{4i+4}, u_{4i+3}, u_{4i+6}, u_{4i+5}$, except for $a - 1$, where we omit the last vertex.

For the remainder, we depict literal edges representing the value **true** in blue and the ones representing **false** in orange, while the edges in F are colored in red; c.f. Figures 5, 6, 8, and 10. The proofs of statements marked with a (\star) are available in [19].

► **Lemma 4** (\star) . *Let v be a variable gadget described as above. Then, in any k -planar drawing containing v , its literal edges, and the edges $F_v \subseteq F$ incident to vertices in v , either all negative or all positive literal edges are crossed.*

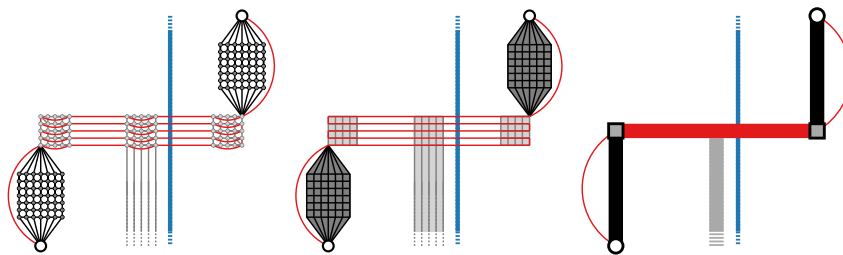
We think of the variable corresponding to the gadget as set to **true** if the negative literal edges are crossed, and to **false** otherwise. We connect the variable gadgets by adding one copy of H^+ with a 1-spanning edge added to F in between them; see Figure 5.

The clause gadget. We describe the construction only for the positive clauses; it works symmetrically for the negative ones. The clause gadget is depicted in Figure 6. It consists of a chain of two copies of H^+ , followed by two edges, followed by two more copies of H^+ . We mark the middle vertices of each of the two copies and the two edges as *variable endpoints* and add one additional edge to them, their *literal edge*. Assume that all literal edges are drawn on the same side as shown in Figure 6 and add edges to F as shown in red. Further, we add three copies of H^- on the top and two on the bottom side of the gadget, and connect the left- and rightmost vertex in the gadget as well as the middle variable endpoint to the corresponding ones. Similar to the variable gadget, these H^- copies will be connected via edges in F as shown in Figure 6.

► **Lemma 5** (★). *Let c be a clause gadget drawn as described above. Then, in any k -planar drawing containing c , its literal edges, and the edges $F_c \subseteq F$ incident to vertices of c , at least one literal edge has to be crossed by an edge in F_c .*

Propagating the variable state. Again, we only describe the construction for layers > 0 as the other side is symmetric. We insert H^+ -chains with 1-spanning edges added to F on every layer > 0 of Γ_ϕ and insert the clause gadgets into the respective layers as shown in Figure 8. Further, we create variable endpoints on all layers > 0 in order to propagate the state of the variable gadgets to clauses in higher layers. Layer 1 thereby ensures that each variable endpoint can be connected to another endpoint, even if the respective literal edge is not used in a clause, as this is crucial to ensure the alternating pattern in the variable gadgets; see, e.g., x_1 in Figure 8. For each pair of corresponding variable endpoints of a variable gadget and clause gadget, we create a variable endpoint at a merged vertex in the H^+ -chain in each layer i with $0 < i < L(c)$ and insert *propagating edges*, by prescribing F to span the two neighboring copies of H^+ . Further, we connect every two consecutive variable endpoints on layer j and $j + 1$ with $1 \leq j < L(c)$ via a literal edge, as illustrated in Figure 8.

Both the variable and the clause gadget require each literal edge to have either $k - 1$ or k crossings. Since the $k_{\geq 1}$ -PIP problem requires Γ to be plane, it is not possible in our construction to create these crossings with edges in G , hence the path formed by edges F has to cross each literal edge $k - 1$ times, in addition to one potential crossing by the propagation edges. To this end, we create a subpath P_i comprised of edges in F between each layer i and layer $i - 1$ (if present), which is passing through copies of H^- and crosses the literal edges $k - 1$ times; see Figure 7 for different levels of abstraction used in our illustrations. The subpath of F on every layer i is joined to P_i and P_{i+1} (if present) by an H^+ -chain and 1-spanning edges. For $k = 1$ we simply connect the subpaths of F on each layer to the next by a H^+ -chain and 1-spanning edges. Note that if k is even, the sides where the H^+ -chain is located alternate, otherwise they connect on the same side of the drawing. Note that in our illustrations showing final the constructions for the graph given in Figure 3, we assume an even k . To ensure that the edges in each P_i do not exceed k crossings, we subdivide between each literal edge by inserting vertically connected copies of H^- as depicted in Figure 8.

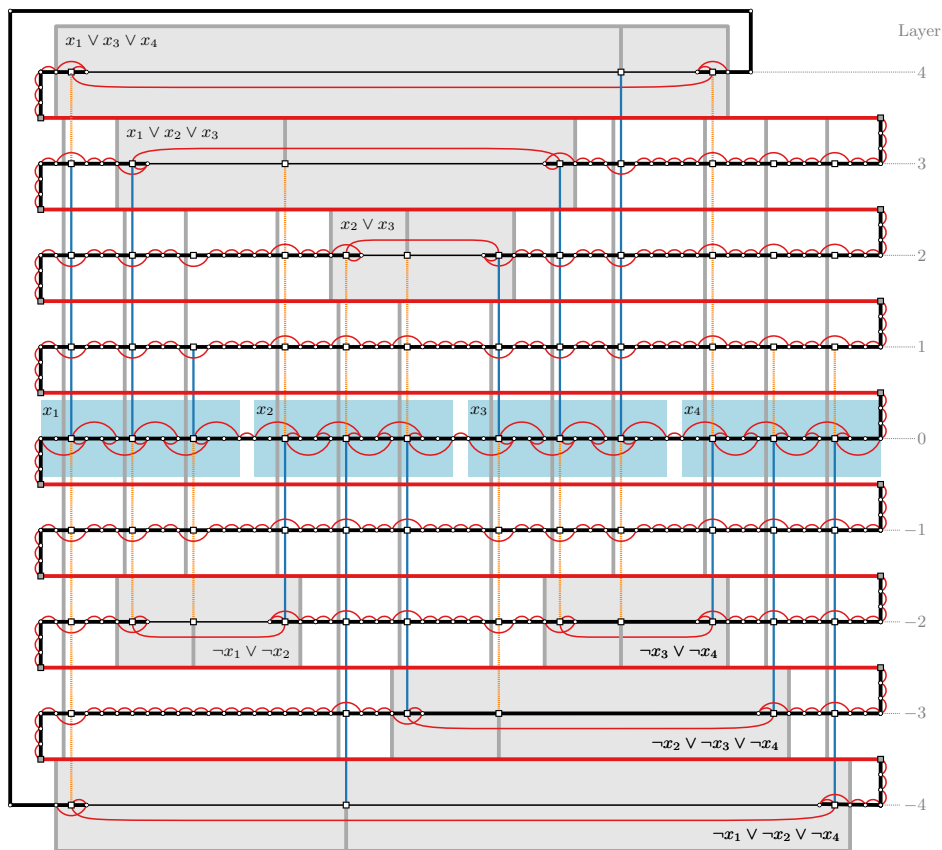


■ **Figure 7** Different representations of the alternating path P_i between layer i and layer $i - 1$.

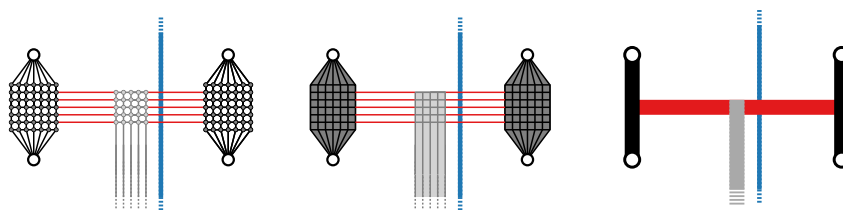
► **Lemma 6** (★). *Let $P = e_1, \dots, e_{L(c)}$ be a path of literal edges such that e_1 is incident to a variable endpoint of variable gadget v and $e_{L(c)}$ to one clause gadget c , if e_1 is crossed in v , then $e_{L(c)}$ is crossed by an edge of F incident to vertices on layer $L(c) - 1$.*

Note that the first edge of P being uncrossed in the variable gadget does not necessarily lead to its last edge being uncrossed in layer $L(c) - 1$. In fact, this is possible when multiple literals evaluate to **true** for a clause gadget; e.g., the top-left orange edge in Figure 8.

► **Theorem 7** (\star). k -PIP is NP-complete for every $k \geq 1$, even if G is biconnected and F forms a path.



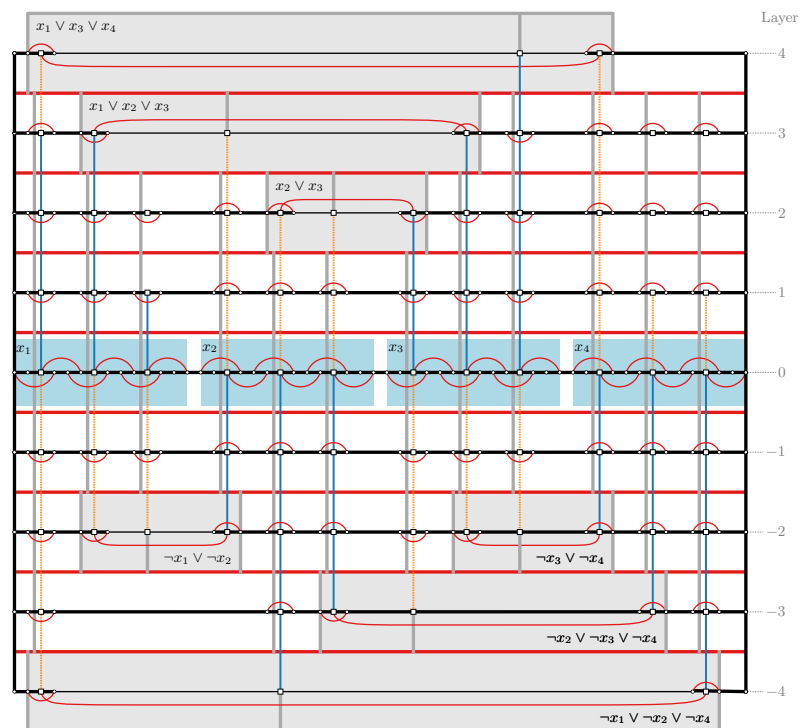
■ **Figure 8** Solution (in red) of the k -PIP instance coming from the graph given in Figure 3.



■ **Figure 9** Different representations of the $k - 1$ matching edges which substitute the alternating path P_i in the case that F is a matching.

We can use essentially the same construction, but replace the alternating connections between the layers by single edges to prove NP-hardness also for the case that F is a matching; see Figures 9 and 10.

► **Corollary 8.** k -PIP is NP-complete for every $k \geq 1$, even if G is biconnected and F forms a matching.



■ **Figure 10** k -PIP instance where F is a matching reduced from the graph given in Figure 3.

4 Conclusion

We introduced the k -PIP problem and showed that it is NP-complete for every $k \geq 1$ even when the given graph is biconnected and the inserted edges form a path or matching. We also presented a linear-time algorithm for 1-PIP when the given graph is triangulated. This naturally raises the question if the triconnected case of 1-PIP is also polynomial-time solvable.

References

- 1 Patrizio Angelini, Giuseppe Di Battista, Fabrizio Frati, Vít Jelínek, Jan Kratochvíl, Maurizio Patrignani, and Ignaz Rutter. Testing planarity of partially embedded graphs. *ACM Transactions on Algorithms*, 11(4):32:1–32:42, 2015. doi:10.1145/2629341.
- 2 Patrizio Angelini, Ignaz Rutter, and Sandhya T. P. Extending partial orthogonal drawings. In *Proceedings of the 28th International Symposium on Graph Drawing and Network Visualization (GD'20)*, volume 12590 of *LNCS*, page 265–278. Springer, 2020. doi:10.1007/978-3-030-68766-3_21.
- 3 Alan Arroyo, Martin Derka, and Irene Parada. Extending simple drawings. In *Proceedings of the 27th International Symposium on Graph Drawing and Network Visualization (GD)'19*, volume 11904 of *LNCS*, pages 230–243. Springer, 2019. doi:10.1007/978-3-030-35802-0_18.
- 4 Alan Arroyo, Fabian Klute, Irene Parada, Birgit Vogtenhuber, Raimund Seidel, and Tilo Wiedera. Inserting one edge into a simple drawing is hard. *Discrete & Computational Geometry*, 69(3):745–770, 2023. doi:10.1007/s00454-022-00394-9.
- 5 Guido Brückner and Ignaz Rutter. Partial and constrained level planarity. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'17)*, page 2000–2011. SIAM, 2017. doi:10.1137/1.9781611974782.130.

- 6 Erin W. Chambers, David Eppstein, Michael T. Goodrich, and Maarten Löffler. Drawing graphs in the plane with a prescribed outer face and polynomial area. *Journal of Graph Algorithms and Applications*, 16(2):243–259, 2012. doi:10.7155/jgaa.00257.
- 7 Timothy M. Chan, Fabrizio Frati, Carsten Gutwenger, Anna Lubiw, Petra Mutzel, and Marcus Schaefer. Drawing partially embedded and simultaneously planar graphs. *Journal of Graph Algorithms and Applications*, 19(2):681–706, 2015. doi:10.7155/jgaa.00375.
- 8 Markus Chimani, Carsten Gutwenger, Petra Mutzel, and Christian Wolf. Inserting a vertex into a planar graph. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms, (SODA'09)*, page 375–383. SIAM, 2009. doi:10.1137/1.9781611973068.42.
- 9 Markus Chimani and Petr Hliněný. Inserting multiple edges into a planar graph. *Journal of Graph Algorithms and Applications*, 27(6):489–522, 2023. doi:10.7155/JGAA.00631.
- 10 Mark de Berg and Amirali Khosravi. Optimal binary space partitions for segments in the plane. *International Journal of Computational Geometry and Applications*, 22(3):187–206, 2012. doi:10.1142/S0218195912500045.
- 11 Walter Didimo, Giuseppe Liotta, and Fabrizio Montecchiani. A survey on graph drawing beyond planarity. *ACM Computing Surveys*, 52(1):4:1–4:37, 2019. doi:10.1145/3301281.
- 12 Reinhard Diestel. *Graph Theory, 5th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012. doi:10.1007/978-3-662-53622-3.
- 13 Eduard Eiben, Robert Galian, Thekla Hamm, Fabian Klute, and Martin Nöllenburg. Extending nearly complete 1-planar drawings in polynomial time. In *Proceedings of the 45th International Symposium on Mathematical Foundations of Computer Science, (MFCS'20)*, volume 170 of *LIPICs*, page 31:1–31:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.MFCS.2020.31.
- 14 Eduard Eiben, Robert Galian, Thekla Hamm, Fabian Klute, and Martin Nöllenburg. Extending partial 1-planar drawings. In *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming (ICALP'20)*, volume 168 of *LIPICs*, page 43:1–43:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.ICALP.2020.43.
- 15 Robert Galian, Thekla Hamm, Fabian Klute, Irene Parada, and Birgit Vogtenhuber. Crossing-optimal extension of simple drawings. In *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPICs*, page 72:1–72:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICs.ICALP.2021.72.
- 16 Carsten Gutwenger, Petra Mutzel, and René Weiskircher. Inserting an edge into a planar graph. *Algorithmica*, 41(4):289–308, 2005. doi:10.1007/S00453-004-1128-8.
- 17 Thekla Hamm and Petr Hliněný. Parameterised partially-predrawn crossing number. In *Proceedings of the 38th International Symposium on Computational Geometry (SoCG'22)*, volume 224 of *LIPICs*, page 46:1–46:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICs.SoCG.2022.46.
- 18 Seok-Hee Hong and Takeshi Tokuyama, editors. *Beyond Planar Graphs, Communications of NII Shonan Meetings*. Springer, 2020. doi:10.1007/978-981-15-6533-5.
- 19 Julia Katheder, Philipp Kindermann, Fabian Klute, Irene Parada, and Ignaz Rutter. On k -plane insertion into plane drawings. *CoRR*, abs/2402.14552, 2024. doi:10.48550/arXiv.2402.14552.
- 20 Donald E. Knuth and Arvind Raghunathan. The problem of compatible representatives. *SIAM Journal on Discrete Mathematics*, 5(3):422–427, 1992. doi:10.1137/0405033.
- 21 Giordano Da Lozzo, Giuseppe Di Battista, and Fabrizio Frati. Extending upward planar graph drawings. *Computational Geometry: Theory and Applications*, 91:101668, 2020. doi:10.1016/j.comgeo.2020.101668.
- 22 Tamara Mchedlidze, Martin Nöllenburg, and Ignaz Rutter. Extending convex partial drawings of graphs. *Algorithmica*, 76(1):47–67, 2016. doi:10.1007/s00453-015-0018-6.

- 23 Petra Mutzel and Thomas Ziegler. The constrained crossing minimization problem. In Jan Kratochvíl, editor, *Proceedings of the 7th International Symposium on Graph Drawing (GD'99)*, volume 1731 of *Lecture Notes in Computer Science*, page 175–185. Springer, 1999. doi:10.1007/3-540-46648-7_18.
- 24 Petra Mutzel and Thomas Ziegler. The constrained crossing minimization problem a first approach. In *Proceedings of the 1999 International Conference on Operations Research*, page 125–134. Springer, 1999. doi:10.1007/978-3-642-58409-1_11.
- 25 Maurizio Patrignani. On extending a partial straight-line drawing. *International Journal of Foundations of Computer Science*, 17(5):1061–1070, 2006. doi:10.1142/S0129054106004261.
- 26 Thomas Ziegler. *Crossing minimization in automatic graph drawing*. PhD thesis, Saarland University, Saarbrücken, Germany, 2001. URL: <https://d-nb.info/961610808>.

Noncrossing Longest Paths and Cycles

Greg Aloupis ✉ 🏠

Khoury College of Computer Sciences,
Northeastern University, Boston, MA, USA

Prosenjit Bose ✉ 🏠 


School of Computer Science,
Carleton University, Ottawa, Canada

David Eppstein ✉ 🏠

Computer Science Department,
University of California, Irvine, CA, USA

Saeed Odak ✉ 🏠

School of Electrical Engineering and Computer
Science, University of Ottawa, Canada

Csaba D. Tóth ✉ 🏠 

Department of Mathematics, California State
University Northridge, Los Angeles, CA, USA
Department of Computer Science,
Tufts University, Medford, USA

Ahmad Biniiaz ✉ 🏠

School of Computer Science,
University of Windsor, Canada

Jean-Lou De Carufel ✉ 🏠

School of Electrical Engineering and Computer
Science, University of Ottawa, Canada

Anil Maheshwari ✉ 🏠 

School of Computer Science,
Carleton University, Ottawa, Canada

Michiel Smid ✉ 🏠

School of Computer Science,
Carleton University, Ottawa, Canada

Pavel Valtr ✉ 🏠 

Department of Applied Mathematics,
Charles University, Prague, Czech Republic

Abstract

Edge crossings in geometric graphs are sometimes undesirable as they could lead to unwanted situations such as collisions in motion planning and inconsistency in VLSI layout. Short geometric structures such as shortest perfect matchings, shortest spanning trees, shortest spanning paths, and shortest spanning cycles on a given point set are inherently noncrossing. However, the longest such structures need not be noncrossing. In fact, it is intuitive to expect many edge crossings in various geometric graphs that are longest.

Recently, Álvarez-Rebollar, Cravioto-Lagos, Marín, Solé-Pi, and Urrutia (Graphs and Combinatorics, 2024) constructed a set of points for which the longest perfect matching is noncrossing. They raised several challenging questions in this direction. In particular, they asked whether the longest spanning path, on any finite set of points in the plane, must have a pair of crossing edges. They also conjectured that the longest spanning cycle must have a pair of crossing edges.

In this paper, we give a negative answer to the question and also refute the conjecture. We present a framework for constructing arbitrarily large point sets for which the longest perfect matchings, the longest spanning paths, and the longest spanning cycles are noncrossing.

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Combinatoric problems; Mathematics of computing → Paths and connectivity problems

Keywords and phrases Longest Paths, Longest Cycles, Noncrossing Paths, Noncrossing Cycles

Digital Object Identifier 10.4230/LIPIcs.GD.2024.36

Funding *Ahmad Biniiaz*: Research supported by NSERC.

Prosenjit Bose: Research supported by NSERC.

Jean-Lou De Carufel: Research supported by NSERC.

David Eppstein: Research supported by NSF grant CCF-2212129.

Anil Maheshwari: Research supported by NSERC.

Saeed Odak: Research supported by NSERC.

Michiel Smid: Research supported by NSERC.

Csaba D. Tóth: Research supported in part by the NSF award DMS-2154347.

Pavel Valtr: Research supported by Czech Science Foundation grant GAČR 23-04949X.



© Greg Aloupis, Ahmad Biniiaz, Prosenjit Bose, Jean-Lou De Carufel, David Eppstein,
Anil Maheshwari, Saeed Odak, Michiel Smid, Csaba D. Tóth, and Pavel Valtr;
licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 36; pp. 36:1–36:17



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Acknowledgements This work was initiated at the 10th Annual Workshop on Geometry and Graphs, held at Bellairs Research Institute in Barbados in February 2023. We thank the organizers and the participants.

1 Introduction

Traversing points in the plane by a polygonal path or cycle possessing a desired property has a rich background. For instance, the celebrated travelling salesperson problem asks for a polygonal path or cycle with minimum total edge length [9, 23, 25]. In recent years, there has been increased interest in paths and cycles with properties such as being noncrossing [2, 16], minimizing the longest edge length [6, 12, 22], maximizing the shortest edge length [7], minimizing the total or largest turning angle [1, 11, 18, 21], and minimizing the number of turns [13, 17, 26] to name a few. The longest cycle – the MaxTSP – is NP-hard in Euclidean spaces of dimension ≥ 3 , but the complexity of the planar MaxTSP is unknown [20, 10]. Paths and cycles that have combinations of these properties have also attracted attention. For example, simultaneously being noncrossing and having maximum total edge length [3, 19] is difficult to satisfy: to achieve a larger length we typically introduce more crossings.

Edge crossings in geometric graphs are usually undesirable as they have the potential of creating unwanted situations such as collisions in motion planning and inconsistency in VLSI layout. They are also undesirable in the context of graph drawing and network visualization as they make drawings more difficult to read and use. Short geometric structures such as shortest perfect matchings, shortest spanning trees, shortest spanning paths, and shortest spanning cycles are inherently noncrossing. This property, however, does not necessarily hold if the structure is not shortest. For long structures such as longest perfect matchings, longest spanning trees, longest spanning paths, and longest spanning cycles – the other end of the spectrum – it seems natural to expect many crossings. Counting crossings in geometric graphs and finding geometric structures with a minimum or maximum number of crossings are active research areas in discrete geometry. The study of this type of problem attracted more attention after the work of Aronov et al. [8] in 1994, who showed that any set of n points in the plane in general position admits a *crossing family* (a set of pairwise intersecting segments) of size $\Omega(\sqrt{n})$. They also conjectured that the true lower bound is linear in n . The current best lower bound, $n^{1-o(1)}$, was established by Pach et al. [24] in 2019.

The noncrossing property of shortest structures is mainly ensured by the triangle inequality. The triangle inequality, as noted by Alon et al. [3], also implies that the longest structures often have crossings because a structure usually gets longer by creating more crossings. Alon et al. [3] studied the problem of finding longest noncrossing structures (such as matchings, paths, or trees). Some of their initial results have been improved and extended by Dumitrescu and Tóth [19] (for matchings, paths, and cycles), by Biniáz et al. [14] and by Cabello et al. [15] (for trees). Along this direction, one might wonder whether a longest structure (defined on an arbitrarily large point set) is necessarily crossing. This was explicitly asked by Álvarez-Rebollar et al. [4]. Among other interesting results, they presented arbitrarily large planar point sets for which the longest perfect matching is noncrossing. They asked the following question and proposed the following conjecture:

► **Question 1** (Álvarez-Rebollar et al. [4]). *For every sufficiently large planar point set, must the longest spanning path have two edges that cross each other?*

► **Conjecture 1** (Álvarez-Rebollar et al. [4]). *The longest spanning cycle on every sufficiently large set of points in the plane has a pair of crossing edges.*

The “sufficiently large” condition in the question and conjecture makes sense, as otherwise one can take any 3 points in general position, or any 4 points that are not in a convex position – for such point sets, all spanning paths and cycles are noncrossing.

In the other direction, one might wonder about maximizing the number of crossings in cycles. Here, we would like to highlight another result of Álvarez-Rebollar et al. [4, 5]. Let $C(n)$ be the largest number such that any set of n points in the plane admits a spanning cycle with at least $C(n)$ pairs of crossing edges. Álvarez-Rebollar et al. [4, 5] established the following lower and upper bounds: $n^2/12 - O(n) < C(n) < 5n^2/18 - O(n)$. In other words, any set of n points in the plane admits a spanning cycle with at least $n^2/12 - O(n)$ crossings, and there is a family of point sets that does not admit any cycle with more than $5n^2/18 - O(n)$ crossings.

1.1 Our contributions

In this paper, we provide negative answers to both Question 1 and Conjecture 1. For any integer $n \geq 1$ we present a set of n points in the plane for which the longest spanning path is unique and noncrossing. Similarly, for any integer $n \geq 4$, we present a set of n points in the plane for which the longest spanning cycle is unique and noncrossing. To build such point sets, we use the following framework: First, we choose a set P of points on the x -axis for which the longest structure may not be unique. Then, we assign new y -coordinates to points in P to obtain a new point set P' for which the longest structure corresponds to one in P and is also unique and noncrossing. In Section 6, we present some structural properties of longest paths and cycles, which may be of independent interest.

1.2 Preliminaries

All point sets considered in this paper are in the Euclidean plane. A *geometric graph* is a graph with vertices represented by points and edges represented by line segments between the points. Let P be a finite point set. A *spanning path* for P is a path drawn with straight-line edges such that every point in P lies at a vertex of the path and every vertex of the path lies at a point in P . A *spanning cycle* is defined analogously. In other words, a spanning path is a Hamiltonian path in the complete geometric graph on P , and a spanning cycle is a Hamiltonian cycle in this graph.

Consider two line segments, each connecting a pair of points in P . If the interiors of the segments intersect, then we say that they *cross*; this configuration is called a *crossing*. A path or a cycle is called *noncrossing* if its edges do not cross each other. We denote the undirected edge between two points p and q by pq , the directed edge from p towards q by (p, q) , and the Euclidean distance between p and q by $|pq|$. The *length* of a geometric graph G is the sum of the lengths of its edges, and we denote it by $|G|$.

2 Longest Paths and Cycles on the Real Line

In this section we characterize longest paths and cycles in dimension one. These observations play a pivotal role in our constructions in the plane (Sections 3 and 4). We say that an edge e *intersects a point* p if the intersection of e and p is not empty (the intersection could be an endpoint of e). For a sorted set of $2k+1$ numbers, the median is the number with rank $k+1$, and for a sorted set of $2k$ numbers, the median is the mean of the two numbers with ranks k and $k+1$.

36:4 Noncrossing Longest Paths and Cycles

► **Lemma 2.** *Let P be a set with an even number of points in \mathbb{R} , i.e., in dimension one. The endpoints of any longest spanning path on P lie on different sides of the median of P .*

Proof. Let $P = \{p_1, \dots, p_n\}$ and assume *w.l.o.g.* that 0 is the median of P (in particular, $0 \notin P$). Let H be a longest spanning path on P . Orient the edges of H to make it a directed path. Let p_s and p_e be the starting and ending points of H , respectively. For the sake of contradiction, assume that p_s and p_e have the same sign, which we may assume, due to symmetry, to be positive. Thus $p_s, p_e > 0$. Then, the sum of degrees of vertices in H to the left of the origin is 2 more than the sum of degrees of vertices to the right. Therefore, H must have a directed edge (p_a, p_b) where $p_a, p_b < 0$. If $p_b < p_a$, then by replacing (p_a, p_b) with the undirected edge $p_s p_b$ we obtain a longer undirected path; and if $p_b > p_a$ by replacing (p_a, p_b) with $p_e p_a$ we obtain a longer undirected path. Both cases lead to a contradiction. ◀

► **Lemma 3.** *Let P be a set with an even number of points in \mathbb{R} , i.e., in dimension one. Let H be a spanning path on P . Then H is a longest spanning path if and only if*

- (i) *every edge of H intersects the median of P , and*
- (ii) *the two endpoints of H are the two points closest to the median of P .*

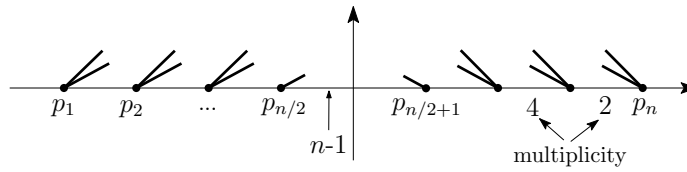
Proof. Let $P = \{p_1, \dots, p_n\}$ so that $p_i < p_j$ for all $i < j \in \{1, \dots, n\}$, and assume *w.l.o.g.* that 0 is the median of P . Note that $0 \notin P$ since n is even. First, we prove by contradiction that if H is a longest spanning path, then (i) and (ii) hold.

Suppose that (i) does not hold. Orient the edges of H to make it a directed path. Let (p_a, p_b) be an edge of H that does not intersect the median. Due to symmetry, assume that $p_a, p_b < 0$. By Lemma 2, the endpoints of H lie on different sides of the median. This implies that both sides have the same sum of vertex degrees. Thus H must have an edge (p_c, p_d) such that $c, d > 0$. By replacing these edges with $p_a p_c$ and $p_b p_d$ we obtain an (undirected) spanning path that is longer than H because $|p_a - p_c| + |p_b - p_d| > |p_a - p_b| + |p_c - p_d|$. This contradicts H being a longest path.

Now suppose that (ii) does not hold: without loss of generality $p_{n/2}$ is not an endpoint of H . (The case for $p_{n/2+1}$ can be handled symmetrically). Then H has an endpoint p_a with $a < n/2$. Orient the edges of H so that the path is directed from p_a towards the other endpoint. Let $(p_{n/2}, p_b)$ be the outgoing edge from $p_{n/2}$. By part (i), we have $p_b \geq 0$. By removing $(p_{n/2}, p_b)$ we obtain two paths, and p_b is an endpoint on one of those paths. Next, join the paths with a new edge (p_a, p_b) . Thus we obtain an (undirected) spanning path that is longer than H because $|p_a - p_b| > |p_{n/2} - p_b|$. This contradicts H being longest.

Finally, we prove that any spanning path H that satisfies (i) and (ii) is longest, using a direct proof. Consider a longest spanning path L on P . By the sufficiency proof, (i) and (ii) hold for L . This implies that the positive interval $[p_{n/2}, p_{n/2+1}]$ is contained in each of the $n-1$ edges, hence it contributes to the length of L with multiplicity $n-1$. Similarly, for any $i \in \{2, \dots, n/2\}$ the positive interval $[p_{n/2+i-1}, p_{n/2+i}]$ contributes to the length of L by multiplicity $n-2i$. A similar argument holds for negative intervals. See Figure 1. On the other hand, any spanning path (including H) that satisfies (i) and (ii) receives the exact same multiplicities from the corresponding intervals. Therefore H and L have the same length, and hence H is also a longest path. ◀

A statement similar to that of Lemma 3 can be proved for paths with an odd number of points (in this case one endpoint is the median itself and the other endpoint is the closest point to the median). However, we will not use this in our construction.



■ **Figure 1** Illustration of a longest path for a point set on a line, for the case where the number of points, n , is even. Numbers below intervals $[p_{n/2+i}, p_{n/2+i+1}]$ represent the multiplicity of the contribution of the corresponding intervals to the length of the longest path.

► **Lemma 4.** *Let P be a finite set in \mathbb{R} , i.e., in dimension one.*

- (i) *A spanning cycle on P is longest iff each of its edges intersects the median of P .*
- (ii) *If P contains an odd number of points, then for any longest spanning cycle the two edges incident to the median lie on opposite sides of it.*
- (iii) *Assume that P contains $n = 2k+1$ points and there is an interval I of length $h > 0$ between the leftmost $k+1$ and the rightmost k points. Then in any longest spanning cycle, $n-1 = 2k$ edges contain the interval I ; and if a spanning cycle has fewer than $2k$ edges that contain I , then it is at least $2h$ shorter than a longest cycle.*

Proof. Let $P = \{p_1, \dots, p_n\}$ so that $p_i < p_j$ for all $i < j \in \{1, \dots, n\}$, and assume *w.l.o.g.* that 0 is the median of P . Note that $0 \notin P$ if n is even, and $p_{\lceil n/2 \rceil} = 0$ if n is odd.

First we prove the sufficiency of (i) by contradiction. Let C be a longest cycle on P , and orient its edges to obtain a directed cycle. Suppose, for the sake of contradiction, that the edge (p_a, p_b) of C does not intersect the median. We may assume *w.l.o.g.* that $p_a, p_b < 0$. The sum of vertex degrees strictly on the left and right side of the median are the same, and the edges that contain 0 in their interior contribute 1 to both sums. Consequently, C contains an edge (p_c, p_d) with $p_c, p_d > 0$; or (when n is odd) there are two edges incident to the median, say $(p_c, 0)$ and $(0, p_d)$ with $p_c, p_d > 0$. In the first case, we can replace edges (p_a, p_b) and (p_c, p_d) with (p_a, p_c) and (p_b, p_d) . In the second case, replace (p_a, p_b) and $(p_c, 0)$ with (p_a, p_c) and $(p_b, 0)$. In both cases, we obtain a longer (undirected) spanning cycle, contradicting the maximality of C .

The necessity of (i) can be proved by a counting argument similar to that of Lemma 3-(i).

Now, we prove (ii) by contradiction. Without loss of generality, let $0 \in P$ be the median of P . Suppose that the median is incident to two edges $(p_c, 0)$ and $(0, p_d)$ with $p_c, p_d > 0$. Then, there is a point in P to the right of 0 incident to an edge of C that does not contain 0 in its interior. Denote this edge by (p_a, p_b) , where $p_a, p_b < 0$. We can replace edges (p_a, p_b) and $(p_c, 0)$ with (p_a, p_c) and $(p_b, 0)$ to obtain a longer spanning cycle, contradicting the maximality of C .

To prove the first part of (iii), note that if $n = 2k+1$, then the median is the $(k+1)$ -st point of P , that we denote by p_0 . Let C be a longest cycle on P . It is implied from (i) and (ii) that exactly one edge of C (which is incident to p_0) does not contain I . The remaining $n-1 = 2k$ edges contain I .

For the second claim in (iii), let C be a spanning cycle on P in which fewer than $2k$ edges contain I . Orient the edges of C to obtain a directed cycle. The sum of degrees of the leftmost $k+1$ (resp., rightmost k) vertices is $2k+2$ (resp., $2k$), and the edges containing I have fewer than $2k$ left (resp., right) endpoints. Consequently, the leftmost $k+1$ (resp., rightmost k) points in P induce at least two edges (resp., one edge) of C . Therefore, C contains two edges, (p_a, p_b) and (p_c, p_d) , such that p_a, p_b are to the left of I and p_b, p_d are to the right of I . We can replace these two edges with (p_a, p_c) and (p_b, p_d) , to obtain a spanning cycle C' that traverses I two more times than C . In particular, we have $|C'| \geq |C| + 2|I| = |C| + 2h$, hence $|C| \leq |C'| - 2h \leq |C_{\max}| - 2h$, where C_{\max} is a longest cycle on P . ◀

3 Noncrossing Longest Paths

Let $n \geq 1$ be an integer. In this section, we construct n points for which the longest spanning path is unique and noncrossing. This can be easily observed for $n < 5$: For example, for $n = 4$, any spanning path of the vertices of a triangle and a point in the interior is noncrossing. Thus, we will now assume that $n \geq 5$. In Section 2, we uncovered some structural properties of longest paths for n points on a line. Here we show how to construct a 2-dimensional point set starting with n points on the x -axis and then assigning y -coordinates to the points. We show that the longest path is unique and noncrossing. We describe our construction for the case where n is even; the construction for the case where n is odd follows with some minor changes. The following theorem summarizes our result in this section.

► **Theorem 5.** *For every integer $n \geq 1$ there exists a set of n points in the plane for which the longest spanning path is unique and noncrossing.*

In Section 3.1 we give an overview of our construction for an even number of points. The details and proofs are given in Section 3.2. The case of odd paths is considered in Section 3.3.

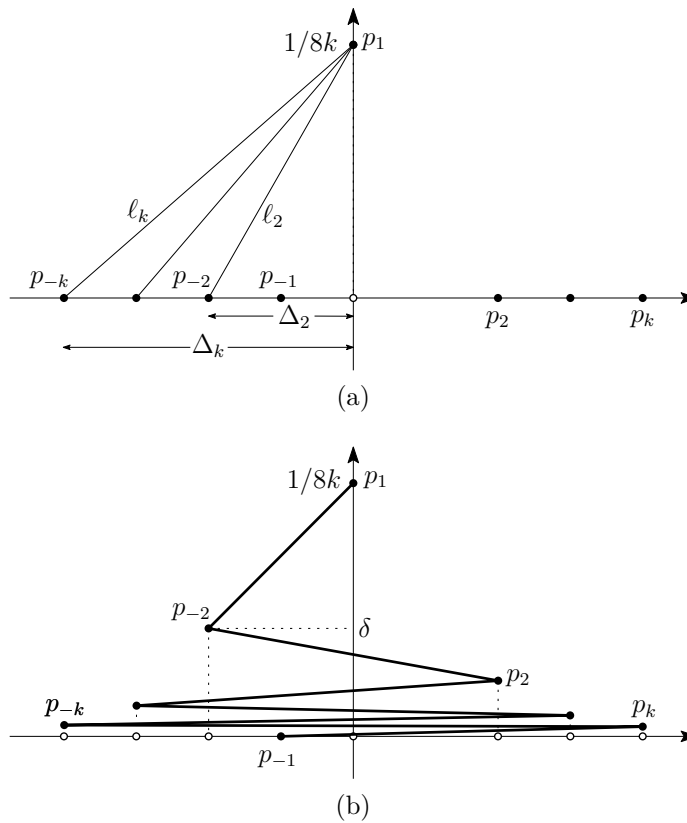
3.1 A path with an even number of points: An overview

For $k \geq 3$, consider a set P of $n = 2k$ points p_i on the x -axis such that $p_1 = (0, 0)$ and $p_i = (i, 0)$ for $i = -1, \pm 2, \dots, \pm k$, as illustrated in Figure 2(a). Our construction would work even if we set $p_1 = (1, 0)$; however, for a reason that will become clear in Section 3.3, we set p_1 differently. The longest spanning path for this point set is not unique. In fact, Lemma 3 implies that any spanning path with endpoints p_1 and p_{-1} and with all edges crossing the y -axis is a longest path. Conversely, any longest path must have endpoints p_1 and p_{-1} , and its edges must cross the y -axis. Let \mathcal{H} be the set of these paths. Let P' be the point set obtained by assigning to each point p_i a y -coordinate y_i such that, as illustrated in Figure 2(b), the following holds:

$$\frac{1}{8k} = y_1 \gg y_{-2} \gg y_2 \gg y_{-3} \gg y_3 \gg \dots \gg y_{-k} \gg y_k \gg y_{-1} = 0.$$

The value y_1 is much larger than y_{-2} , which is in turn much larger than y_2 and so on. Notice that the largest y -coordinate y_1 is $1/8k$ which is much smaller than 1. Due to the small y -coordinates, a longest path H' on P' corresponds to a path $H \in \mathcal{H}$. The length of H' is roughly the length of H plus a very small value $\Delta(H')$, which depends on the new y -coordinates. Let e_1 be the only edge of H' incident to p_1 . Since p_1 has a very large y -coordinate compared to other points, the contribution of e_1 to $\Delta(H')$ is larger than the contribution of other edges. The contribution of e_1 is maximized if it connects p_1 to the nearest plausible neighbor, which is p_{-2} ; this can be observed from Figure 2(b). Therefore $e_1 = p_1 p_{-2}$. By a similar argument, p_{-2} gets connected to p_2 , and so on. It follows that the path H' is unique and it is $p_1, p_{-2}, p_2, p_{-3}, p_3, \dots, p_{-k}, p_k, p_{-1}$. This path is y -monotone, and hence noncrossing; see Figure 2(b).

Note. Figures 2(a) and 2(b) are not to scale. The y -coordinates should be small enough so that all points lie almost on the x -axis (We exaggerated the y -coordinates to facilitate readability). Moreover, if we orient the path from p_1 towards p_{-1} , then the extension of every directed edge intersects all edges that follow.



■ **Figure 2** Illustration of the construction of a longest path for $2k$ points. The figure is not to scale as the real y -coordinates are very small so that the points lie almost on the x -axis. (a) Lifting p_1 to the y -coordinate $1/8k$. (b) The final longest path.

3.2 A path with an even number of points: Details

Recall the set P of $2k$ points, $k \geq 3$, on the x -axis, described in the previous section and illustrated in Figure 2(a). We say that an edge e intersects the y -axis if the intersection of e and the y -axis is not empty (the intersection could be an endpoint of e). The longest paths for points on a line were characterized in Lemma 3. Denote by \mathcal{H} the set of all longest spanning paths on P .

► **Lemma 6.** *Let $0 \leq \varepsilon \leq \frac{1}{8k}$ be a real number. Suppose that every point in P is perturbed by a distance of at most ε . Let P' be the new point set after perturbation. Then, the order of the points along any longest path for P' is the same as the order of the points along some path in \mathcal{H} .*

Proof. The length of any path on P is an integer. Therefore, any path in \mathcal{H} is at least 1 unit longer than any path not in \mathcal{H} .

Let H' be any longest path on P' . The difference between its length and the length of any path in \mathcal{H} is at most $(2k-1) \cdot 2\varepsilon$ because H' has $2k-1$ edges, each edge has 2 endpoints, and each endpoint is at distance at most ε from its corresponding point in P . Since $\varepsilon \leq \frac{1}{8k}$ the difference is less than $1/2$. Therefore, H' cannot correspond to a path that is not in \mathcal{H} , so H' corresponds to a path in \mathcal{H} with the same order of points. ◀

Our plan is to assign new y -coordinates to the points of P to obtain a point set P' for which the longest path is y -monotone and unique. The new y -coordinates will be at most $\frac{1}{8k}$, and thus, by Lemma 6, the longest path H' of P' will correspond to a path in \mathcal{H} . We will make H' correspond to the path $p_1, p_{-2}, p_2, p_{-3}, p_3, \dots, p_k, p_{-1}$, which is in \mathcal{H} (by Lemma 3) and depicted in Figure 2(b). We assign to each point p_i the y -coordinate y_i such that the following holds:

$$y_1 \gg y_{-2} \gg y_2 \gg y_{-3} \gg y_3 \gg \dots \gg y_{-k} \gg y_k \gg y_{-1}.$$

We set $y_1 = \frac{1}{8k}$, $y_{-1} = 0$, and use the following lemma to identify the remaining y -coordinates.

► **Lemma 7.** *There exists a real number δ , with $0 < \delta < y_1$, such that if $0 \leq y_i \leq \delta$ for each $i \neq 1$ then the longest path on P' connects p_1 to p_{-2} .*

Proof. Since each y_i is at most $1/8k$, Lemma 6 implies that any longest path H' on P' corresponds to a path H in \mathcal{H} . Due to small y -coordinates, we have $|H'| = |H| + \Delta(H')$ for some small value $\Delta(H') \geq 0$ which depends on the new y -coordinates. Specifically, we have

$$\begin{aligned} |H'| &= \sum_{(p_i, p_j) \in E(H')} |p_i p_j| = \sum_{(p_i, p_j) \in E(H')} \sqrt{|i-j|^2 + |y_i - y_j|^2} \\ &= |H| + \sum_{(p_i, p_j) \in E(H')} \left(\sqrt{|i-j|^2 + |y_i - y_j|^2} - |i-j| \right) = |H| + \Delta(H'), \end{aligned}$$

where $0 \leq \sqrt{|i-j|^2 + |y_i - y_j|^2} - |i-j| \leq |y_i - y_j| \leq \max\{y_i, y_j\}$.

Recall from Lemma 3 that p_1 is an endpoint of any longest path in \mathcal{H} . Moreover, p_1 is connected to a point (different from p_{-1}) to the left of the y -axis. For $j \in \{2, \dots, k\}$ let ℓ_j be the Euclidean distance between p_1 and the point $(-j, 0)$, and let Δ_j be the difference of their x -coordinates as in Figure 2(a). The contribution of $p_1 p_{-j}$ to $|H'|$ would be at least $\ell_j - \delta$ (when p_{-j} has y -coordinate δ) and at most ℓ_j (when p_{-j} has y -coordinate 0). The contribution of the corresponding edge to $|H|$ would be Δ_j . Hence the contribution of $p_1 p_{-j}$ to $\Delta(H')$ would be at least $\ell_j - \delta - \Delta_j$ and at most $\ell_j - \Delta_j$. An easy calculation shows that

$$\ell_2 - \Delta_2 > \ell_3 - \Delta_3 > \dots > \ell_k - \Delta_k;$$

this is also implied by the fact that $\Delta_{i+1} - \Delta_i = 1$ while $\ell_{i+1} - \ell_i < 1$. If we set $\delta < (\ell_2 - \Delta_2) - (\ell_3 - \Delta_3)$, then the contribution of $p_1 p_{-2}$ to $\Delta(H')$ is at least

$$\ell_2 - \delta - \Delta_2 > \ell_2 - \Delta_2 - ((\ell_2 - \Delta_2) - (\ell_3 - \Delta_3)) = \ell_3 - \Delta_3,$$

which is larger than the contribution of any other plausible edge $p_1 p_{-j}$. Since the y -coordinates of all other points are less than δ , any other edge of H' contributes less than δ to $\Delta(H')$. By setting

$$\delta = \frac{(\ell_2 - \Delta_2) - (\ell_3 - \Delta_3)}{2k - 1},$$

the contribution of $p_1 p_{-2}$ exceeds the sum of the contributions of the remaining $2k - 2$ edges of H' . Thus, for this choice of δ the longest path H' connects p_1 to p_{-2} . ◀

By Lemma 7, we have a specific value δ such that the longest path includes edge (p_1, p_{-2}) . Now we set $y_{-2} = \delta$ and repeat the arguments of Lemma 7, with y_{-2} and p_{-2} (instead of y_1 and p_1). This implies that the next edge of the longest path will connect p_{-2} to p_2 . Repeating this $2k-5$ more times, we obtain the unique longest path $p_1, p_{-2}, p_2, p_{-3}, p_3, \dots, p_k, p_{-1}$, as in Figure 2(b); in each of the last two steps, there is only one remaining plausible edge (namely, $p_{-k} p_k$ from p_{-k} , and $p_k p_{-1}$ from p_k). This path is y -monotone and hence is noncrossing.

3.3 A path with an odd number of points

In this section, we obtain a noncrossing longest path with an odd number of points. Here is the place where we use the coordinate $(0,0)$ of the point $p_1 \in P$. We show that our construction for even paths leads to a construction for odd paths by simply removing p_1 . Thus we do not need to repeat the lemmas of Section 3.2 for the odd case.

We claim that if we remove the point p_1 from the path H' constructed on P' in the previous section, the remaining path, i.e., $H'' = p_{-2}, p_2, p_{-3}, p_3, \dots, p_k, p_{-1}$, is the longest path for the remaining $2k-1$ points. By construction, $|H'| = |H''| + |p_1 p_{-2}|$. Assume, for the sake of proof by contradiction, that the longest path L for the remaining points is longer than H'' . Among the two endpoints of L , let p_i be an endpoint that is not p_{-1} . Due to our choices of the x - and y -coordinates we have $|p_1 p_i| \geq |p_1 p_{-2}|$. Therefore the concatenation of L and $p_1 p_i$ would give a path on P' of length $|L| + |p_1 p_{-2}|$ which is larger than $|H'|$. This contradicts H' being the longest path on P' .

4 Noncrossing Longest Cycles

Let $n \geq 3$ be an integer. In this section, we construct a set of n points for which the longest spanning cycle is unique and noncrossing. For $n = 3$, every spanning cycle is noncrossing. For $n = 4$, we take three vertices of a triangle and a point in the interior. Thus, we assume that $n \geq 5$.

► **Theorem 8.** *For every integer $n \geq 3$ there exists a set of n points in the plane for which the longest spanning cycle is unique and noncrossing.*

In Section 4.1 we give an overview of our construction for an even number of points. The details and proofs are given in Section 4.2. For an odd number of points we sketch a construction in Section 4.3.

4.1 A cycle with an even number of points: An overview

Let $n \geq 6$ be an even number. Then either $n = 4k$ or $n = 4k-2$ for some integer k . To simplify the indexing (of points and y -coordinates) in our construction, from now on we assume that $n = 4k-2$. Let P be a set of n points, consisting of $2k$ points $p_i = (i, 0)$ for $i = \pm 1, \pm 2, \dots, \pm k$ and $2k-2$ points $p'_i = (i+\epsilon, 0)$ for $i = -1, \pm 2, \dots, \pm(k-1), k$, where $\epsilon > 0$ is a small value to be determined; see Figure 3. (The construction for $n = 4k$ is similar; it consists of P and two additional points $p_{k+1} = (k+1, 0)$ and $p'_{-k} = (-k+\epsilon, 0)$.) Our construction for cycles is somewhat similar to that of paths in the sense that our cycle consists of two y -monotone interior-disjoint paths between p_1 and p_{-k} (or between p_1 and p_{k+1} when n is a multiple of 4). Although the main idea sounds simple, the noncrossing property of the longest cycle is not straightforward and involves a more detailed analysis.

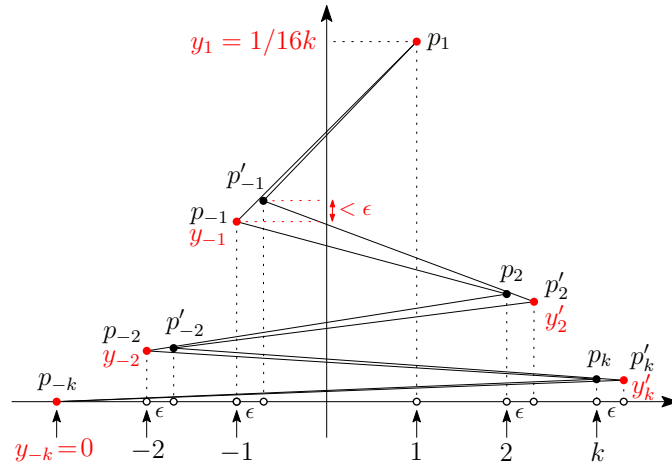
Lemma 4 implies that a spanning cycle on P is longest if and only if each of its edges intersects the y -axis. Let \mathcal{C} be the set of all longest spanning cycles on P . As illustrated in Figure 3, we obtain a point set P' by assigning to each point p_i and p'_i the respective y -coordinates y_i and y'_i such that:

$$\frac{1}{16k} = y_1 \gg y_{-1} \gg y'_2 \gg y_{-2} \gg y'_3 \gg \dots \gg y'_k \gg y_{-k} = 0.$$

For each $i \in \{2, 3, \dots, k\}$ we choose y_i such that p_i lies just below (almost on) the segment $p'_{-i+1} p'_i$, and for each $i \in \{-1, -2, \dots, -(k-1)\}$ we choose y'_i such that p'_i lies just below (almost on) the segment $p_{-i} p_i$.

36:10 Noncrossing Longest Paths and Cycles

Due to the small y -coordinates, any longest cycle C' on P' corresponds to a cycle $C \in \mathcal{C}$. Moreover $|C'| = |C| + \Delta(C')$ for some small value $\Delta(C')$ which depends on the new y -coordinates. Since p_1 has the largest y -coordinate, the contribution of the two edges of C' that are incident to p_1 (say e_1 and e_2) is maximized when they are connected to the nearest plausible neighbors which are p_{-1} and p'_{-1} . We will choose the y -coordinates in such a way that the contribution of e_1 and e_2 is larger than the sum of the contributions of the remaining edges of the cycle. Thus C' must connect p_1 to p_{-1} and p'_{-1} . Similarly, by a suitable choice of y -coordinates, we enforce C' to connect p_{-1} and p'_{-1} to the nearest plausible neighbors which are p_2 and p'_2 , and so on. By repeating this process, the longest cycle C' would be the concatenation of two paths $p_1, p_{-1}, p_2, p_{-2}, \dots, p_{-k}$ and $p_1, p'_{-1}, p'_2, p'_{-2}, \dots, p'_k, p_{-k}$.



■ **Figure 3** Illustration of the construction of a longest cycle for $4k-2$ points. The figure is not to scale. The y -coordinates should be small enough so that all points lie almost on the x -axis.

4.2 A cycle with an even number of points: Details

Recall the point set P from the previous section (the y -coordinates and the value of $\epsilon > 0$ will be determined in this section). The longest cycles for points on a line were characterized in Lemma 4. Let \mathcal{C} be the set of all longest cycles on P .

► **Lemma 9.** *Any cycle in \mathcal{C} is at least 1 unit longer than any cycle not in \mathcal{C} .*

Proof. Consider any cycle D that is not in \mathcal{C} . Lemma 4 implies that D has an edge that does not intersect the y -axis. Orient the edges of D to make it a directed cycle. Since the number of points to the left of the y -axis is the same as the number of points to its right, D has two directed edges (p_a, p_b) and (p_c, p_d) such that $a, b \leq -1$ and $c, d \geq 1$. By replacing these edges with $p_a p_c$ and $p_b p_d$ we obtain an (undirected) spanning cycle D' such that

$$|D'| - |D| = (|p_a p_c| + |p_b p_d|) - (|p_a p_b| + |p_c p_d|) \geq 2|p_1 p'_{-1}| = 2(2 - \epsilon) > 1.$$

Since the length of any cycle C in \mathcal{C} is at least $|D'|$, we get $|C| > |D| + 1$. ◀

► **Lemma 10.** *Let $0 \leq \epsilon \leq 1/16k$ be a real number. Suppose that every point of P is perturbed by a distance of at most ϵ . Then the order of the points along any longest cycle of the new point set is the same as the order of the points along some cycle in \mathcal{C} .*

Proof Sketch. The proof is similar to that of Lemma 6 and uses Lemma 9. The parameter ε is small enough such that the total change in the length of any spanning cycle on P is less than $1/2$. Together with Lemma 9, this implies that any longest cycle on the perturbed points corresponds to a cycle in \mathcal{C} . ◀

To obtain P' we only need to describe the following y -coordinates:

$$y_1 \gg y_{-1} \gg y'_2 \gg y_{-2} \gg y'_3 \gg \cdots \gg y'_k \gg y_{-k}.$$

The y -coordinates of the remaining points would then follow as outlined in the previous section (more details are given after Lemma 12). We set $y_1 = \frac{1}{16k}$ and $y_{-k} = 0$. We use the following lemma (which can be proven similarly to Lemma 7) to assign the y -coordinates.

► **Lemma 11.** *There exists a real number δ , $\varepsilon \leq \delta < y_1$, such that if $0 \leq y_i \leq \delta$ for $i \neq 1$ and $0 \leq y'_i \leq \delta$ for $i \neq -1$, then every longest cycle of P' connects p_1 to p_{-1} and p'_{-1} .*

Proof. Lemma 10 implies that any longest cycle C' on P' corresponds to a cycle C in \mathcal{C} . Due to small y -coordinates, we have $|C'| = |C| + \Delta(C')$ for some small value $\Delta(C') \geq 0$ which depends on the new y -coordinates. Lemma 4 implies that C' connects p_1 to two points to the left of the y -axis. Similar to Lemma 7, for $j \in \{1, \dots, k\}$ define ℓ_j as the Euclidean distance between p_1 and the point $(-j, 0)$, and define Δ_j as the difference of their x -coordinates. Analogously, for $j \in \{1, \dots, k-1\}$ define ℓ'_j and Δ'_j for p_1 and the point $(0, -j + \varepsilon)$. Every edge that connects p_1 to a point to the left of the y -axis has the following contributions to $|C|$, $|C'|$ and $\Delta(C')$.

- For $j \in \{1, \dots, k\}$ the contribution of $p_1 p_{-j}$ to $|C'|$ is at least $\ell_j - \delta$ and at most ℓ_j . The contribution of the corresponding edge to $|C|$ is Δ_j . Hence the contribution of $p_1 p_{-j}$ to $\Delta(C')$ is at least $\ell_j - \delta - \Delta_j$ and at most $\ell_j - \Delta_j$.
- For $j \in \{2, \dots, k-1\}$ the contribution of $p_1 p'_{-j}$ to $|C'|$ is at least $\ell'_j - \delta$ and at most ℓ'_j . The contribution of the corresponding edge to $|C|$ is Δ'_j . Thus the contribution of $p_1 p'_{-j}$ to $\Delta(C')$ is at least $\ell'_j - \delta - \Delta'_j$ and at most $\ell'_j - \Delta'_j$.
- The contribution of $p_1 p'_{-1}$ to $|C'|$ is at least $\ell'_1 - \delta - \varepsilon$ because the y -coordinate of p'_{-1} is at most $\delta + \varepsilon$; to verify this observe that $y_{-1} \leq \delta$ and $y'_{-1} - y_{-1} < \varepsilon$ because p'_{-1} is almost on $p_1 p_{-1}$ whose slope is less than 1; also see Figure 3 (recall that the figure is not to scale). The contribution of the corresponding edge to $|C|$ is Δ'_1 . Therefore the contribution of $p_1 p'_{-1}$ to $\Delta(C')$ is at least $\ell'_1 - \delta - \varepsilon - \Delta'_1$ and at most $\ell'_1 - \Delta'_1$.

Observe that

$$\ell'_1 - \Delta'_1 > \ell_1 - \Delta_1 > \ell'_2 - \Delta'_2 > \ell_2 - \Delta_2 > \cdots > \ell_k - \Delta_k.$$

If we set $\delta < \frac{1}{2}((\ell_1 - \Delta_1) - (\ell'_2 - \Delta'_2))$, then the contributions of $p_1 p_{-1}$ and $p_1 p'_{-1}$ to $\Delta(C')$ would respectively be at least

$$\ell_1 - \delta - \Delta_1 > \ell_1 - 2\delta - \Delta_1 > \ell_1 - \Delta_1 - ((\ell_1 - \Delta_1) - (\ell'_2 - \Delta'_2)) = \ell'_2 - \Delta'_2, \text{ and}$$

$$\ell'_1 - \delta - \varepsilon - \Delta'_1 \geq \ell'_1 - 2\delta - \Delta'_1 > \ell'_1 - \Delta'_1 - ((\ell_1 - \Delta_1) - (\ell'_2 - \Delta'_2)) > \ell'_2 - \Delta'_2,$$

which are larger than the contribution of any other edge $p_1 p_{-j}$ and $p_1 p'_{-j}$. By setting

$$\delta = \frac{1}{2} \frac{(\ell_1 - \Delta_1) - (\ell'_2 - \Delta'_2)}{4k - 2}$$

the contribution of each of $p_1 p_{-1}$ and $p_1 p'_{-1}$ would be even larger than the sum of the contributions of the remaining $4k-4$ edges of C' . Thus, for this choice of δ the longest cycle C' connects p_1 to p_{-1} and p'_{-1} . ◀

36:12 Noncrossing Longest Paths and Cycles

We choose δ as in the proof of Lemma 11, and set $y_{-1} = \delta$. Then we set y'_{-1} so that p'_{-1} lies just below (almost on) the segment p_1p_{-1} , as in Figure 3. Notice that $\delta < y'_{-1} < \delta + \epsilon = y_{-1} + \epsilon$. Then, by Lemma 11 the longest cycle connects p_1 to p_{-1} and p'_{-1} . By Lemma 4, the other edges incident to p_{-1} and p'_{-1} must cross the y -axis.

► **Lemma 12.** *There exists a real δ , $\epsilon \leq \delta < y_{-1}$, such that if $0 \leq y_i \leq \delta$ for $i \neq -1, 1, 2$ and $0 \leq y'_i \leq \delta$ for $i \neq -1$, then every longest cycle of P' connects p_{-1} to p_2 and p'_{-1} to p'_2 .*

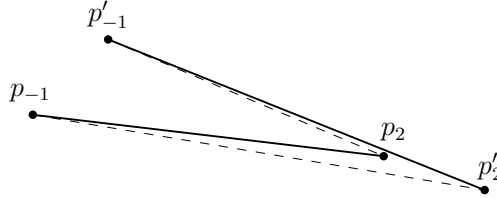
Proof. Recall the longest cycle C' from the proof of Lemma 11. We choose δ small enough such that the contribution of each of $p_{-1}p_2$, $p_{-1}p'_2$, $p'_{-1}p_2$, and $p'_{-1}p'_2$ to $\Delta(C')$ is larger than the sum of the contributions of the remaining $4k-6$ edges of C' . This would force C' to connect p_{-1} and p'_{-1} to p_2 and p'_2 .

By an argument similar to that of Lemma 11 we can find a parameter δ_1 that forces C' to connect p_{-1} to p_2 or p'_2 ($\delta_1, y_{-1}, p_{-1}, p_2$, and p'_2 play the roles of $\delta, y_1, p_1, p'_{-1}$, and p_{-1} , respectively). Similarly, we can find a parameter δ'_1 that forces C' to connect p'_{-1} to p_2 or p'_2 (where $\delta'_1, y'_{-1}, p'_{-1}, p_2$, and p'_2 play the roles of $\delta, y_1, p_1, p'_{-1}$, and p_{-1} , respectively). Then we choose $\delta = \min\{\delta_1, \delta'_1\}$.

Our choice of δ ensures that C' connects p_{-1} and p'_{-1} to p_2 and p'_2 . Notice that p_{-1} and p'_{-1} cannot both connect to p_2 or to p'_2 because it closes the cycle. Thus C' must use $p_{-1}p_2$ and $p'_{-1}p'_2$ or $p_{-1}p'_2$ and $p'_{-1}p_2$. We show that C' uses $p_{-1}p_2$ and $p'_{-1}p'_2$. See Figure 4. Recall that p_2 is almost on the edge $p'_{-1}p'_2$, and hence $|p'_{-1}p'_2| \approx |p'_{-1}p_2| + |p_2p'_2|$. By the triangle inequality we get $|p_{-1}p_2| + |p_2p'_2| > |p_{-1}p'_2|$. Adding these two yields

$$|p_{-1}p_2| + |p'_{-1}p'_2| > |p_{-1}p'_2| + |p'_{-1}p_2|, \quad (1)$$

which means that C' connects p_{-1} to p_2 and p'_{-1} to p'_2 . ◀



■ **Figure 4** The longest cycle connects p_{-1} to p_2 and p'_{-1} to p'_2 .

We choose our new δ as in the proof of Lemma 12, and set $y'_2 = \delta$. Now that the point p'_2 is fixed we can choose the y -coordinate of p_2 in the triangle $\Delta p_{-1}p'_{-1}p'_2$ and very close to the segment $p'_{-1}p'_2$ such that (1) holds. This forces the longest cycle to use $p_{-1}p_2$ and $p'_{-1}p'_2$. By repeatedly applying Lemma 12, the longest cycle will use the edges $p_i p_{-i}$ and $p'_i p'_{-i}$ (for $i > 0$) and the edges $p_i p_{-i+1}$ and $p'_i p'_{-i+1}$ (for negative $i < 0$). Therefore the longest cycle on P' is the concatenation of two paths: $p_1, p_{-1}, p_2, p_{-2}, \dots, p_{-k}$ and $p_1, p'_{-1}, p'_2, p'_{-2}, \dots, p'_k, p_{-k}$. This cycle is unique and noncrossing.

Each time we apply Lemma 12 we obtain a new value for δ . In each application we need δ to be greater than or equal to our fixed parameter ϵ . For this purpose, we choose ϵ to be the parameter δ that is obtained in the last application of Lemma 12, i.e., $\delta = y'_k$.

4.3 A cycle with an odd number of points: An overview

Our construction uses the longest paths of Section 3.2. First we observe that our path construction can be generalized to any set of x -coordinates.

► **Lemma 13.** For every even integer $n \geq 4$, every set P of n real numbers, and every $\delta > 0$ such that the δ -neighborhood of the median of P does not contain any points in P , there exists a set P' of n points in the plane with the following properties:

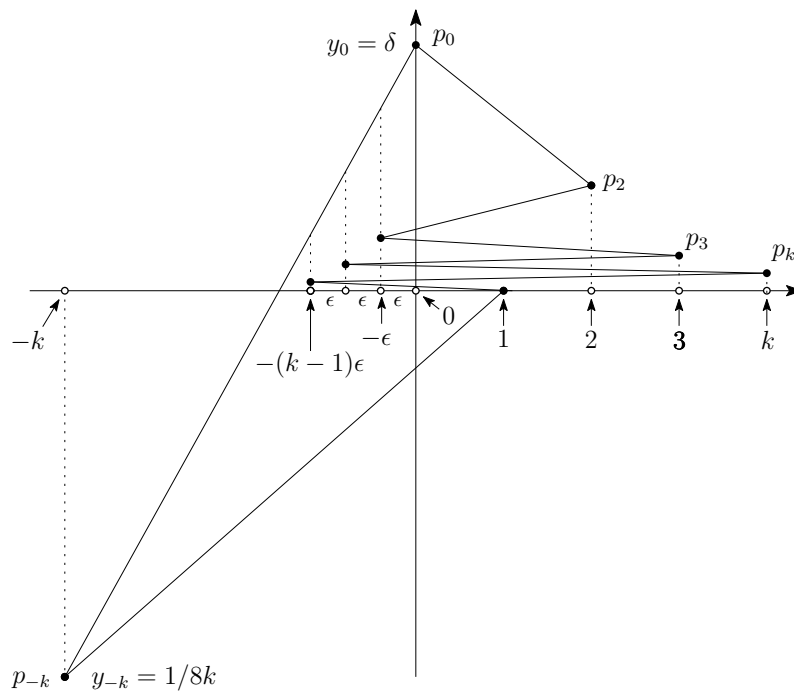
1. the x -projection of P' is P ;
2. all y -coordinates are in the interval $[0, \delta]$;
3. the x -projection of any longest path on P' is a longest path on P ;
4. the longest spanning path on P' is unique and noncrossing; and
5. the y -coordinates of the two endpoints of the longest path are 0 and δ .

Proof sketch. We choose the points in P' such that their x -coordinates are the same as the numbers in P and their y -coordinates are in $[0, \delta]$, and thus (1) and (2) follow.

By an argument similar to the proof of Lemma 3(i) one can show that the difference of lengths of a longest and a non-longest path on P is at least 2δ . Therefore Lemma 6 would imply that by choosing the y -coordinates in the interval $[0, 2\delta/8k]$, any longest path on P' corresponds to a longest path on P , and thus (3) follows. Items (4) and (5) follow by proper choices of y -coordinates similar to that of Lemma 7. ◀

We can now outline the construction; see Figure 5 for an illustration. Let $n = 2k+1$, for $k \geq 2$. We choose a set of x -coordinates as $P = \{-k, -(k-1)\epsilon, -(k-2)\epsilon, \dots, -\epsilon, 0, 1, 2, \dots, k\}$, where $\epsilon \in (0, 1/16k^2)$ will be specified later. Note that 0 is the median of P , and the set $A = \{-i \cdot \epsilon : i = 0, 1, \dots, k-1\} \subset [-1/16k, 0]$ forms a small cluster. By Lemma 4(ii), all edges of any longest cycle on P intersect the y -axis; and Lemma 4(iii) implies the following.

► **Observation 14.** The length of any cycle on P that connects p_{-k} to two points in A is at least 2 units shorter than a longest cycle on P .



■ **Figure 5** Illustration of the construction of a longest cycle for $2k+1$ points.

36:14 Noncrossing Longest Paths and Cycles

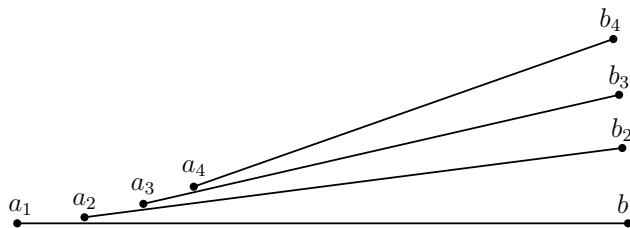
Below, we will specify a y -coordinate for each element in P . This will result in the point set P' for which the longest spanning cycle is unique and noncrossing. We will denote by A' the set of points in P' corresponding to A .

It remains to specify the y -coordinates of the points in P' and the parameter ϵ . Let p_x denote the point in P' with x -coordinate $x \in P$. We first choose the y -coordinate for the leftmost point: Let $y_{-k} = -1/16k$; this is the only negative y -coordinate. We assume that $|y_i| \ll 1/16k$ for all other points. This ensures that the longest cycle on P' corresponds to a longest cycle on the 1-dimensional multiset where 0 represents the entire cluster A (cf. Lemma 4(iii) and Lemma 10). By Lemma 4(ii), for any longest cycle on P' , the two edges incident to p_{-k} intersect the y -axis (i.e., the median). Furthermore, there is a threshold $\delta > 0$ such that if $0 \leq y_i \leq \delta$ for all remaining points, then p_{-k} must be adjacent to the two closest points on or to the right of the y -axis: That is, p_{-k} is adjacent to a point in cluster A' and to p_1 (cf. Observation 14 and Lemma 10). Next, we set $y_0 = \delta$ and find a threshold $\delta_1 \in (0, \delta)$ such that if $0 \leq y_i \leq \delta_1$ for all remaining points and $0 < \epsilon < \delta_1$, then the contribution of edge $p_{-k}p_0$ exceeds the sum of contributions of all remaining edges of a spanning cycle. Consequently, the longest cycle must include the edge $p_{-k}p_0$. Now both p_{-k} and p_0 are fixed, and we choose a sufficiently small $\epsilon \in (0, \delta_1)$ such that all remaining points in the cluster A' are below $p_{-k}p_0$ for all possible y -coordinates.

A longest cycle on P' comprises of $p_{-k}p_0$, $p_{-k}p_1$, and the longest path H' on $P' \setminus \{p_{-k}\}$ (from p_0 to p_1 cf. Lemma 3). By Lemma 13, we can choose y -coordinates for the remaining points such that H' is unique and noncrossing; and $y_1 = 0$. In particular, edge $p_{-k}p_1$ lies below the x -axis, hence below the entire path H' ; and $P' \setminus \{p_{-k}, p_0\}$ lies below the supporting line of $p_{-k}p_0$. Consequently, the concatenation of $p_{-k}p_0$, $p_{-k}p_1$ and H' is noncrossing.

5 Noncrossing Longest Matchings

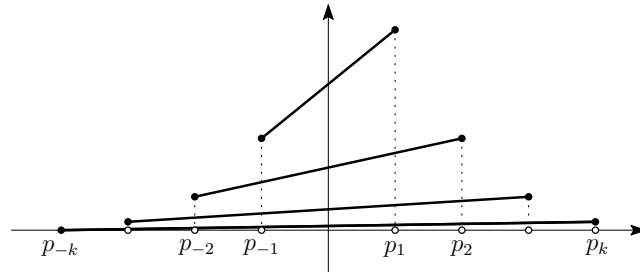
Álvarez-Rebollar et al. [4] showed that there exist point sets for which the longest perfect matchings are noncrossing. Their example is attributed to Kåre P. Villanger in a paper by Tverberg [27]. As illustrated in Figure 6, it consists of a set S of k segments with endpoints in $A = \{a_1, \dots, a_k\}$ and $B = \{b_1, \dots, b_k\}$. The distance between any two points $a_i \in A$ and $b_j \in B$ is larger than the distance between any two points in A , or the distance between any two points in B . The points in B are roughly on a vertical line. Álvarez-Rebollar et al. [4] have provided a precise description of the construction along with a detailed proof that S is a longest matching for $A \cup B$.



■ **Figure 6** Villanger's configuration as illustrated in [4].

Here, we exhibit an alternative point set for which the longest perfect matching is noncrossing. Our construction follows the same framework as for paths and cycles. Let P be a set of $2k$ points $p_i = (i, 0)$ for $i = \pm 1, \pm 2, \dots, \pm k$. One can verify that a perfect matching on P is longest if and only if all edges cross the y -axis. One such matching is

$M = \{p_{-i}p_i : i = 1, \dots, k\}$. Using ideas similar to those used for paths and cycles, one can assign to each p_i a new y -coordinate y_i to make M longest and noncrossing at the same time; see Figure 7. The new y -coordinates are of the following form: $y_1 \gg y_{-1} = y_2 \gg y_{-2} = y_3 \gg \dots = y_k \gg y_{-k}$.



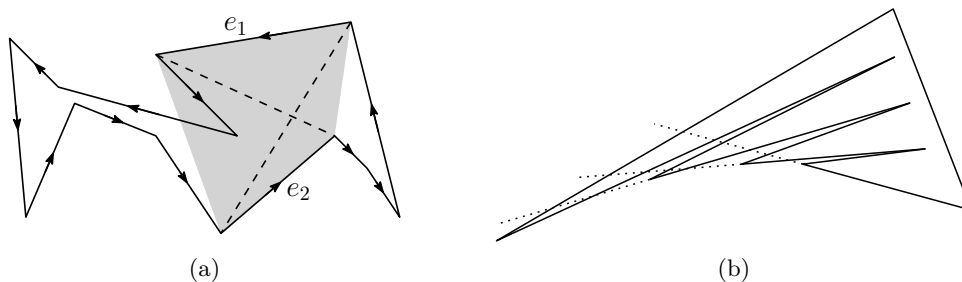
■ **Figure 7** Illustration of our construction of a longest matching.

6 Some Properties of Longest Paths and Cycles

In this section we give some structural properties of longest paths and cycles, possibly of independent interest. We state these properties only for cycles, but they hold for paths as well. Two edges are in *convex position* if they are edges of their convex hull. Two directed edges in convex position have *the same orientation* if they are both directed clockwise or counterclockwise along their convex hull.

► **Observation 15.** *Suppose that we orient the edges of a longest cycle C to make it a directed cycle. Then C cannot have pair of non-adjacent edges that are in convex position and have the same orientation along their convex hull.*

To verify this, note that if C has two such edges, say e_1 and e_2 , then flipping them (replacing e_1 and e_2 by the two diagonals of the convex hull of e_1 and e_2) would produce a longer undirected cycle as in Figure 8(a). Since e_1 and e_2 have the same orientation along their convex hull, the flip does not break the cycle into two components. If every directed simple polygon S contained a pair of non-adjacent edges in convex position with the same orientation along their convex hull, Observation 15 would imply Conjecture 1. However, some simple polygons do not have edges that can be flipped in this way; see e.g., Figure 8(b).



■ **Figure 8** (a) Flipping two edges in convex position. (b) A simple polygon with no pair of edges in convex position that have the same orientation, no matter how we direct the polygon.

► **Observation 16.** *The longest cycle need not contain an edge between diametric points.*

To verify this observation consider an isosceles right triangle abc whose right angle is at b . Place one point at a , one point at c , and two or more points very close to b . Then, the longest cycle does not contain the diametric point pair $\{a, c\}$. This observation implies that a longest cycle may not be achieved by greedily choosing longest edges.

The following proposition implies that if the longest cycle is noncrossing, it contains some edge whose length is among the smallest three-quarters of all distances defined by its vertices.

► **Proposition 17.** *Let S be a simple polygon (a noncrossing cycle) on n points. Then S has an edge whose length is among the smallest $3n^2/8 + n/8$ distances of the $\binom{n}{2}$ point pairs.*

Proof. Let e and e' be two edges of S such that their distance along S (in terms of the number of edges) is at least 2. Since S is a simple polygon, e and e' do not cross. Thus, there is an endpoint p of e and an endpoint p' of e' such that $|pp'|$ is larger than the length of the shorter of e and e' , and pp' is not an edge of S . The number of pairs of edges at distance 0 is n , and the number of pairs of edges at distance 1 is also n . Thus, the total number of pairs of edges at a distance at least 2 is $\binom{n}{2} - 2n$. Each such pair of edges yields a pair $\{p, p'\}$. Each $\{p, p'\}$ can be counted for 4 different pairs of edges that are obtained by combining the two edges incident to p and the two edges incident to p' . Therefore the total number of distinct pairs $\{p, p'\}$ is at least $\frac{1}{4}(\binom{n}{2} - 2n)$. Subtracting this from the total number $\binom{n}{2}$ of point pairs yields the claimed bound. ◀

References

- 1 Alok Aggarwal, Don Coppersmith, Sanjeev Khanna, Rajeev Motwani, and Baruch Schieber. The angular-metric traveling salesman problem. *SIAM Journal on Computing*, 29(3):697–711, 1999. doi:10.1137/S0097539796312721.
- 2 Oswin Aichholzer, Sergio Cabello, Ruy Fabila Monroy, David Flores-Peñaloza, Thomas Hackl, Clemens Huemer, Ferran Hurtado, and David R. Wood. Edge-removal and non-crossing configurations in geometric graphs. *Discrete Mathematics & Theoretical Computer Science*, 12(1):75–86, 2010. doi:10.46298/dmtcs.525.
- 3 Noga Alon, Sridhar Rajagopalan, and Subhash Suri. Long non-crossing configurations in the plane. *Fundam. Inform.*, 22(4):385–394, 1995. doi:10.3233/FI-1995-2245.
- 4 Jose Luis Álvarez-Rebollar, Jorge Cravioto-Lagos, Nestaly Marín, Oriol Andreu Solé-Pi, and Jorge Urrutia. Crossing and intersecting families of geometric graphs on point sets. *Graphs Comb.*, 40(1):17, 2024. doi:10.1007/S00373-023-02734-9.
- 5 José Luis Álvarez-Rebollar, Jorge Cravioto-Lagos, and Jorge Urrutia. Crossing families and self crossing Hamiltonian cycles. In *Abstracts of XVI Encuentros de Geometría Computacional*, page 13, 2015.
- 6 Hyung-Chan An, Robert Kleinberg, and David B. Shmoys. Approximation algorithms for the bottleneck asymmetric traveling salesman problem. *ACM Trans. Algorithms*, 17(4):35:1–35:12, 2021. doi:10.1145/3478537.
- 7 Esther M. Arkin, Yi-Jen Chiang, Joseph S. B. Mitchell, Steven Skiena, and Tae-Cheon Yang. On the maximum scatter traveling salesperson problem. *SIAM Journal on Computing*, 29(2):515–544, 1999. doi:10.1137/S0097539797320281.
- 8 Boris Aronov, Paul Erdős, Wayne Goddard, Daniel J. Kleitman, Michael Klugerman, János Pach, and Leonard J. Schulman. Crossing families. *Combinatorica*, 14(2):127–134, 1994. doi:10.1007/BF01215345.
- 9 Sanjeev Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998. doi:10.1145/290179.290180.

- 10 Alexander Barvinok, Edward Kh. Gimadi, and Anatoliy I. Serdyukov. The maximum TSP. In Gregory Gutin and Abraham P. Punnen, editors, *The Traveling Salesman Problem and Its Variations*, pages 585–607. Springer, Boston, MA, 2007. doi:10.1007/0-306-48213-4_12.
- 11 Ahmad Biniáz. Acute tours in the plane. In *Proc. 38th International Symposium on Computational Geometry (SoCG)*, volume 224 of *LIPICs*, pages 16:1–16:8. Schloss Dagstuhl, 2022. doi:10.4230/LIPICs.SOCG.2022.16.
- 12 Ahmad Biniáz. Euclidean bottleneck bounded-degree spanning tree ratios. *Discrete & Computational Geometry*, 67(1):311–327, 2022. doi:10.1007/s00454-021-00286-4.
- 13 Ahmad Biniáz. Improved bounds for covering paths and trees in the plane. In *Proc. 39th International Symposium on Computational Geometry (SoCG)*, volume 258 of *LIPICs*, pages 19:1–19:15. Schloss Dagstuhl, 2023. doi:10.4230/LIPICs.SOCG.2023.19.
- 14 Ahmad Biniáz, Prosenjit Bose, Kimberly Crosbie, Jean-Lou De Carufel, David Eppstein, Anil Maheshwari, and Michiel H. M. Smid. Maximum plane trees in multipartite geometric graphs. *Algorithmica*, 81(4):1512–1534, 2019. doi:10.1007/S00453-018-0482-X.
- 15 Sergio Cabello, Michael Hoffmann, Katharina Klost, Wolfgang Mulzer, and Josef Tkadlec. Long plane trees. In *Proc. 38th International Symposium on Computational Geometry (SoCG)*, volume 224 of *LIPICs*, pages 23:1–23:17. Schloss Dagstuhl, 2022. doi:10.4230/LIPICs.SOCG.2022.23.
- 16 Jakub Cerný, Zdenek Dvorák, Vít Jelínek, and Jan Kára. Noncrossing Hamiltonian paths in geometric graphs. *Discrete Applied Mathematics*, 155(9):1096–1105, 2007. doi:10.1016/j.dam.2005.12.010.
- 17 Adrian Dumitrescu, Dániel Gerbner, Balázs Keszegh, and Csaba D. Tóth. Covering paths for planar point sets. *Discrete & Computational Geometry*, 51(2):462–484, 2014. doi:10.1007/s00454-013-9563-4.
- 18 Adrian Dumitrescu, János Pach, and Géza Tóth. Drawing Hamiltonian cycles with no large angles. *The Electronic Journal of Combinatorics*, 19(2):P31, 2012. doi:10.37236/2356.
- 19 Adrian Dumitrescu and Csaba D. Tóth. Long non-crossing configurations in the plane. *Discrete & Computational Geometry*, 44(4):727–752, 2010. doi:10.1007/s00454-010-9277-9.
- 20 Sándor P. Fekete. Simplicity and hardness of the maximum traveling salesman problem under geometric distances. In *Proc. 10th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 337–345, 1999. URL: <http://dl.acm.org/citation.cfm?id=314500.314586>.
- 21 Sándor P. Fekete and Gerhard J. Woeginger. Angle-restricted tours in the plane. *Computational Geometry: Theory and Applications*, 8:195–218, 1997. doi:10.1016/S0925-7721(96)00012-0.
- 22 Ming-Yang Kao and Manan Sanghi. An approximation algorithm for a bottleneck traveling salesman problem. *J. Discrete Algorithms*, 7(3):315–326, 2009. doi:10.1016/J.JDA.2008.11.007.
- 23 Joseph S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k -MST, and related problems. *SIAM J. Comput.*, 28(4):1298–1309, 1999. doi:10.1137/S0097539796309764.
- 24 János Pach, Natan Rubin, and Gábor Tardos. Planar point sets determine many pairwise crossing segments. *Advances in Mathematics*, 386:107779, 2021. doi:10.1016/j.aim.2021.107779.
- 25 Christos H. Papadimitriou. The Euclidean traveling salesman problem is NP-complete. *Theoretical Computer Science*, 4(3):237–244, 1977. doi:10.1016/0304-3975(77)90012-3.
- 26 Clifford Stein and David P. Wagner. Approximation algorithms for the minimum bends traveling salesman problem. In *Proceedings of the 8th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, volume 2081 of *LNCS*, pages 406–422. Springer, 2001. doi:10.1007/3-540-45535-3_32.
- 27 Helge Tverberg. A separation property of plane convex sets. *Mathematica Scandinavica*, 45:255–260, 1979. doi:10.7146/math.scand.a-11840.

Rectilinear Crossing Number of Graphs Excluding a Single-Crossing Graph as a Minor

Vida Dujmović  

University of Ottawa, Canada

Camille La Rose 

University of Ottawa, Canada

Abstract

The rectilinear crossing number of G is the minimum number of crossings in a straight-line drawing of G . A single-crossing graph is a graph whose crossing number is at most one. We prove that every n -vertex graph G that excludes a single-crossing graph as a minor has rectilinear crossing number $O(\Delta n)$, where Δ is the maximum degree of G . This dependence on n and Δ is best possible. The result applies, for example, to K_5 -minor-free graphs, and bounded treewidth graphs. Prior to our work, the only bounded degree minor-closed families known to have linear rectilinear crossing number were bounded degree graphs of bounded treewidth as well as bounded degree $K_{3,3}$ -minor-free graphs. In the case of bounded treewidth graphs, our $O(\Delta n)$ result is again tight and it improves on the previous best known bound of $O(\Delta^2 n)$ by Wood and Telle, 2007.

2012 ACM Subject Classification Mathematics of computing → Graph theory; Mathematics of computing → Graphs and surfaces; Theory of computation → Computational geometry

Keywords and phrases (rectilinear) crossing number, graph minors, maximum degree, clique-sums

Digital Object Identifier 10.4230/LIPIcs.GD.2024.37

Related Version *Previous Version*: <https://arxiv.org/abs/2402.15034>

Funding *Vida Dujmović*: Funded by NSERC and a University of Ottawa Research Chair.

Camille La Rose: Funded by NSERC.

1 Introduction

In this article graphs are undirected, simple, and finite, unless stated otherwise. For a graph G , with vertex set $V(G)$ and edge set $E(G)$, let $|G| := |V(G)|$ and $\|G\| := |E(G)|$.¹

A *drawing* of a graph represents each vertex by a distinct point in the plane, and represents each edge by a simple closed curve between its endpoints, such that the only vertices an edge intersects are its own endpoints, and no three edges intersect at a common point (except at a common endpoint). A drawing is *rectilinear* if each edge is a line segment.² A *crossing* is a point of intersection between two edges (other than a common endpoint). A drawing with no crossings is *crossing-free*. A graph is *planar* if it has a crossing-free drawing. The *crossing number* of a graph G , denoted by $\text{cr}(G)$, is the minimum number of crossings in any drawing of G . The *rectilinear crossing number* of a graph G , denoted by $\overline{\text{cr}}(G)$, is the minimum number of crossings in any rectilinear drawing of G .

Crossing number is a fundamental and extensively studied graph parameter with wide ranging applications and rich history (see the survey by Schaefer [37] for over 700 references on the crossing number and its variants). Computationally the problem of determining the

¹ For each vertex v of G , let $N_G(v) := \{w \in V(G) : vw \in E(G)\}$ be the neighbourhood of v in G . The *degree* of v , denoted by $\text{deg}_G(v)$, is $|N_G(v)|$. Let $\Delta(G)$ be the maximum degree of G . When the graph is clear from the context, we will sometimes write $\text{deg}(v)$ instead of $\text{deg}_G(v)$ and Δ instead of $\Delta(G)$.

² Rectilinear drawings are also known as *straight-line drawings* in the literature.



© Vida Dujmović and Camille La Rose;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 37; pp. 37:1–37:18

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

crossing number of a given graph, is notoriously difficult. Computing the crossing number is NP-hard by Garey and Johnson [19], even for planar graph plus an edge [8]. It is hard to approximate even for cubic graphs [7] and until recently there were no approximation algorithms with sub-polynomial in n approximation factor even for bounded degree graphs [10]. On the positive side, Kawarabayashi and Reed [23] give an $O(f(k) \cdot |G|)$ algorithm for deciding whether a given graph G has crossing number at most k .

Since computing the exact, or even an asymptotic, crossing number of a graph is hard, a great deal of past research has been focused on deriving asymptotic bounds. Regardless of the applications, be it visualization or circuit design [4, 26, 27], having as few crossings as possible is a desirable property in a drawing of a graph. This naturally leads to a study on upper bounds and lower bounds on the crossing number of various graph families.

Trivially, the (rectilinear) crossing number of every graph G is at most $O(\|G\|^2)$. For some graphs this bound is asymptotically tight, including for example, the complete graph and a random cubic graph. For others, it is far from optimal. We would like to understand what classes of graphs have rectilinear crossing number better than $O(\|G\|^2)$ and by how much. To understand that, we consider next the known lower bounds on the crossing number, which clearly apply to rectilinear crossing number.

Every planar triangulation G is known to have $3|G| - 6$ edges (if $|G| \geq 3$). Consequently, for every $n \geq 3$, there are n -vertex graphs with $3n - 6$ edges that can be drawn with zero crossings. The following result, known as the *crossing lemma*, tells us that as soon as a graph has a little more than $3|G|$ edges, it must have vastly more than zero crossings in every drawing. Specifically, the crossing lemma, proposed by Erdős and Guy [18], proved by Leighton [28] and Ajtai et al. [3]; and, subsequently improved [2, 29, 31] states the following.

► **Theorem 1.** *For any $\epsilon > 0$, there exists c_ϵ such that, every graph G with $\|G\| > (3 + \epsilon) \cdot |G|$ edges, $\text{cr}(G) \geq c_\epsilon \frac{\|G\|^3}{|G|^2}$*

An immediate consequence of this theorem is that all graphs G that have at least $(3 + \epsilon) \cdot |G|$ edges have crossing number at least $\Omega(|G|)$. We say that a family of graphs has a *linear crossing number* if there exists a constant c such that every graph G in the family has $\text{cr}(G) \leq c \cdot |G|$. The crossing lemma tells us that for most graph families the best one can hope of is that they have linear crossing number. By the crossing lemma, any family of graphs that has members with the number of edges superlinear in the number of vertices cannot have linear crossing number. Thus the only candidates for linear crossing number are families of graphs whose members all have a linear number of edges. One example of such a family is a family of graphs whose members have the maximum degree bounded by a constant. Another example is a family of graphs that exclude some fixed graph H as a minor.³ Kostochka [24] and Thomason [39] proved independently that H -minor free graphs G have $O(|H| \sqrt{\log |H|} \cdot |G|)$, and thus a linear, number of edges.

It turns out that neither of these two families of graphs have linear crossing number. Consider, for example, the graph $K_{3,n}$. It has a linear number of edges and it is K_5 -minor-free, yet it is known to have crossing number $\Omega(n^2)$ [30, 32]. Similarly, consider the family of all cubic graphs. They have a linear number of edges and yet, for every large enough n , there is a cubic n -vertex graph whose crossing number is $\Omega(n^2)$ [25, 13, 14, 28].

³ Let vw be an edge of a graph G . Let G' be the graph obtained by identifying the vertices v and w , deleting loops, and replacing parallel edges by a single edge. Then G' is obtained from G by *contracting* vw . A graph X is a *minor* of a graph G if X can be obtained from a subgraph of G by contracting edges. A graph G is *X -minor-free* if X is not a minor of G . A family of graphs \mathcal{F} is *minor-closed* if $G \in \mathcal{F}$ implies that every minor of G is in \mathcal{F} . \mathcal{F} is *proper* if it is not the family of all graphs.

Thus to admit a linear crossing number, it is not enough for a family of graphs to have a bounded degree only or to only exclude a fixed graph as a minor. Having both of these properties however is enough. Böröczky et al. [22] first showed such a result for bounded Euler genus graphs (see Theorem 2). Note that by the above mentioned result by Kostochka [24] and Thomason [39], for every graph G from a proper minor closed family of graphs $\|G\| \in O(|G|)$. That fact will be used throughout this article, starting with the next theorem.

► **Theorem 2** ([1, 22]). *For every integer $\gamma \geq 0$, there is a function f such that every graph G with Euler genus γ has crossing number $\text{cr}(G) \leq f(\gamma) \cdot \sum_{v \in V(G)} \deg(v)^2 \leq 2 \cdot f(\gamma) \cdot \Delta(G) \cdot \|G\| \in O(f(\gamma) \cdot \Delta(G) \cdot |G|)$*

An improvement on the dependence on γ in Theorem 2 for orientable surfaces was shown by Djidjev and Vrto [15], with $\text{cr}(G) \leq c \cdot \gamma \cdot \Delta(G) \cdot |G|$ for some constant c . Wood and Telle [41] were the first to show that excluding a minor and bounding the maximum degree were sufficient to ensure a linear crossing number, as stated in the next theorem.

► **Theorem 3** ([41]). *For every graph H , there is a constant $c := c(H)$ such that every H -minor-free graph G has crossing number $\text{cr}(G) \leq c \cdot \Delta(G)^2 \cdot |G|$*

Theorem 3 was improved by Dujmović et al. [16] by reducing $\Delta(G)^2$ to $\Delta(G)$.

► **Theorem 4** ([16]). *For every graph H , there is a constant $c := c(H)$ such that every H -minor-free graph G has crossing number $\text{cr}(G) \leq c \cdot \Delta(G) \cdot |G|$*

In addition, the result in Theorem 4 was shown to have the best possible dependence of $\Delta(G)$ and $|G|$. These results show that we know very strong, in fact best possible, bounds on the crossing number of all proper minor-closed families of graphs of bounded degree.

Much less is known about the rectilinear crossing number. Fáry [21] and Wagner [40] proved independently that every planar graph has a rectilinear drawing with no crossings. Hence, every planar graph G has the rectilinear crossing number 0, and thus for planar graphs G , $\overline{\text{cr}}(G) = \text{cr}(G)$. One may be tempted to conjecture that the rectilinear crossing number and crossing number are tied. However, that is not the case. In particular, Bienstock and Dean [5] proved that for every m and every $k \geq 4$, there exists a graph G with $\text{cr}(G) = k$, but $\overline{\text{cr}}(G) \geq m$. Therefore, Theorem 3 and Theorem 4 do not imply that bounded degree proper minor-closed families of graphs have linear rectilinear crossing number.

In fact, in addition to planar graphs, we are only aware of the following two minor-closed families of bounded degree admitting linear rectilinear crossing number. The first is the result on $K_{3,3}$ -minor-free graphs by Dujmović et al. [16].

► **Theorem 5** ([16]). *Every $K_{3,3}$ -minor-free graph G has rectilinear crossing number $\overline{\text{cr}}(G) \leq \sum_{v \in V(G)} \deg(v)^2 \leq 2 \cdot \Delta(G) \cdot \|G\| \in O(\Delta(G) \cdot |G|)$*

The second is a result on the convex crossing number of bounded treewidth graphs by Wood and Telle [41]. Rectilinear drawings where vertices are required to be in convex positions are called *convex drawings*. For a graph G , the minimum number of crossings over all convex drawings of G is called *convex crossing number* of G and is denoted by $\text{cr}^*(G)$. Clearly, for every G , $\text{cr}(G) \leq \overline{\text{cr}}(G) \leq \text{cr}^*(G)$.

► **Theorem 6** ([41]). *Every graph G of treewidth k has convex crossing number $\text{cr}^*(G) \in O(k^2 \cdot \Delta(G)^2 \cdot \|G\|) \in O(k^3 \cdot \Delta(G)^2 \cdot |G|)$*

In the case of the rectilinear crossing number a stronger bound is known but still with a quadratic dependence on $\Delta(G)$ in the worst case.

► **Theorem 7** ([16]). *Every graph G of treewidth k has rectilinear crossing number $\overline{\text{cr}}(G) \in O(k \cdot \Delta(G) \cdot \sum_{v \in V(G)} \deg(v)^2) \in O(k \cdot \Delta(G)^2 \cdot |G|)$*

Our goal in this article is to extend this result to much wider minor-closed families of graphs of bounded degree, in particular, a family of graphs that exclude a single-crossing graph as a minor. A *single-crossing graph* is a graph whose crossing number is at most one. Single-crossing minor-free graphs have been studied by the algorithms community [17, 12, 9] where at times these results were precursors to algorithms and techniques applicable to more general minor-closed classes [12, 11]. $K_{3,3}$, K_5 and every planar graph are examples of single-crossing graphs. Note however that a minor of a single-crossing graph is not necessarily a single-crossing graph itself (see [12] for easy examples). Note finally that a graph excluding a single-crossing graph as a minor may have arbitrarily large crossing number. For example, any n -vertex graph G , composed of disjoint union of $\lfloor \frac{n}{6} \rfloor$ copies of $K_{3,3}$, excludes K_5 as a minor (K_5 is a single-crossing graph) and yet the crossing number of G is $\Theta(n)$.

The following theorem is our main result.

► **Theorem 8.** *Let X be a single-crossing graph. There exists a constant $c := c(X)$, such that every X -minor-free graph G has the rectilinear crossing number of at most $c \cdot \Delta(G) \cdot |G|$.*

The dependence on Δ and $|G|$ in the above theorem is best possible. A standard lower bound constructions implies it (see for example [22, 41]). Specifically, consider a graph comprised of the disjoint union of $\Omega(n/\Delta)$ copies of $K_{3,3}$ where each $K_{3,3}$ is transformed into a maximum degree Δ graph by adding $\Omega(\Delta)$ paths of length two between every pair of adjacent vertices in each copy of $K_{3,3}$. This graph has maximum degree Δ and is still K_5 -minor-free (and thus single-crossing minor-free) and yet has crossing number $\Omega(\Delta \cdot n)$. It also has treewidth at most 5. Thus the following two corollaries of Theorem 8 are both tight. Since K_5 is a single-crossing graph, the following is an immediate corollary of Theorem 8.

► **Corollary 9.** *There exists a constant c such that every K_5 -minor-free graph G has a rectilinear crossing number of at most $c \cdot \Delta(G) \cdot |G|$.*

It is known that the family of graphs of treewidth at most k excludes a planar grid of size k^c as a minor (for some constant c) [35]. Since every planar graph is a single-crossing graph, Theorem 8 implies the following result.

► **Corollary 10.** *For every integer $k > 0$, there exists a constant c_k such that every graph G of treewidth at most k has a rectilinear crossing number at most $c_k \cdot \Delta(G) \cdot |G|$.*

In the process of proving the main result, Theorem 8, we establish a more precise upper bound for treewidth- k graphs: $k \cdot (k + 2) \cdot \Delta(G) \cdot ||G||$. See Theorem 19.

This corollary improves the previous best known bound on rectilinear crossing number of bounded treewidth graphs from $O(\Delta(G)^2 \cdot |G|)$ (see Theorems 6 and 7 above) to the optimal $O(\Delta(G) \cdot |G|)$ bound. It should be noted however that Theorem 6 by Wood and Telle [41] gives an $O(\Delta(G)^2 \cdot |G|)$ bound for the convex crossing number of bounded treewidth graphs, and that bounds still stands as the best known for convex drawings.

In the next section, Section 2, we introduce key concepts that will be helpful in proving Theorem 8. In Section 3, we present the proof of Theorem 8. Finally, we conclude in Section 4.

2 Preliminaries

Recall that the definition of drawing of a graph has the following condition: “no three edges intersect at a common point (except at a common endpoint)”. In relation to this, we define a set of points P to be in *general position* if no three points of P lie on one line and if no three line segments between pairs of points in P intersect in one point unless all three share a common endpoint. For the ease of presentation we will add to the definition of rectilinear drawings the condition that all endpoints of G are in general position.

2.1 Multigraphs

Our proof of the main result, Theorem 8, will require the use of multigraphs. Recall that a multigraph is a graph that may have parallel edges but no loops. For the remainder of this paper, we will use the term “multigraph” when parallel edges are allowed, and “graph” when they are not, i.e., when the graph is simple. The degree of a vertex v in a multigraph Q , denoted by $\deg_Q(v)$, is the number of edges of Q incident to v . However, unlike in simple graphs, $\deg_Q(v)$ is not necessarily equal to $|N_Q(v)|$.

A *rectilinear drawing* of a multigraph Q represents vertices, $V(Q)$, by a set of $|V(Q)|$ points in the plane in general position and represents each edge by a line segment between its endpoints. The general position assumption implies that the only vertices an edge intersects are its own endpoints, and no point in the drawing is in three *distinct* line segments (unless all three share a common endpoint). It should be noted that parallel edges between the same pair of vertices in such a drawing overlap, as they are represented by the same line segment. A *crossing-pair* is a pair of edges in a rectilinear drawing that do not have a common endpoint and whose line segments intersect at a common point. The number of crossings in a rectilinear drawing of a multigraph is the number of crossing-pairs in the drawing. The *rectilinear crossing number* of a multigraph Q , denoted by $\overline{cr}(Q)$, is the minimum number of crossings over all rectilinear drawings of Q .

Note that by these definitions, a pair of overlapping edges in a rectilinear drawing of a multigraph is not considered a crossing-pair. This is because, in our main proof, we start with a rectilinear drawing of a certain multigraph Q . We eventually replace parallel edges (and their overlapping line segments) with incident edges (and thus line segments that have one endpoint in common) and such edges can never cross. Notice also that if one is allowed to replace line segments by arcs in a rectilinear drawing of a multigraph Q , then it is trivial to redraw Q such that the resulting “arc” drawing of Q has no overlapping edges and has the same number of crossings as the starting rectilinear drawing of Q . Finally, if Q is a simple graph, these definitions of rectilinear drawing and rectilinear crossing number are equivalent to the earlier ones for simple graphs.

2.2 Decompositions and Treewidth

For graphs G and H , an H -*decomposition* of G is a collection $(B_x \subseteq V(G) : x \in V(H))$ of sets of vertices in G (called *bags*) indexed by the vertices of H , such that

1. for every edge vw of G , some bag B_x contains both v and w , and
2. for every vertex v of G , the set $\{x \in V(H) : v \in B_x\}$ induces a non-empty connected subgraph of H .

The *width* of a decomposition is the size of the largest bag minus 1. The *adhesion* of a decomposition is the size of the largest intersection between two bags that share an edge in H . If H is a tree, then an H -decomposition is called a *tree decomposition*. The *treewidth* of

a graph G is the minimum width of any tree decomposition of G . Tree decomposition and treewidth are key concepts in graph minor structure theory and they have been extensively studied ever since their introduction by Halin [20] and by Robertson and Seymour [34].

2.3 Rectilinear Drawings

In the process of proving our main result, Theorem 8 in Section 3.1, we will construct drawings of graphs where at one stage we will replace a point (representing some vertices) with disks that fulfill certain criteria. The following lemma will be helpful for that stage.

For any positive integer h , let $[h]$ denote the sequence of numbers $[1, \dots, h]$. When it is clear from the context, we will make no distinction between a vertex v of a graph and the point that represents it in a drawing. Specifically, we will refer to both as v when no confusion can arise. The same will be true for an edge e and the line segment representing it in a drawing.

► **Lemma 11.** *Let D be a rectilinear drawing of any graph G . Then for each vertex $w \in V(G)$, there exists a disk C_w of positive radius centered at w such that the following is true. Let v_1, \dots, v_d be the neighbours of w in G . Let P_w be any set of at most d points in C_w such that $V(G) \cup P_w$ is in general position. For each $i \in [d]$, replace the line segment $\overline{wv_i}$ of D by a line segment between v_i and any point in P_w . Denote that point by p_i . The resulting drawing D' (of the resulting graph G') has the following properties:*

1. Any two edges in G , neither of which is incident to w , cross in D' if and only if they cross in D .
2. For each $i \in [d]$, the edge wv_i and any edge xy where $\{x, y\} \subseteq V(G) - \{w, v_i\}$ cross in D if and only if $v_i p_i$ and xy cross in D' .
3. All the remaining crossings in D' are crossings between pairs of segments with distinct endpoints in P_w .

It should be noted that if $|P_w| = 1$, that is if P_w has exactly one point, then D' is a rectilinear drawing of G where a pair of edges of G cross in D' if and only if they cross in D .

Proof. Start with the drawing D of G and a disk C centered at w such that the only parts of D that intersect C are w and the edges incident to w . Then, for each $i \in [d]$, let S_i be the union of all possible line segments from v_i to any point in C . Let S denote the union of all S_i , $i \in [d]$. By reducing the radius of C to some positive radius r and then redefining S accordingly, the following becomes true for D and C .

- No vertex of G is in S other than w, v_1, \dots, v_d .
- For each $i \in [d]$, the only vertices of G that are in S_i are v_i and w .
- For each $i \in [d]$, the only crossing points of D in S_i are crossings between wv_i and the edges not incident to w in G .
- No segment between two crossings in D is fully contained in S , unless it lies on one of the edges wv_i , $i \in [d]$.

Such a positive radius r exists by continuity and the resulting disk meets the conditions imposed on C_w . ◀

3 Main Result

In order to prove our main result, Theorem 8, we will use, as one of the tools, Robertson and Seymour's structure theorem for graphs that exclude a single-crossing graph as a minor [33]. This structure theorem uses the notion of clique-sum, that we define next. Let G_1 and G_2 be

two disjoint graphs. Let $C_1 = \{v_1, v_2, \dots, v_k\}$ be a clique in G_1 and $C_2 = \{w_1, w_2, \dots, w_k\}$ be a clique in G_2 , each of size k , for some integer $k \geq 1$. Let G be a graph obtained from G_1 and G_2 by identifying v_i and w_i for each $i \in [k]$ and possibly deleting some of the edges $u_i u_j$ in the resulting clique $C = \{u_1, u_2, \dots, u_k\}$ of G . Then we say that G is *obtained by k -clique-sums* of graphs G_1 and G_2 (at C_1 and C_2). A $(\leq k)$ -clique-sum is an l -clique-sum for any $l \leq k$. The following theorem by Robertson and Seymour [33] describes a structure of graphs that exclude a single-crossing graph as a minor.⁴

► **Theorem 12** ([33]). *For every single-crossing graph X , there exists a positive integer $t := t(|X|)$ such that if G is an X -minor-free graph, then G can be obtained by (≤ 3) -clique-sums of graphs G_1, \dots, G_h such that for each $i \in [h]$, G_i is a planar graph (with no separating triangles) or the treewidth of G_i is at most t .*

The graphs G_1, \dots, G_h in Theorem 12 are called the *pieces* of the decomposition.

Theorem 12 is equivalent to stating that every X -minor-free graph G has a tree decomposition of adhesion at most 3 such that the vertices in each bag of the decomposition induce in G either a planar graph (with no separating triangles) or a graph of treewidth at most t . Armed with these notions, we are now ready to state a more precise version of our main result.

► **Theorem 13.** *Let X be a single-crossing graph. Let G be an X -minor-free graph and let $t := t(|X|)$ be the integer from Theorem 12. Then $\overline{\text{cr}}(G) \leq 3 \cdot (t^2 + 2t + 2) \cdot \Delta(G) \cdot \|G\|$.*

Theorem 13 is a strengthened version of Theorem 8 by Theorem 12 and the fact that $\|G\| \in O(|X| \sqrt{\log |X|} \cdot |G|)$ [24, 39] (as discussed earlier). Hence, the remainder of this section will be dedicated to proving Theorem 13. To do so, one has to be able to produce rectilinear drawings of the pieces, G_1, \dots, G_h , of the decomposition (from Theorem 12) with the claimed number of crossings and then combine these drawings by conducting clique-sums. The following is a sketch of the two main steps our proof will take.

Step 1. Foremost, Theorem 13 has to be true for the pieces G_i of the decomposition, namely the planar graphs and bounded treewidth graphs. By the Fáry-Wagner theorem [21, 40], we know that Theorem 13 is true for all planar graphs. In fact, it is true with bound zero for the rectilinear crossing number. On the contrary, if G_i is a bounded treewidth graph, the required $O(\Delta(G_i) \cdot |G_i|)$ bound on its rectilinear crossing number was not known prior to our work. Thus one of the goals of this paper is to prove that bound for bounded treewidth graphs as one of the necessary steps in the proof of Theorem 13.

Step 2. Suppose now that for each piece, G_i of the decomposition, we have already established the $O(\Delta(G_i) \cdot |G_i|)$ bound for the rectilinear crossing of G_i . The main goal then becomes demonstrating that the rectilinear drawings of G_1, G_2, \dots, G_h can be joined by performing clique-sums without increasing the number of crossings in the final drawing of G by too much. Prior to this work it was not known how to conduct clique-sums on rectilinear drawings while achieving that goal. In particular, we need to join rectilinear drawings of G_1, G_2, \dots, G_h in such a way that the resulting number of crossings in the rectilinear drawing of G is $O(\Delta(G) \cdot |G|)$.

⁴ Note that the original statement of Theorem 12 by Robertson and Seymour [36] does not mention separating triangles. The reason such a statement can be made is that any planar graph G containing a separating triangle can itself be obtained by 3-clique-sums of two strictly smaller planar graphs, G_1 and G_2 , where the clique-sum is performed on that separating triangle.

The main challenge for proving Theorem 13 is Step 2 above. To overcome that challenge, we introduce the notion of simplicial blowups of graphs. The use of these simplicial blowups however impacts Step 1. In particular, it is not enough anymore to prove that the pieces of the decomposition have $O(\Delta(G_i) \cdot |G_i|)$ rectilinear crossing number. We must prove a stronger condition, namely that the simplicial blowups of the pieces have such a rectilinear crossing number.

In Section 3.1 we introduce simplicial blowups and demonstrate how to achieve Step 2. In Section 3.2 we introduce graph partitions and present a helpful lemma for producing rectilinear drawings. In Section 3.3 and 3.4, we then prove that Step 1 above can be accomplished, or more precisely that simplicial blowups of planar graphs and bounded treewidth graphs have the desired rectilinear crossing number. Once those two steps have been achieved, we will conclude the proof of Theorem 13 in Section 3.5.

3.1 Bound for Rectilinear Crossing Number Using Clique-Sums

A multigraph Q is called a $(\leq k)$ -*simplicial blowup* of a graph G if Q can be obtained from G by adding an independent set of vertices S to G , and performing the following steps for each vertex u in S :

1. Make u adjacent to all the vertices of some clique of size at most k of G
 2. Add zero or more parallel edges between u and its neighbours in G .
- and finally, once Steps 1 and 2 are conducted on all vertices of S , delete zero or more edges from each clique of G involved in Step 1.

Theorem 14 is the key technical tool of this paper. It shows how rectilinear drawings (of simplicial blowups) of the pieces of a decomposition can be combined into a rectilinear drawing of a graph obtained by clique-sums of the pieces, all while not increasing the final number of crossings by too much. The previous result on the crossing number of minor-closed families (Theorem 4) by Dujmović et al. [16], also had to deal with performing clique-sums on drawings while controlling the crossing number. Our proof of Theorem 14 is inspired by their proof. However the drawings produced by their theorem have many bends per edge and are thus far from rectilinear drawings.

The following theorem is stated in a form that is more general than we will require. Specifically, the theorem does not require the pieces G_i of the decomposition to be planar or of bounded treewidth. As such, Theorem 14 may be useful in future work on rectilinear crossing numbers of X -minor-free graphs where X is not necessarily a single-crossing graph and thus the pieces of the decomposition are the almost embeddable graphs from the Robertson and Seymour graph minor theory.

We say that a graph R is (k, c) -*agreeable* if for every induced subgraph R' of R and every $(\leq k)$ -simplicial blowup R^* of R' , $\overline{cr}(R^*) \leq c \cdot \Delta(R^*) \cdot ||R^*||$.

► **Theorem 14.** *Let c be a positive number, k a positive integer, and G_1, \dots, G_h a collection of graphs such that every G_i is (k, c) -agreeable. Then every graph G that can be obtained by $(\leq k)$ -clique-sums of graphs G_1, \dots, G_h has rectilinear crossing number $\overline{cr}(G) \leq k \cdot (c+2) \cdot \Delta(G) \cdot ||G||$.*

Proof. Since clique-sums identify vertices, to avoid confusion, we will assume that the vertices of the final graph G have names and that each vertex in each piece G_i , $i \in [h]$ inherits its name from G . Thus vertices that are identified by clique-sums have the same name in the pieces involved. Consequently, there may be multiple vertices with the same name in the disjoint union of G_1, G_2, \dots, G_h .

We may assume that the indices $1, \dots, h$ are such that for all $j \geq 2$, there exists a minimum i such that $i < j$ where G_i and G_j are joined at some clique C of G_i when constructing G . We define G_i to be the *parent* of G_j , with $P_j = V(C)$ being the *parent clique* of G_j . The parent clique of G_1 is the empty set. Note that, by the introductory paragraph of this proof, it makes sense to talk about the clique P_j as existing in both in G_i and in G_j .

Let T be a rooted tree with vertex set $\{1, \dots, h\}$, where ij is an edge of T if and only if G_j is a child of G_i . Let T_i denote the subtree of T rooted at i and U_i be the set of the children of i in T .

For each $i \in [h]$, let $G_i^- = G_i - P_i$. Note that for each $v \in V(G)$, there is exactly one $i \in [h]$ such that v is in $V(G_i^-)$. Thus $V(G_1^-), \dots, V(G_h^-)$ is a partition of $V(G)$. We say that a vertex v of G *belongs* to vertex i of T if $v \in G_i^-$. For each $i \in [h]$, let $G[T_i]$ denote the graph induced in G by the vertices of G that belong to the vertices of T_i , that is the graph induced in G by $\bigcup\{V(G_j^-) : j \in T_i\}$.

Defining the ($\leq k$)-simplicial blowups of pieces. To prove the theorem, we now define, for each G_i^- , $i \in [h]$, a specific ($\leq k$)-simplicial blowup, denoted by Q_i^- . To define Q_i^- , start with G_i^- . For each child G_j of G_i , add a new vertex c_j to G_i^- . We call c_j a *dummy* vertex and say that c_j *represents* G_j in Q_i^- . Note that for all $j \in [2, \dots, h]$, there is exactly one $i < j$ such that Q_i^- has a vertex that represents G_j (namely, the vertex c_j). For the clarity of the next statement, note first that $V(G_i^-) \cap P_j$ is not empty as otherwise P_j would also exist in some G_f where $f < i$ and G_j would not be a child of G_i . For each edge $vw \in E(G)$, where $v \in V(G_i^-) \cap P_j$ and $w \in G_\ell^-$, where $i < \ell$ and $\ell \in V(T_j)$, connect v to c_j by an edge. Label that edge with the triple (v, w, \mathcal{P}_{vw}) , where \mathcal{P}_{vw} is the path in T from i to ℓ . We call the edge labelled (v, w, \mathcal{P}_{vw}) in Q_i^- an *isthmus* edge. It represents the edge vw in the final drawing of G . We consequently refer to the edge vw of G as isthmus edge as well. We say that two isthmus edges are *siblings* if they are adjacent to the same dummy vertex. For a vertex u in Q_i^- such that u is in P_j for some child G_j of G_i , we say that u is *involved* in a clique-sum in Q_i^- . Thus each isthmus edge of Q_i^- has an endpoint in G_i^- that is involved in some clique-sum in Q_i^- . We finally remove from Q_i^- the edges in P_j that are not in G . We set the resulting multigraph to be Q_i^- .

Notice that Q_i^- is a ($\leq k$)-simplicial blowup of G_i^- . Since G_i is (k, c) -agreeable (by the assumption) and since G_i^- is an induced subgraph of G_i , it follows that $\overline{cr}(Q_i^-) \leq c \cdot \Delta(Q_i^-) \cdot \|Q_i^-\|$.

For $i \in [h]$, consider a rectilinear drawing of Q_i^- with at most $c \cdot \Delta(Q_i^-) \cdot \|Q_i^-\|$ crossings. We will construct the desired rectilinear drawing of G by joining these rectilinear drawings of Q_i^- . Consider for a moment solely the disjoint union of these rectilinear drawings. The resulting rectilinear drawing of the disjoint union has at most $\sum_{i \in [h]} c \cdot \Delta(Q_i^-) \cdot \|Q_i^-\|$ crossings.

Notice that there is one-to-one mapping between the edges of G and the edges in the union of all $Q_1^-, Q_2^-, \dots, Q_h^-$, that is in $\bigcup_{i \in [h]} E(Q_i^-)$ (where the isthmus edges of G map to the isthmus edges in the union and where the non-isthmus edges of G map to the non-isthmus edges of the union). Thus $\|G\| = \sum_{i \in [h]} \|Q_i^-\|$. Hence, if for all $i \in [h]$, $\Delta(Q_i^-) \leq k \cdot \Delta(G)$, the above sum would be upper bounded by $c \cdot k \cdot \Delta(G) \cdot \sum_{i \in [h]} \|Q_i^-\| = c \cdot k \cdot \Delta(G) \cdot \|G\|$. This is akin to the upper bound that we want on the rectilinear crossing number of G . Thus we want to first bound the degree of each vertex in Q_i^- by $k \cdot \Delta(G)$. This is not completely obvious due to the addition of the dummy vertices in the construction of Q_i^- and also due to the fact that clique-sums allow for edge deletions from the cliques.

▷ **Claim 1.** For every $i \in [h]$ and every $v \in Q_i^-$, $\deg_{Q_i^-}(v) \leq k \cdot \Delta(G)$.

Proof. There are three cases to consider.

Case 1. v is a dummy vertex of Q_i^- .

By construction, for some $j \in U_i$, v represents some G_j and is adjacent to at most k vertices of the parent clique P_j in G_i . Each edge between a vertex $u \in P_j$ and v corresponds to an (isthmus) edge in G adjacent to u . Since $\deg_G(u) \leq \Delta(G)$, v is incident to at most $k \cdot \Delta(G)$ edges, giving $\deg_{Q_i^-}(v) \leq k \cdot \Delta(G)$.

Case 2: v is in G_i^- (and thus not a dummy vertex) and v is not involved in any clique-sums. Then, it follows that $\deg_{Q_i^-}(v) \leq \deg_G(v)$.

Case 3: v is in G_i^- (and thus not a dummy vertex) and is involved in at least one clique-sum.

Consider every $j \in U_i$ such that $v \in P_j$. Then v has a least one neighbour in $G[T_j]$ and thus at least one edge connecting it to c_j , otherwise the clique-sum could have omitted v . Additionally, there exists a one-to-one mapping between the set of edges in G between v and its neighbours in $G[T_j]$ and the set of (parallel isthmus) edges between v and c_j in Q_i^- . In other words, there is a one-to-one mapping between the isthmus edges incident to v in Q_i^- and the isthmus edges incident to v in G . Finally, consider the non-isthmus edges incident to v in G_i^- . Each edge of G_i that has been removed in the construction of Q_i^- (namely the edges removed from P_j) was also removed in G , thus $\deg_{Q_i^-}(v) \leq \deg_G(v)$. \triangleleft

With degrees of the vertices of Q_i^- sorted out, we are ready to describe how to construct a rectilinear drawing of G from the rectilinear drawings of $Q_1^-, Q_2^-, \dots, Q_h^-$.

Constructing the rectilinear drawing of G from the rectilinear drawings of $Q_1^-, Q_2^-, \dots, Q_h^-$.

Since for each $i \in [h]$, Q_i^- is (k, c) -agreeable, $\overline{cr}(Q_i^-) \leq c \cdot \Delta(Q_i^-) \cdot \|Q_i^-\|$. By Claim 1, $\overline{cr}(Q_i^-) \leq c \cdot k \cdot \Delta(G) \cdot \|Q_i^-\|$. Let $D(Q_i^-)$ denote a rectilinear drawing of Q_i^- with at most $c \cdot k \cdot \Delta(G) \cdot \|Q_i^-\|$ crossings. For the remainder of the proof, we will show how to construct a rectilinear drawing, $D(G)$, of G by combining the rectilinear drawings $D(Q_i^-)$ of Q_i^- , $i \in [h]$, such that the resulting number of crossings in $D(G)$ is as claimed in the theorem.

Note that removing dummy vertices (and their incident isthmus edges) from $D(Q_i^-)$ gives a rectilinear drawing of G_i^- . Denote these rectilinear drawings by $D(G_i^-)$. In the final drawing, $D(G)$, the drawing of each G_i^- will be identical to $D(G_i^-)$, possibly scaled and/or rotated. In other words, in $D(G)$, the implied rectilinear drawing of the disjoint union of $G_1^-, G_2^-, \dots, G_h^-$ will be the disjoint union of $D(Q_1^-), D(Q_2^-), \dots, D(Q_h^-)$ without the isthmus edges. The isthmus edges will be redrawn in this construction.

We will join the rectilinear drawings $D(Q_i^-)$, $i \in [h]$ in the order of their indices. For $\ell \in [h]$, D_ℓ denotes the rectilinear drawing obtained by joining $D(Q_1^-), D(Q_2^-), \dots, D(Q_\ell^-)$ (joining is detailed below). The rectilinear drawing D_h will thus be the desired rectilinear drawing $D(G)$ of G . While joining these drawings, we will maintain the invariant that for each $j > \ell$, such that the parent of G_j is some G_i with $i \in [\ell]$, the rectilinear drawing D_ℓ contains the representative dummy vertex (c_j) of each G_j . Furthermore we maintain that D_ℓ minus the dummy vertices (that is $D_\ell - \cup_{j>\ell} c_j$) is isomorphic to $G - \cup_{j>\ell} V(G[T_j])$.

We start by defining $D_1 = D(Q_1^-)$. D_1 satisfies the above invariant. For $j \in [2, \dots, h]$ we construct D_ℓ from $D_{\ell-1}$ and $D(Q_\ell^-)$ as follows. By the invariant, $D_{\ell-1}$ has a dummy vertex c_ℓ representing G_ℓ . Let C_ℓ be a disk centered in the point c_ℓ in $D_{\ell-1}$ that meets the conditions of Lemma 11. Let v_1, v_2, \dots, v_d be the neighbours of c_ℓ in $D_{\ell-1}$. Construct D_ℓ by the following steps.

1. Remove c_ℓ and its incident (isthmus) edges.
2. Scale down $D(Q_\ell^-)$. Place it inside C_ℓ and rotate it such that all the vertices of D_ℓ are in general position.
3. For each isthmus edge labelled with (x, y, P_{xy}) that was incident to c_ℓ , (re)draw it as the line segment from x to y if y in Q_ℓ^- . Otherwise, by construction, $D(Q_\ell^-)$ has a point c_j , $j > \ell$ and $y \in G[T_j]$. In that case, draw a line segment between x and c_j . By Lemma 11, the only new crossings (pairs) that this introduces are crossings between (a) a pair of (re)drawn sibling isthmus edges (that were both incident to c_ℓ) or (b) one such isthmus edge (incident to c_ℓ) and edges strictly inside the disk C_ℓ (that is, edges in G_ℓ^-).

The resulting drawing D_ℓ satisfies the invariant. Note that at the end of this process, when $\ell = h$, there are no more dummy vertices and each edge labelled (x, y, P_{xy}) in D_h is an actual line segment connecting vertex x and y in G and thus actually represents the isthmus edge xy of G . The final drawing D_h is a rectilinear drawing $D(G)$ of G . It remains to prove that $D(G)$ has the claimed number of crossings.

Before joining the drawings $D(Q_i^-)$, $i \in [h]$, the total number of crossings in the disjoint union of all drawings was at most $c \cdot k \cdot \Delta(G) \cdot \|G\|$, as argued earlier. We name this quantity the *initial sum*. We now prove that joining these drawings into a drawing of G does not increase the initial sum by much. Specifically, we will show that all new crossings can be charged to the edges of G such that each edge is charged at most $2 \cdot k \cdot \Delta(G)$ new crossings, which will complete the proof.

By the construction, the new crossings must involve at least one isthmus edge. Consider such an isthmus edge e labelled (v, w, P_{vw}) , where $v \in Q_i^-$ and $w \in Q_p^-$, $i < p$ and $w \in G[T_j]$ where $j \in U_i$ (and thus $p \in T_j$). There are four cases to consider.

Case 1. Consider first a crossing in $D(G)$ between e and a non-isthmus edge e' in Q_i^- . That crossing is already accounted for in the initial sum by the crossing in $D(Q_i^-)$ between e' and edge vc_j labelled (v, w, P_{vw}) .

Case 2. Consider next a crossing between e and an isthmus edge e' labelled (x, y, P_{xy}) , where $x \in Q_i^-$ and $y \in G[T_r]$, with $r \in U_i$. 2a) If $r \neq j$ (so e and e' are not sibling isthmus edges), then crossing between e and e' was accounted for as well in the initial sum by the crossing in $D(Q_i^-)$ between the edge vc_j labelled (v, w, P_{vw}) and the edge xc_r labelled (x, y, P_{xy}) . 2b) If $r = j$, it must be that $v \neq x$ as otherwise e and e' cannot cross. By construction both w and y are in the disk C_r . In the construction of G , $G[T_j]$ is added via a ($\leq k$)-clique-sum to G_i (with parent clique P_j). Thus at most $k \cdot \Delta(G)$ (isthmus) edges cross the cycle bounding C_r . Thus e can be crossed by at most $k\Delta(G)$ such edges e' . We charge these at most $k \cdot \Delta(G)$ crossings to e .

Case 3. Consider next a crossing between e and any edge e' where both endpoints of e' are in $G[T_j]$. The endpoints of e' are thus in Q_a^- and Q_b^- where $j \leq a \leq b$. We charge the crossing to e' . (Think of that crossing being charged to e' in Q_a^-). As argued above, at most $k \cdot \Delta(G)$ (isthmus) edges cross the cycle C_a that replaced the dummy vertex c_a thus each such edge e' is charged at most $k \cdot \Delta(G)$ new crossings.

Case 4. Finally consider a crossing between e and an isthmus edge e' labelled (x, y, P_{xy}) , where $x \in Q_f^-$ with $f < i$. Then there exists $g \in U_f$ such that $i \in T_g$. In that case both endpoints of e are in $G[T_g]$ and we are in Case 3 with the roles of e and e' reversed. Thus at most $k \cdot \Delta(G)$ crossings are charged to e .

By the arguments above, each edge of G is charged at most $2 \cdot k \cdot \Delta(G)$ new crossings (at most $k \cdot \Delta(G)$ in Case 2b and at most $k \cdot \Delta(G)$ in Case 4). Together with the initial sum that results in at most $(c + 2) \cdot k \cdot \Delta(G) \cdot \|G\|$ crossings. ◀

3.2 Rectilinear Drawings of Multigraphs via Graph Partitions

As mentioned previously, in order to prove our main result, (Theorem 13), we will use as a main tool the theorem that we have just proved, Theorem 14. Theorem 12 tells us that in order to use Theorem 14, we need to show that planar graphs and bounded treewidth graphs are $(3, c)$ -agreeable for some constant c . In this section, we define graph partitions and prove a lemma that will be helpful in proving that planar graphs are $(3, c)$ -agreeable in Section 3.3 and that bounded treewidth graphs are (k, c) -agreeable for any $k \geq 1$ in Section 3.4.

An H -partition of a (multi)graph G is comprised of a graph H and a partition of vertices of G such that

- each vertex of H is a non-empty set of vertices of G (called a *bag*),
- every vertex of G is in exactly one bag of H , and
- if an edge of G has one endpoint in A and the other endpoint in B and A and B are distinct, then AB is an edge of H .

The *width* of a partition is the maximum number of vertices in a bag. The *density* of a bag of an H -partition is the number of edges of G with at least one endpoint in that bag. The *density* of an H -partition is the maximum density over all bags of H . A bag is said to be *solitary* if it contains exactly one vertex of G .

The proof of the following lemma is a slight modification of a similar result by Wood and Telle [41].

► **Lemma 15.** *Let K be a multigraph and H a simple graph such that K has an H -partition of width w and density d . Let X be the set of all vertices of K that are not in solitary bags of H . Then we have the following.*

1. $\overline{\text{cr}}(K) \leq \overline{\text{cr}}(H) \cdot w^2 \cdot \Delta(K)^2 + (w - 1) \cdot \sum_{v \in X} \deg_K(v)^2$
2. if H is planar, then
 - (a) there exists a rectilinear drawing of K with at most $2 \cdot d$ crossings per edge.
 - (b) if in addition, the non-solitary bags of H form an independent set in H , then there is a rectilinear drawing of K with at most d crossings per edge.

Proof. We start with a rectilinear drawing $D(H)$ of H with $\overline{\text{cr}}(H)$ crossings. Consider any vertex (bag) B of H . Let $C_\epsilon(B)$ be a disk of radius $\epsilon > 0$ centered at B in $D(H)$. For each edge AB of H , let $C_\epsilon(AB)$ be the region defined by the union of all the line segments with one endpoint in $C_\epsilon(A)$ and the other in $C_\epsilon(B)$. Note that there exists an ϵ small enough such that all of the following conditions are met:

- $C_\epsilon(A) \cap C_\epsilon(B) = \emptyset$ for all distinct bags A and B of H ;
- $C_\epsilon(AB) \cap C_\epsilon(PQ) = \emptyset$ for every pair of edges AB and PQ of H that have no endpoints in common and do not cross in $D(H)$;
- $C_\epsilon(AB) \cap C_\epsilon(Q) = \emptyset$ for every triple of distinct bags A, B, Q of H where AB is an edge of H ;
- For each crossing-pair of edges AB and PQ in $D(H)$, $C_\epsilon(AB) \cap C_\epsilon(PQ)$ is non-empty. We call that region, $C_\epsilon(AB) \cap C_\epsilon(PQ)$, of the plane *busy region* of pair AB and PQ . Finally, the busy regions of all distinct pair of edges are pairwise disjoint.

For each vertex v of K such that v is in a bag B of H , draw v as a point in $C_\epsilon(B)$ such that the final set of points representing $V(K)$ is in general position. Draw every edge of K straight. This defines a rectilinear drawing $D(K)$ of K , since no edge in $D(K)$ contains a vertex other than its own endpoints and no three edges of $D(K)$ cross at one point.

We first prove that the number of crossings in $D(K)$ is at most $\bar{c}(H) \cdot w^2 \cdot \Delta(K)^2 + (w - 1) \cdot \sum_{v \in X} \deg_K(v)^2$ which will prove the first part of the theorem. Consider two crossing edges e and f in $D(K)$. There are two cases to consider (based on two types of crossings that can occur in $D(K)$).

- Case 1: there is bag B of H that has at least one endpoint of e and at least one endpoint of f . Order all the vertices of $B = \{v_1, v_2, \dots, v_\ell\}$, $l \leq w$ such that $\deg_K(v_1) \leq \dots \leq \deg_K(v_\ell)$. Let v_i be an endpoint of e and v_j and endpoint of f , $i < j$. We charge the crossing between e and f to v_j .

Thus the number of crossings charged to v_j is at most

$$\sum_{i < j} \deg_K(v_i) \cdot \deg_K(v_j) \leq \sum_{i < j} \deg_K(v_j)^2 \leq (\ell - 1) \cdot \deg_K(v_j)^2 \leq (w - 1) \cdot \deg_K(v_j)^2$$

The vertices in the solitary bags of H are charged 0 crossings, rendering the total number of crossings in Case 1 is at most $(w - 1) \sum_{v \in X} \deg_K(v)^2$.

- Case 2: there is no bag of H that has both an endpoint of e and an endpoint of f . This implies that four endpoints of e and f are in four distinct bags, A, B, P, Q of H . Let $e \in C_\epsilon(AB)$ and $f \in C_\epsilon(PQ)$. Since e and f cross, their crossing point must be the busy region of AB and PQ . Denote that region by R . There are at most $\Delta(K) \cdot w$ edges of K drawn inside $C_\epsilon(AB)$ that intersect R and at most $\Delta(K) \cdot w$ edges of K drawn inside $C_\epsilon(PQ)$ that intersect R . We charge the crossings between these pairs of edges to the busy region R . Thus the number of crossings charged to R is at most $w\Delta(K) \cdot w\Delta(K) = w^2 \cdot \Delta(K)^2$. Since $D(H)$ has $\bar{c}(H)$ crossings, there are exactly $\bar{c}(H)$ busy regions determined by crossing edges in $D(H)$. Thus the total number of crossings in Case 2 is at most $\bar{c}(H) \cdot w^2 \cdot \Delta(K)^2$.

Thus $\bar{c}(K) \leq \bar{c}(H) \cdot w^2 \cdot \Delta(K)^2 + (w - 1) \cdot \sum_{v \in X} \deg_K(v)^2$ as stated in part 1.

We now prove the second part of the theorem. In this case, H is planar. By the Fáry-Wagner theorem [40, 21], there is a rectilinear drawing $D(H)$ of H with no crossings. Starting with such crossing-free drawing $D(H)$, we produce a rectilinear drawing $D(K)$ of K using the algorithm described above. Let e be an edge of K with an endpoint in some bag A of H . We now prove that the number of crossings on e in $D(K)$ is at most $2d$ as claimed in part 2a. There are two cases to consider:

- Case 1: both endpoints of e are in A . Then, e is only crossed by the edges that have at least one endpoint in A . As there are at most d such edges, there is at most d crossings on e in $D(K)$.
- Case 2: the other endpoint of e is in a bag B of K distinct from A . Then, since $D(H)$ is crossing-free, e can only be crossed by the edges that have at least one endpoint in A or in B . There is at most $2d$ such edges, thus there is at most $2d$ crossings on e in $D(K)$.

In either case, e is crossed by at most $2d$ edges in $D(K)$ as required by part 2a.

Finally, consider the case when the non-solitary bags of H form an independent set in H . Let e be an edge of K . If two endpoints of e are in two distinct solitary bags of H then no edge of K crosses e since $D(H)$ is crossing-free. Therefore, in that case, trivially, there are at most d crossings on e in $D(K)$. Thus we may assume that at least one endpoint of e is in a non-solitary bag of H . Let A denote that bag. If the other endpoint of e is also in A , the result follows from Case 1 above. Therefore, we may assume that the other endpoint, v , of e is in a bag B of H distinct from A . B is then a solitary bag (by the independent set assumption). Since the edges incident to the same vertex (v in this case) cannot cross, the only edges that can cross e are those with an endpoint in A . There is at most d edges with endpoints in A and thus there are at most d crossings on e in $D(K)$. ◀

3.3 Rectilinear Crossing Number of Simplicial Blowups of Planar Graphs

Theorem 12 tells us that in order to use Theorem 14, it is enough to consider (≤ 3)-simplicial blowups of planar graphs with no separating triangles. In other words, it is enough to prove that planar graphs with no separating triangles are $(3, c)$ -agreeable for some constant c . The next lemma achieves that.

► **Lemma 16.** *Every planar graph G that has no separating triangles is $(3, 3)$ -agreeable.*

Proof. Since every induced subgraph of G is also planar and with no separating triangles, it is enough to show that every (≤ 3)-simplicial blowup Q of G has rectilinear crossing number $\overline{\text{cr}}(Q) \leq 3 \cdot \Delta(Q) \cdot \|Q\|$.

Let $S = V(Q) - V(G)$. Since adding a 1-simplicial or 2-simplicial vertex to a planar graph results in a planar graph, we may assume that each vertex in S has exactly 3 neighbours in G . We now define an H -partition of Q . To start, we make H isomorphic to G and put each $v \in V(G)$ in the bag B_v in H . Currently, all the bags in H are solitary bags. Since G , and therefore the current H , has no separating triangles and since S is an independent set in Q , we have that for each $v \in S$, $N_Q(v)$ induces a face in an embedding of G and thus it is a face in the equivalent embedding of H . For each vertex set $\{x, y, z\}$ in H that forms such a face, we add a bag B_{xyz} adjacent to x , y and z in H . The resulting graph H is simple and planar. For each vertex $v \in S$ adjacent to x , y and z in Q , add v to the corresponding bag B_{xyz} in H . Thus the defined graph H and the assignment of the vertices of Q to its bags defines an H -partition of Q .

As every vertex of Q in bag B_{xyz} is adjacent to all vertices in $\{x, y, z\}$, the maximum number of edges of Q with an endpoint in a non-solitary bag B_{xyz} is at most $\deg_Q(x) + \deg_Q(y) + \deg_Q(z) \leq 3 \cdot \Delta(Q)$. The maximum number of edges of Q with an endpoint in a solitary bag of H is clearly $\Delta(Q)$. Thus the density of the H -partition is at most $3 \cdot \Delta(Q)$. Additionally, the non-solitary bags of H form an independent set in H which, by Lemma 15 (2b), implies that Q has a rectilinear drawing with at most $3 \cdot \Delta(Q)$ crossings per edge, giving the desired result, $\overline{\text{cr}}(Q) \leq 3 \cdot \Delta(Q) \cdot \|Q\|$. ◀

3.4 Rectilinear Crossing Number of Simplicial Blowups of Treewidth- k Graphs

In this section, we prove that bounded treewidth graphs are (k, c) -agreeable for some constants k and c . We start with the following trivial bound applicable to all graphs.

► **Lemma 17.** *Every graph G is $(|G|, |G| - 1)$ -agreeable.*

Proof. If $|G| = 1$, the statement is trivial since every (≤ 1)-simplicial blowup of G is a star thus the crossing number of every such blowup is zero. Assume now that $|G| \geq 2$. Since every induced subgraph of G is also in the class of all graphs, it is enough to show that every ($\leq |G|$)-simplicial blowup Q of G has rectilinear crossing number $\overline{\text{cr}}(Q) \leq (|G| - 1) \cdot \Delta(Q) \cdot \|Q\|$.

Let $S = V(Q) - V(G)$. We build an H -partition of Q as follows. Start with $H := K_2$ with $V(H) = \{v, w\}$. Place one vertex of G in B_v and all the remaining vertices of G in B_w . Add an independent set of $|S|$ of vertices to H and make each connected to v and w . It is simple to verify that H is a simple planar graph. Place each vertex of S in a new vertex (bag) of H . That defines an H -partition of Q where H is a simple planar graph and where all bags of H are solitary except for one bag, that is B_w . Trivially, that one non-solitary bag forms an independent set in H . Since H is planar and since the density of H is at most $(|G| - 1) \cdot \Delta(Q)$, we obtain the desired result, $\overline{\text{cr}}(Q) \leq (|G| - 1) \cdot \Delta(Q) \cdot \|Q\|$ by Lemma 15 (2b). ◀

The following result, obtained by setting $|G| = t$, is an immediate corollary of Lemma 17.

► **Corollary 18.** *The complete graph, K_t , is $(t, t - 1)$ -agreeable.*

We are now ready to prove that every bounded treewidth graph G has $\overline{cr}(G) \in O(\Delta(G) \cdot |G|)$.

► **Theorem 19.** *For $k \geq 1$, let \mathcal{G} denote a family of graphs of treewidth at most k . For every graph $G \in \mathcal{G}$, $\overline{cr}(G) \leq k \cdot (k + 2) \cdot \Delta(G) \cdot ||G||$.*

Proof. It is well known (see [6] for example) that G can be obtained by $(\leq k)$ -clique-sums on graphs G_1, G_2, \dots , where each G_i is the complete graph on at most $k+1$ vertices. Corollary 18 implies that, for each i , $i \in [h]$, G_i is $(k+1, k)$ -agreeable and thus (k, k) -agreeable. This fulfills the sole condition of Theorem 14. Thus $\overline{cr}(G) \leq k \cdot (k + 2) \cdot \Delta(G) \cdot ||G||$. ◀

Theorem 19 gives an $O(\Delta(G) \cdot |G|)$ bound for the rectilinear crossing number of bounded treewidth graphs G . As discussed in the introduction, the bound is optimal and it improves on the previously known bounds (see Theorems 6 and 7).

Since every k -simplicial blowup of any graph of treewidth at most k itself has treewidth at most k , we get the following immediate corollary of Theorem 19.

► **Lemma 20.** *For every positive integer k , every graph of treewidth at most k is $(k, k \cdot (k+2))$ -agreeable.*

3.5 Proof of Theorem 13

Recall that Theorem 13 of Robertson and Seymour states that each piece in the decomposition is either a graph of treewidth at most t or it is a planar graph with no separating triangle. Lemma 16 then implies that every planar piece G_i of the decomposition is $(3, 3)$ -agreeable. Consider the non-planar pieces of the decomposition. By Theorem 12, they have treewidth at most t , where $t \geq 3$, as graphs of treewidth at most 2 are planar [6]. Lemma 20 states that every treewidth at most t graph is $(t, t \cdot (t + 2))$ -agreeable. Since every non-planar piece of the decomposition has treewidth at most t with $t \geq 3$, these pieces are $(3, t \cdot (t + 2))$ -agreeable. Since $t \geq 1$ for all pieces of the decomposition, if we choose $c := t \cdot (t + 2)$ all the pieces of the decomposition are $(3, c)$ -agreeable. Theorem 14 (and Theorem 12 by Robertson and Seymour) then implies that G has rectilinear crossing number at most $3 \cdot (t \cdot (t + 2) + 2) \cdot \Delta(G) \cdot ||G|| = 3 \cdot (t^2 + 2t + 2) \cdot \Delta(G) \cdot ||G||$, as claimed.

4 Conclusion and Open Problems

In this article, we proved that n -vertex bounded degree single-crossing minor-free graphs have $O(n)$ rectilinear crossing number. More strongly we proved that for any single-crossing graph X , every n -vertex X -minor-free graph G has rectilinear crossing number at most $O(\Delta(G) \cdot n)$ and the bound is best possible. The result represents a strong improvement over the previous state of the art on the rectilinear crossing numbers of minor-closed families of graphs, as argued in the introduction.

The ultimate goal for future work would be to obtain the above result for any fixed graph X . For such families an $O(f(\Delta) \cdot n)$ bound is not known for any function f . In fact, the best known bound on the rectilinear crossing number of bounded degree proper minor-closed families is $O(n \log n)$ [38].

In order to attempt to prove an $O(f(\Delta) \cdot n)$ bound, that is, a linear rectilinear crossing number for all proper minor-closed families of graphs of bounded degree, Robertson and Seymour's graph minor theory tells us that one should provide two ingredients. The first

ingredient is to prove the result for k -almost embeddable graphs. The second is to be able to handle clique-sums of those. Proving the result for almost embeddable graphs entails proving it for bounded Euler genus graphs, that is, proving a result akin to Theorem 2 by Pach and Tóth [1] but with the crossing number replaced by the rectilinear crossing number. However, such a result is not even known for all bounded degree toroidal graphs.

The second ingredient however, handling the clique-sums of rectilinear drawings, can be achieved by our Theorem 14. In particular, one can change the definition of (k, c) -agreeable to $(k, f(\Delta))$ -agreeable so as to allow for any function $f(\Delta)$ and not just the linear function, $c \cdot \Delta$, and then recall that the proof of Theorem 14 in fact shows that the rectilinear drawings of $(k, f(\Delta))$ -agreeable graphs can be joined by $(\leq k)$ -clique sums into a rectilinear drawing of the resulting graph G while only increasing the total number of crossings by $2 \cdot k \cdot \Delta(G) \cdot \|G\|$. Suppose, in the future, one could provide the first ingredient above, that is, show that almost embeddable n -vertex graphs G have linear rectilinear crossing number, that is $\overline{cr}(G) \leq g(\Delta) \cdot n$ for some function g . In that case the following lemma, Lemma 21, would imply that simplicial blowups of almost embeddable graphs are $(k, f(\Delta))$ -agreeable with $f(\Delta) \in O(\Delta^4) \cdot g(\Delta)$. That and Theorem 14, as discussed in this paragraph, would imply that all proper minor-closed families of graphs of bounded degree have linear rectilinear crossing number.

► **Lemma 21.** *For every graph G and every $(\leq k)$ -simplicial blowup Q of G , $\overline{cr}(Q) \leq (\Delta(Q) + 1)^2 \cdot \overline{cr}(G) + \Delta(Q)^4 \cdot \|Q\|$.*

Proof. Let $S = V(Q) - V(G)$. We now define an H -partition of Q . To start, we make H isomorphic to G and put each $v \in V(G)$ in the bag B_v in H . For each vertex $u \in S$, u is adjacent to all the vertices of some clique C in G . Place u in a bag B_v where $v \in C$. This does not change H since v is adjacent to all the neighbours of u in G . This defines an H -partition of multigraph Q .

For each $v \in H$, each vertex of S in B_v is adjacent to v in Q thus the width of H is at most $\Delta(Q) + 1$. Thus by Lemma 15, $\overline{cr}(Q) \leq (\Delta(Q) + 1)^4 \cdot \overline{cr}(H) + \Delta(Q)^2 \cdot \|Q\|$ which is equal to $(\Delta(Q) + 1)^4 \cdot \overline{cr}(G) + \Delta(Q)^2 \cdot \|Q\|$ since H is isomorphic to G . ◀

References

- 1 Géza . Crossing number of toroidal graphs. In *Topics in discrete mathematics*, pages 581–590. Springer, 2006.
- 2 Martin Aigner and Günter M. Ziegler. *Proofs from The Book*. Springer, third edition, 2004.
- 3 M. Ajtai, V. Chvátal, M.M. Newborn, and E. Szemerédi. Crossing-free subgraphs. In *Theory and Practice of Combinatorics*, volume 60 of *North-Holland Mathematics Studies*, pages 9–12. North-Holland, 1982.
- 4 Sandeep N. Bhatt and F. Thomson Leighton. A framework for solving VLSI graph layout problems. *J. Comput. System Sci.*, 28(2):300–343, 1984. doi:10.1016/0022-0000(84)90071-0.
- 5 Daniel Bienstock and Nathaniel Dean. Bounds for rectilinear crossing numbers. *Journal of graph theory*, 17(3):333–348, 1993. doi:10.1002/JGT.3190170308.
- 6 Hans L. Bodlaender. A partial k -arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1):1–45, 1998. doi:10.1016/S0304-3975(97)00228-4.
- 7 Sergio Cabello. Hardness of approximation for crossing number. *Discret. Comput. Geom.*, 49(2):348–358, 2013. doi:10.1007/S00454-012-9440-6.
- 8 Sergio Cabello and Bojan Mohar. Adding one edge to planar graphs makes crossing number and 1-planarity hard. *SIAM J. Comput.*, 42(5):1803–1829, 2013. doi:10.1137/120872310.
- 9 Erin W. Chambers and David Eppstein. Flows in one-crossing-minor-free graphs. *J. Graph Algorithms Appl.*, 17(3):201–220, 2013. doi:10.7155/JGAA.00291.

- 10 Julia Chuzhoy and Zihan Tan. A subpolynomial approximation algorithm for graph crossing number in low-degree graphs. In *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 303–316. ACM, 2022. doi:10.1145/3519935.3519984.
- 11 Erik D. Demaine, Fedor V. Fomin, Mohammad Taghi Hajiaghayi, and Dimitrios M. Thilikos. Subexponential parameterized algorithms on bounded-genus graphs and H -minor-free graphs. *J. ACM*, 52(6):866–893, 2005. doi:10.1145/1101821.1101823.
- 12 Erik D. Demaine, Mohammad Taghi Hajiaghayi, and Dimitrios M. Thilikos. Exponential speedup of fixed-parameter algorithms for classes of graphs excluding single-crossing graphs as minors. *Algorithmica*, 41(4):245–267, 2005. doi:10.1007/S00453-004-1125-Y.
- 13 Josep Diaz, Norman Do, Maria J Serna, and Nicholas C Wormald. Bounds on the max and min bisection of random cubic and random 4-regular graphs. *Theoretical computer science*, 307(3):531–547, 2003. doi:10.1016/S0304-3975(03)00236-6.
- 14 Hristo N Djidjev and Imrich Vrt'o. Crossing numbers and cutwidths. *Journal of Graph Algorithms and Applications*. v7, pages 245–251, 2006.
- 15 Hristo N Djidjev and Imrich Vrt'o. Planar crossing numbers of graphs of bounded genus. *Discrete & Computational Geometry*, 48(2):393–415, 2012. doi:10.1007/S00454-012-9430-8.
- 16 Vida Dujmović, Ken-ichi Kawarabayashi, Bojan Mohar, and David R. Wood. Tight upper bounds on the crossing number in a minor-closed class. *CoRR*, abs/1807.11617, 2018. arXiv:1807.11617.
- 17 David Eppstein and Vijay V. Vazirani. NC algorithms for computing a perfect matching and a maximum flow in one-crossing-minor-free graphs. *SIAM J. Comput.*, 50(3):1014–1033, 2021. doi:10.1137/19M1256221.
- 18 P. Erdos and R. K. Guy. Crossing number problems. *The American Mathematical Monthly*, 80(1):52–58, 1973.
- 19 M. R. Garey and D. S. Johnson. Crossing number is np-complete. *SIAM Journal on Algebraic Discrete Methods*, 4(3):312–316, 1983.
- 20 Rudolf Halin. S -functions for graphs. *J. Geometry*, 8(1-2):171–186, 1976.
- 21 Fáy István. On straight-line representation of planar graphs. *Acta scientiarum mathematicarum*, 11(229-233):2, 1948.
- 22 Károly J., János Pach, and Géza Tóth. Planar crossing numbers of graphs embeddable in another surface. *Int. J. Found. Comput. Sci.*, 17(5):1005–1016, 2006. doi:10.1142/S0129054106004236.
- 23 Ken-ichi Kawarabayashi and Bruce Reed. Computing crossing number in linear time. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 382–390. ACM, 2007. doi:10.1145/1250790.1250848.
- 24 Alexandr V. Kostochka. Lower bound of the Hadwiger number of graphs by their average degree. *Combinatorica*, 4(4):307–316, 1984. doi:10.1007/BF02579141.
- 25 AV Kostochka and LS Mel'nikov. On bounds of the bisection width of cubic graphs. In *Annals of Discrete Mathematics*, volume 51, pages 151–154. Elsevier, 1992.
- 26 F. Thomson Leighton. *Complexity Issues in VLSI*. MIT Press, 1983.
- 27 F. Thomson Leighton. New lower bound techniques for VLSI. *Math. Systems Theory*, 17(1):47–70, 1984. doi:10.1007/BF01744433.
- 28 Frank Thomson Leighton. *Complexity issues in VLSI: optimal layouts for the shuffle-exchange graph and other networks*. MIT Press, Cambridge, MA, USA, 1983.
- 29 Bernard Montaron. An improvement of the crossing number bound. *J. Graph Theory*, 50(1):43–54, 2005. doi:10.1002/JGT.20090.
- 30 Nagi H. Nahas. On the crossing number of $K_{m,n}$. *Electron. J. Comb.*, 10, 2003. doi:10.37236/1748.
- 31 János Pach, Radoš Radoičić, Gábor Tardos, and Géza Tóth. Improving the crossing lemma by finding more crossings in sparse graphs. *Discrete Comput. Geom.*, 36(4):527–552, 2006. doi:10.1007/S00454-006-1264-9.

- 32 R. Bruce Richter and J Širáň. The crossing number of $K_{3,n}$ in a surface. *Journal of Graph Theory*, 21(1):51–54, 1996. doi:10.1002/(SICI)1097-0118(199601)21:1<51::AID-JGT7>3.0.CO;2-L.
- 33 Neil Robertson and Paul Seymour. Excluding a graph with one crossing. In *Graph structure theory. Proceedings of the AMS-IMS-SIAM joint summer research conference on graph minors*, pages 669–675. Providence, RI: American Mathematical Society, 1993.
- 34 Neil Robertson and Paul D. Seymour. Graph minors. II. algorithmic aspects of tree-width. *J. Algorithms*, 7(3):309–322, 1986. doi:10.1016/0196-6774(86)90023-4.
- 35 Neil Robertson and P.D Seymour. Graph minors. V. excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 41(1):92–114, 1986. doi:10.1016/0095-8956(86)90030-4.
- 36 Neil Robertson and P.D Seymour. Graph minors. XVI. excluding a non-planar graph. *Journal of Combinatorial Theory. Series B*, 89(1):43–76, 2003. doi:10.1016/S0095-8956(03)00042-X.
- 37 Marcus Schaefer. The graph crossing number and its variants: a survey. *Electron. J. Comb.*, DS21:90, 2013.
- 38 Farhad Shahrokhi, Ondrej Šýkora, László A. Székely, and Imrich Vrto. Bounds for convex crossing numbers. In Tandy J. Warnow and Binhai Zhu, editors, *Computing and Combinatorics, 9th Annual International Conference, COCOON 2003*, volume 2697 of *LNCS*, pages 487–495. Springer, 2003. doi:10.1007/3-540-45071-8_49.
- 39 Andrew Thomason. An extremal function for contractions of graphs. *Math. Proc. Cambridge Philos. Soc.*, 95(2):261–265, 1984. doi:10.1017/S0305004100061521.
- 40 Klaus Wagner. Bemerkungen zum vierfarbenproblem. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 46:26–32, 1936.
- 41 David R. Wood and Jan Arne. Planar decompositions and the crossing number of graphs with an excluded minor. *New York J. Math.*, 13:117–146, 2007.

Harborth's Conjecture for 4-Regular Planar Graphs

Daniel J. Chang

Department of Computer Science, San Francisco State University, CA, USA

Timothy Sun 

Department of Computer Science, San Francisco State University, CA, USA

Abstract

We show that every 4-regular planar graph has a straight-line embedding in the plane where all edges have integer length. The construction extends earlier ideas for finding such embeddings for 4-regular planar graphs with diamond subgraphs or small edge cuts.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Planar graph, straight-line embedding, Diophantine equation

Digital Object Identifier 10.4230/LIPIcs.GD.2024.38

1 Introduction

In this paper, graphs are *simple*, i.e., they do not have self-loops or parallel edges. A *Fáry embedding* of a graph is an embedding in the plane where each edge is drawn as a straight line segment. It is a well-known fact that such drawings exist for all planar graphs [6, 12, 15], and a longstanding conjecture of Harborth (see [8]) posits that every planar graph has a Fáry embedding where every edge has integer (or equivalently, rational) length. Geelen, Guo, and McKinnon [7], using a result of Berry [2], showed that all planar graphs of maximum degree 3 satisfy Harborth's conjecture. Since then, there have been simpler proofs of the cubic case [3, 13] and extensions of their application of Berry's theorem to graphs of higher maximum degree [1, 3, 5].

Sun [14] showed how these ideas can be used to find rational Fáry embeddings of two families of 4-regular graphs. We correct an error in one of those constructions and show that those techniques can be strengthened to cover all 4-regular planar graphs:

► **Theorem 1.** *Every 4-regular planar graph has a rational Fáry embedding.*

2 Background

Given a planar graph $G = (V, E)$, we treat Fáry embeddings of G as injective mappings $\phi: V \rightarrow \mathbb{R}^2$. We say that a Fáry embedding is *rational* if all its edge lengths are rational and *fully rational* if it is rational and the vertices are at rational coordinates. Two Fáry embeddings ϕ and ϕ' of G are said to be ε -close if, for every vertex $v \in V$, the Euclidean distance $d(\phi(v), \phi'(v))$ between the images of v is less than ε . A planar graph G is said to be *fully approximable* if, for every Fáry embedding ϕ of G and every $\varepsilon > 0$, there exists a fully rational Fáry embedding ϕ' that is ε -close to ϕ . Geelen, Guo, and McKinnon [7] were the first to consider such a property because of the following result on Diophantine equations:

► **Theorem 2** (Berry [2]). *Let x, y , and z be non-collinear points in the plane such that $d(x, y)$, $d(y, z)^2$, and $d(z, x)^2$ are rational. Then, the set of points at rational distance from all three points is dense in the plane.*



© Daniel J. Chang and Timothy Sun;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 38; pp. 38:1–38:9

Leibniz International Proceedings in Informatics



LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

We note that the squares of the distances are automatically rational if the points have rational coordinates. Using Theorem 2, Geelen et al. constructed fully rational Fáry embeddings of planar graphs of maximum degree 3, but Biedl [3] was the first to observe that their method extends to a more general family of graphs. Let G be a graph on n vertices. An ordering of its vertices v_1, \dots, v_n is said to be a *3-elimination order* if:

- G has only one vertex, i.e., $n = 1$, or
- v_1 has degree at most 2 and v_2, \dots, v_n is a 3-elimination order for $G - v_1$, or
- v_1 has degree 3 and there are two neighbors u and u' of v_1 such that v_2, \dots, v_n is a 3-elimination order for $(G - v_1) \cup \{uu'\}$.

Given a 3-elimination order, one can recursively construct a fully rational Fáry embedding of $G - v_1$ or $(G - v_1) \cup \{uu'\}$, and then add v_1 using Theorem 2:

► **Theorem 3** (Geelen et al. [7], Biedl [3]). *Every planar graph with a 3-elimination order is fully approximable.*

A graph is called $(2, 1)$ -sparse if every nonempty induced subgraph H satisfies $|E(H)| \leq 2|V(H)| - 1$. Biedl [3] identified the $(2, 1)$ -sparse graphs as a rich family of graphs with 3-elimination orders, but unfortunately, the original proof has an error. For our purposes, we do not explicitly need $(2, 1)$ -sparseness (a discussion and a corrected proof are deferred to Appendix A). Instead, we rely on just a special case: a graph is said to be *subquartic* if it has maximum degree 4, but is not 4-regular. Biedl [3] showed that connected subquartic graphs are $(2, 1)$ -sparse while Benediktovich [1] directly proved that such graphs have 3-elimination orders:

► **Theorem 4** (Benediktovich [1], Biedl [3]). *Every connected subquartic graph has a 3-elimination order. Hence, planar connected subquartic graphs are fully approximable.*

4-regular planar graphs cannot have 3-elimination orders because there is no possible first vertex v_1 . To circumvent this, Sun [14] considered two approaches. If the graph has low edge connectivity, then a rational Fáry embedding can be pieced together from its subquartic blocks. The other idea is to apply a theorem of Kemnitz and Harborth [8] to remove a vertex of a diamond subgraph. In Theorems 5 and 14, we improve upon each approach.

3 The low connectivity case

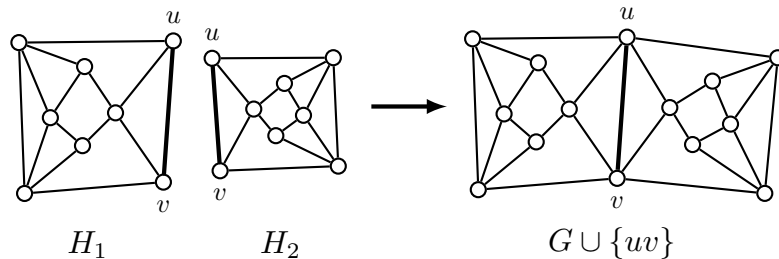
Using rigidity theory, Sun [14] proved that 4-regular planar graphs with edge connectivity 2 have rational Fáry embeddings. We upgrade this result to *vertex* cuts of size 2:

► **Theorem 5.** *Every connected 4-regular planar graph that is not 3-connected has a rational Fáry embedding.*

Proof. By the handshaking lemma, minimal edge cuts in 4-regular graphs must have even size. If the graph has an edge cut of size 2 (which includes when the graph has a cutvertex), then the result of Sun [14] applies. Now suppose the graph G is 4-edge-connected and has a vertex cut $\{u, v\}$. Deleting the cut disconnects the graph into exactly two connected components C_1 and C_2 . Furthermore, since G does not have an edge cut of size 2, u and v both have exactly two neighbors in both C_1 and C_2 .

For $i = 1, 2$, define H_i to be the induced subgraph $G[V(C_i) \cup \{u, v\}]$, with the edge uv added. Since C_{3-i} is connected, there is a path in C_{3-i} from u to v , so H_i is planar. Choose Fáry embeddings of H_1 and H_2 where uv is on the boundary of the embeddings' convex hulls. By applying Theorem 4, we obtain two nearby rational Fáry embeddings of H_1 and

H_2 . As shown in Figure 1, after applying a rational scaling to one of the embeddings so that the length of the edge uv is the same in both, they can be glued together at this edge to obtain a rational Fáry embedding of G . ◀



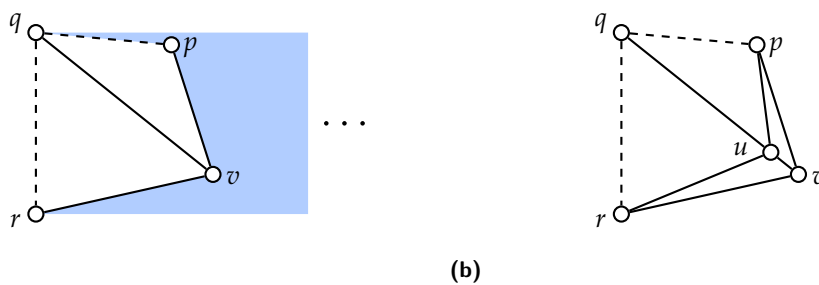
■ **Figure 1** Pasting together two rational Fáry embeddings at two vertices.

4 The 3-connected case

4.1 The geometric part

Let $vpqr$ be a quadrilateral where vq is an internal diagonal and $d(v, p)$, $d(v, q)$, $d(v, r)$, $d(p, q)^2$, and $d(q, r)^2$ are rational. Kemnitz and Harborth [8] applied the theory of Diophantine equations to find a point u on the line \overline{vq} at rational distance from v, p, q, r . If v, p, q, r are vertices of a graph, then the lengths required to be rational form a complete bipartite graph $K_{1,3}$, where v is the vertex of degree 3. We say that a Fáry embedding of $K_{1,3}$ is *permissible* if the rational constraints above are satisfied and Kemnitz and Harborth’s solution for u lies on the interior of the line segment vq .

Sun [14] gave an explicit example of a permissible quadrilateral: we call a Fáry embedding of $K_{1,3}$ a *good kite* if $vpqr$ forms a convex quadrilateral, $d(p, q) = d(p, v) = 3$, $d(r, q) = d(r, v) = 4$, and $d(q, v) = 5$. A good kite and the location of the new vertex u are shown in Figure 2(a) and (b), respectively.



■ **Figure 2** A special “one-sided” polygon (a) enables a useful solution to a certain system of Diophantine equations (b).

▶ **Proposition 6** (Sun [14]). *Let ϕ be a permissible embedding of $K_{1,3}$. Then, there exists $\varepsilon_0 > 0$ such that any fully rational Fáry embedding ϕ' that is ε_0 -close to ϕ has a point on the interior of the line segment qv at rational distance to each of v, p, q, r .*

However, when trying to use Proposition 6, Sun [14] erroneously claimed that any drawing of an internal face as a convex polygon can be extended to a Fáry embedding of the entire graph. A simple counterexample is the wheel graph on 5 vertices whose quadrangular face is drawn as a square: the only way to add the remaining vertex is inside of the square.

Fortunately, the convex quadrilateral found in a good kite can be extended. Using the terminology in Mchedlidze, Nöllenburg, and Rutter [10], given a maximal plane graph with a specified unbounded face, we say that a cycle is *outerchordless* if all of its chords are in the interior of the cycle. A convex polygon is said to be *one-sided* if there is a point in the exterior of the polygon which is visible from every vertex of the polygon.

► **Theorem 7** (Mchedlidze et al. [10], Theorem 2). *Let G be a maximal plane graph, and let C be an outerchordless cycle in G . Then any drawing of C as a one-sided polygon can be extended to a Fáry embedding ϕ of G , possibly with a different unbounded face, where a face is internal to C in ϕ if and only if it is internal to C in G .*

The original statement of the above result in [10] does not mention preserving internal faces, but it is implicit in their proof. We note that this is not their main result (Theorem 4 in [10]), which applies to arbitrary convex polygons and is able to preserve the unbounded face, though it has additional requirements on so-called “petals” of the cycle.

► **Lemma 8.** *Let G be a plane graph, where the rotation at some vertex v is of the form $(\dots p, q, r \dots)$, and p and r are not adjacent. Then, there is a Fáry embedding of G where the restriction to the $K_{1,3}$ subgraph formed by the edges vp , vq , and vr forms a good kite, and vq is the only edge that intersects the interior of the convex hull of the $K_{1,3}$ subgraph.*

Proof. Triangulate the graph to obtain a maximal plane graph G' so that there are faces $[v, q, p]$ and $[v, r, q]$ (which may require changing the locations of the edges pq and qr , if they already exist) and the edge pr is still missing, introducing additional vertices if necessary. Draw the $K_{1,3}$ subgraph as a good kite and consider the cycle $vpqr$. Since the edge vq is in the interior of the cycle and there is no edge pr , the cycle is outerchordless. In Figure 2(a), there is a point in the shaded region that extends perpendicularly from qr where all four vertices are visible from that point, so the cycle has been drawn as a one-sided quadrilateral. Thus, we may apply Theorem 7 to $vpqr$ to obtain a Fáry embedding of all of G' . Since G' is 3-connected, Whitney's theorem ensures that the new drawing has the same set of rotations, and hence the same set of faces. Consequently, no other edge besides vq intersects the interior of the convex quadrilateral $vpqr$. ◀

Adding a vertex using Proposition 6 creates what we call a *diamond*, two triangular faces meeting at an edge. We call that intersecting edge the *central* edge and its endpoints *central vertices*. As seen in Figure 2, undoing the vertex addition is combinatorially equivalent to contracting the edge uv . We can subsequently summarize the aforementioned geometric results in a graph-theoretic manner:

► **Lemma 9.** *Let G be a plane graph with a diamond where one of its central vertices has degree 4, and the two non-central vertices are not adjacent. Let G' be the graph after contracting the diamond's central edge. If G' has a 3-elimination order, then G has a rational Fáry embedding.*

Proof. Reusing earlier notation, let the rotation at u be (v, p, q, r) and let the diamond's faces be $[u, p, v]$ and $[u, v, r]$. Contract the edge uv and call the new vertex v . Apply Lemma 8 to obtain a Fáry embedding ϕ of G' . There exists $\varepsilon_1 > 0$ such that, in any Fáry embedding

ε_1 -close to ϕ , the quadrilateral $vpqr$ is convex, and no other edge intersects the interior of $vpqr$ (i.e., the same guarantees as in Lemma 8). In particular, adding a new vertex anywhere in the interior of $vpqr$ and connecting it to v, p, q, r would not create any crossings. Since G' has a 3-elimination order, use Theorem 3 to find a fully rational Fáry embedding $\min(\varepsilon_0, \varepsilon_1)$ -close to ϕ . Finally, use Proposition 6 to find a point in the interior of vq to add back vertex u and its incident edges so that those edges have rational length. ◀

Guaranteeing non-adjacency for applications of Lemma 9 is aided by the following facts:

► **Proposition 10** (Sun [14]). *In a 4-edge-connected 4-regular plane graph, every 3-cycle is facial.*

► **Corollary 11.** *In a 4-edge-connected 4-regular plane graph, if the rotation at a vertex is of the form (a, b, c, d) , then a and c are not adjacent.*

We note that, for 4-regular graphs, 3-connectivity implies 4-edge-connectivity by the same parity argument mentioned in the proof of Theorem 5.

4.2 The combinatorial part

Given a 3-connected 4-regular plane graph, we say that a vertex is of *type* (t_1, t_2, t_3, t_4) , where $t_1 \leq t_2 \leq t_3 \leq t_4$, if the lengths of the faces $f_1 \leq f_2 \leq f_3 \leq f_4$ incident with the vertex satisfy $f_i \leq t_i$, for $i = 1, \dots, 4$.

► **Theorem 12** (Lebesgue [9]). *Every 3-connected 4-regular plane graph has a vertex of type $(3, 3, 3, \infty)$, $(3, 3, 4, 11)$, $(3, 3, 5, 7)$, or $(3, 4, 4, 5)$.*

We extract a few configurations of faces from Lebesgue's criterion. A *bowtie* consists of two triangular faces intersecting at a vertex, and a *house* consists of a triangular face and a quadrangular face intersecting at an edge.

► **Corollary 13.** *Every 3-connected 4-regular plane graph contains a diamond, a bowtie, or a house.*

Proof. If the vertex in Theorem 12 is incident with at least two triangular faces, then there is a diamond or a bowtie. Otherwise, the vertex is of type $(3, 4, 4, 5)$. At least one of the quadrangular faces intersects the triangular face at an edge, forming a house. ◀

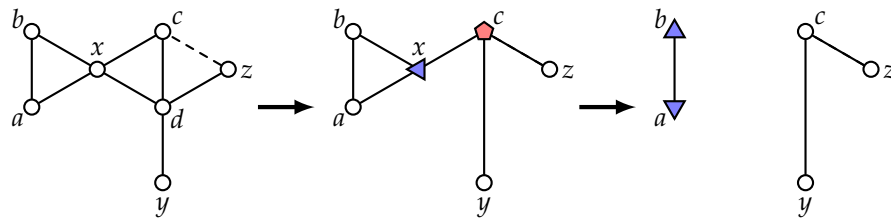
Note that the latter two configurations are necessary: of the three configurations above, the medial graphs of the dodecahedron and cuboctahedron graphs have only bowties and houses, respectively.

► **Theorem 14.** *Every planar 3-connected 4-regular graph has a rational Fáry embedding.*

Proof. For each such graph G , we will create a diamond (if one does not already exist) by adding an edge near a triangular face, and then verify that this graph satisfies the conditions in Lemma 9. In each case, the diamond's non-central vertices will not be adjacent by Corollary 11, so it remains to show that contracting the diamond's central edge yields a graph G' with a 3-elimination order. If G already has a diamond, then G' would be connected and subquartic, so assume otherwise.

We note that in the remaining cases, the inclusion of another edge causes G' to have degree sequence $3, 4, \dots, 4, 5$, which implies that it has too many edges to be $(2, 1)$ -sparse. Instead, we will have to specify the first few vertices of the 3-elimination order until we are able to invoke Theorem 4 to generate the rest of the ordering.

By Corollary 13, G must have a bowtie or a house. If there is a bowtie, let its two triangular faces be $[x, a, b]$ and $[x, c, d]$, as seen on the left of Figure 3. In the subsequent drawings, blue triangles and red pentagons denote vertices of degree 3 and 5, respectively. Suppose that the rotation at vertex d is (c, x, y, z) . By Corollary 11, x and z are not adjacent. Adding the edge cz creates a diamond where the edge cd is shared between two triangular faces, and vertex d has degree 4. After contracting cd , x and c are the vertices of degree 3 and 5, respectively. Let x be the first vertex in the 3-elimination order for G' , and choose a and b to be its neighbors. The subsequent graph $(G' - x) \cup \{ab\} = G' - x$ is subquartic and connected, since it is a spanning supergraph of $G - \{x, d\}$, and G is 3-connected.



■ **Figure 3** Reducing a bowtie by specifying the first vertex in the 3-elimination order.

If there is a house, label the vertices so that its two faces are $[w_1, w_\ell, b, c]$ and $[w_1, c, d]$, the rotation at vertex d is (c, w_1, y, z) , and $[w_1, w_2, \dots, w_\ell]$ is the face sharing the edge $w_1 w_\ell$ with the quadrangular face. Since the graph is simple, there are four distinct neighbors of w_1 . If, say, $c = w_i$, for some $i \in \{2, \dots, \ell - 1\}$, then deleting w_1 and c would disconnect w_2 from w_ℓ , contradicting the assumption that G is 3-connected. Similarly, $d \neq w_i$ as well. Thus, the vertices c, d, w_1, \dots, w_ℓ are all distinct.

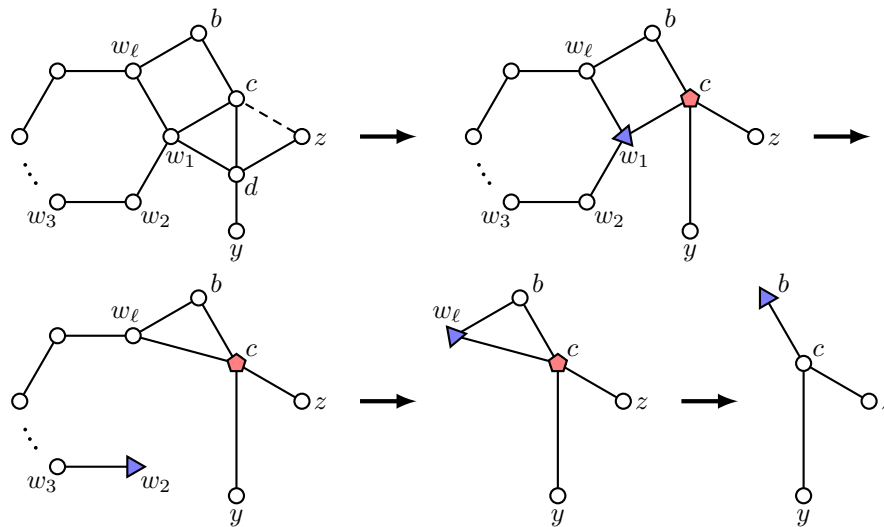
Like in the bowtie case, add the edge cz and contract the edge cd . As depicted in Figure 4, the first vertices in the 3-elimination order are w_1, w_2, \dots, w_ℓ , where the neighbors u, u' of w_i , for each $i = 1, \dots, \ell - 1$, are chosen to be the two that are not w_{i+1} . Each vertex w_i has degree 3 by the time it is deleted, since they each started with degree 4. For w_ℓ , we choose b and c as its neighbors. The edges bc and bw_ℓ are in the original graph, but cw_ℓ was only added when deleting vertex w_1 .

The resulting graph is now subquartic, so it remains to check that it is connected. It is a spanning supergraph of $(G/cd) - \{w_1, \dots, w_\ell\}$, but since $c, d \neq w_i$, we obtain the same graph if we reverse the order of these two operations. Face boundaries in 3-connected plane graphs are non-separating cycles (see, e.g., Proposition 2.4.7 of Diestel [4]). Thus, deleting w_1, \dots, w_ℓ does not disconnect the graph, and neither would contracting the edge cd afterwards. ◀

5 Future Directions

We showed that the solutions to certain Diophantine equations [2, 8] can be used to construct rational Fáry embeddings for all 4-regular planar graphs. However, the proof of Theorem 2 is quite complicated and does not give an explicit method for calculating the locations of such points. Is there a simpler construction, perhaps exploiting the additional condition that vertices are placed at rational coordinates?

Because the proof of Theorem 5 possibly applies a rotation to a Fáry embedding, the vertices are not guaranteed to be at rational coordinates. Is it possible to use the proof technique for 3-connected graphs (or other methods) to find fully rational Fáry embeddings in the low connectivity case?



■ **Figure 4** Reducing a house by transforming the quadrangular face into a triangular one.

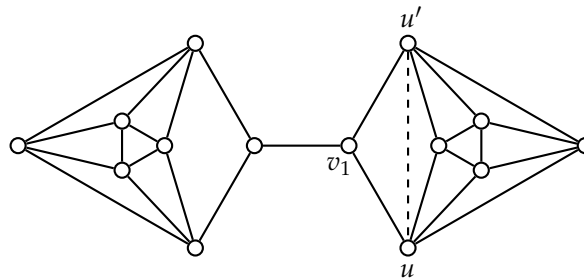
Finally, Harborth's conjecture is still wide open. Are there any methods for finding a point at rational distance to families of five-point sets? What other interesting families of graphs, especially those with $3n - O(1)$ edges, have 3-elimination orders?

References

- 1 Vladimir I. Benediktovich. On rational approximation of a geometric graph. *Discrete Mathematics*, 313(20):2061–2064, 2013. doi:10.1016/J.DISC.2013.06.018.
- 2 T. G. Berry. Points at rational distance from the vertices of a triangle. *Acta Arithmetica*, 62:391–398, 1992.
- 3 Therese Biedl. Drawing some planar graphs with integer edge-lengths. In *Canadian Conference on Computational Geometry*, pages 291–296, 2011.
- 4 Reinhard Diestel. *Graph Theory*, volume 173. Springer, 2017.
- 5 Artūras Dubickas. On some rational triangles. *Mediterranean Journal of Mathematics*, 9(1):95–103, 2012.
- 6 István Fáry. On straight-line representation of planar graphs. *Acta Sci. Math. (Szeged)*, 11:229–233, 1948.
- 7 Jim Geelen, Anjie Guo, and David McKinnon. Straight line embeddings of cubic planar graphs with integer edge lengths. *Journal of Graph Theory*, 58(3):270–274, 2008. doi:10.1002/JGT.20304.
- 8 Arnfried Kemnitz and Heiko Harborth. Plane integral drawings of planar graphs. *Discrete Mathematics*, 236:191–195, 2001. doi:10.1016/S0012-365X(00)00442-8.
- 9 Henri Lebesgue. Quelques conséquences simples de la formule d'Euler. *Journal de Mathématiques Pures et Appliquées*, 19(1-4):27–43, 1940.
- 10 Tamara Mchedlidze, Martin Nöllenburg, and Ignaz Rutter. Drawing planar graphs with a prescribed inner face. In *International Symposium on Graph Drawing*, pages 316–327. Springer, 2013. doi:10.1007/978-3-319-03841-4_28.
- 11 Anthony Nixon and John Owen. An inductive construction of $(2,1)$ -tight graphs. *arXiv preprint arXiv:1103.2967*, 2011.
- 12 Sherman K. Stein. Convex maps. *Proceedings of the American Mathematical Society*, 2(3):464–466, 1951.

- 13 Timothy Sun. Rigidity-theoretic constructions of integral Fary embeddings. In *Canadian Conference on Computational Geometry*, pages 287–290, 2011.
- 14 Timothy Sun. Drawing some 4-regular planar graphs with integer edge lengths. In *Canadian Conference on Computational Geometry*, pages 193–198, 2013.
- 15 Klaus Wagner. Bemerkungen zum Vierfarbenproblem. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 46:26–32, 1936.

A (2, 1)-sparse graphs have 3-elimination orders



■ **Figure 5** A potential dead end in a graph with a 3-elimination order.

The original proof in [3] attempts to show that $(2, 1)$ -sparse graphs have 3-elimination orders via induction. In particular, the inductive step claimed that when deleting a vertex v_1 of degree 3, adding *any* edge between its neighbors still results in a $(2, 1)$ -sparse graph. Figure 5 illustrates a case where the choice of neighbors is important: deleting vertex v_1 and adding an edge between its neighbors u and u' create a 4-regular component. Consequently, this new graph is not $(2, 1)$ -sparse and does not have a 3-elimination order, even though the original graph is connected and subquartic.

Given a subset of vertices $X \subseteq V$, let $e(X)$ denote the number of edges in the subgraph induced by X . In this notation, a graph is $(2, 1)$ -sparse if for all nonempty $X \subseteq V$, $e(X) \leq 2|X| - 1$. The key argument in the following proof is due to Nixon and Owen [11], who used it to characterize the $(2, 1)$ -sparse graphs G with exactly $|E(G)| = 2|V(G)| - 1$ edges.

► **Lemma 15** (Biedl [3]). *Every $(2, 1)$ -sparse graph has a 3-elimination order.*

Proof. We induct on the number of vertices. Given a $(2, 1)$ -sparse graph G , the result is true when G has one vertex. Since G has average degree strictly less than 4, there is a vertex v of degree at most 3. If v has degree at most 2, or has degree 3 and two of its neighbors are adjacent, then $G - v$ is also $(2, 1)$ -sparse by definition. The remaining case is when v has degree 3, and none of its neighbors w_1, w_2, w_3 are adjacent.

We show that there is at least one choice of neighbors w_i and w_j such that $(G - v) \cup \{w_i w_j\}$ is still $(2, 1)$ -sparse. Assume that no such choice exists, so that for each pair of neighbors w_i and w_j , there is a subset of vertices X_{ij} such that $v \notin X_{ij}$, $w_i, w_j \in X_{ij}$, and $e(X_{ij}) = 2|X_{ij}| - 1$ (i.e., a subset that cannot afford having another edge). Then, consider the subgraph induced by $X' = X_{12} \cup X_{23} \cup \{v\}$. This graph has

$$\begin{aligned} e(X') &= e(X_{12} \cup X_{23}) + 3 \\ &= e(X_{12}) + e(X_{23}) - e(X_{12} \cap X_{23}) + 3 \\ &= (2|X_{12}| - 1) + (2|X_{23}| - 1) - e(X_{12} \cap X_{23}) + 3 \\ &\geq (2|X_{12}| - 1) + (2|X_{23}| - 1) - (2|X_{12} \cap X_{23}| - 1) + 3 \\ &= 2|X_{12} \cup X_{23}| + 2 \\ &= 2|X'| \end{aligned}$$

edges, violating $(2, 1)$ -sparseness. The inequality relies on the fact that $X_{12} \cap X_{23}$ is nonempty (since w_2 is in it), which allows us to apply the definition of $(2, 1)$ -sparseness to it. ◀


Drawing Planar Graphs and 1-Planar Graphs Using Cubic Bézier Curves with Bounded Curvature

David Eppstein ✉ 🏠

University of California, Irvine, CA, USA

Michael T. Goodrich ✉ 🏠 

University of California, Irvine, CA, USA

Abraham M. Illickan ✉ 🏠 

University of California, Irvine, CA, USA

Abstract

We study algorithms for drawing planar graphs and 1-planar graphs using cubic Bézier curves with bounded curvature. We show that any n -vertex 1-planar graph has a 1-planar RAC drawing using a single cubic Bézier curve per edge, and this drawing can be computed in $O(n)$ time given a combinatorial 1-planar drawing. We also show that any n -vertex planar graph G can be drawn in $O(n)$ time with a single cubic Bézier curve per edge, in an $O(n) \times O(n)$ bounding box, such that the edges have $\Theta(1/\text{degree}(v))$ angular resolution, for each $v \in G$, and $O(\sqrt{n})$ curvature.

2012 ACM Subject Classification Theory of computation

Keywords and phrases graph drawing, planar graphs, Bézier curves, and RAC drawings

Digital Object Identifier 10.4230/LIPIcs.GD.2024.39

Funding This work was supported in part by NSF Grant 2212129.

1 Introduction

A Bézier curve is a parametric curve defined by a set of *control* points that determine a smooth, continuous curve in the plane [14, 30]. For example, one of the most common types, a *cubic Bézier* curve, is defined by four points, P_0, P_1, P_2, P_3 , such that the curve starts at P_0 tangent to the line segment $\overline{P_0P_1}$ and ends at P_3 tangent to the line segment $\overline{P_2P_3}$, with the lengths of $\overline{P_0P_1}$ and $\overline{P_2P_3}$ determining “how fast” the curve turns towards P_1 before turning towards P_2 . Formally, a cubic Bézier curve, f , has the following explicit form (see Figure 1):

$$f(t) = (1-t)^3P_0 + 3(1-t)^2tP_1 + 3(1-t)t^2P_2 + t^3P_3, \quad \text{for } 0 \leq t \leq 1.$$

The speed in which a curve turns can be characterized by its *curvature*, which is a measure of the instantaneous rate of change of direction of a point that moves on the curve; hence, the larger the curvature, the larger this rate of change. For example, the curvature of a line is zero, the curvature of a polygonal chain with a bend is $+\infty$, and the curvature of a circle is the reciprocal of its radius.

Formally, the curvature of a twice-differentiable parameterized curve, $\mathbf{c}(t) = (x(t), y(t))$, can be defined as follows (e.g., see [33, p. 890]):

$$\kappa(t) = \frac{|x'y'' - y'x''|}{(x'^2 + y'^2)^{3/2}},$$

where x' and x'' are the first and second derivatives of x , and y' and y'' are the first and second derivatives of y , with respect to t . For this to be well-defined the curve must be smooth enough to have a second derivative, which is not true for polylines. In such cases the curvature can be thought of as infinite, as a limiting case of smooth perturbations of the given curve.



© David Eppstein, Michael T. Goodrich, and Abraham M. Illickan;
licensed under Creative Commons License CC-BY 4.0

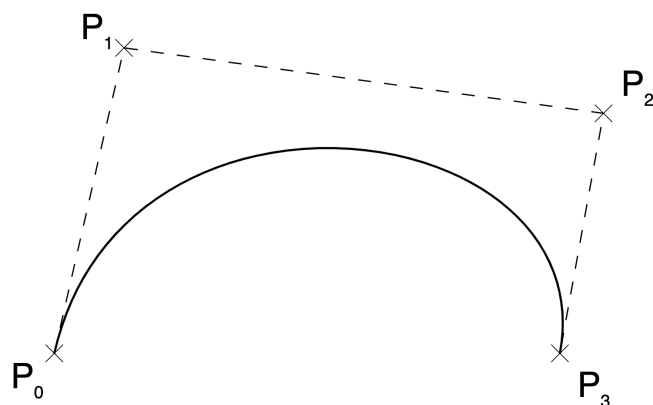
32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 39; pp. 39:1–39:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** An example cubic Bézier curve. Public domain image by MarianSigler.

For a graph drawing, we desire the curvatures of the edges to be small, where we define the curvature of a drawing of a graph G to be the maximum curvature for a non-vertex point on an edge of G , taken over all edges in the drawing of G . For example, Xu, Rooney, Passmore, Ham, and Nguyen [36] empirically show that user performance on network tasks is better for low-curvature drawings than for high-curvature drawings.

Unfortunately, minimizing curvature may conflict with other goals for a drawing. For example, curvature can conflict with angular resolution for planar drawings of planar graphs.¹ For example, we can draw an n -vertex planar graph without crossings using straight edges (i.e., with curvature 0) but this can cause angular resolution to be $O(1/n)$, even for low-degree vertices, e.g., see [9, 28, 31], or even worse, e.g., see Tutte [35]. Indeed, Garg and Tamassia [22] show that, in general, the best angular resolution that any algorithm for drawing a degree- d planar graph G using straight-line drawing can achieve is $O(\sqrt{(\log d)/d^3})$. Ideally, we would like the angular resolution for a drawing of a graph G , to be $\Omega(1/\text{degree}(v))$, for each $v \in G$, which Goodrich and Wagner show how to achieve [23], but their methods for achieving this bound either use polylines with bends (hence, with infinite curvature) or with Bézier curves that the authors admit have high curvature, and they pose as an open problem whether one can simultaneously achieve good angular resolution and relatively low curvature for planar graph drawings with edges represented with cubic Bézier curves.²

In terms of another trade-off for drawings with curves, Huang, Eades, Hong, and Duh [25] empirically show that users performing network tasks were quickest with drawings with curved crossing edges rather than mixed drawings with no crossings, and the authors conclude that for better human graph comprehension, it might be better to use curves to increase crossing angle, rather than to remove them completely. Similarly, Huang, Hong, and Eades [26] report on user studies showing that crossings with large angles are much less harmful to the readability of drawings than shallow crossings. Relatedly, there is considerable prior work on right angle crossing (RAC) drawings, where every pair of crossing edges must cross at right angles, but these drawings are typically achieved by using polygonal paths with bends, e.g.,

¹ Recall that a planar graph can be drawn in the plane without edge crossings and a 1-planar graph can be drawn in the plane so that each edge crosses at most one other edge. Also recall that the angular resolution for each vertex is the minimum angle between two edges incident on v in the drawing.

² For the sake of normalization of the curvature parameter, we assume in this paper that a drawing has an $O(n) \times O(n)$ bounding box, as is common for drawings of planar and 1-planar graphs.

see [1, 2, 10, 11, 32, 34]; hence, these drawings can have unbounded curvature. Therefore, we are interested in methods for producing RAC drawings of 1-planar graphs using curves having bounded curvature, e.g., such as can be achieved with cubic Bézier curves.

Bézier curves are used extensively in computer graphics applications, where it is common to concatenate Bézier curves together to form a composite Bézier curve, e.g., see [30]. As long as each connection point is collinear with its two adjacent control points, then the resulting composite Bézier curve will be C^1 continuous, but it will not necessarily have continuous curvature. In addition, such representations can be quite complex, depending on the number of pieces used, and the curvature at connection points might not be well-defined or, even if it exists, it might not be easy to bound. Thus, we are interested in this paper on studying drawings of planar graphs and 1-planar graphs using cubic Bézier curves where each edge is represented with a single cubic Bézier curve, so that each edge has bounded curvature. In the case of 1-planar graph drawings, we desire edge crossings to be at right angles, and in the case of planar graph drawings, we would like to simultaneously achieve good angular resolution and low curvature.

1.1 Related Prior Work

There is some notable previous work on using Bézier curves for graph drawing, which we review below, but we are not aware of previous work on using Bézier curves to draw planar graphs with low curvature per edge and optimal angular resolution or for RAC drawings of 1-planar graphs.³

In addition to the work cited above, there is some interesting prior work on using Bézier curves in graph drawing systems. For example, the Graphviz system of Gansner [21] can render edges using Bézier curves. Finkel and Tamassia [20] describe a force-directed graph drawing implementation that uses Bézier curves to render graph edges by integrating control points into the force equations. Brandes and Wagner [3] visualize railroad systems with some edges rendered using Bézier curves, and Fink, Haverkort, Nöllenburg, Roberts, Schuhmann, and Wolff [19] provide a similar type of system for drawing metro maps. The GDot system of Hong, Eades, and Torkel [24] uses Bézier curves to draw edges in graphs visualized as dot paintings. In addition, the CelticGraph system of Eades, Gröne, Klein, Eades, Schreiber, Hailer, and Schreiber [15] draws graphs using Celtic knots with edges represented as Bézier curves with limited curvature.

In terms of additional theoretical work, Eppstein, Goodrich, and Meng [17] show how to draw confluent layered drawings using Bézier curves that combine multiple edges, and Eppstein and Simons provide a similar result for Hasse diagrams [18]. In addition, there is considerable prior work on Lombardi drawings, where edges are drawn using circular arcs, e.g., see [5, 12, 13, 16, 18, 29], which we consider to be related work even though circular arcs are not Bézier curves. Cheng, Duncan, Goodrich, and Kobourov [4] show how to draw an n -vertex planar graph G with asymptotically optimal angular resolution, $O(1/\text{degree}(v))$, for each $v \in G$, using 1-bend polylines or circle-arc chains, both of which have unbounded curvature.

³ At a workshop affiliated with GD 2023 to celebrate the 60th birthday of Beppe Liotta, Peter Eades advocated for more research on the topic of using Bézier curves to draw graphs, including results involving curvature guarantees.

1.2 Our Results

In this paper, we show how to draw 1-planar graphs as RAC drawings using a single cubic Bézier curve for each edge; hence, with bounded curvature. We also show how to draw planar graphs in an $O(n) \times O(n)$ grid with good angular resolution by rendering each edge using a cubic Bézier curve with $O(\sqrt{n})$ curvature. Our methods involve careful constructions and proof techniques for proving bounded curvature results, which may be applicable in other settings.

Our constructions are also based in part on refinements of the *convex hull property* of Bézier curves, which is that every point of a Bézier curve lies inside the convex hull of its defining control points, e.g., see [14]. In our results, however, the convex hull property is not sufficient, since the regions in which we desire Bézier curves to traverse are more restrictive than just the convex hulls of control points. Moreover, the convex hull property says nothing about right-angle crossing points for pairs of Bézier curves, which is an important component of our work, and one that requires considerable work, as we show.

2 Constrained Constructions for Pairs of Bézier Curves

We show in this paper that we can draw any 1-planar graph in the plane with right angle crossings, i.e., a RAC drawing, using Bézier curves for every pair of intersecting edges and straight line segments for the rest.

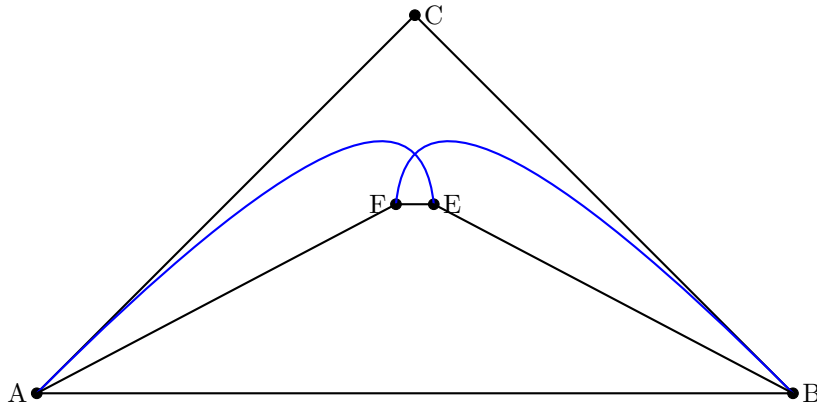
Bekos, Didimo, Liotta, Mehrabi, and Montecchiani [2] show that one can draw any 1-planar graph as a RAC drawing where each edge is represented by a polyline that has at most one bend. Their algorithm starts from a 1-planar (combinatorial) embedding of a 1-planar graph G , and proceeds via an induction proof involving augmentation and contraction steps to produce a RAC drawing of G with edges represented with polylines with at most one bend per edge. We show how to adapt their proof to use a single cubic Bézier curve per edge in place of a 1-bend polyline. To achieve this, we develop a number of constructions for pairs of Bézier curves that cross at right angles in specific ways while fitting in specified polygonal regions. As mentioned above, our constructions go well beyond the convex hull property for Bézier curves. We describe each of these constructions in this section and we show in the subsequent section how to use these constructions to prove our main result for RAC drawings of 1-planar graphs, which is the following.

► **Theorem 1.** *Any n -vertex 1-planar graph has a 1-planar RAC drawing using a single cubic Bézier curve per edge. Further, if a 1-planar embedding of the graph has been provided, a 1-planar RAC drawing using such cubic Bézier curves can be computed in $O(n)$ time.*

Since Bézier curves have bounded curvature, we achieve a RAC drawing of any 1-planar graph using edges with bounded curvature. We give each of our constructions for constrained pairs of Bézier curves in the subsections that follow. Our constructions make use of another property of Bézier curves; namely, that applying an affine transformation (e.g., a rotation, reflection, translation, or scaling) to a Bézier curve is equal to the Bézier curve defined by applying that same transformation to the original control points, e.g., see [14].

2.1 Right-angle Crossing in a Triangle and Outside a Quadrilateral

Our first construction is for defining two cubic Bézier curves that have a right-angle crossing inside a triangle, each with two endpoints that form a quadrilateral with the base of the triangle, such that the curves lie outside of that quadrilateral. The curves that we describe are actually quadratic Bézier curves but any quadratic Bézier curve is also a cubic Bézier curve by de Casteljau's algorithm [7, 8]. See Figure 2.



■ **Figure 2** Triangle ABC with quadrilateral $ABEF$ and pair of cubic Bézier curves that cross in a right angle outside the quadrilateral but inside ABC .

► **Theorem 2.** *For any triangle ABC , there is a quadrilateral $ABEF$ contained in ABC such that there is a pair of Bézier curves with pairs of endpoints $\{A, E\}$ and $\{B, F\}$ that intersect each other at right angles, are contained within ABC , and lie outside of $ABEF$. The coordinates of E and F and the control points of the Bézier curves can be computed efficiently, given the coordinates of A , B and C .*

Proof. It is enough to consider the case of an isosceles triangle with base \overline{AB} , since such a triangle can be found within any given triangle ABD . By the equivalence property for Bézier curves under angle-preserving affine transformations, let us assume $A = (0, 0)$, $B = (1, 0)$, $C = (1/2, C_y)$. Let $E = \left(\frac{1}{4} \left(4C_y^2 + 6C_y + \sqrt{16C_y^4 + 48C_y^3 + 40C_y^2 + 12C_y + 1} + 3\right), C_y/2\right)$ with $C_y > 0$ and $F = (1 - E_x, C_y/2)$. Let us define $g_1(t) = At^2 + 2Ct(1 - t) + E(1 - t)^2$ and $g_2(t) = Bt^2 + 2Ct(1 - t) + F(1 - t)^2$. In the range $(0, 1)$, these curves only meet at one point,

$$\left(\frac{1}{2}, \frac{\left(-\sqrt{4C_y^2 + 8C_y + 1} + 6C_y + 1\right) \left(\sqrt{4C_y^2 + 8C_y + 1} + 2C_y - 1\right)}{24C_y}\right),$$

when $t = \frac{4C_y + 1}{6C_y} - \frac{\sqrt{4C_y^2 + 8C_y + 1}}{6C_y}$ for both curves. At this point, the slope of one curve is -1 and the other is $+1$, so they intersect at a right angle. At $t = \frac{\sqrt{-4E_x^2 + 8E_x - 3} - 2E_x + 1}{2(1 - E_x)}$, g_2 is at

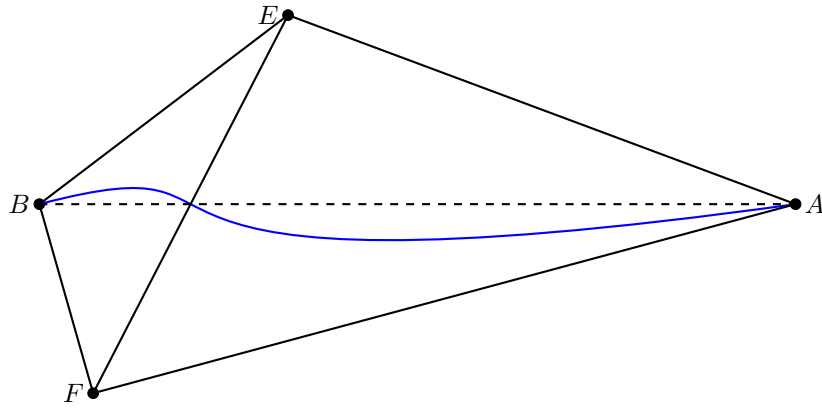
$$\left(E_x, -\frac{C_y \left(\sqrt{-4E_x^2 + 8E_x - 3} - 1\right) \left(3\sqrt{-4E_x^2 + 8E_x - 3} - 8E_x + 5\right)}{8(E_x - 1)^2}\right),$$

which is always higher than $E_y = C_y/2$. Thus, g_2 cannot intersect \overline{EB} or \overline{EF} except at B and F respectively because it is a parabola and E is inside it. Similarly, g_1 does not intersect \overline{AF} or \overline{EF} except at A and F respectively. ◀

2.2 Right-angle Crossing of a Diagonal of a Convex Quadrilateral

Our next construction is for a cubic Bézier curve f that replaces a diagonal of a convex quadrilateral so that f has a right-angle crossing with the other diagonal. See Figure 3.

We actually prove a slightly stronger result, for which, w.l.o.g., we assume the diagonal to be replaced is horizontal.



■ **Figure 3** Pair of intersecting diagonals, \overline{EF} and \overline{AB} , inside a convex quadrilateral, $AEBF$. We replace \overline{AB} with a Bézier curve that intersects \overline{EF} at a right angle.

► **Theorem 3.** For any convex quadrilateral with horizontal diagonal, \overline{AB} , a point X on \overline{AB} , and any real number, m , there is a simple cubic Bézier curve with endpoints A and B such that the Bézier curve intersects \overline{AB} at X and makes slope m at X and the curve is contained in the quadrilateral and also stays in a pair of opposite quadrants around X . For $m \neq 0$, the curve intersects \overline{AB} only at A , X , and B . The control points of the Bézier curve can be computed efficiently, given the vertices of the quadrilateral.

We will work with the points A and B being at $(1, 0)$ and $(0, 0)$ respectively. For any other pair of points we can apply an angle-preserving affine transformation. W.l.o.g., we also work with the point of intersection being on the line segment from $(0, 0)$ to $(8/9, 0)$, the portion of \overline{AB} closer to B , by symmetry. We will first show a curve that is perpendicular at X in Lemma 4. This first curve can be bounded by any arbitrary quadrilateral with \overline{AB} as diagonal and also leaves two opposite quadrants around X free for the other diagonal to be drawn as a straight line segment. Then, in Lemma 5, we show a curve that is a monotonic straight line that reaches X at the same value of the parameter, t , as defines the first curve. (See Figure 4.) Finally, in Lemma 6, we take a convex combination of these curves to get the required slope. This third curve is the one that we use.

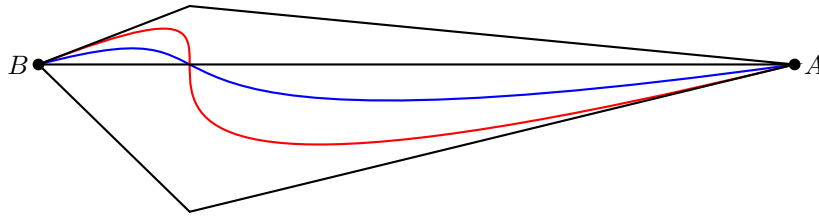
► **Lemma 4.** Let $A = (1, 0)$, $B = (0, 0)$, and $X = (x_0, 0)$ be a point on the line segment AB . For any convex quadrilateral with \overline{AB} as diagonal, there is a cubic Bézier curve with A and B as endpoints and C_1 and D_1 as control points that intersects the line segment \overline{AB} at X at a right angle at the parameter value $t = t_0 = \frac{C_{1,x} - 2D_{1,x}}{-1 + 3C_{1,x} - 3D_{1,x}}$, and is contained within this quadrilateral and also within any pair of opposite quadrants around the point of intersection.

Proof. Assume $x_0 < 8/9$. Otherwise mirror the plane. Let $D_1 = (D_{1,x}, -r)$ and $C_1 = (\frac{1}{2}(D_{1,x} - \sqrt{4D_{1,x} - 3D_{1,x}^2}), \frac{1 - 2C_{1,x} + D_{1,x}}{2D_{1,x} - C_{1,x}}r)$ where $r \in \mathbb{R}$. We will define $D_{1,x}$ later. Let $f_1(t) = At^3 + 3C_1t^2(1-t) + 3D_1t(1-t)^2 + B(1-t)^3$ be the Bézier curve we are constructing. Then the y -coordinate function for f_1 is

$$f_{1,y}(t) = -3 \frac{1 - 2C_{1,x} + D_{1,x}}{2D_{1,x} - C_{1,x}} rt^2(1-t) + 3rt(1-t)^2.$$

At $t = t_0$, $f_{1,y}(t) = 0$. Note that there are three roots for the cubic polynomial $f_{1,y}$ and the other two are at $t = 0$ and $t = 1$. The x -coordinate function for f_1 is

$$f_{1,x}(t) = t^3 + 3 \frac{1}{2} (D_{1,x} - \sqrt{4D_{1,x} - 3D_{1,x}^2}) t^2(1-t) + 3D_{1,x} t(1-t)^2.$$



■ **Figure 4** Bézier curves intersecting the line segment, \overline{AB} , between the endpoints at different angles. The red curve is a curve which meets the line segment at that point at right angles. The blue curve is obtained by taking a convex combination with a curve that remains on the line segment and has the same parameter t as the red curve at the point of intersection.

At $t = t_0$, $f_{1,x}(t) = \frac{D_{1,x}(3D_{1,x} + \sqrt{-D_{1,x}(3D_{1,x}-4)})}{3D_{1,x} + 3\sqrt{-D_{1,x}(3D_{1,x}-4)} + 2}$ and we choose $D_{1,x}$ to be $x_0 + \sqrt[3]{x_0^2 - x_0^3}$ which is a root of $f_{1,x}(t_0) = x_0$ when treated as an equation in $D_{1,x}$. This root is valid whenever $C_{1,x}$ is real, which happens when $4D_{1,x} - 3D_{1,x}^2 \geq 0$, which is true when $0 \leq D_{1,x} \leq 4/3$. This happens when $0 \leq x_0 \leq 8/9$. Moreover,

$$\frac{df_{1,y}}{dt} = -\frac{3r(C_{1,x}(9t^2 - 8t + 1) + D_{1,x}(-9t^2 + 10t - 2) + (2 - 3t)t)}{C_{1,x} - 2D_{1,x}} \quad \text{and}$$

$$\frac{df_{1,x}}{dt} = 3(t(C_{1,x}(2 - 3t) + t) + D_{1,x}(3t^2 - 4t + 1)).$$

At $t = t_0$,

$$\frac{df_{1,x}}{df_{1,y}} = \frac{\frac{df_{1,x}}{dt}}{\frac{df_{1,y}}{dt}} = 0,$$

which means that the angle at the point of intersection is $\pi/2$. Also, the value of $C_{1,x}$ is such that $\frac{df_{1,x}}{dt} = 3t^2(-3C_{1,x} + 3D_{1,x} + 1) + 3t(2C_{1,x} - 4D_{1,x}) + 3D_{1,x}$ is a quadratic polynomial in terms of t with discriminant 0. This means that t_0 is a repeated root and, hence, $f_{1,x}$ is monotonic. Thus, the curve remains in a pair of opposite quadrants, determined by the sign of r . Next, we show that this curve is bounded by the quadrilateral $(0, 0), (x_0, \frac{-rx_0}{D_{1,x}}), (1, 0), (x_0, \frac{r(1-2C_{1,x}+D_{1,x})(x_0-1)}{(C_{1,x}-1)(2D_{1,x}-C_{1,x})})$. This is obtained by the convex hull property of Bézier curves in conjunction with the fact that $f_{1,x}$ is monotonic. For any value of x_0 , these values are bounded. The value of r can be adjusted such that this quadrilateral is contained within any quadrilateral with the diagonal \overline{AB} . ◀

We next address the case for the point of intersection X having x -coordinate in $(0, 8/9)$ and forming an angle of 0.

► **Lemma 5.** *Let $A = (1, 0)$, $B = (0, 0)$, and $X = (x_0, 0)$ a point on the line segment \overline{AB} . There is a cubic Bézier curve with A and B as endpoints that is at X when $t = t_0$ as obtained from Lemma 4 for the same X and monotonically traces the straight line segment \overline{AB} .*

39:8 Drawing Planar Graphs and 1-Planar Graphs Using Cubic Bézier Curves

Proof. Assume $x_0 < 8/9$. Otherwise, mirror the plane. Let $C_2 = D_2 = (\frac{x_0 - t_0^3}{3(1-t_0)t_0}, 0)$ and $f_2(t) = At^3 + 3C_2t^2(1-t) + 3D_2t(1-t)^2 + B(1-t)^3$ be the Bézier curve we are constructing. It is easy to see that for f_2 's y -coordinate function, $f_{2,y} = 0$ throughout. The x -coordinate function is

$$f_{2,x}(t) = t^3 + \frac{(1-t)t(x_0 - t_0^3)}{(1-t_0)t_0}.$$

At $t = t_0$, $f_{2,x}(t) = x_0$. Also, C_2 and D_2 , the repeated control points are between A and B .

This means that the curve is always moving from B to A when t goes from 0 to 1, without turning back or overshooting A . ◀

Given the previous two lemmas, we now can use them in convex combination.

► **Lemma 6.** *Let $A = (1, 0)$, $B = (0, 0)$, $X = (x_0, 0)$ a point on the line segment \overline{AB} and m a real number. For any convex quadrilateral with \overline{AB} as a horizontal diagonal, there is a simple cubic Bézier curve with A and B as endpoints that intersects the line segment \overline{AB} at X with slope m for the parameter value $t = t_0$ as obtained from Lemma 4 for the same X and is contained within this quadrilateral and also within any pair of opposite quadrants around the point of intersection.*

Proof. Let $C_3 = kC_1 + (1-k)C_2$ and $D_3 = kD_1 + (1-k)D_2$ for $k \in [0, 1]$ where C_1, D_1 are control points of a curve $f_1(t)$ obtained from the proof of Lemma 4 and C_2, D_2 are control points of a curve $f_2(t)$ obtained from the proof of Lemma 5. Let

$$f_3(t) = At^3 + 3C_3t^2(1-t) + 3D_3t(1-t)^2 + B(1-t)^3 = kf_1(t) + (1-k)f_2(t)$$

be the more-general Bézier curve we are now constructing as a convex combination of f_1 and f_2 . Clearly, $f_3(t)$ is at X when $t = t_0$. In addition, we can write the slope of f_3 at t_0 as follows:

$$\frac{df_{3,y}}{df_{3,x}} = \frac{\frac{df_{3,y}}{dt}}{\frac{df_{3,x}}{dt}} = \frac{k \frac{df_{1,y}}{dt}}{k \frac{df_{1,x}}{dt} + (1-k) \frac{df_{2,x}}{dt}}.$$

To get a positive slope, we use a positive value of r . To get a negative slope, we use a negative value of r . Values of k in $[0, 1]$ will span the full range of slopes (either positive or negative) for every value of r . To get a specific slope m at $t = t_0$, we set

$$k = \frac{m \frac{df_{2,x}}{dt}}{m \left(\frac{df_{1,x}}{dt} - \frac{df_{2,x}}{dt} \right) + \frac{df_{1,y}}{dt}}.$$

Now, we need to show that this curve does not self-intersect. $f_{3,x}$ is monotonic as it is a convex combination of monotonic functions. This means that no x -coordinate is repeated and the curve does not self intersect. The curve, f_3 , is also bounded by the quadrilateral, $(0, 0), (x_0, \frac{-rx_0}{D_{1,x}}), (1, 0), (x_0, \frac{r(1-2C_{1,x}+D_{1,x})(x_0-1)}{(C_{1,x}-1)(2D_{1,x}-C_{1,x})})$, as it is a convex combination of the curves from Lemma 4 and Lemma 5, which are bounded by the same. The value of r can be adjusted such that this quadrilateral is contained within any arbitrary quadrilateral. Again this curve also remains in a pair of opposite quadrants around X since $f_{3,x}$ is monotonic. ◀

3 RAC Drawings of 1-Planar Graphs with Bézier Curves

We are now ready to prove Theorem 1, which is the following.

► **Theorem 1.** *Any n -vertex 1-planar graph has a 1-planar RAC drawing using a single cubic Bézier curve per edge. Further, if a 1-planar embedding of the graph has been provided, a 1-planar RAC drawing using such cubic Bézier curves can be computed in $O(n)$ time.*

The proof goes through three stages, adapting a proof of Bekos, Didimo, Liotta, Mehrabi, and Montecchiani [2] for RAC drawings of 1-planar graphs with 1-bend polygonal edges.

3.1 Augmentation

We start with a 1-plane combinatorial drawing G of the graph. We call every connected region of the plane bounded by edges and parts of edges a face. The number of such edges or parts is called the length of the face. The induction will be using triangulated 1-plane multigraphs, that is, 1-plane multigraphs in which every face is of length 3. For every pair of crossing edges ab, cd , add edges ac, cb, bd, ad such that the only edges contained within the cycle $acbd$ are ab and cd . We call the subgraph consisting of these edges an empty kite. If there are 2-length faces in this drawing, remove one of the edges recursively until there are no more. Also remove any parallel edge that was crossed. Now all faces in this drawing are either of length three with 2 vertices and a crossing point or bounded only by vertices and no crossing points. In every face longer than 3 add a new vertex and connect it to all the vertices on the face. We call this 1-plane multigraph G^+ .

3.2 Contraction

A separation pair is a pair of vertices $\{u, v\}$ whose removal disconnects the graph. Lemma 5 in [2] states that between any separation pair $\{u, v\}$, there exist two parallel edges e, e' such that $\{u, v\}$ is not a separation pair for the graph obtained by removing everything inside the cycle $\langle e, u, e', v \rangle$. We call this removed subgraph along with the cycle G_{uv} . Replace G_{uv} with a thick edge and iterate until there are no separation pairs. We call this graph G^* . G^* is a simple triangulated 1-plane graph.

3.3 Drawing

Obtain graph H^* by removing all crossing pairs in G^* . All the faces of H^* have either three or four vertices. Lemma 7 in [2] states that H^* is 3-connected. We can draw it with all faces convex and the outer face as a trapezoid or triangle using Tutte's method [35] or the method of Chiba *et al.* [6] to do it in linear time. Insert all the crossing edges in the interior four length faces by first drawing one of the edges as a straight line segment and then the other with the required slope using Theorem 3. If the outer face is of length four, use Theorem 2 with any large triangle.

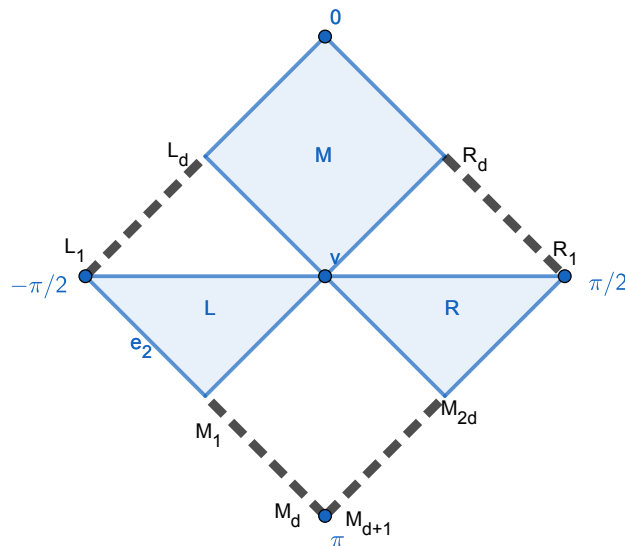
For any thick edge (u, v) with a triangle uvx that G_{uv} could be contained in (here x need not be a vertex, it could be any point), consider H_{uv} obtained by removing all crossing pairs. This is also 3-connected like H^* . If the outer face of H_{uv} is of length three and of the form uvw , recursively draw it inside uvx . If the outer face is of length four, use Theorem 2 to obtain the trapezoid and crossing pair and continue recursively.

4 Drawing Planar Graphs Using Bézier curves

Suppose we are given an n -vertex planar graph G . In this section, we describe a method of drawing G with asymptotically optimal angular resolution, $\Theta(1/\text{degree}(v))$, for each $v \in G$, using a single cubic Bézier curve of curvature $O(\sqrt{n})$ for each edge. We start from an $O(n) \times O(n)$ grid drawing D with asymptotically optimal angular resolution obtained by algorithm `OneBend` from Cheng, Duncan, Goodrich, and Kobourov [4], which uses one-bend edges. We describe some properties of this drawing in Section 4.1. We describe our new drawing using Bézier curves in Section 4.2. We show that the edges do not cross each other in Section 4.2.1. We show that the vertices in our drawing have asymptotically optimal angular resolution in Section 4.2.2. We show that the curvature of our drawing is $O(\sqrt{n})$ in Section 4.2.3.

4.1 The Drawing Obtained by the `OneBend` Algorithm

In the `OneBend` algorithm, G is drawn in a drawing, D , in an $O(n) \times O(n)$ grid such that every vertex, v , has a **joint box** – a square rotated $\pi/4$ of width and height $4\text{degree}(v) + 4$, centered at v . Each joint box is divided into six regions. (See Figure 5.)



■ **Figure 5** A joint box for a degree- d vertex, v , with left ports, L_1, \dots, L_d , right ports, R_1, \dots, R_d , and middle ports, M_1, \dots, M_{2d} , and free regions, L , M , and R . Image from [4].

The regions of a joint box are of two types, free regions and port regions. The free regions are as follows – M is $\pi/4$ on either side of the top corner and L (resp., R) is $\pi/4$ below the left (resp., right) corner. The port regions are as follows – M is opposite the free M region, L (resp., R) is between the free M and L (resp., R) regions. The sides of the square on port regions have d evenly spaced ports. Every edge in the drawing is between a free M region and a port M region, a free L region and a port R region, or a free R region and a port L region. Every edge is drawn as two line segments starting at the endpoints of the edge and meeting at a distinct port on the port region [4].

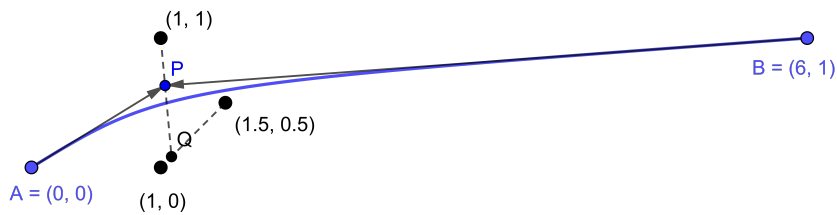
In our construction, we treat each of the M regions as two regions such that we have 8 regions which are congruent. However, only one of the two free M regions will have an edge.

4.2 Drawing a Planar Graph using Cubic Bézier Curves

We take the same embedding of the vertices as given by the `OneBend` algorithm. Since these curves can be rotated, translated and reflected without changing the curvature, for any edge (A, B) which is a 1-bend polyline with the port region at A and free region at B , without loss of generality, we may assume the following: A is at the origin, $(0, 0)$, and $B = (b_1, b_2)$ is in the region bounded below by the x -axis and above by the line $y = x - 1$. After this transformation the edges through the same port region of A in the drawing obtained by the `OneBend` algorithm are ported through $(d + 1, d + 1) + (i, -i)$ for distinct values of i which follow the same order along the vertex and $1 \leq i \leq d$. In our construction, we replace the polyline edges of the `OneBend` algorithm with cubic Bézier curves of the form

$$\gamma(t) = Bt^3 + 3Pt^2(1 - t) + 3Qt(1 - t)^2,$$

where $P = (1 - k)(1, 1) + kQ$, $Q = (1 - s)(1, 0) + s(3/2, 1/2)$, $s = b_2/b_1$, and $k = i/(d + 1)$. The repeated control point, P , is a convex combination of $(1, 1)$ and Q depending on the parameter, k . Q , in turn, is a convex combination of $(1, 0)$ and $(3/2, 1/2)$, depending on the angle b_2/b_1 that B makes with the x -axis. See Figure 6.



■ **Figure 6** Bézier curve for an edge, (A, B) .

4.2.1 Planarity

In this subsection, we show that the cubic Bézier curves representing edges with a common endpoint do not cross each other. First we show that edges that meet at a common endpoint but through different ports do not intersect. We show that an edge through the ported R region does not intersect with an edge through the free R region or the free M region of the same vertex. A similar argument shows that an edge through the ported L region or either half of the ported M region does not intersect with an edge through an adjacent region.

► **Lemma 7.** $\gamma(t)$ lies in the region $x > y > 0$ for $t \in (0, 1)$.

Proof. The proof follows from the convex hull property of Bézier curves and the fact that this region itself is convex. The control points except A are within this region. Only A could be on the boundary of the region. Since all points on the curve except for the endpoints will have a positive coefficient for the two intermediate control points, these points will lie within the region. ◀

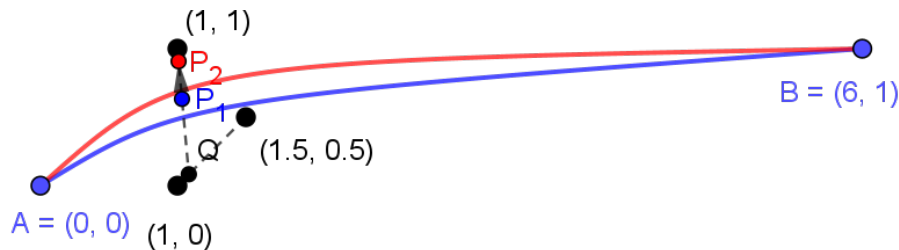
In other words, an edge ported through the R region will remain in the wedge of this R region extended to infinity. Let us next consider whether such an edge will intersect another one in the free M region. The control points of any edge through the free M region lie in the region $y \geq x$ and therefore can't intersect with γ . The control points of any edge through the free R region lie in the region $y \leq 0$ and therefore can't intersect with γ .

39:12 Drawing Planar Graphs and 1-Planar Graphs Using Cubic Bézier Curves

Next we show that two curves through the same ported region do not intersect. We will show this for the ported R region but the argument is similar for ported L region and both halves of the ported M region. We actually prove a stronger result, namely, that even if we allowed for parallel edges with the same two endpoints, they would not cross.

► **Lemma 8.** *Suppose γ_1 and γ_2 are two edge curves with the same endpoints using different values of k , $k_1 > k_2$, respectively. The curves γ_1 and γ_2 do not intersect except at the endpoints, and, except at the endpoints, γ_2 is above γ_1 .*

Proof. For the common endpoint of γ_1 and γ_2 , which has the same $s = b_2/b_1$, the different values of k give different values of P along a line of slope $1 - 2/s$. For the values of B under consideration, $0 \leq s < 1$ and hence the slope of this line is less than -1 . (See Figure 7.) Consider a rotation such that this line is now perpendicular to the x -axis and the two corresponding values of P such that P_2 is above P_1 . All points of γ_2 would lie directly above the corresponding points of γ_1 . Both curves are monotonic in the horizontal direction since the repeated control point is between the two endpoints in the horizontal direction. The curves do not intersect except at the endpoints. γ_2 is above γ_1 . ◀



■ **Figure 7** Parallel edges would not cross. As k decreases, all point except the endpoints move up in a direction parallel to P_1P_2 . The curve itself moves monotonically in the perpendicular direction with t .

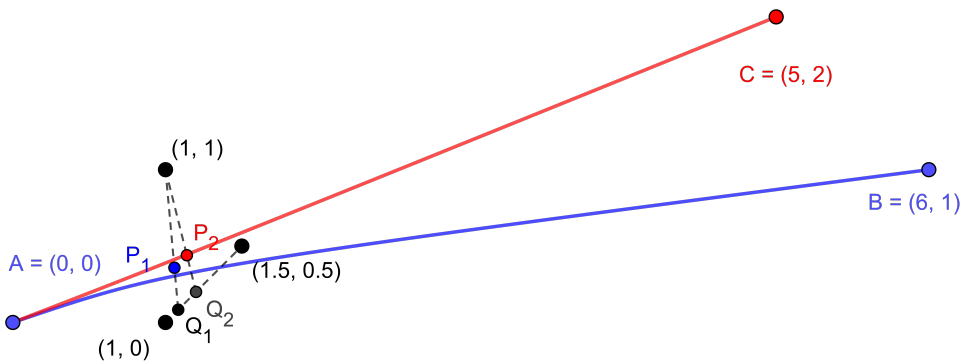
We next show that two curves with exactly one common endpoint and ported through the same R region do not cross.

► **Lemma 9.** *Let γ_1 and γ_2 be two curves with the same value of k but different vertices B and C which are endpoints of edges incident on A through the same region in the **OneBend** algorithm, with B coming counterclockwise first. The curves intersect only at A and γ_2 lies above γ_1 everywhere else.*

Proof. Let P_1 and P_2 be the corresponding values of P . Since the 1-bend edge BC has to pass through a pair of ports as described in Section 4.1, the straight line BC has to make an angle between $\pi/4$ and $3\pi/4$ with the x -axis. When moving from B to C , s increases or remains the same. This means that P_2 is the same as P_1 or is further away along a line making angle $\pi/4$ with the x -axis. (See Figure 8.) Consider the transformation $\{x, y\} \mapsto \{x - y, x + y\}$. Within this proof, unless noted otherwise, everything will be considered under this transformation. For any value of P , the x -coordinate is k , which is between 0 and 1 and the y -coordinate is $2 - k(1 - s)$ which is between 1 and 2. A has both coordinates less than those of P_1 and P_2 . B and C have both coordinates greater than those of P_1 and P_2 respectively. This means that both curves are monotonic along both axes. $C_x \leq B_x$ and $C_y \geq B_y$ with at least one of the inequalities strict. Suppose there are t_1, t_2 such that $\gamma_1(t_1) = \gamma_2(t_2)$. $\gamma_2(t_1)_y \geq \gamma_1(t_1)_y = \gamma_2(t_2)_y$ with equality only when

$B_y = C_y$. Monotonicity implies that $t_1 \geq t_2$. Similarly, $\gamma_2(t_1)_x \leq \gamma_1(t_1)_x = \gamma_2(t_2)_x$ with equality only when $B_x = C_x$. Monotonicity implies that $t_1 \leq t_2$. The curves do not intersect unless $B = C$ which is not the case. γ_2 lies above γ_1 because $\gamma_2(t)_y \geq \gamma_1(t)_y$ and the curves do not intersect. ◀

For two curves through the same ported region, they will have different endpoints and different values of k . The curve with the same value of k as the upper edge and the endpoint the same as the lower edge, lies below the curve representing the upper edge and above the curve representing the lower edge. Therefore, the curves representing the upper and lower edges do not intersect.



■ **Figure 8** When the endpoint changes from B to C and k remains the same, all points except A move in the direction P_1P_2 which is the same as Q_1Q_2 or remain where they are. In the perpendicular direction, they move to the left, away from γ_1 .

4.2.2 Angular Resolution

We next show that for each vertex, $v \in G$, in this drawing, we achieve angular resolution $\Omega(1/\text{degree}(v))$.

► **Lemma 10.** *Let $v \in G$ be a vertex with degree $\text{degree}(v)$. The angular resolution of v in our drawing of G using cubic Bézier curves is $\Omega(1/\text{degree}(v))$.*

Proof. Without loss of generality, let us assume that the vertex v is at $A = (0, 0)$ and has degree d . We will consider the angle between any pair of adjacent edges, at least one of which is ported through the R region. The other cases are similar. We first consider the case when both edges are ported through the R region. The tangent of each Bézier curve for an edge incident to v , at the point of contact is the same direction as the line segment between A and the control point P . For a pair of curves with the same pair of endpoints but different values of P , that is, P_1 and P_2 due to different values of k due to different values of i , consider the triangle P_1AP_2 . P_1P_2 has length at least $1/(\sqrt{2}d)$. From the law of sines,

$$\frac{1/(\sqrt{2}d)}{\sin \angle P_1AP_2} = \frac{P_1A}{\sin \angle P_1P_2A},$$

$$\frac{1}{\sqrt{2}d \sin \angle P_1AP_2} \leq \frac{\sqrt{10}/2}{1/\sqrt{2}}, \quad \text{and}$$

$$\sin \angle P_1AP_2 \geq \frac{1}{\sqrt{10}d}.$$

From, the Taylor expansion of \sin^{-1} ,

$$\angle P_1AP_2 \geq \frac{1}{\sqrt{10d}}.$$

Thus, we have $\Omega(1/d)$ angle between this pair of parallel edges. From Lemma 9, we know that the second curve would be above these edges and would have an even larger angle. Note that these curves also leave at least the same angle with the boundaries of ported R region since we use $k = i/(d+1)$ and i ranges from 1 to d and 0 and 1 correspond to the corners. This means that this minimum angle is maintained with any edge incident to v through any other region as well. \blacktriangleleft

4.2.3 Curvature

We next show that the curvature of our construction is $O(\sqrt{n})$.

► **Lemma 11.** *The curvature of γ is $O(\sqrt{b_1}) \subseteq O(\sqrt{n})$.*

Proof. We denote the curvature by κ . By the definition of curvature,

$$\frac{\partial(\kappa^2/b_1)}{\partial b_1} = \frac{8(t-1)^2t^2(k(s^2-s+2)+2(s-1))^2 \left(\begin{array}{c} -2(5b_1^2(s^2+1)t^4 - 8b_1(s+1)t^3 + 4t^2(b_1s+b_1-2) + 8t-2) \\ +4k(s-1)(2t-1)(b_1st^2+2t-1) + k^2(s^2-2s+2)(1-2t)^2 \end{array} \right)}{9 \left(\begin{array}{c} 2b_1^2(s^2+1)t^4 - 2k(s-1)(2t-1)(b_1st^2-4t+2) \\ -8b_1(s+1)t^3 + 4t^2(b_1s+b_1+4) + k^2(s^2-2s+2)(1-2t)^2 - 16t+4 \end{array} \right)^4}.$$

In terms of b_1 , the numerator of $\frac{\partial(\kappa^2/b_1)}{\partial b_1}$ is a quadratic polynomial with the coefficient of the degree-2 term being negative. The denominator is a fourth power and hence always non-negative. For values of b_1 , s , k and t that we care about, $\frac{\partial(\kappa^2/b_1)}{\partial b_1}$ has at most one root,

$$\frac{-(2t-1) \left(\begin{array}{c} \sqrt{2k^2(7s^4-14s^3+17s^2-10s+10) + 8k(7s^3-5s^2+3s-5) + 8(7s^2+4s+7)} \\ -2ks^2 + 2(k-2)s - 4 \end{array} \right)}{10(s^2+1)t^2},$$

if $t < 0.222$ and no roots otherwise. It is negative everywhere except possibly before this root. In addition, κ^2/b_1 decreases as b_1 increases after this point for fixed k , s and t , because this a quadratic with a negative leading coefficient. At $b_1 = 4$, the curvature is

$$\frac{128(t-1)^2t^2(k(s^2-s+2)+2(s-1))^2}{9 \left(\begin{array}{c} k^2(s^2-2s+2)(1-2t)^2 \\ -4k(s-1)(2t-1)(2st^2-2t+1) \\ +4(8(s^2+1)t^4 - 8(s+1)t^3 + 4(s+2)t^2 - 4t+1) \end{array} \right)^3}.$$

This is less than 3 for all values of s , k and t between 0 and 1. We evaluate κ^2/b_1 at the possible root of $\frac{\partial(\kappa^2/b_1)}{\partial b_1}$ and get the following:⁴

⁴ We verified this expression, as well as the others in this proof, using Mathematica [27].

$$\frac{12500 \left(\frac{(s^2 + 1)^2 (k(s^2 - s + 2) + 2(s - 1))^2}{2ks^2 - 2(k - 2)s + 4} \right) (t - 1)^2}{243 \left(\begin{array}{l} \left(\frac{4k^2 (2s^4 - 4s^3 + 7s^2 - 5s + 5)}{2k^2 (7s^4 - 14s^3 + 17s^2 - 10s + 10)} \right. \\ \left. + k(s - 1) \left(s \left(\sqrt{\frac{2k^2 (7s^4 - 14s^3 + 17s^2 - 10s + 10)}{+8k (7s^3 - 5s^2 + 3s - 5) + 8 (7s^2 + 4s + 7)}} - 8 \right) \right) \right. \\ \left. + 2 \left(\begin{array}{l} \left(\frac{+32s^2 + 40}{2k^2 (7s^4 - 14s^3 + 17s^2 - 10s + 10)} \right. \right. \\ \left. \left. + \sqrt{\frac{2k^2 (7s^4 - 14s^3 + 17s^2 - 10s + 10)}{+8k (7s^3 - 5s^2 + 3s - 5) + 8 (7s^2 + 4s + 7)}} + 16s^2 + 16 \right) \right) \right) \end{array} \right)^3 (2t - 1)^5.$$

This is a product of two expressions, one independent of t and the other entirely in t . The magnitude of the expression in t is bounded by 12 for all $0 \leq t \leq 0.222$. The magnitude of the other expression is bounded by $1/128$ for all values of s and k between 0 and 1. Putting these together we get that κ^2/b_1 is bounded by $12/128$. The curvature of these curves is $O(\sqrt{b_1})$ and hence $O(\sqrt{n})$ since b_1 is $O(n)$. ◀

To sum up, we have the following theorem.

► **Theorem 12.** *Given an n -vertex planar graph, G , we can draw G in an $O(n) \times O(n)$ grid and $\Omega(1/\text{degree}(v))$ angular resolution, for each vertex $v \in G$, using a single cubic Bézier curve with curvature $O(\sqrt{n})$ per edge in $O(n)$ time.*

5 Conclusion

In this paper, we have studied methods for drawing 1-planar and planar graphs using cubic Bézier curves with bounded curvature. Possible directions for future work and open problems include the following:

- Can the curvature for drawing an n -vertex planar graph using a single Bézier curve for each edge be improved from $O(\sqrt{n})$ while still maintaining an angular resolution of $\Omega(1/\text{degree}(v))$, for each vertex v in the drawing?
- Our algorithm used to produce a RAC drawing of an n -vertex 1-planar graph G , given a combinatorial 1-planar drawing of G , is based on the recursive construction of Bekos, Didimo, Liotta, Mehrabi, and Montecchiani [2]. This allows us to achieve bounded curvature for each edge in the drawing, but it does not give us a bound on the curvature in terms of n . Is it possible to achieve such a bound?
- Our algorithm used to produce a RAC drawing of an n -vertex 1-planar graph uses “S”-shaped Bézier curves. Can the same result be achieved with “C”-shaped Bézier curves, e.g., quadratic Bézier curves, which are arguably more aesthetically pleasing?
- As mentioned above, Lombardi drawings are drawings where edges are drawn using circular arcs, e.g., see [5, 12, 13, 16, 18, 29], but circular arcs are not Bézier curves. Can every graph with a Lombardi drawing also be drawn with the same edge crossings using a single Bézier curve of bounded curvature for each edge?

References

- 1 Patrizio Angelini, Michael A. Bekos, Henry Förster, and Michael Kaufmann. On RAC drawings of graphs with one bend per edge. *Theoretical Computer Science*, 828–829:42–54, 2020. doi:10.1016/j.tcs.2020.04.018.
- 2 Michael A. Bekos, Walter Didimo, Giuseppe Liotta, Saeed Mehrabi, and Fabrizio Montecchiani. On RAC drawings of 1-planar graphs. *Theoretical Computer Science*, 689:48–57, 2017. doi:10.1016/j.tcs.2017.05.039.
- 3 Ulrik Brandes and Dorothea Wagner. Using graph layout to visualize train interconnection data. *Journal of Graph Algorithms and Applications*, 4(3):135–155, 2000. doi:10.1007/3-540-37623-2_4.
- 4 C. C. Cheng, Christian A. Duncan, Michael T. Goodrich, and Stephen G. Kobourov. Drawing planar graphs with circular arcs. *Discrete & Computational Geometry*, 25:405–418, 1999. doi:10.1007/s004540010080.
- 5 Roman Chernobelskiy, Kathryn I. Cunningham, Michael T. Goodrich, Stephen G. Kobourov, and Lowell Trott. Force-directed Lombardi-style graph drawing. In Marc van Kreveld and Bettina Speckmann, editors, *Graph Drawing (GD)*, pages 320–331. Springer, 2012. doi:10.1007/978-3-642-25878-7_31.
- 6 Norishige Chiba. Linear algorithms for convex drawings of planar graphs. *Progress in graph theory*, pages 153–173, 1984.
- 7 Paul de Casteljau. Outillage methodes calcul. *Enveloppe Soleau 40.040*, 1959.
- 8 Paul de Casteljau. Courbes et surfaces à pôles. *Microfiche P 4147-1*, 1963.
- 9 Hubert De Fraysseix, János Pach, and Richard Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10:41–51, 1990. doi:10.1007/BF02122694.
- 10 Walter Didimo, Peter Eades, and Giuseppe Liotta. Drawing graphs with right angle crossings. *Theoretical Computer Science*, 412(39):5156–5166, 2011. doi:10.1016/j.tcs.2011.05.025.
- 11 Walter Didimo, Giuseppe Liotta, Saeed Mehrabi, and Fabrizio Montecchiani. 1-bend RAC drawings of 1-planar graphs. In Yifan Hu and Martin Nöllenburg, editors, *Graph Drawing and Network Visualization (GD)*, pages 335–343. Springer, 2016. doi:10.1007/978-3-319-50106-2_26.
- 12 Christian Duncan, David Eppstein, Michael Goodrich, Stephen Kobourov, and Martin Nöllenburg. Lombardi drawings of graphs. *Journal of Graph Algorithms and Applications*, 16(1):85–108, 2012. doi:10.7155/jgaa.00251.
- 13 Christian A. Duncan, David Eppstein, Michael T. Goodrich, Stephen G. Kobourov, Maarten Löffler, and Martin Nöllenburg. Planar and poly-arc Lombardi drawings. *Journal of Computational Geometry*, 9(1):328–355, 2018. doi:10.20382/jocg.v9i1a11.
- 14 Marsh Duncan. Bézier curves I. In *Applied Geometry for Computer Graphics and CAD*, pages 135–160. Springer, 2005. doi:10.1007/1-84628-109-1_6.
- 15 Peter Eades, Niklas Gröne, Karsten Klein, Patrick Eades, Leo Schreiber, Ulf Hailer, and Falk Schreiber. CelticGraph: Drawing graphs as Celtic knots and links. In Michael A. Bekos and Markus Chimani, editors, *Graph Drawing and Network Visualization (GD)*, pages 18–35. Springer, 2023. doi:10.1007/978-3-031-49272-3_2.
- 16 David Eppstein. Planar Lombardi drawings for subcubic graphs. In Walter Didimo and Maurizio Patrignani, editors, *Proc. 20th Int. Symp. Graph Drawing*, volume 7704 of *Lecture Notes in Computer Science*, pages 126–137. Springer, 2012. doi:10.1007/978-3-642-36763-2_12.
- 17 David Eppstein, Michael T Goodrich, and Jeremy Yu Meng. Confluent layered drawings. *Algorithmica*, 47:439–452, 2007. doi:10.1007/s00453-006-0159-8.
- 18 David Eppstein and Joseph Simons. Confluent Hasse diagrams. *Journal of Graph Algorithms and Applications*, 17(7):689–710, 2013. doi:10.7155/jgaa.00312.
- 19 Martin Fink, Herman Haverkort, Martin Nöllenburg, Maxwell Roberts, Julian Schuhmann, and Alexander Wolff. Drawing metro maps using Bézier curves. In Walter Didimo and Maurizio Patrignani, editors, *Graph Drawing (GD)*, pages 463–474. Springer, 2013. doi:10.1007/978-3-642-36763-2_41.

- 20 Benjamin Finkel and Roberto Tamassia. Curvilinear graph drawing using the force-directed method. In János Pach, editor, *Graph Drawing (GD)*, pages 448–453. Springer, 2005. doi:10.1007/978-3-540-31843-9_46.
- 21 Emden R. Gansner. Drawing graphs with Graphviz. Technical report, AT&T Bell Laboratories, 2009.
- 22 Ashim Garg and Roberto Tamassia. Planar drawings and angular resolution: Algorithms and bounds. In Jan van Leeuwen, editor, *Algorithms – ESA '94*, volume 855 of *Lecture Notes in Computer Science*, pages 12–23. Springer, 1994. doi:10.1007/BFB0049393.
- 23 Michael T. Goodrich and Christopher G. Wagner. A framework for drawing planar graphs with curves and polylines. *Journal of Algorithms*, 37(2):399–421, 2000. doi:10.1006/jagm.2000.1115.
- 24 Seok-Hee Hong, Peter Eades, and Marnijati Torkel. Gdot: Drawing graphs with dots and circles. In *2021 IEEE 14th Pacific Visualization Symposium (PacificVis)*, pages 156–165, 2021. doi:10.1109/PacificVis52677.2021.00029.
- 25 Weidong Huang, Peter Eades, Seok-Hee Hong, and Henry Been-Lirn Duh. Effects of curves on graph perception. In *2016 IEEE Pacific Visualization Symposium (PacificVis)*, pages 199–203, 2016. doi:10.1109/PACIFICVIS.2016.7465270.
- 26 Weidong Huang, Seok-Hee Hong, and Peter Eades. Effects of crossing angles. In *2008 IEEE Pacific Visualization Symposium*, pages 41–46, 2008. doi:10.1109/PACIFICVIS.2008.4475457.
- 27 Wolfram Research, Inc. Mathematica, Version 14.0. Champaign, IL, 2024. URL: <https://www.wolfram.com/mathematica>.
- 28 Goos Kant. Drawing planar graphs using the canonical ordering. *Algorithmica*, 16:4–32, 1996. doi:10.1007/BF02086606.
- 29 Philipp Kindermann, Stephen Kobourov, Maarten Löffler, Martin Nöllenburg, André Schulz, and Birgit Vogtenhuber. Lombardi drawings of knots and links. In *Graph Drawing and Network Visualization (GD)*, pages 113–126. Springer, 2017. doi:10.1007/978-3-319-73915-1_10.
- 30 Michael E. Mortenson. *Mathematics for Computer Graphics Applications*. Industrial Press Inc., 1999.
- 31 Walter Schnyder. Embedding planar graphs on the grid. In *1st ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 138–148, 1990. URL: <http://dl.acm.org/citation.cfm?id=320176.320191>.
- 32 Yusuke Suzuki. Re-embeddings of maximum 1-planar graphs. *SIAM Journal on Discrete Mathematics*, 24(4):1527–1540, 2010. doi:10.1137/090746835.
- 33 George B. Thomas and Ross L. Finney. *Calculus and Analytic Geometry*. Addison-Wesley, 9th ed. edition, 1996.
- 34 Csaba D. Tóth. On RAC drawings of graphs with two bends per edge. In Michael A. Bekos and Markus Chimani, editors, *Graph Drawing and Network Visualization (GD)*, pages 69–77. Springer, 2023. doi:10.1007/978-3-031-49272-3_5.
- 35 William T. Tutte. How to draw a graph. *Proceedings of the London Mathematical Society*, s3-13(1):743–767, 1963. doi:10.1112/plms/s3-13.1.743.
- 36 Kai Xu, Chris Rooney, Peter Passmore, Dong-Han Ham, and Phong H. Nguyen. A user study on curved edges in graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2449–2456, 2012. doi:10.1109/TVCG.2012.189.



Morphing Planar Graph Drawings via Orthogonal Box Drawings

Therese Biedl  

University of Waterloo, Canada

Anna Lubiw  

University of Waterloo, Canada

Jack Spalding-Jamieson  

University of Waterloo, Canada

Abstract

We give an algorithm to morph planar graph drawings that achieves small grid size at the expense of allowing a constant number of bends on each edge. The input is an n -vertex planar graph and two planar straight-line drawings of the graph on an $O(n) \times O(n)$ grid. The planarity-preserving morph is composed of $O(n)$ linear morphs between successive pairs of drawings, each on an $O(n) \times O(n)$ grid with a constant number of bends per edge. The algorithm to compute the morph runs in $O(n^2)$ time on a word RAM model with standard arithmetic operations – in particular no square roots or cube roots are required.

The first step of the algorithm is to morph each input drawing to a planar orthogonal box drawing where vertices are represented by boxes and each edge is drawn as a horizontal or vertical segment. The second step is to morph between planar orthogonal box drawings. This is done by extending known techniques for morphing planar orthogonal drawings with point vertices.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases morphing, graph morphing, linear morph, planar graph drawing, orthogonal box drawing, orthogonal drawing, algorithm

Digital Object Identifier 10.4230/LIPIcs.GD.2024.40

Related Version This work has appeared as the third author’s Master’s thesis.

Thesis: <https://hdl.handle.net/10012/20172>

Full Version: <https://arxiv.org/abs/2409.04074>

Funding *Therese Biedl:* Research supported by NSERC FRN RGPIN-2020-03958.

Anna Lubiw: Research supported by NSERC.

1 Introduction

Algorithms to compute a straight-line drawing of a planar n -vertex graph on an $O(n) \times O(n)$ grid have been known since the 1980’s [7, 12], but it is an open problem to achieve such straight-line small-grid results for planar graph morphing. To make this precise, let P and Q be two planar straight-line drawings of a graph G that are *compatible*, meaning that they have the same faces and the same outer face. A *linear morph sequence* from P to Q is a sequence of *explicit intermediate drawings* of G starting with P and ending with Q . By taking a *linear morph*, i.e., a linear interpolation of vertex positions, between each successive pair of explicit intermediate drawings, we obtain a continuous *piece-wise linear morph* from P to Q indexed by time $t \in [0, 1]$. The morph is *planarity-preserving* if the drawing at every time $t \in [0, 1]$ is planar. A straight-line drawing of a graph *lies on a grid* if the points representing the vertices lie at grid points; in case edges are drawn as poly-lines with bends, the bends must also lie at grid points. The following problem is open:



© Therese Biedl, Anna Lubiw, and Jack Spalding-Jamieson;
licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 40; pp. 40:1–40:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

► **Open Problem.** For a planar graph G with n vertices and a compatible pair of planar straight-line drawings P and Q of G on an $O(n) \times O(n)$ grid, is there a planarity-preserving piece-wise linear morph from P to Q where each explicit intermediate drawing is a straight-line drawing on an $O(n) \times O(n)$ grid?

The morphing algorithm of Alamdari, Angelini, Barrera-Cruz, Chan, Da Lozzo, Di Battista, Frati, Haxell, Lubiw, Patrignani, Roselli, Singla, and Wilkinson [1] – which is based on Cairns’ edge contraction method – finds a planarity-preserving linear morph sequence of length $O(n)$ but with no guarantee on the grid size of the explicit intermediate drawings. In fact, the vertex coordinates are computed using cube roots on a real RAM model, and vertices may become almost coincident to imitate edge contractions. The recent morphing algorithm of Erickson and Lin [9] achieves the same bound of $O(n)$ linear morphs for the subclass of 3-connected graphs and avoids the edge-contraction paradigm by following Floater’s method of interpolating the matrix of barycentric coordinates/weights. However, computing the barycentric weights requires square roots, and, even if that were avoided, the reliance on Tutte/Floater drawings means that the vertex coordinates can require $\Omega(n)$ bits of precision, and thus a grid of size $\Omega(2^n \times 2^n)$ [8]. The open problem was solved for the special case of Schnyder drawings by Barrera-Cruz, Haxell, and Lubiw [2].

Note that the Open Problem asks about the *existence* of a small-grid morph, regardless of algorithmic considerations. Of course, one would also like a fast algorithm to find the morph, and current research has been much more successful with regard to run-time than grid-size. The general morphing algorithm of Alamdari et al. [1] runs in $O(n^3)$ time. This was improved to $O(n^2 \log n)$, and to $O(n^2)$ for 2-connected graphs, by Klemz [10]. The morphing algorithm of Erickson and Lin [9], which involves solving linear systems and only applies to 3-connected graphs, runs in $O(n^{1+\omega/2})$ time, where $\omega < 2.371552$ is the matrix multiplication exponent [14]. Note that a runtime of $O(n^2)$ is optimal if all the explicit intermediate drawings must be given as output, since $\Omega(n)$ such drawings may be required [1].

We approach the Open Problem by insisting that explicit intermediate drawings lie on an $O(n) \times O(n)$ grid, but we relax the condition that edges be drawn as straight line segments, and allow a constant number of bends per edge. Our main result is as follows.

► **Theorem 1.** Let G be a connected planar graph with n vertices. For a compatible pair of planar straight-line drawings P and Q of G on an $O(n) \times O(n)$ grid, there exists a planarity-preserving linear morph sequence from P to Q of length $O(n)$, where each explicit intermediate drawing lies on an $O(n) \times O(n)$ grid and has $O(1)$ bends per edge. Moreover, this sequence can be found in $O(n^2)$ time.

This result improves the algorithm of Lubiw and Petrick [11] that finds a linear morph sequence of length $O(n^6)$ with explicit intermediate drawings on an $O(n^3) \times O(n^3)$ grid with $O(n^5)$ bends per edge and has run-time $O(n^6)$. Besides the bad bounds, the algorithm introduces bends at non-grid points, which our present result avoids.

For Theorem 1, the way we keep vertices of the explicit intermediate drawings on the grid is to impose an even stronger condition that every edge is drawn as a poly-line with all segments along grid lines (in particular, orthogonal) except for the two segments at its endpoints. The non-orthogonal segments incident to a vertex will live in an orthogonal rectangle called a **box**. Expressed differently, the first step of the algorithm is to morph each of the input straight-line drawings to a **planar orthogonal box drawing** where a vertex is represented by a box, and an edge is drawn as an **orthogonal poly-line**, a sequence of horizontal and vertical segments, joining two vertex boxes. See Figure 1. In this way, we reduce our problem to the problem of morphing planar orthogonal box drawings. In fact, we only need the special case where the input orthogonal box drawings have no bends.

Our second main result is an algorithm to find a planarity-preserving morph between two compatible planar orthogonal box drawings. For a precise definition of such a morph, see Section 1.1. When the drawings lie on a small grid with few bends, our analysis of grid size, bends, and run-time allows us to prove Theorem 1.

► **Theorem 2.** *Let G be a connected planar graph with n vertices. If P and Q are a compatible pair of planar orthogonal box drawings of G on an $O(n) \times O(n)$ grid with $O(1)$ bends per edge, then there exists a planarity-preserving linear morph sequence from P to Q of length $O(n)$ where each explicit intermediate drawing is an orthogonal box drawing that lies on an $O(n) \times O(n)$ grid with $O(1)$ bends. Moreover, this sequence can be found in $O(n^2)$ time.*

We prove Theorem 2 by building on algorithms to morph between planar orthogonal drawings where a vertex is drawn as a point and an edge is drawn as an orthogonal poly-line, a more limited setting since vertices must have degree at most 4. For these algorithms, n is the number of vertices plus bends. The algorithm of Biedl, Lubiw, Petrick, and Spriggs [4] uses $O(n^2)$ linear morphs and runs in $O(n^3)$ time. Van Goethem, Speckmann, and Verbeek [13] improved the number of linear morphs to $O(n)$ by performing operations simultaneously, however without a run-time analysis. We follow the two phases of Biedl et al.: (1) morph so that for each directed edge, its sequence of directions of edge segments (ignoring segment lengths) is the same in both drawings; and (2) morph between such “parallel” orthogonal graph drawings. We use the second phase as-is. However, we must do some work to extend the first phase to the setting of orthogonal box drawings, and to improve the number of linear morphs to $O(n)$ by performing many operations simultaneously as in Van Goethem et al., while keeping the grid size small and the run-time fast.

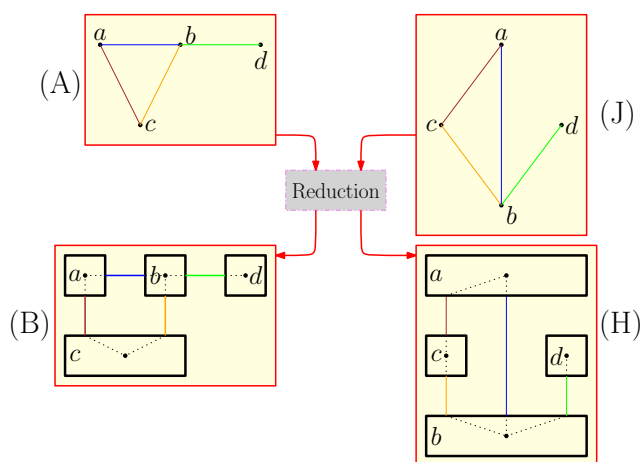
1.1 Preliminaries

Most graph drawing methods represent the vertices as points. In a *straight line* drawing an edge is drawn as a line segment between the vertex points, whereas in a *poly-line* drawing an edge is drawn as a simple poly-line – a non-self-intersecting path of line segments joined at *bends*.

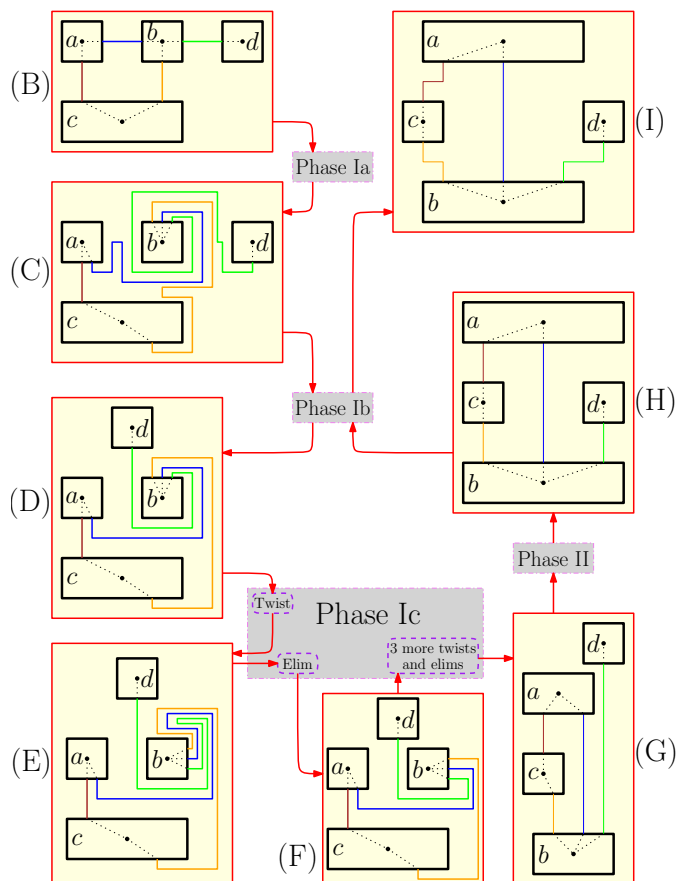
In a *linear morph* from drawing P to Q , the position of a *defining point* representing a vertex or bend is linearly interpolated between its position in P and in Q , so the point travels on a straight line at constant speed. A *unidirectional morph* is a special case of a linear morph where the lines along which the points move are all parallel. In a *horizontal [vertical] morph*, these lines are horizontal [vertical].

Our algorithm works with poly-line drawings, and the algorithm may add *degenerate bends* at existing vertices/bends or at a grid point along an edge segment. The algorithm may also delete such degenerate bends. We separate the steps of a linear morph sequence into: steps that add/delete degenerate bends; and steps that linearly interpolate between drawings that have the same number of bends along every edge, in which case their orders of appearance along the edge in the two drawings determines their correspondence.

In an *orthogonal point drawing* vertices are represented by points and edges by orthogonal poly-lines. In an *orthogonal box drawing* a vertex is represented by a positive area *box* with two horizontal sides and two vertical sides, and edges are represented by orthogonal poly-lines. The point where an edge attaches to a vertex box is called a *port*. An orthogonal box drawing is *planar* if there are no extraneous feature intersections: no coincident ports, no intersecting vertex boxes, no intersections between an edge and another edge, no intersection between an edge and a vertex box except at a port. One exception is that we will allow a port at a corner of a vertex box. For orthogonal box drawings, there is one more type of *degenerate bend* in addition to those defined above: a bend at a port.



■ **Figure 1** The two straight-line point drawings (A) and (J) are (respectively) morphed to admitted drawings of the orthogonal box drawings (B) and (H), by introducing two bends to each edge. A morph from (B) to (H) (see the relevant part of Figure 2) then induces a morph from (A) to (J).



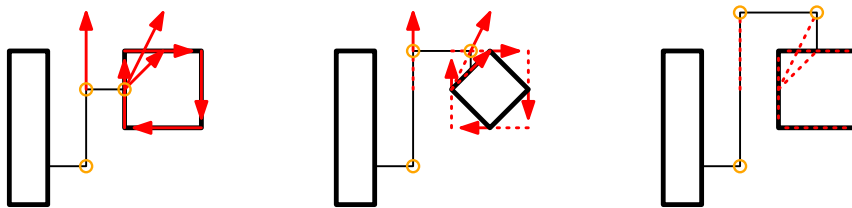
■ **Figure 2** Morphing between orthogonal box drawings where one drawing has bends. The input drawings are (B) and (I). Phase Ia morphs (B) to (C), port-aligned with (I). Phase Ib morphs (C) to (D) and (I) to (H), eliminating zig-zags. In (D) we need 4 clockwise twists of b to match spirality with (H). Twisting once morphs (D) to (E), and eliminating zig-zags gives (F). Three more twists and zig-zag eliminations give (G), which is morphed to the parallel drawing (H) in Phase II.

We use the following relationship between box drawings and poly-line drawings. Any planar orthogonal box drawing has a corresponding planar *admitted poly-line drawing*, where the vertex box is replaced by a vertex point at the center of the box joined by straight segments to the ports, which become bends in the edges. Thus an edge drawn with k bends in the orthogonal box drawing becomes an edge with $k + 2$ bends in the admitted drawing. Figure 2 shows extra segments for the admitted drawings with dotted lines.

A morph of an orthogonal box drawing D is specified in terms of its *defining points* which are the vertex box corners, the ports, and the edge bends. In a *linear morph* between two orthogonal box drawings the positions of the defining points are linearly interpolated. We restrict to *structure-preserving* linear morphs that meet the following conditions:

1. At every time during the morph the positions of the four corners of a vertex box determine a *vertex rectangle* which is a rectangle of positive area, though not necessarily with horizontal and vertical sides, and ports stay attached to their vertex rectangle.
2. For an edge from u to v , every segment along the edge remains orthogonal (horizontal or vertical) and does not change its direction (upward, downward, rightward or leftward). If a segment has length zero at a strictly intermediate time during the morph, then it has length zero throughout the morph.

By allowing non-axis-aligned vertex rectangles in condition (1) we can “twist” a square vertex box as shown in Figure 3¹. The figure shows a clockwise twist where each corner moves to the position of its clockwise neighbouring corner, so two of the corners move horizontally and the other two move vertically. Ports and bends may move non-orthogonally. Twists will be defined formally in Section 4.3.1. Apart from twists, all our morphs of orthogonal box drawings will in fact be horizontal or vertical.



■ **Figure 3** An example of a twist at a vertex box. Red arrows show motion to come; dashed portions show past motion.

A linear morph of an orthogonal box drawing is *planarity-preserving* if the drawing is planar at every time during the morph. When we say a *planarity-preserving linear morph of an orthogonal box drawing* we always mean that the morph is also structure-preserving. We prove in Section 3 that a planarity-preserving linear morph of an orthogonal box drawing induces a planarity-preserving linear morph of its admitted poly-line drawing.

We morph orthogonal box drawings using a sequence of intermediate goals. Two orthogonal box drawings of the same graph are *port-aligned* if, for each vertex v , when we compare the two vertex boxes representing v , then for every side (top, bottom, right left) the sequence of edges attached to that side of the vertex box is the same in both drawings. Two drawings without degenerate bends are *parallel* if they are port aligned and for every directed edge the two directed poly-lines representing the edge have the same sequence of left/right turns. To make drawings parallel, we use twists and zig-zag eliminations (to be defined), where a *zig-zag* is a sequence of two opposite turns, left then right, or right then left.

¹ This is similar to the “rotation” steps used to morph rectangular duals [5].

2 Algorithm Overview

This section gives the main steps of the algorithm that will prove Theorem 1. The input is a pair of compatible straight-line planar drawings of a graph G on n vertices.

► **Algorithm 1** (Reduction to orthogonal box drawings).

Morph each of the straight-line drawings to an admitted poly-line drawing of an orthogonal box drawing in which edges are drawn as single horizontal/vertical segments. See Figure 1 for an example. We prove in Lemma 3 that a morph of an orthogonal box drawing induces a morph of its admitted poly-line drawing, so this step reduces the problem to that of morphing orthogonal box drawings with zero bends per edge. Details of the reduction are in Section 3.

► **Algorithm 2** (Morphing between planar orthogonal box drawings).

The input consists of a pair of compatible planar orthogonal box drawings P and Q of a graph G on n vertices. P and Q may have bends, which is more general than the output of Algorithm 1. We describe steps of the morph “from both ends”, modifying P and Q until they meet in the middle, which is legitimate since linear morphs are reversible.

Phase I: Morph P and Q to become parallel. Building on the approach of Biedl et al. [4] for the case of orthogonal point drawings, we use the following steps:

Phase Ia: Morph P to become port-aligned with Q . This adds bends in edges of P .

Phase Ib: Morph P and Q to eliminate zig-zags. (In the situation required to prove Theorem 1, Q has no bends and thus no zig-zags.)

Phase Ic: Morph P so that for every directed edge the difference between its number of left and right turns (the “spirality”) is the same in P as in Q , while preserving port-alignment. This is accomplished via twists, which are interspersed with zig-zag eliminations. We use the property that if P and Q are port-aligned with matching spirality and no zig-zags or degenerate bends, then they are parallel.

Details of Phase I can be found in Section 4.

Phase II: Morph between the two parallel orthogonal box drawings. To do this, we imagine ports, corners, and bends as vertices and appeal directly to the result of Biedl et al. [4] for morphing parallel orthogonal point drawings. See the full version for details.

See Figure 2 for a sketch of all these steps. Although most of the detailed proofs that our morphs are planarity-preserving appear only in the full version, it is worth noting that, apart from twists (which are handled in Section 4.3.1) and the reduction to orthogonal box drawings, all of our morphs are horizontal or vertical, which simplifies the task. We prove in the full version that a horizontal morph between planar orthogonal box drawings P and Q is planarity-preserving so long as every horizontal line intersects the same sequence of defining points and edge/box segments.

3 Reduction to Orthogonal Box Drawings

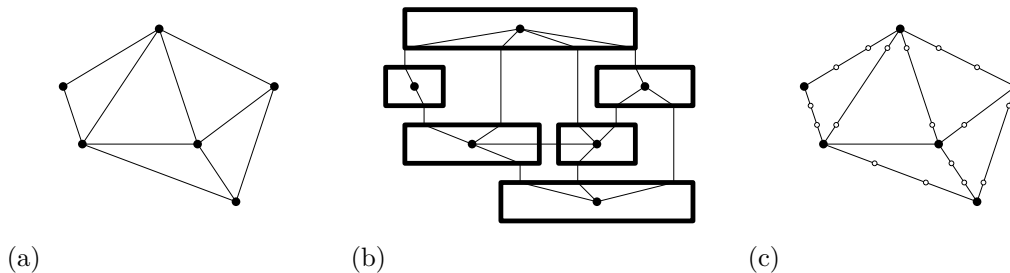
In this section we morph a straight-line planar graph drawing to an admitted poly-line drawing of an orthogonal box drawing, and we prove that a morph between two orthogonal box drawings induces a morph between their admitted drawings. In all cases “morph” means satisfying all our conditions, as detailed below. We begin with the second result:

► **Lemma 3.** *Let P and Q be planar orthogonal box drawings of the same graph. Let P' and Q' be the admitted planar poly-line drawings of P and Q , respectively. Suppose there is a planarity-preserving linear morph from P to Q . Then the linear morph from P' to Q' is also planarity-preserving.*

The idea is that a linear combination is preserved throughout a linear morph, and the center of a vertex box/rectangle is a linear combination (the average) of the four corners. For details, see the full version.

We now turn to the problem of morphing a straight-line planar drawing to an admitted drawing of an orthogonal box drawing.

► **Theorem 4.** *Let P be a planar straight-line drawing of a graph G with n vertices drawn on an $O(n) \times O(n)$ integer grid. Then there is a planar orthogonal box drawing D of G on an $O(n) \times O(n)$ grid with no bends, and there is a planarity-preserving linear morph sequence of length $O(n)$ from P to P' , the admitted poly-line drawing of D , where each explicit intermediate drawing lies on an $O(n) \times O(n)$ grid. Moreover, this sequence can be computed in $O(n^2)$ time.*



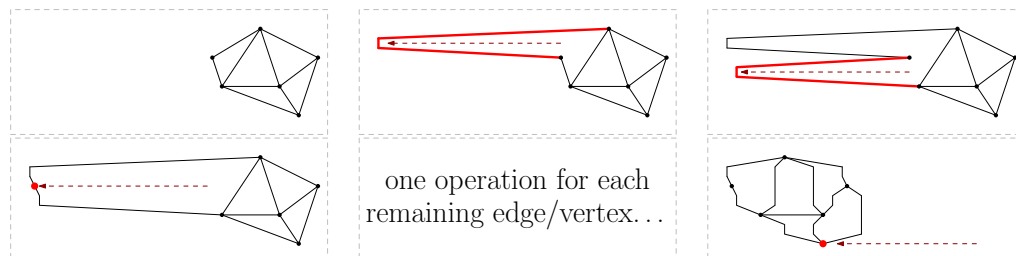
■ **Figure 4** (a) a straight-line drawing P ; (b) the corresponding orthogonal box drawing D and admitted drawing P' with every vertex at the same y -coordinate as in P ; (c) adding two bends to each edge of P would allow a horizontal morph to P' , but this is forbidden since the bends are not on the grid.

We prove Theorem 4 in two steps: (1) construct an orthogonal box drawing D with certain properties as described below; (2) show how to morph from P to P' .

For (1) a vertex v of P at y -coordinate c will be replaced in D by a vertex box that is skinny in the y -direction: we add a new grid line just below c and a new grid line just above c and construct a vertex box between these new grid lines. Adding the grid lines *refines* the grid in the y direction by a factor of 3. See Figure 4. The properties for D are that edges have no bends, each vertex has the same y -coordinate in P and in the refined grid of P' , and every horizontal line intersects the same sequence of edges in P and P' . We construct D in $O(n \log n)$ time by turning an existence result due to Biedl [3] into an efficient algorithm. Her result says that any planar straight-line drawing can be redrawn so that each vertex becomes a horizontal segment (or “bar”) with the same y -coordinate, and each non-horizontal edge becomes a vertical segment incident to the bars representing its endpoints, while maintaining the order of edges and vertices crossed by any horizontal line. Thickening the vertex bars to boxes gives the result we need. For details, see the full version.

For (2), we show how to morph from P to the admitted drawing P' . It would be straightforward to add degenerate bends where each edge crosses the newly added horizontal grid lines (see Figure 4(c)), and perform a single horizontal morph to move those bends to the position of the corresponding ports. However, our rule is that new bends can only be added

at grid points, so we are forced to a more complicated solution. Imagine P' drawn strictly to the left of P . Order the edges and vertices of P left-to-right, with priority given to edges. One by one, in this order, pull the edges and vertices of P to their positions in P' . To handle one edge of P we add two new bends at its endpoints, and morph them to the positions of the corresponding ports. See Figure 5 for a sketch of the algorithm and the full version for more details.



■ **Figure 5** Morphing from a straight-line drawing on the right to (the admitted drawing of) an orthogonal box drawing on the left by moving edges/vertices one-by-one. Each pane shows the newly morphed edge/vertex in red.

4 Morphing Orthogonal Box Drawings

This section contains the algorithm to morph between orthogonal box drawings (Algorithm 2). The algorithm works for any two compatible planar orthogonal box drawings, but the bounds needed to prove our main theorem (number of linear morphs, run-time, grid size, and number of bends) apply only when the input drawings lie on an $O(n) \times O(n)$ grid and have $O(1)$ bends per edge. Informally, we call these *small* drawings but lemma statements are precise. Let $f(D)$ denote the number of defining points of orthogonal box drawing D . A count of box corners, ports, and bends shows that $f(D)$ is $O(n)$ if the number of bends is $O(n)$.

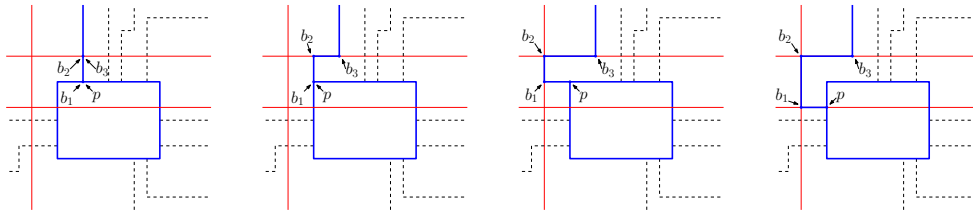
Recall from Section 2 that the algorithm has two phases. Phase I morphs the drawings P and Q to become parallel, and is described in this section. Phase II morphs between the resulting parallel drawings, and is deferred to the full version, since it follows directly from Biedl et al. [4].

Phase Ia, port alignment, is in Section 4.1. Port aligned drawings are parallel if each directed edge has the same sequence of left/right turns in both drawings. Following Biedl et al. [4], we define the *spirality* of a directed orthogonal poly-line to be its number of left turns minus its number of right turns² ignoring degenerate bends. An edge drawn without zig-zags has only left turns or only right turns. Thus if the two drawings of an edge have the same spirality, no degenerate bends, and no zig-zags, then they have the same sequence of left/right turns. We eliminate zig-zags in Section 4.2 and adjust spirality in Section 4.3. A main novel contribution is the twist operation for boxes in Section 4.3.1.

4.1 Phase Ia: Port Alignment

We morph P to be port-aligned with Q by successively “turning” a port around the corner of a vertex box as shown in Figure 6. Each turn operation uses $O(1)$ planarity-preserving linear morphs, and adds 3 bends and 3 grid lines. We prove in the full version that $O(1)$ such turn

² This is different from the definition of spirality in [13].



■ **Figure 6** A sketch of how a port can be morphed around a corner by introducing 3 bends b_i and 3 new grid lines (in red).

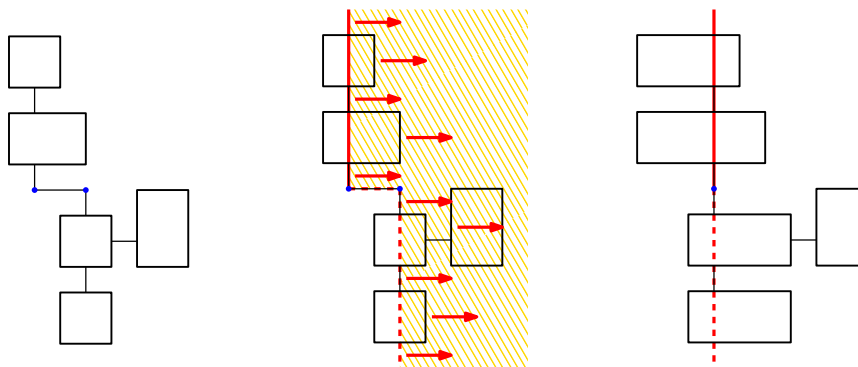
operations per edge suffice. For small drawings, we use in total $O(n)$ linear morphs, and remain on an $O(n) \times O(n)$ grid with $O(1)$ bends per edge. The run-time is $O(n^2)$. Note that a port may be coincident with a corner of a vertex box in explicit intermediate drawings. However, after Phase Ia, we never allow a port at a corner.

4.2 Phase Ib: Zig-zag Elimination

Zig-zag elimination is used in Phases Ib and Ic. It performs a sequence of horizontal and vertical morphs to get rid of all the zig-zags in an orthogonal box drawing, and produces a drawing D with grid size $f(D)$. This is $O(n)$ if each edge has $O(1)$ bends, but the exact bound $f(D)$ will be important when we apply zig-zag elimination $O(n)$ times in Phase Ic.

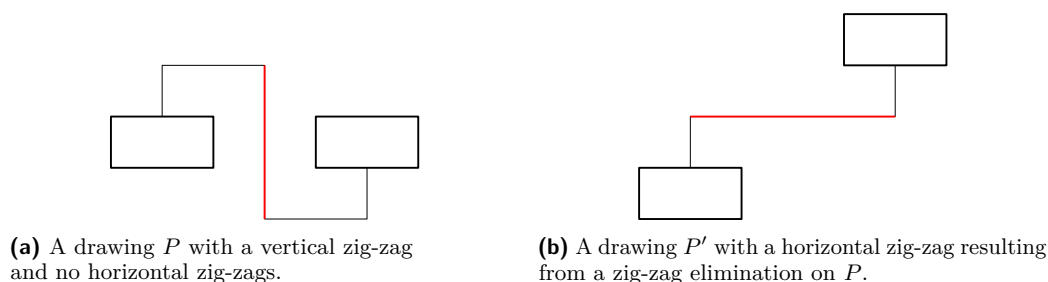
In Phase Ib, we apply zig-zag elimination once to P , and, if necessary, once to Q . Note that during Phase I, the drawing Q is altered only in Phase Ib, and is not altered at all in the situation arising from Theorem 1.

A zig-zag is *horizontal* [*vertical*] if the segment between the two turns is horizontal [vertical]. Biedl et al. [4] were the first to show that a horizontal linear morph can be used to eliminate one horizontal zig-zag in an orthogonal point drawing. Figure 7 shows the idea as applied to an orthogonal box drawing. Van Goethem et al. [13] observed that all horizontal zig-zags in a drawing P can be eliminated simultaneously with a single horizontal linear morph to a drawing P' . Their result also applies to orthogonal box drawings, but we must improve the grid size and runtime, see below.



■ **Figure 7** An orthogonal box drawing with a horizontal zig-zag, and the zig-zag elimination that converts its segment into two coinciding degenerate bends.

One could determine P' by pretending to perform $O(n)$ individual zig-zag eliminations in P , but this would be too slow and result in a large grid. Instead we compute P' via *compaction techniques*. We encode the relative x -order of all maximal vertical segments of



■ **Figure 8** An example of a vertical zig-zag elimination that results a new horizontal zig-zag.

a drawing P in an auxiliary directed acyclic graph $G_T(P)$ of size at most $f(P)$. Then we modify $G_T(P)$ to capture that the segments of horizontal zig-zags must vanish, and determine the target-drawing P' simply by computing longest paths in $G_T(P)$ to get the x -coordinates of defining points (y -coordinates remain unchanged). Graph $G_T(P)$ can be computed in $O(n \log n)$ time using a so-called trapezoidal map; the run-time can be reduced to $O(n)$ when P represents a connected graph using Chazelle's triangulation results [6]. The computed x -coordinates are at most the number of defining points of P' , not counting degenerate bends (which we are about to eliminate). More details of this morph to eliminate all horizontal zig-zags are given in the full version.

By repeatedly eliminating horizontal zig-zags and then eliminating vertical zig-zags we can eliminate all zig-zags, which yields the main result of this section:

► **Lemma 5.** *Let G be a planar graph with n vertices. Let P be a planar orthogonal box drawing of G on an $O(n) \times O(n)$ grid with $O(1)$ bends per edge. Then there is a planarity-preserving linear morph sequence of length $O(1)$ from P to a zig-zag-free orthogonal box drawing P' with the same port alignment and edge spiralities as P such that P' is drawn on an $f(P') \times f(P')$ (hence $O(n) \times O(n)$) grid. Furthermore, during the linear morph sequence, the number of bends never increases, and the width and height of the grid may increase but never beyond $f(P)$. Finally, the linear morph sequence can be computed in $O(n)$ time.*

Proof. Repeat the following four steps (a “round”) until no bends have been removed for an entire round:

1. eliminate all horizontal zig-zags with one horizontal morph as described above,
2. remove degenerate bends,
3. eliminate all vertical zig-zags with one vertical morph,
4. remove degenerate bends.

Note that we need multiple rounds in general because eliminating vertical zig-zags may create horizontal ones, and vice versa (see Figure 8). Except for the last round, each round eliminates at least one zig-zag – and thus at least two bends – from every edge that has zig-zags. No bends are ever added. Thus the number of rounds (hence also the number of morphs) is $O(1)$. Each drawing in the linear morph sequence can be computed in $O(n)$ time, so the run-time to compute all of them is also in $O(n)$.

Each morph preserves port alignment and edge spiralities. Steps 1 and 2 do not change the grid height and result in a drawing of width at most the (current) number of defining points, which is at most the initial number $f(P)$. Symmetrically, steps 3 and 4 do not change the grid width and result in a drawing of height at most the current number of defining points, which is at most $f(P)$. At the end, the final drawing P' lies on an $f(P') \times f(P')$ grid since the last round did not decrease the number of defining points. ◀

4.3 Phase Ic: Adjusting Spirality

In Phase Ic we morph P so that the spirality of each edge matches its spirality in Q . The basic operation we use is a “twist”, a morph that essentially turns all the ports around a vertex box: in a clockwise twist the ports on the top of the box become ports on the right, while ports on the right become ports on the bottom, and so on around the box. (Note that doing this by turning edges around corners as in Section 4.1 would require a linear number of morphs.) We describe twists in Section 4.3.1.

How do twists adjust spirality? We will see that a clockwise twist of a vertex box adds two left turns and one right turn to all edges leaving the vertex box. This guides the calculation of how many twists to apply at each vertex box. We also need a multiple of four twists for each vertex box in order to maintain port alignment. The question of how many twists to apply and what order to apply them in is addressed in Section 4.3.2. We also keep the drawing zig-zag free, and on a small grid.

4.3.1 Performing twists

In this section we focus on how to perform one twist simultaneously on a set of vertex boxes. We need a preliminary set-up step that makes all the vertex boxes square and clears some space around them. More precisely, the *k -proximal region* of the vertex box of vertex v consists of all points within L_∞ distance $k \cdot \deg(v)$ from the box, where $\deg(v)$ is the degree of v (thus the k -proximal region is a rectangle formed by enlarging the vertex box by $k \cdot \deg(v)$ on each side). The vertex boxes of a drawing are *k -spaced* if their k -proximal regions are disjoint and the only edge segments intersecting a k -proximal region are those incident to the vertex box. With a constant number of linear morphs we can make all vertex boxes square and 2-spaced (as shown in the first pane of Figure 9), while keeping the drawing port-aligned, and preserving the spirality of each edge. See the full version.

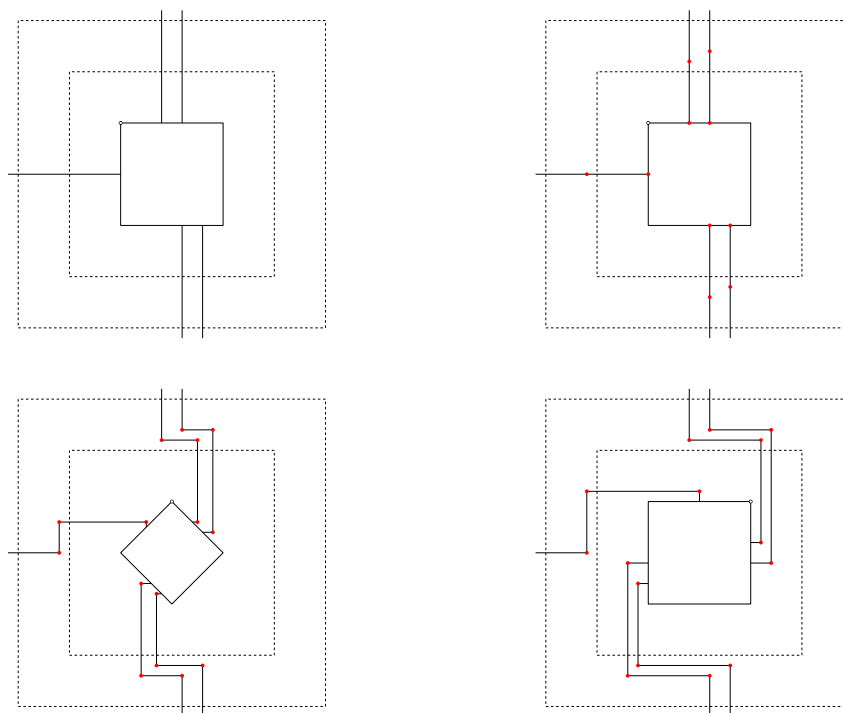
We next describe how to take an orthogonal box drawing that has square 2-spaced vertex boxes, and perform one twist of one vertex box. The twist will only alter the drawing inside the 2-proximal region of the vertex box, so we can twist multiple vertices simultaneously, see Lemma 7 below.

A *clockwise twist* operation is defined as follows, see Figure 9. (Counterclockwise twists are defined analogously.) Let v be a vertex of degree $\deg(v)$ and let S be its square vertex box. In the first step of the morph we add 3 degenerate bends in each edge incident to S . The second step is the actual linear morph. We give exact coordinates for the bends added to the i^{th} edge e_i (in clockwise order) that is attached to the top side of S , say at port p_i . Other coordinates are analogous. Bend $b_{i,1}$ is placed at p_i . Bends $b_{i,2}$ and $b_{i,3}$ are placed on the edge e_i , at distance $\deg(v) + i$ above p_i .

During the linear morph, each corner of S moves to the next clockwise corner. Each port on S is a linear combination of two box corners; this is maintained throughout the morph. It remains to describe the positions of the bends. As before, we give exact coordinates only for edge e_i attached to the top side. Let p'_i be the new position for port p_i . Bend $b_{i,3}$ does not move, so $b'_{i,3} = b_{i,3}$. Bend $b_{i,1}$ moves to a point $b'_{i,1}$ that is i units to the right of p'_i . Bend $b_{i,2}$ moves horizontally to a point $b'_{i,2}$ directly above $b'_{i,1}$.

► **Lemma 6.** *The clockwise twist is structure-preserving and planarity-preserving.*

Proof outline. We need only focus on the 2-proximal region of S . We first show that the twist is structure-preserving. Because S is square, it remains square throughout the morph. Also, each port stays attached to its side of S . The edge segment $p_i b_{i,1}$ remains horizontal,



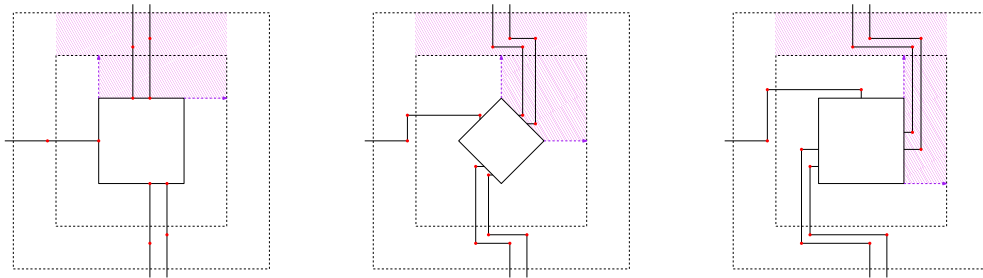
■ **Figure 9** Performing a clockwise twist. Bends are drawn as small red circles. The top left box corner (marked with a hollow circle) moves to the top right.

starting at length zero and ending at length i . Segment $b_{i,1}b_{i,2}$ remains vertical of positive length, and segment $b_{i,2}b_{i,3}$ remains horizontal, starting at length zero and ending at positive length.

We next show that the twist is planarity-preserving. It follows from the above that edges do not intersect the vertex box during the morph. It remains to show that no two edges intersect. For edges incident to the same side (say the top side) of S , observe that: the vertical segments $b_{i,1}b_{i,2}$ maintain their left-to-right ordering; the horizontal segments $b_{i,2}b_{i,3}$ remain at constant, distinct y -coordinates, and the horizontal segments $p_i b_{i,1}$ maintain their top-to-bottom ordering. For edges incident to different sides of S , we define four disjoint subregions of the 2-proximal region of S as shown in Figure 10, and argue that edges remain in their subregion as we morph the edges and the subregions (details are in the full version). ◀

Because a twist does not change the drawing outside the 2-proximal region of a vertex box, we can indeed (as promised before) twist multiple vertices simultaneously, which immediately gives the following:

► **Lemma 7.** *Let G be a connected planar graph with n vertices. Let P be a planar orthogonal box drawing of G drawn on an $O(n) \times O(n)$ grid with $O(1)$ bends per edge. Let $t : V(G) \rightarrow \{-1, 0, 1\}$ specify the twists to be performed at vertices, where -1 indicates a clockwise twist and 1 indicates a counterclockwise twist. Then there is a planarity-preserving sequence of length $O(1)$ from P to an orthogonal box drawing P' that effects the twists of t , where each explicit intermediate drawing is drawn on an $O(n) \times O(n)$ grid with $O(1)$ bends per edge. Moreover, the linear morph sequence can be computed in $O(n^2)$ time.*



■ **Figure 10** The subregion of the 2-proximal region of S corresponding to the top side of an initial box, throughout the morph.

4.3.2 Planning twists

There are two aspects to planning twists in P : (1) decide how many twists to perform at each vertex box; and (2) schedule the twists into rounds where each round twists each vertex box at most once, so that one round can be performed simultaneously as in Lemma 7. For (1) we solve a set of equations, the same as in Biedl et al. [4]. Step (2) is new, as are the improved bounds on the number of linear morphs and the grid size that complete Phase Ic.

Numbers of twists. We need some notation. For edge $e = uv$ and any orthogonal box drawing D , let $s_D(u, v)$ denote the spirality of the edge in D . Our goal is to morph P (while leaving Q unchanged) until for every edge uv its *spirality difference*, $\Delta s_P(u, v) := s_P(u, v) - s_Q(u, v)$, becomes zero. The final P must also be port-aligned with Q (equivalently, port-aligned with the initial P), although this need not be true of the explicit intermediate drawings. These two conditions together constitute the requirements for our twists.

Observe that a clockwise twist (as defined in the previous section) at u 's vertex box adds two left turns and one right turn to every edge leaving u , for a net change of $+1$ in the spirality of edges uv leaving u , and a net change of -1 for edges vu entering u . These are reversed for a counterclockwise twist. For example, if $\Delta s_P(u, v)$ is positive, we should apply counterclockwise twists to u and/or clockwise twists to v . For vertex v , let $t(v)$ be the number of twists to be performed on v 's vertex box in P , where a positive value indicates counterclockwise twists and a negative value indicates clockwise twists. Necessary and sufficient conditions to meet the requirements for our twists are:

$$\text{for each edge } uv \quad \Delta s_P(u, v) = t(u) - t(v) \quad (1)$$

$$\text{for each vertex } v \quad t(v) \equiv 0 \pmod{4} \quad (2)$$

► **Lemma 8.** *A solution to the above equations with $|t(v)| \in O(n)$ can be found in $O(n)$ time.*

Proof outline. The lemma was proved by Biedl et al. [4] for orthogonal point drawings. As a main ingredient, they prove that it suffices to impose condition (2) for a single vertex v_0 , and to impose condition (1) for the edges of a spanning tree T rooted at v_0 . The idea is that a non-tree edge e creates a cycle with T which corresponds to a face of the drawing. Condition (1) is then proved for e using the fact that a planar orthogonal cycle has 4 more right turns than left turns when traversed clockwise. We prove that this main ingredient carries over to orthogonal box drawings – the difference is that a face of the point drawing has edges meeting at vertices, whereas a face of a box drawing contains portions of box sides. See the full version.

40:14 Morphing Planar Graph Drawings via Orthogonal Box Drawings

Given this main ingredient, we assign $t(v_0) := 0$, and assign $t(v)$ to be $t(v_0)$ plus the sum of the spirality differences on edges of the path from v_0 to v in T . This takes $O(n)$ time, and gives $t(v) \in O(n)$ since the spirality difference of each edge is constant. ◀

Allocating twists to rounds. We allocate the twists $t(v)$ to rounds r_i , where i ranges from 1 to $k := \max_v |t(v)|$ so that each vertex box is twisted at most once in each round. For vertex v , we put its twists into rounds 1 through $|t(v)|$.

Algorithm for Phase Ic. Let $P'_0 := P$. For $i = 1, \dots, k$, perform the simultaneous twists of round r_i on input P'_{i-1} according to Lemma 7 to obtain P_i , and then do zig-zag elimination on P_i according to Lemma 5 to obtain P'_i . Output P'_k .

Analyzing the algorithm. Note that the algorithm does $O(n)$ linear morphs. It remains to bound the grid size, the number of bends, and the run-time for Phase Ic. Since zig-zag elimination is called $O(n)$ times it becomes important to give absolute bounds to avoid hiding a “growing constant” inside the $O(n)$ bound on grid size. The main idea is to analyze the number of bends, since that determines the grid size after zig-zag elimination.

► **Lemma 9.** *There are constants c and d , independent of i , such that P_i and P'_i , $i = 1, \dots, k$ lie on a grid of size $cn \times cn$ with at most d bends per edge.*

Proof. We first prove a bound on bends for P'_i for $i = 0, \dots, k$. The drawing P'_i is zig-zag-free so the number of bends on edge uv is the absolute value of its spirality, which is the same as in P_i since zig-zag elimination preserves spirality (Lemma 5). Our method of allocating twists to rounds ensures that the spirality of an edge stays the same except during a constant number of rounds where it improves. More formally:

▷ **Claim 10 (Proof in the full version).** After each round of simultaneous twists, the spirality of an edge remains the same or gets closer to its spirality in Q , i.e., $s_{P_i}(u, v)$ lies in the closed interval between $s_{P_{i-1}}(u, v)$ and $s_Q(u, v)$.

Thus $|s_{P_i}(u, v)| \leq \max\{|s_P(u, v)|, |s_Q(u, v)|\}$ which is bounded by some constant d' since edges in P and Q each have a constant number of bends.

We now turn to the grid size. For $i = 0, \dots, k$ drawing P'_i is the result of applying Lemma 5 and hence its grid size is $f(P_i) \times f(P'_i)$, where $f(P'_i)$ is the number of defining points of P'_i . Counting box corners, ports and bends, gives $f(P'_i) \leq 4n + 2m + d'm$, where G has $m \leq 3n - 6$ edges, so $f(P'_i) \leq c'n$ for some $c' \leq 10 + 3d'$. P_i is obtained from P'_{i-1} by applying one round of simultaneous twists as in Lemma 7. This may add a constant term to the bound d' on the number of bends, and may increase the bound $c'n \times c'n$ on the grid size by a constant factor, which gives the final values of d and c . ◀

We summarize Phase Ic as follows:

► **Lemma 11.** *Let G be a connected planar graph with n vertices. Let P and Q be compatible port-aligned zig-zag-free planar orthogonal box drawings of G on an $O(n) \times O(n)$ grid with $O(1)$ bends per edge. Then there is a planarity-preserving linear morph sequence of length $O(n)$ from P to a zig-zag-free orthogonal box drawing P' with the same port-alignment and edge spirality as Q (thus P' and Q are parallel) such that all explicit intermediate drawings are on an $O(n) \times O(n)$ grid with $O(1)$ bends per edge. Moreover, the linear morph sequence can be computed in $O(n^2)$ time.*

5 Conclusions

We have now assembled all the ingredients for proving our two main theorems. In summary, Theorem 2 (morphing planar orthogonal box drawings) is proved by the steps of Algorithm 2: Phase Ia (port-alignment) is sketched in Section 4.1, with details in the full version; Phase Ib (zig-zag elimination in both drawings) is handled by Lemma 5; Phase Ic (spirality) is handled by Lemma 11; and Phase II (morphing parallel drawings) simply appeals to [4] (details are in the full version).

Theorem 1 (morphing straight-line drawings on a small grid) is proved by the reduction to morphing orthogonal box drawings (see Theorem 4 in Section 3) plus Theorem 2.

In conclusion, our main result is an algorithm to morph compatible planar straight-line drawings with a linear number of linear morphs for which all explicit intermediate drawings lie on small grids, have few bends per edge, and can be computed quickly. Extending the result to disconnected graphs involves further details, and is left for future work.

The biggest remaining open problem is the one from the introduction: Is there a piece-wise linear morph with small explicit intermediate drawings *without* bends? If so, can we limit to a short linear morph sequence? And if not, can we at least limit the number of bends per edge to a very small constant, such as 1 or 2?

References

- 1 Soroush Alamdari, Patrizio Angelini, Fidel Barrera-Cruz, Timothy M Chan, Giordano Da Lozzo, Giuseppe Di Battista, Fabrizio Frati, Penny Haxell, Anna Lubiw, Maurizio Patrignani, Vincenzo Roselli, Sahil Singla, and Bryan T Wilkinson. How to morph planar graph drawings. *SIAM Journal on Computing*, 46(2):824–852, 2017. doi:10.1137/16M1069171.
- 2 Fidel Barrera-Cruz, Penny Haxell, and Anna Lubiw. Morphing Schnyder drawings of planar triangulations. *Discrete & Computational Geometry*, 61:161–184, 2019. doi:10.1007/s00454-018-0018-9.
- 3 Therese Biedl. Height-preserving transformations of planar graph drawings. In Christian Duncan and Antonios Symvonis, editors, *Graph Drawing: 22nd International Symposium, GD 2014, Würzburg, Germany, September 24-26, 2014*, pages 380–391, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg. doi:10.1007/978-3-662-45803-7_32.
- 4 Therese Biedl, Anna Lubiw, Mark Petrick, and Michael Spriggs. Morphing orthogonal planar graph drawings. *ACM Transactions on Algorithms (TALG)*, 9(4):1–24, 2013. doi:10.1145/2500118.
- 5 Steven Chaplick, Philipp Kindermann, Jonathan Klawitter, Ignaz Rutter, and Alexander Wolff. Morphing rectangular duals. In *Graph Drawing and Network Visualization: 30th International Symposium, GD 2022, Tokyo, Japan, September 13–16, 2022*, pages 389–403. Springer, 2023. doi:10.1007/978-3-031-22203-0_28.
- 6 Bernard Chazelle. Triangulating a simple polygon in linear time. *Discrete & Computational Geometry*, 6(3):485–524, 1991. doi:10.1007/BF02574703.
- 7 Hubert De Fraysseix, János Pach, and Richard Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10:41–51, 1990. doi:10.1007/BF02122694.
- 8 Giuseppe Di Battista and Fabrizio Frati. From Tutte to Floater and Gotsman: on the resolution of planar straight-line drawings and morphs. In *Graph Drawing and Network Visualization: 29th International Symposium, GD 2021, Tübingen, Germany, September 14–17, 2021*, pages 109–122. Springer, 2021. doi:10.1007/978-3-030-92931-2_8.
- 9 Jeff Erickson and Patrick Lin. Planar and toroidal morphs made easier. *Journal of Graph Algorithms and Applications*, 27(2):95–118, 2023. doi:10.7155/jgaa.00616.

40:16 Morphing Planar Graph Drawings via Orthogonal Box Drawings

- 10 Boris Klemz. Convex Drawings of Hierarchical Graphs in Linear Time, with Applications to Planar Graph Morphing. In Petra Mutzel, Rasmus Pagh, and Grzegorz Herman, editors, *29th Annual European Symposium on Algorithms (ESA 2021)*, volume 204 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 57:1–57:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ESA.2021.57.
- 11 Anna Lubiw and Mark Petrick. Morphing planar graph drawings with bent edges. *Journal of Graph Algorithms and Applications*, 15(2):205–227, 2011. URL: <https://jgaa.info/index.php/jgaa/article/view/paper223/2737>, doi:10.7155/JGAA.00223.
- 12 Walter Schnyder. Embedding planar graphs on the grid. In *SODA '90: Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 138–148, 1990. URL: <https://dl.acm.org/doi/pdf/10.5555/320176.320191>.
- 13 Arthur van Goethem, Bettina Speckmann, and Kevin Verbeek. Optimal morphs of planar orthogonal drawings. *Journal of Computational Geometry*, 13(1):263–297, 2022. doi:10.20382/jocg.v13i1a11.
- 14 Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. New bounds for matrix multiplication: from alpha to omega. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3792–3835, 2024. doi:10.1137/1.9781611977912.134.

Graph Drawing Contest Report

Sara Di Bartolomeo  

TU Wien, Austria

Fabian Klute  

Universitat Politècnica de Catalunya, Barcelona, Spain

Debajyoti Mondal  

University of Saskatchewan, Saskatoon, Canada

Jules Wolms  

TU Eindhoven, The Netherlands

Abstract

This report describes the 31st Annual Graph Drawing Contest, held in conjunction with the 32nd International Symposium on Graph Drawing and Network Visualization (GD'24) at TU Wien, Vienna, Austria. The mission of the Graph Drawing Contest is to monitor and challenge the current state of the art in graph-drawing technology. This year's edition featured two categories, a creative track in which participants visualized a dataset based on the Olympic medal track-record of countries and a live challenge held at the conference where participants had to draw a graph on a given point-set with as few crossings as possible.

2012 ACM Subject Classification Human-centered computing → Graph drawings; Human-centered computing → Visualization systems and tools; Human-centered computing → Visual analytics

Keywords and phrases Information Visualization, Graph Drawing Contest

Digital Object Identifier 10.4230/LIPIcs.GD.2024.41

Category Graph Drawing Contest Report

Acknowledgements The contest committee would like to thank the organizing and program committee of the conference; the organizers who provided us with a room with hardware for the live challenge and monetary prizes; the generous sponsors of the symposium; and all the contestants for their participation. Further details including all submitted drawings and challenge graphs can be found on the contest website. Finally, the contest committee wants to thank Wouter Meulemans for his help and support in running this years contest.

1 Introduction

Following the tradition of the past years, the Graph Drawing Contest was divided into two parts: the *creative topic* and the *live challenge*.

As in the 2023 edition, the creative contest focused on only one dataset this year. In this year the topic was *Medals Won by Countries at the Olympic Games*: The data consisted of countries and their gold, silver, and bronze medals won per category of sports. Additionally, the contestant had access to a wide set of metadata which we describe below. The data set was published about half a year in advance, and contestants submitted their visualizations before the conference started.

The live challenge took place during the conference in a format similar to a typical programming contest. Teams were presented with a collection of *challenge graphs* and had one hour to submit their highest scoring drawings. As is tradition, this year's topic was repeated from the last year: given an undirected simple graph and a point set, find straight-line drawing of the graph with the vertices drawn on top of the points such that the number of crossing edges is minimized.



© Sara Di Bartolomeo, Fabian Klute, Debajyoti Mondal, Wouter Meulemans, and Jules Wolms; licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 41; pp. 41:1–41:13

Leibniz International Proceedings in Informatics



LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

41:2 Graph Drawing Contest Report

Overall, we received 33 submissions: 9 submissions for the creative topics and 24 submissions for the live challenge (17 manual and 7 automatic).

2 Creative Topic

The dataset for the Creative Topic represents countries participating and their medals won at the modern Olympic games since 1896 and until 2020. In the final dataset each country formed a vertex and was connected via an edge to a category of sports if it ever won any medal in a discipline contained in that category. The categories we chose were:

- Athletics
- Boating
- Equestrian
- Fighting
- Gymnastics
- Racquet
- Shooting
- Swimming
- Teams
- Other

Similar to the directly preceding contests we decided to keep a large set of metadata attached to each edge. In contrast though, we explicitly asked contestants to formulate a question or hypothesis about the data and try to answer or explore it with their visualization.

Vertices came with an `id` and `name`. The ones representing countries additionally contained a `feel_noc` with the international abbreviation of that country. Every edge had a large record connected to it containing the following information for each medal won by the country in this category:

```
{
  "athlete": {
    "name": "<NAME OF ATHELETE>",
    "sex": "<MALE OR FEMALE>",
    "born": "<DATE OF BIRTH yyyy-mm-dd>",
    "height": "<HEIGHT IN cm OR na IF NOT AVAILABLE>",
    "weight": "<WEIGHT IN kg OR na IF NOT AVAILABLE>"
  },
  "sport": "<OLYMPIC SPORT>",
  "event": "<NAME OF EVENT>",
  "year": "<YEAR OF THE RESULT>",
  "city": "<CITY>",
  "medal": "<MEDAL TYPE>"
}
```

The total graph had 163 vertices and 700 edges.

The raw data forming the basis of the dataset was taken from a Kaggle repository.¹ Using Python scripts we formed the categories mentioned above and extracted the final dataset.

¹ <https://www.kaggle.com/datasets/josephcheng123456/olympic-historical-dataset-from-olympediaorg/data>

The general goal of the creative topic was to visualize the dataset with complete artistic freedom, and with the aim of answering a question or hypothesis about the data which the contestants were free to form themselves. For inspiration we provided examples on the contest website:

Temporal. How has the graph structure evolved over time with each Olympic event? Are there any noticeable changes in the graph's topology across different Olympic years?

Comparative. How do the subgraphs of male and female athletes differ in terms of structure and connectivity?

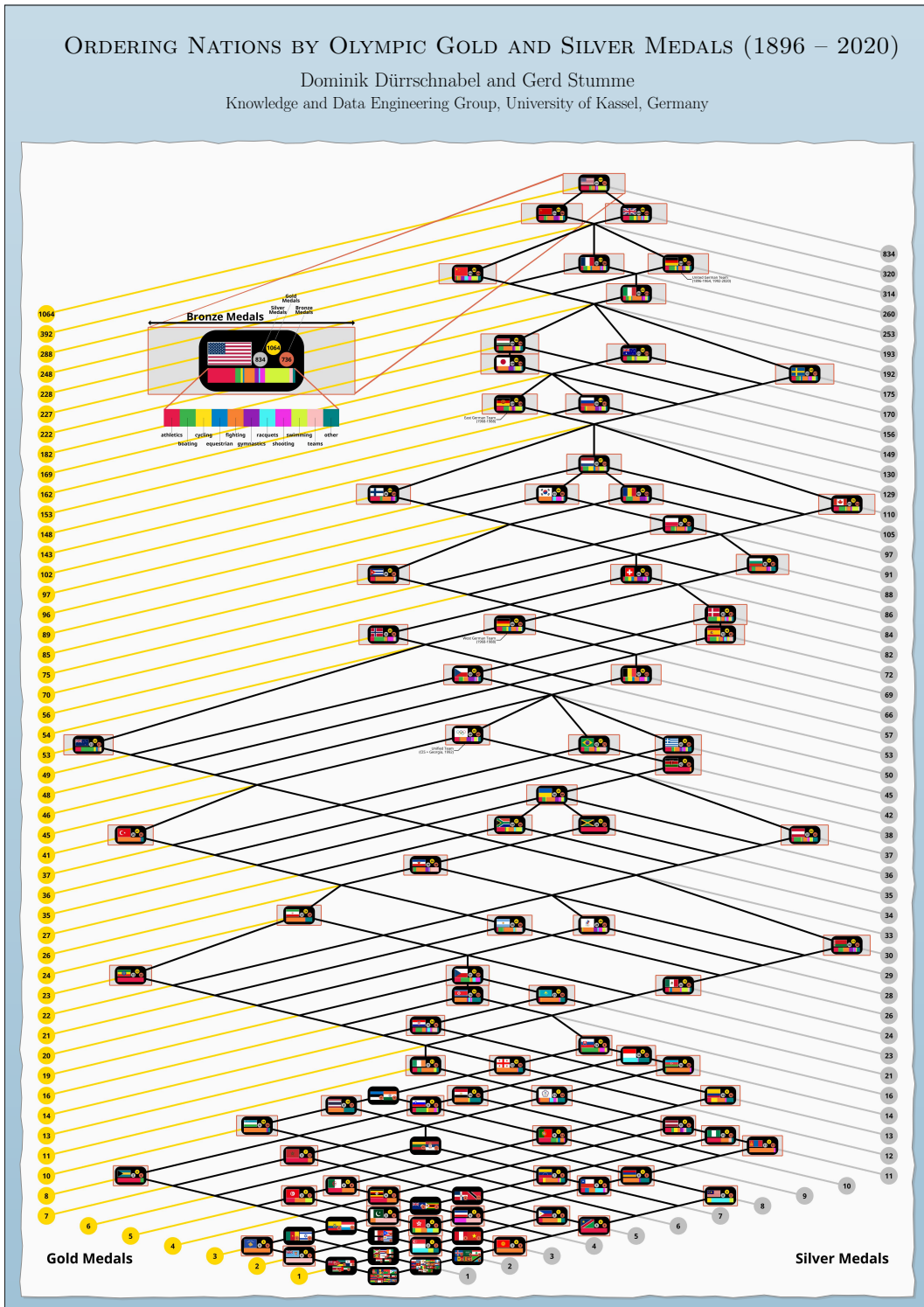
Clustering. Can clusters or communities of countries with similar Olympic success profiles be identified within the graph? Are there distinct communities within the graph based on geographical or cultural similarities? For instance, are countries that excel in swimming close to the sea?

We received 9 submissions for the creative challenge. Submissions were evaluated according to four criteria:

1. readability and clarity of the visualization,
2. aesthetic quality,
3. novelty of the visualization concept, and
4. design quality.

We noticed overall that it is a complex combination of several aspects that make a submission stand out. These aspects include but are not limited to the understanding of the structure of the data, investigation of the additional data sources, applying intuitive and powerful data visual metaphors, careful design choices, combining automatically created visualizations with post-processing by hand, as well as keeping the visualization, especially the text labels, readable. We selected the top six submissions before the conference, which were printed on large poster boards and presented at the Graph Drawing Symposium. We also made all the submissions available on the contest website in the form of a virtual poster exhibition. During the conference, we presented the top six submissions and announced the winners. For a complete list of submissions, refer to <https://www.graphdrawing.org/gdcontest/2024/results/>.

3rd Place: Dominik Dürschnabel and Gerd Stumme (University of Kassel)

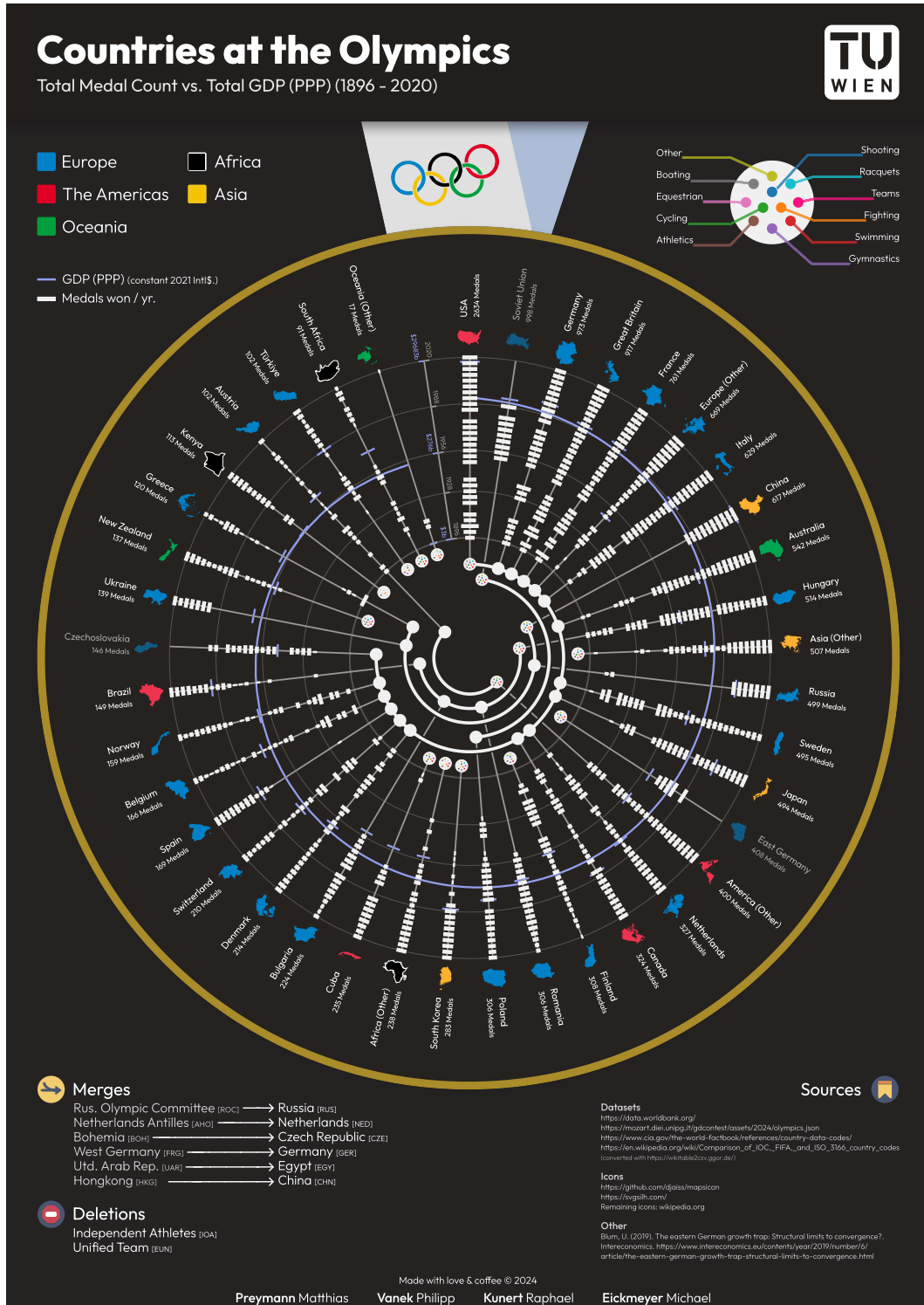


This submission stood out for its unique interpretation of the data: With no universal metric available to rank nations, based on the medals they won, this submission suggests to use a partial ordering of the nations. The result is a drawing that has many properties of a classic Hasse diagram. The contest committee appreciates that even with the abundance of meta data to visualize, this submissions' focal point is a drawing with nodes and links.

“ Our submission uses order as the guiding principle to rank nations based on their gold and silver medal counts, ensuring a two-dimensional layout. Each nation is represented by a pill-shaped dot containing its flag, medal tally, and a scarf plot showing the distribution of medals across sports. Bronze medals are indicated by the width of a box surrounding each nation's pill. This approach is an application of formal concept analysis, which is applied to compute the ranking. ”

D. Dürrschnabel and G. Stumme

2nd Place: Matthias Preymann, Philipp Vanek, Michael Eickmeyer, and Raphael Kunert (TU Wien)



The committee found that this submission was of excellent visual quality and appreciated the idea of contrasting GDP and medals at Olympia. The radial style makes the drawing intuitive and well legible without sacrificing space. The small hypergraph visualization in the center is nice and the committee found it well executed, but would have liked it to be integrated better and maybe more prominently into the visualization. The committee also found that the GDP data could have been better focused or put in a more interesting relation as the medal count is given per year, but the GDP isn't.

“ Our poster is shaped by our main design decision to use a radial figure which emulates a medal. The large amount of data prompted combining countries and regions, which was further guided by taking various geopolitical events into account. By overlaying the medal count timeline with a GDP regression spiral, clear trends can be observed. Finally, to incorporate the full bipartite nature of the input graph, a very compact country-category mapping is included in the center. ”

M. Preymann, P. Vanek, M. Eickmeyer, and R. Kunert

1st Place: Hoang An Nguyen, Nico Martin, Jannik Brandstetter, and Micha Fauth (University of Tübingen)



The Olympic Times



RIISING CONTENDERS

In recent years, Asia has continuously increased its medal count and has emerged as a leading contender in combat sports and shooting. Meanwhile, Africa has demonstrated its growing prowess in athletics. These trends highlight the rising influence of Asia and Africa in the Olympics, ushering in a new era of global competition and top-tier athletes. Will they be able to continue this upward trajectory during the 2024 Olympics in Paris?

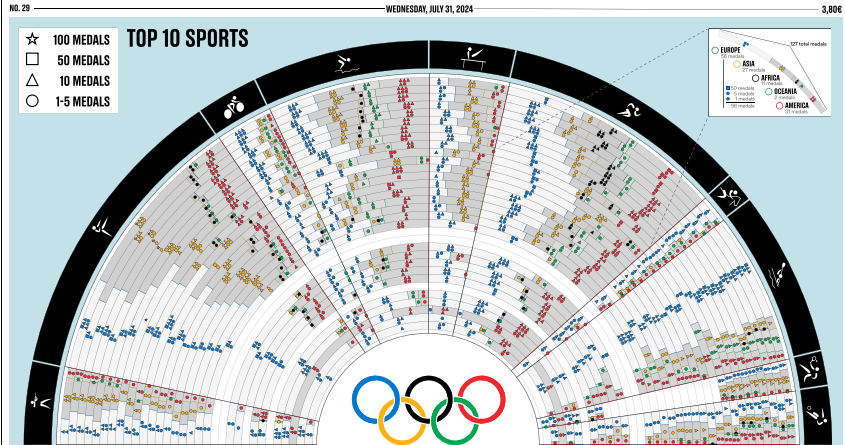
FIERCE RIVALRY

The Olympics spotlight intense rivalries, with Europe, Asia, and America clashing in combat sports to determine the top medal holders in these relentless disciplines. Meanwhile, in athletics, Europe's versatile competitors continually challenge America's top athletes for the most medals, but Africa is quickly catching up. Swimming is another attention-worthy discipline, with almost all continents amassing similar numbers of medals. Only time will tell if the balance of power will shift in the upcoming Olympics this year.

MEDAL STATISTIC



CLASH OF THE CONTINENTS

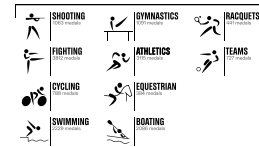


MEDAL HEAVYWEIGHT

Europe stands out as the dominant force in Olympic success, amassing an exceptionally high number of medals across various disciplines, nearly doubling the medal count of the second-highest continent. This remarkable achievement highlights the region's dedication to excellence in sports and its emphasis on unity through diversity, with competitors from a wide array of nationalities all representing Europe as a symbol of camaraderie and mutual respect.

UNDERDOGS NO MORE

Underdog stories are an integral part of the Olympic narrative. Even underestimated continents like Africa have shown remarkable achievements in athletics and continue to improve their dedication to this sport, holding most of the world records in running disciplines. Another prime example of a fierce competitor, once only sparsely present, Asia now holds a prominent position in the medal count for swimming disciplines.



The committee valued the graphical choices of this submission. The presentation and execution of the idea of a newspaper article as a drawing point works well and the small articles are interesting to read. The visualization itself is well executed. It is arguably more reduced than the preceding contenders, but it accomplishes its goals cleanly and with good ideas to overcome the challenge of presenting a very dense dataset. The committee thinks that the glyphs used for the amounts of medals won could have been improved by more coarse categories, removing them for too small amounts, or actually using none at all as the diagrams are for the most part sufficient in communicating the amounts.

“The goal of our visualization is to offer users an intuitive and visually engaging way to understand the data without sacrificing key information. At first glance, it provides a high-level overview of the “Clash of the Continents,” highlighting which ones excel in specific sports categories. As viewers explore further, the visualization reveals more granular details, such as absolute figures, providing clear reference points for better understanding and comparing performances across countries. Maintaining a cohesive design was also essential, leading us to choose a newspaper theme to emphasize the timeliness and relevance of the topic, particularly in light of the upcoming Paris Olympics 2024, further enhancing reader engagement.”

H. An Nguyen, N. Martin, J. Brandstetter, and M. Fauth

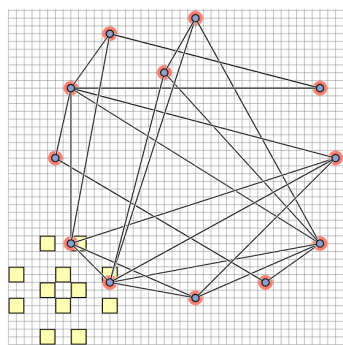
3 Live Challenge

The live challenge took place during the conference and lasted exactly one hour. During this hour, local participants of the conference could take part in the manual category (in which they could attempt to draw the graphs using a supplied tool: <http://graphdrawing.org/gdcontest/tool/>), or in the automatic category (in which they could use their own software to draw the graphs). As in the last year, we allowed everybody in both categories to participate remotely. To coordinate the contest, give a brief introduction, answer questions, and give participants the possibility to form teams, we were kindly provided with both a room in the conference building, and a Zoom stream for the conference. A small bug emerged during the contest related to the submission system. The contest committee determined that the manual category could best be evaluated by each team sending a screenshot. The automatic category turned out to be not affected after all. The error has by now been found and corrected.

The challenge focused on placing the vertices of an undirected simple graph on a given point set with the goal to minimize the edge crossings in the resulting straight-line drawing. We allowed for points of the point set to be collinear and for vertices to lie on top of edges. For each proper crossings we added one to the quality measure and for each vertex-edge overlap we added n to the quality measure where n was the number of vertices. Embedding vertices at fixed or constraint locations is a researched topic in information visualization and graph drawing often with a focus on achieving plane drawings. With this challenge we hope to point to the possibility in this topic to also look at classic quality measures, such as edge crossings.

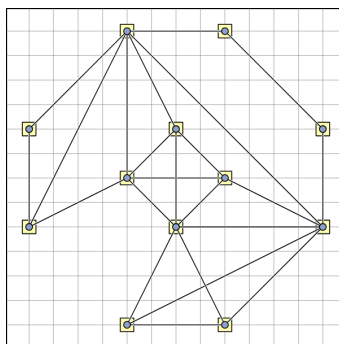
3.1 The Graphs

In the manual category, participants were presented with seven graphs. These were arranged from small to large with the exception of the last graph and chosen to contain different types of graph structures. In the automatic category, participants had to draw the same seven graphs as in the manual category, and in addition another eight larger graphs. Again, the graphs were constructed to have different structures.



Provided drawing

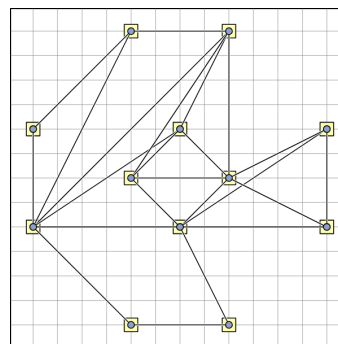
invalid drawing



Best manual solution

ThePointless

2 crossings



Best automatic solution

Graph Gladiators

2 crossings

For illustration, we include below the third graph, where the contestants were given a planar graph plus one edge on a symmetric pointset. The best manual and automatic solutions managed to find drawings with 2 crossings. While the best manual and automatic

41:10 Graph Drawing Contest Report

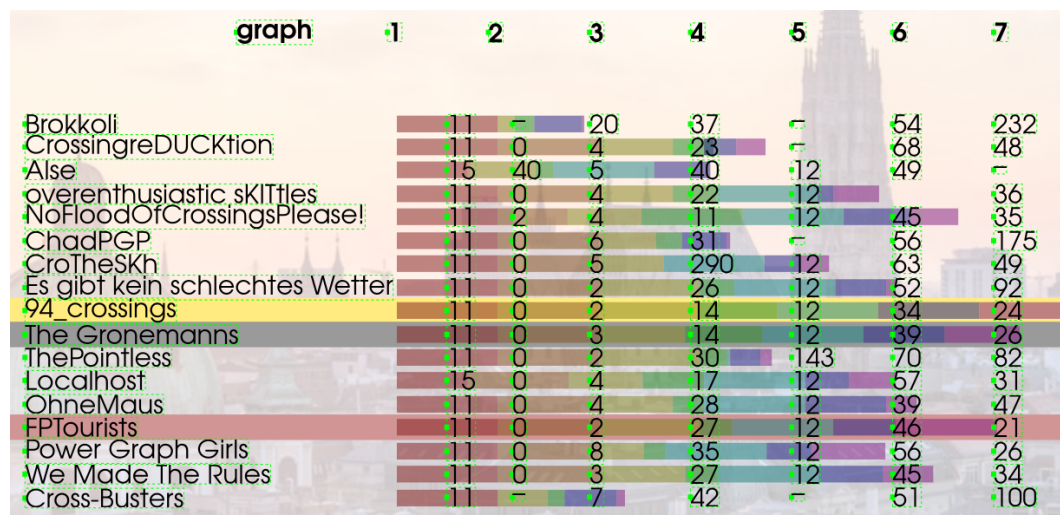
solutions reached the same number of crossings, the manual ones show on average a better and clearer embedding of the graph on the pointset. For example in the drawings we show below, the automatic solution has a worse angular resolution around the vertices than the manual one.

This example fits into similar observations throughout the past years. The committee sees repeatedly that manual (human) drawings of graphs display a deeper understanding of the underlying graph structure than automatic and therefore gain in readability. Moreover, on all but three of the smaller graphs the humans were able to find a solution with the same number of crossings (presumably the best possible) as the automatic solutions.

For the complete set of graphs and submissions, refer to the contest website at <https://www.graphdrawing.org/gdcontest/2024/results/>. The graphs are still available for exploration and solving Graph Drawing Contest Submission System: <https://www.graphdrawing.org/gdcontest/tool/>.

3.2 Results: Manual Category

Below we present the full list of scores for all teams. The numbers listed are the edge-length ratios of the drawings; the horizontal bars visualize the corresponding scores.



Third place: **FPTourists**, consisting of Mathis Rocton and Vaishali Surianarayanan.

“ In this contest, we aimed to minimize edge crossings by manually rearranging the vertices of a given graph. We started by analyzing the graph’s structure, identifying dense and sparse regions, and distinguishing low-degree from high-degree vertices to extract as much visual structure as possible. This gave us an intuitive understanding of the graph’s shape – like connected components and symmetries – before considering the specific point set.

From there, we worked on finding an embedding with fewer crossings and used local optimization, swapping small sets of vertices to further reduce crossings. While we didn’t actually use fixed-parameter techniques, our team, humorously called the “FPTourists,” did a pretty good job of improving the graph’s visual clarity!

M. Rocton and V. Surianarayanan ”

Second place: **The Gronemanns**, consisting of Fouli Argyriou, Mirko Wagner, and Henry Förster

“In this year’s contest, we applied a three-phase approach. The first phase, called the “Pressure phase,” focused on quickly submitting solutions to assert pressure on other contestants by achieving a high score on the leader board. The second phase, known as “Drag around until no edge is colored orange,” used greedy heuristics to reduce edge-vertex overlaps, which incurred penalties, though this year’s instances made it easier to find solutions without overlaps. In the third phase, a vertex swap heuristic and a pattern recognition approach were employed to refine the layout. The vertex swaps iteratively improved the solution, while pattern recognition suggested macro adjustments, yielding near-optimal solutions even for the most difficult challenges.”

F. Argyriou, M. Wagner, and H. Förster

Winner: **94_crossings**, consisting of Tim Hegemann and Johannes Zink.

“We asked ChatGPT to write us a victory speech. Here is what we got: GD’24, day two, Vienna 17:45. With just the two of us, Team 94_crossings assembles once more for another exciting challenge. 17:50. Armed with geometry, strategy, and a shared birth year (but let’s keep that a secret!), we dive into the vibrant contest. 18:00. Navigating through vertices and the intricate dance of crossing minimization, we remain focused on our ambition. Ending with 97 crossings – slightly above our namesake goal – the outcome is clear: another victory in a field brimming with exceptional teams. Here’s to close contests and the spirit of collaboration in beautiful Vienna!”

T. Hegemann and J. Zink

3.3 Results: Automatic Category

In the following we present the full list of scores for all teams that participated in the automatic category. The numbers listed are the number of crossings of the drawings; the horizontal bars visualize the corresponding scores. Given that node-edge overlaps added the number of vertices to the number of crossings relatively high numbers of crossings were present in the results.

Third place: **Baseline**, consisting of Maximilian Pfister.

“As the title already suggests, the approach is as straightforward as you can think of: An initial (random) assignment of the vertices to the points is generated, which is consequently improved by either using a (i) “swap-operation”, where the position of two vertices is exchanged, or a (ii) “replace-operation”, where a vertex is moved to an unused point. Random restarts were deployed to escape local minima and new assignments were accepted in a greedy fashion, i.e., whenever they did not increase the number of crossings. The decent performance of the algorithm can be attributed to the efficient update of the number of crossings (enabled by the small local changes) which allowed to perform many iterations in a short number of time.”

M. Pfister

41:12 Graph Drawing Contest Report

■ **Table 1** Results of the automatic live challenge.

Graph	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Geometry	11	0	10	28	143	41	—	—	324248	—	—	—	—	—	—
Rustlings	11	0	2	9	19	37	21	2.1E7	2091036	1412682	26382	2.1E7	2820156	1.1E7	306049
Graph Gladiators	11	0	2	8	12	24	15	183516	307724	25309	5450	2113030	598224	831078	12619
Baseline	11	0	2	9	12	27	18	63786	334434	14886	18092	3420933	709006	1663029	13246
Irp	11	0	—	10	74	27	16	2.1E7	851337	119752	22559	1.75E7	—	—	—
OptimizationGroup2	11	0	2	46	1353	51	80	2.1E7	1291315	1369448	424034	2.1E7	3065353	1.1E7	2822936
OMEGA	11	0	2	8	12	24	15	4468	309244	3961	4	65486	3242986	65947	3583

Second place: **Graph Gladiators**, consisting of Philipp Kindermann, Alexander Kutscheid, and Jan-Niclas Loosen.

“ We started with an ILP formulation to solve the problem exactly, which as expected turned out to be too slow, even for the larger manual graphs; but we could solve the problem for convex point sets and double chains quickly with a specialized ILP. For the larger instances, we first created random and force-directed layouts (Tutte, Eades and FruchtermannReingold) and matched the vertices to the closest points. It turned out that FruchtermannReingold works the best, and computing a greedy matching is more effective than finding the optimum one. We then used a simulated annealing approach to move either a single vertex or a vertex plus its closest neighbors to different points. We selected the vertices to move randomly weighted by the number of edge crossings it is involved in. The main difficulty for us was to update the vertex weights and the number of crossings during the movements without recomputing them from scratch. ”
P. Kindermann, A. Kutscheid, and J.-N. Loosen

Winner: **OMeGA**, consisting of Julien Bianchetti, Pauline Blavy, Guilhem Gerouille, Laurent Moalic, and Dominique Schmitt.

“ The algorithm we used this year is the one we implemented for last year’s challenge, with some improvements. We generate a first embedding of the graph with the FMME algorithm from the OGDF library. Every node of the embedding is then assigned to its closest available point. Different assignments are tested, and the one providing the best score is kept. Using a simulated annealing approach, the nodes are then randomly moved to other locations. The move is always accepted if it improves the current solution. If the solution is degraded, the move is accepted with a certain probability depending on the temperature reached by the simulated annealing. The initial value of the temperature and its variations are automatically computed to be best adapted to the graph being processed. We ran our program on a 10-core CPU, simultaneously on the 15 given graphs. It crashed on graphs 3 and 13. We solved graph 3 manually and submitted graph 13 in its original version (without improvement). ”
J. Bianchetti, P. Blavy, G. Gerouille, L. Moalic, and D. Schmitt




4 Conclusion

The 2024 edition of the Graph Drawing Contest was again a success in participation and result. The high numbers from the 2023 edition could almost be replicated, which the committee, given that in the 2023 iteration several participants of the yWorks company were present due to an overlap in events in Sicily, values as a success. The participation in the automatic category especially was stronger the last two iterations than the iterations before. The committee nonetheless believes that some changes should be made to the format to make it even more attractive for participants from the Graph Drawing community. The manual live challenge is in a good spot the committee believes, the participation numbers are high and from talking at the conference the committee gathers that the participants enjoy the format. Finally, the creative contest has seen an increase in submissions and, arguably, overall quality in the last years. The turn to only one category made the contest more focused and targeted, the committee believes.

From Planar via Outerplanar to Outerpath – Engineering NP-Hardness Constructions

Joshua Geis

Universität Würzburg, Germany

Johannes Zink   

Universität Würzburg, Germany

Abstract

A typical question in graph drawing is to determine, for a given graph drawing style, the boundary between polynomial-time solvability and NP-hardness. For two examples from the area of drawing graphs with few slopes, we sharpen this boundary. We suggest a framework for a certain type of NP-hardness constructions where graphs have some parts that can only be realized as rigid structures, whereas other parts allow a controllable degree of flexibility. Starting with an NP-complete problem involving planarity (here, we use planar monotone rectilinear 3-SAT), we consider first a reduction to a planar graph, which can be adjusted to an outerplanar graph, and finally to an outerpath. An *outerplanar graph* is a graph admitting an *outerplanar drawing*, that is, a crossing-free drawing where every vertex lies on the outer face, and an *outerpath* is a graph admitting an outerplanar drawing where the weak dual is a path. The (*weak*) *dual* of a graph drawing is the graph that has a node for every (inner) face and a link if two faces share an edge.

Specifically, we first show that, for every upward-planar directed outerpath G , it is NP-hard to decide whether G admits an upward-planar straight-line drawing where every edge has one of three distinct slopes, and we second show that, for every undirected outerpath G , it is NP-hard to decide whether G admits a proper level-planar straight-line drawing where every edge has one of two distinct slopes. For both problems, NP-hardness has been known before for outerplanar graphs.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases NP-hardness, outerplanar, outerpath

Digital Object Identifier 10.4230/LIPIcs.GD.2024.42

Category Poster Abstract

1 The Framework

Consider a problem where the input is a graph G with specific properties and the task is to draw G in a specific drawing style without crossings. When reducing from planar monotone rectilinear 3-SAT [2], we model variables by variable gadgets, and clauses by clause gadgets. The gadgets are subgraphs of G that can only be drawn in a specific way – the boundaries (/frames/skeletons) are rigid building blocks, while other parts can be drawn with a small degree of flexibility allowing a mapping of truth values to variable gadgets and a crossing-free drawing of a clause gadget only if the clause is satisfied. First, we construct G such that it is planar and connected by following the planar incidence graph of the 3-SAT instance.

Second, we try to make G an outerplanar graph G' while the reduction remains applicable. To this end, we add gaps to the rigid boundary such that every vertex in a drawing of G' lies on the outer face but G' stays connected. Note that not every NP-hardness construction in this flavor is directly suited here because the rigid and flexible structures should be “thin”.

Third, we try to make G' an outerpath G'' while the reduction remains applicable. To this end, we trace the boundary of the embedding of G' . Note that this boundary is a cycle that encounters every vertex of G' . Replace the boundary by a chain of the same rigid building



© Joshua Geis and Johannes Zink;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 42; pp. 42:1–42:3

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

blocks as before (just “smaller”). For the flexible parts, one needs to carefully re-design them as (multiple parts of) outerpaths such that the functionality in the reduction is preserved. We keep a gap in the boundary such that the weak dual is not a wheel graph but a path.

2 Upward-Planar 3-Slope Drawings

An upward-planar graph is a directed graph that admits a planar drawing where every directed edge uv is drawn as a y -monotone curve such that $y(u) < y(v)$. Determining for a directed graph whether it is upward planar is NP-hard in general [4], but polynomial-time solvable if an embedding is given [1] or if the input graph is outerplanar [13]. We consider the setting of straight-line drawings with a limited number of slopes. Determining the minimum number of slopes is $\exists\mathbb{R}$ -hard for undirected and directed graphs [7, 14]. For three slopes (w.l.o.g., 45, 90, 135 degrees w.r.t. the x -axis), Klawitter and Zink [9, 10] observe that a specific (sub)graph can only be drawn as a square with a diagonal (we call them *rigid square* here). Using them as the rigid structure, and using as the flexible structure *sliders*, which are two parallel edges attached to rigid squares that can extend only in one dimension, they show NP-hardness for directed outerplanar graphs. We extend their result as follows.

► **Theorem 1.** *Given a directed outerpath G , which optionally can be equipped with an upward-planar outerpath embedding, it is NP-hard to decide whether G admits an upward-planar straight-line drawing where every edge has one of three distinct slopes.*

According to the framework, we arrange rigid squares and triangles along the boundary. We replace each slider by two sliders that are attached to two chains of rigid squares that are twisted into each other such that we cannot move the two sliders much and the replacement of the slider behaves like the original. For details see the theses by Geis [5] and Zink [15].

3 Proper Level-Planar 2-Slope Drawings

A *level-planar* graph drawing is crossing free and every vertex is placed at a specific level (levels are equidistant horizontal lines). A level-planar drawing is *proper* if, for every edge e , the endpoints of e lie on consecutive levels. The setting where a leveling of the vertices is given in addition to the graph is most common. This is testing upward planarity with prescribed y -coordinates, which is polynomial-time solvable [3, 6, 8]. We focus on the case of straight-line edges with a limited set of slopes. Brückner, Krisam, and Mchedlidze show for this case that deciding if a graph given with a leveling and an arbitrary number of slopes admits a level-planar drawing can be solved in polynomial time if the leveling is proper, but otherwise it is NP-hard even for just two slopes. Here, we study the case where no leveling (or edge directions) are given but the generated drawing shall be proper. The problem becomes NP-hard with two slopes even for outerplanar graphs as shown by Kraus [11]. Again, squares are used as rigid structure and the flexible structure are edges going either a level up or down, and edges having either the first or the second slope. We extend this result to outerpaths.

► **Theorem 2.** *Given an outerpath G , it is NP-hard to decide whether G admits a proper level-planar straight-line drawing where every edge has one of two distinct slopes.*

4 Conclusion and Open Questions

The framework seems to be suited for settings with a small number of edge slopes or edge lengths. The two NP-hardness constructions presented here are very similar to each other and to Nöllenburg’s [12] NP-hardness proof for planar octilinear metro maps. However, it is not obvious how to change his construction to work for outerplanar graphs or even outerpaths. Are there more examples of graph drawing problems where NP-hardness for (outer)planar graphs is known, while the “simpler” trees and cactus graphs are polynomial-time solvable, and the known construction can be adjusted to outerplanar graphs or outerpaths?

References

- 1 Paola Bertolazzi, Giuseppe Di Battista, Giuseppe Liotta, and Carlo Mannino. Upward drawings of triconnected digraphs. *Algorithmica*, 12(6):476–497, 1994. doi:10.1007/BF01188716.
- 2 Mark de Berg and Amirali Khosravi. Optimal binary space partitions for segments in the plane. *International Journal of Computational Geometry and Applications*, 22(3):187–206, 2012. doi:10.1142/s0218195912500045.
- 3 Giuseppe Di Battista and Enrico Nardelli. Hierarchies and planarity theory. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(6):1035–1046, 1988. doi:10.1109/21.23105.
- 4 Ashim Garg and Roberto Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM Journal on Computing*, 31(2):601–625, 2001. doi:10.1137/S0097539794277123.
- 5 Joshua Geis. Aufwärtsplanare Zeichnungen mit drei Steigungen von Außenpfaden. Bachelor’s thesis, Institut für Informatik, Universität Würzburg, 2022. (in German). URL: <https://www1.pub.informatik.uni-wuerzburg.de/pub/theses/2022-geis-bachelor.pdf>.
- 6 Lenwood S. Heath and Sriram V. Pemmaraju. Recognizing leveled-planar dags in linear time. In *Proc. Symposium on Graph Drawing (GD’95)*, pages 300–311. Springer, 1995. doi:10.1007/BFb0021813.
- 7 Udo Hoffmann. On the complexity of the planar slope number problem. *Journal of Graph Algorithms and Applications*, 21(2):183–193, 2017. doi:10.7155/jgaa.00411.
- 8 Michael Jünger, Sebastian Leipert, and Petra Mutzel. Level planarity testing in linear time. In *Proc. 6th International Symposium on Graph Drawing (GD’98)*, pages 224–237. Springer, 1998. doi:10.1007/3-540-37623-2_17.
- 9 Jonathan Klawitter and Johannes Zink. Upward planar drawings with three and more slopes. In *Proc. 29th International Symposium on Graph Drawing & Network Visualization (GD’21)*, pages 149–165. Springer, 2021. doi:10.1007/978-3-030-92931-2_11.
- 10 Jonathan Klawitter and Johannes Zink. Upward planar drawings with three and more slopes. *Journal of Graph Algorithms and Applications*, 27(2):49–70, 2023. doi:10.7155/jgaa.00617.
- 11 Rebecca Kraus. Level-außenplanare Zeichnungen mit wenigen Steigungen. Bachelor’s thesis, Institut für Informatik, Universität Würzburg, 2020. (in German).
- 12 Martin Nöllenburg. Automated drawing of metro maps. Diploma thesis, University of Karlsruhe (TH), 2005. URL: <https://i11www.iti.kit.edu/extra/publications/n-admm-05da.pdf>.
- 13 Achilleas Papakostas. Upward planarity testing of outerplanar DAGs. In *Proc. DIMACS International Workshop on Graph Drawing (GD’94)*, pages 298–306, 1994. doi:10.1007/3-540-58950-3_385.
- 14 Valentin Quapil. Upward and upward-planar drawings with limited slopes. Bachelor’s Thesis, Karlsruhe Institute of Technology, 2021. URL: https://i11www.iti.kit.edu/_media/teaching/theses/ba-quapil-21.pdf.
- 15 Johannes Zink. *Algorithms for Drawing Graphs and Polylines with Straight-Line Segments*. PhD thesis, Universität Würzburg, 2024. doi:10.25972/OPUS-35475.

Minimizing Switches in Cased Graph Drawings

Robert Ganian ✉ 🏠 

Algorithms and Complexity Group, TU Wien, Austria

Martin Nöllenburg ✉ 🏠 

Algorithms and Complexity Group, TU Wien, Austria

Sebastian Röder ✉ 

TU Wien, Austria

Abstract

In cased drawings of graphs, edges are drawn in front of others in order to decrease the negative impact of crossings on readability. In this context, a switch on an edge is defined as two consecutive crossings, where the edge is drawn in the front at one crossing and behind another edge at the next crossing. We investigate the problem of minimizing the maximum number of switches on any edge – both in a fixed drawing as well as for non-embedded graphs. We resolve an open question by Eppstein, van Kreveld, Mumford, and Speckmann (2009) by establishing the NP-hardness of minimizing the number of switches in a fixed drawing, provide a fixed-parameter algorithm for this problem, and obtain a full characterization of the problem for non-embedded graphs.

2012 ACM Subject Classification Mathematics of computing → Graph theory; Human-centered computing → Graph drawings

Keywords and phrases beyond planarity, complexity theory, non-planar drawings, crossings

Digital Object Identifier 10.4230/LIPIcs.GD.2024.43

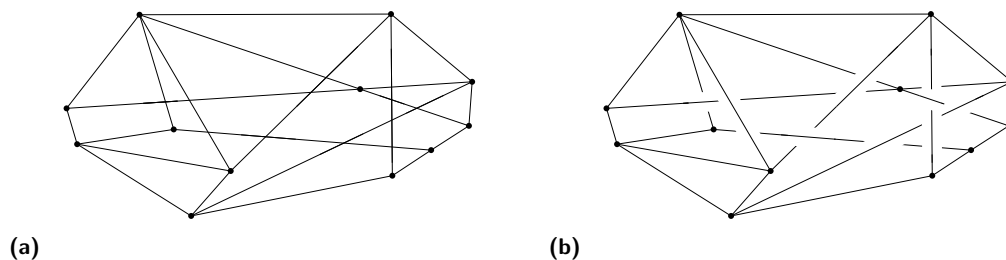
Category Poster Abstract

Funding The authors acknowledge support from the Vienna Science and Technology Fund (WWTF) [10.47379/ICT22029].

Robert Ganian: Austrian Science Fund (FWF) [10.55776/Y1329].

1 Introduction

Edge casing is a frequently used visual rendering method to improve the readability of crossings in non-planar graph drawings. In a cased drawing – introduced by Eppstein et al. [2] – two crossing edges are locally ordered into an upper and a lower edge and the curve representing the lower edge (called the *tunnel*) is locally interrupted to let the upper edge (called the *bridge*) pass through the created gap. This can be particularly important for graph drawings with regions of high feature density and many edge crossings, which, in non-cased drawings, are hard to discern from the small disk symbols typically representing the vertices; see Figure 1.



■ **Figure 1** (a) A drawing of a graph with crossings; (b) The same drawing with edge casing.



© Robert Ganian, Martin Nöllenburg, and Sebastian Röder;
licensed under Creative Commons License CC-BY 4.0

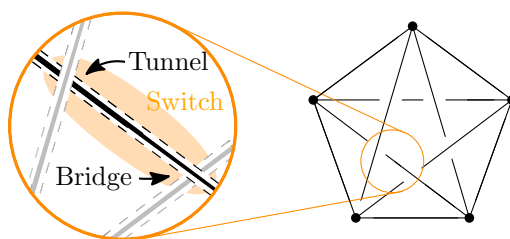
32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 43; pp. 43:1–43:3

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 2** Cased drawing of K_5 ; the close-up shows a tunnel, bridge, and switch of the black edge.

Eppstein et al. defined several optimization problems, either concerning the number, length, or distance of the tunnels per edge, or concerning the number of *switches*, which are pairs of consecutive crossings of an edge e , one of which is a tunnel for e and the other one a bridge (see Figure 2). We revisit cased drawings and focus on the problem of minimizing the number of switches.

2 Results

We resolve an open question of Eppstein et al. [2] on the complexity of the MINMAXSWITCHES problem of minimizing the maximum number of switches per edge for a given graph drawing. We show that this problem is NP-complete even when the target number of switches per edge is 1, i.e., when we need to decide whether a given drawing δ_G can be embedded with at most 1 switch per edge. On the other hand, it is known that deciding whether δ_G can be embedded with 0 switches per edge is polynomial-time solvable [2]. We complement our hardness proof with a fixed-parameter algorithm. The structure of the input graph can be assumed to be trivial as every instance can be transformed into an equivalent matching. Therefore, we use the vertex cover number of the cell adjacency graph of the input drawing as a parameter. All results obtained also directly carry over to the straight-line setting.

► **Theorem 1.** MINMAXSWITCHES is NP-complete.

Proof Sketch. Inclusion in NP is immediate; we show hardness by reducing from NAE 3-SAT, i.e., the NP-complete variant of 3-SAT where clauses are required to contain at least one satisfied and at least one unsatisfied literal [4]. ◀

The structural parameter we use to achieve our tractability result is the *vertex cover number* ℓ of χ_G . The *cell adjacency graph* $\chi_G = (\mathcal{F}, E')$ is the graph, whose vertices are precisely the *cells* \mathcal{F} of δ_G (i.e., the connected regions of $\mathbb{R}^2 \setminus \delta_G$) and where two cells are adjacent if and only they touch, i.e., share a edge segment or crossing on their boundary.

► **Theorem 2.** MINMAXSWITCHES is fixed-parameter tractable when parameterized by the vertex cover number ℓ of χ_G .

Proof Sketch. Two reduction rules are applied. First, remove any edge e with at most 2 crossings. Secondly, split the drawing into *bridgeless* subproblems. A drawing δ_G is *bridgeless* if there is no edge segment between two crossing points whose removal would disconnect δ_G . We claim that – after exhaustively applying the aforementioned reduction rules – the size of the obtained kernel is upper-bounded by $O(\ell^4)$. ◀

Moreover, following the definition of k -gap-planar graphs by Bae et al. [1], we define a graph to be k -switch-planar if it admits a cased drawing with at most k switches per edge. We give a full characterization of this notion. Recall that a graph G is biplanar if it has thickness at most 2, i.e., its edge set can be partitioned into two planar subgraphs.

► **Theorem 3.** *A graph G is 0-switch-planar if and only if it is biplanar.*


Theorem 3 implies that determining whether a graph admits a cased drawing without switches is NP-complete, as this is the case for testing a graph for biplanarity [3].

► **Theorem 4.** *Every graph G is 1-switch-planar.*

References

- 1 Sang Won Bae, Jean-Francois Baffier, Jinhee Chun, Peter Eades, Kord Eickmeyer, Luca Grilli, Seok-Hee Hong, Matias Korman, Fabrizio Montecchiani, Ignaz Rutter, and Csaba D. Tóth. Gap-planar graphs. *Theoretical Computer Science*, 745:36–52, October 2018. doi:10.1016/j.tcs.2018.05.029.
- 2 David Eppstein, Marc Van Kreveld, Elena Mumford, and Bettina Speckmann. Edges and switches, tunnels and bridges. *Computational Geometry*, 42(8):790–802, October 2009. doi:10.1016/j.comgeo.2008.05.005.
- 3 Anthony Mansfield. Determining the thickness of graphs is NP-hard. *Mathematical Proceedings of the Cambridge Philosophical Society*, 93(1):9–23, 1983. doi:10.1017/S030500410006028X.
- 4 Thomas J. Schaefer. The complexity of satisfiability problems. In *Theory of Computing (STOC'78)*, pages 216–226. ACM Press, 1978. doi:10.1145/800133.804350.

Graph-Drawing Supported Identification of Influential Students at Schools

Markus Chimani ✉ 

Theoretical Computer Science, Osnabrück University, Germany

Lea Kröger ✉

Theoretical Computer Science, Osnabrück University, Germany

Juliane Liedtke ✉

VNB (Association of Education Initiatives in Lower Saxony), Hannover, Germany

Jonah Mevert ✉

Theoretical Computer Science, Osnabrück University, Germany

Maor Shani ✉ 

Developmental Psychology, Osnabrück University, Germany

Maarten van Zalk ✉ 

Developmental Psychology, Osnabrück University, Germany

Abstract

We consider the real-world problem of identifying a set of “influential” students at schools for a workshop on tolerance. We report on a tool that visualizes the networks of social connections between students, identifies sets of influential students, and lets one explore and understand the solution space with a focus on usability for teachers who are untrained in network analysis.

2012 ACM Subject Classification Applied computing → Law, social and behavioral sciences

Keywords and phrases social network tool, force-directed graph drawing, group centrality

Digital Object Identifier 10.4230/LIPIcs.GD.2024.44

Category Poster Abstract

Funding This research was supported by the Volkswagen Stiftung (AZ: 9B060) and the prevention council of the ministry of justice of Lower Saxony in Germany (4209 1.88/2022).

1 Problem Scenario

The “Together for Tolerance” (T4T) intervention (a subproject of the “INCLUSIVITY” project [5, 6], led by developmental psychologists and education workers) aims at promoting intergroup tolerance in high schools, especially under polarized conditions where schools become increasingly divided into groups with contrasting opinions, beliefs about social issues, and conflictual behavior. The central idea is to conduct special week-long workshops for the students, led by (school-external) trained personnel.

Since a typical German high school has about 1000 students, it is cost-prohibitive to do so with all students. Thus, the workshops are done with only a comparably small group of students, between 15 and 20, called *social referents* [4], who are highly connected to others via social relationships and have outstanding network positions within the school. Their network positions are thought to make them most influential, as their behavior is exemplary for and observed most by other students. The formally “best” way to select students is an active field of research and part of the T4T project lead by developmental psychologists.

We know, however, that teachers are not accurate in identifying social referents; one intervention even showed that teacher-selected social referents (instead of selections done by other adolescent students) had no or even detrimental influences on their fellow students [2].



© Markus Chimani, Lea Kröger, Juliane Liedtke, Jonah Mevert, Maor Shani, and Maarten van Zalk; licensed under Creative Commons License CC-BY 4.0

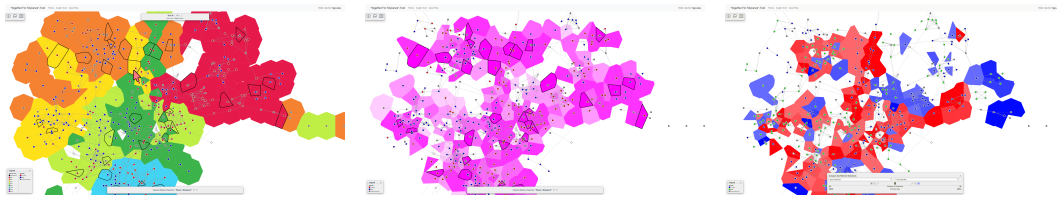
32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 44; pp. 44:1–44:3

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1 (left)** Network of a school with 1000 students, regions are colored according to class year, nodes according to gender, social referent selection in black borders. **(middle)** Visualizing distances to the selected students. **(right)** Comparing two different selections (red=better and blue=worse than second selection).

Thus, we design a web-tool to not only select students, but equally important to visually and intuitively “explain” the selection to the involved teachers. Overall, our web-tool can be seen as the core of step 2 in a 3-step pipeline:

1. Acquiring the social network between the students via questionnaires by developmental psychologists [5]. The network is a multi-graph with node attributes, and edges are mapped to their originating question (“with whom do you spend time?”, “with whom would you like to spend time?”, “whom do you avoid?”, etc.).
 2. The T4T WebTool allows to visualize and inspect the network. It also computes one or more social referent selections (independent of the specific visualization), and allows to investigate, modify, and compare them.
 3. The students of the final social referent selection are invited to the workshop. Their knowledge and behavior acquired at the workshop should permeate through the network.
- In the current test schools, step 2 is performed by researchers. With the project being conducted in more schools, this task should be done by teachers who will be trained in T4T workshops, but have no further experience in network analysis. Furthermore, several practical decisions – e.g., the “best” questions to ask in step 1, how to aggregate the questions into edges, which measure to use for the selection process, etc. – are still ongoing research in the psychological community. As such, the T4T WebTool shall also play a double role for the researchers to help them investigate these questions. They, of course, expect and can deal with a much richer network visualization interface.

2 T4T WebTool

The frontend is written in JavaScript using Vue.js; the backend server in Python using the Django framework. Within the backend server, there is a SQLite database and a C++-bridge to the *Open Graph Drawing Framework* [1]. The tool supports the computation and numerical comparison of several different group centrality measures (sometimes enriched by node-attribute specific adaptations) to use for the social referent selection; despite being an interesting topic in itself, this is beyond the scope of this abstract. Apart from typical operations like zooming, panning, node inspection, etc., the tool has some distinctive features:

- The graphs are drawn using OGDF’s implementation of the FM³ algorithm [3]. To obtain context-aware drawings, we temporarily extend the network with suitably weighted dummy edges to, e.g., (a) cluster students within the same class, and (b) spread the students of the social referent selection over the drawing area, cf. Fig. 1(left). Clearly, we can obtain different drawings for different social referent selections.
- Within the drawing, we compute Voronoi regions around the nodes which are used both as clickable regions corresponding to the nodes, as well as for coloring purposes.

- The social referent selection can be visualized, Fig. 1(middle), where the saturation of a region's color encodes the distance between the student and their closest social referent.
- Two selections can be visually compared, Fig. 1(right), where a region's color encodes which selection is better for the respective student; the saturation encodes by how much.

References

- 1 Markus Chimani, Carsten Gutwenger, Michael Jünger, Gunnar W. Klau, Karsten Klein, and Petra Mutzel. The open graph drawing framework (OGDF). In Roberto Tamassia, editor, *Handbook of Graph Drawing and Visualization*, chapter 17. CRC press, 2014. See www.ogdf.net.
- 2 Robin Gomila, Hana Shepherd, and Elizabeth L. Paluck. Network insiders and observers: who can identify influential people? *Behavioural Public Policy*, 7(1):115–142, 2023.
- 3 Stefan Hachul and Michael Jünger. Drawing large graphs with a potential-field-based multilevel algorithm. In Janos Pach, editor, *Proc. Graph Drawing 2004*, volume 3383 of *LNCS*, pages 285–295. Springer-Verlag, 2004. doi:10.1007/978-3-540-31843-9_29.
- 4 Elizabeth L. Paluck and Hana Shepherd. The salience of social referents: a field experiment on collective norms and harassment behavior in a school social network. *Journal of Personality and Social Psychology*, 103(6):899–915, 2012.
- 5 Maor Shani, Sophie de Lede, Stefanie Richters, Malin Kleuker, Wilma Middendorf, Juliane Liedtke, Sandrine Witolla, and Maarten van Zalk. A social network intervention to improve adolescents' intergroup tolerance via norms of equality-based respect: the “together for tolerance” feasibility study. *International Journal of Developmental Science*, 17(1–3):93–110, 2023.
- 6 Maarten van Zalk, Oliver Christ, Eva Jaspers, Miranda J. Lubbers, Marcin Bukowski, Maor Shani, Laura F. Schäfer, Alejandro Ciordia, Anna Potoczek, Jan-Willem Simons, Nina Tegeler, Lucía Estevan-Reina, and Wilma Middendorf. Inclusivity norms to counter polarization in european societies (inclusivity), 2023. Project description. URL: <https://osf.io/n7c4y/>.

GdMetriX – A NetworkX Extension For Graph Drawing Metrics

Martin Nöllenburg   

Algorithms and Complexity Group, TU Wien, Austria

Sebastian Röder  

TU Wien, Austria

Markus Wallinger   

Chair for Efficient Algorithms, Technical University of Munich, Germany

Abstract

networkX is a well-established Python library for network analysis. With *gdMetriX*, we aim to extend the functionality of *networkX* and provide common quality metrics used in the field of graph drawing, such as the number of crossings or the angular resolution. In addition, the package provides easy-to-use access to the graph datasets provided by the ‘Graph Layout Benchmark Datasets’ project from the Northeastern University Visualization Lab.

2012 ACM Subject Classification Human-centered computing → Graph drawings

Keywords and phrases Graph Drawing Metrics

Digital Object Identifier 10.4230/LIPIcs.GD.2024.45

Category Poster Abstract

Supplementary Material *Text (Documentation)*: <https://livus.github.io/gdMetriX/>

Funding *Martin Nöllenburg*: Vienna Science and Technology Fund (WWTF) [10.47379/ICT19035].

Markus Wallinger: Vienna Science and Technology Fund (WWTF) [10.47379/ICT19035].

1 Introduction

networkX is a well-established and commonly used Python package for working with graph structures [1]. In their own words, *networkX* is intended for “the creation, manipulation, and study of the structure, dynamics, and functions of complex networks” [1]. It supports various data structures and algorithms useful in graph theory [4]. *networkX* also supports some basic graph drawing algorithms, such as for drawing planar graphs or spring embedding. Its focus however does not lie on graph drawing, but rather on graph analytics.

With the *gdMetriX* package, we aim to extend the functionality of *networkX* and provide common quality metrics used in graph drawing, such as the number of crossings or the angular resolution. In addition, the package provides easy-to-use access to the graph datasets provided by the “Graph Layout Benchmark Datasets” project from the Northeastern University Visualization Lab [3].

The project is published on the Python packaging index (see <https://pypi.org/project/gdMetriX/>). More information about all implemented metrics and additional features can be found at the project homepage (see <https://livus.github.io/gdMetriX/>) or the GitHub repository (see <https://github.com/livus/gdMetriX>).



© Martin Nöllenburg, Sebastian Röder, and Markus Wallinger;

licensed under Creative Commons License CC-BY 4.0

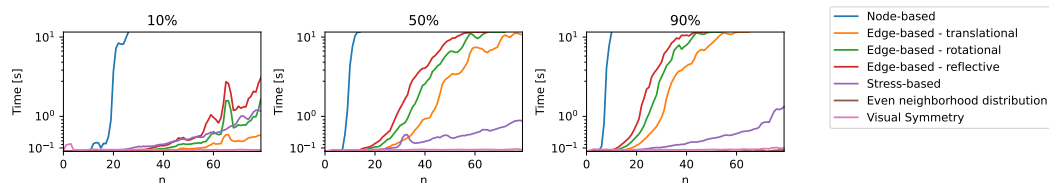
32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 45; pp. 45:1–45:3

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** Runtime of symmetry metrics of random graphs with edge densities ranging from 10% – 90%.

■ **Table 1** Overview of all implemented graph drawing metrics.

Category	Metrics
Crossings	Crossing number, percentage of crossings compared to maximum number of crossings, crossing angles, crossing angular resolution
Area and boundary	Area, tight area, height, width, aspect ratio
Node distribution	Center of mass, closest pair of points, closest pair of elements, concentration, homogeneity, horizontal balance, vertical balance, node orthogonality, Gabriel ratio
Edge directions	Angular resolution, average flow, upwards flow, coherence to average flow, edge orthogonality
Symmetry	Node-based symmetry, edge-based symmetry, stress, even neighborhood distribution, visual symmetry

2 Graph Drawing Quality Metrics

Graph drawing quality measures are a practical and important tool to evaluate the aesthetics and readability of existing drawings and guide the creation of new drawings as well as the development of new drawing algorithms. However, for a large amount of metrics, no publicly accessible Python implementation is available. This increases implementation efforts for new projects and impedes standardization. Indispensable for collecting a set of suitable metrics were the publications by Taylor and Rodgers [7], Purchase [6], Bennett et al. [2], as well as the recent paper by Mooney et al. [5]. A list of all metrics can be found in Table 1.

The framework aims to provide efficient implementations for all metrics. Especially the symmetry metrics offer measures with similar properties but vastly different runtimes (see Figure 1). As some symmetry metrics are infeasible to compute for larger instances, we provide a novel metric – called “visual symmetry” – with linear runtime, aimed to quickly assess the symmetry of a drawing. The metric works by creating a constant-size image of a graph drawing. From this image, rotated and mirrored versions are subtracted and the ink left in the drawing is summed together. The less ink is left, the more symmetric the graph drawing is assumed to be. A more in-depth analysis can be found on the project homepage.



3 Access to Common Benchmark Datasets

Di Bartolomeo et al. [3] collected commonly used benchmark datasets for evaluating graph layout algorithms. To ease the usage of the datasets and make the collection more visible, we have implemented a module to download and parse the provided datasets automatically with a single line of code. The module downloads requested datasets, caches them for reuse, and parses the data for use with networkX.



References

- 1 networkX homepage. <https://networkx.org/>. Accessed: 2024-07-20.
- 2 Chris Bennett, Jody Ryall, Leo Spalteholz, and Amy Gooch. The Aesthetics of Graph Visualization. *Computational Aesthetics in Graphics*, 2007. doi:10.2312/COMPAESTH/COMPAESTH07/057-064.
- 3 Sara Di Bartolomeo, Eduardo Puerta, Connor Wilson, Tarik Crnovrsanin, and Cody Dunne. A Collection of Benchmark Datasets for Evaluating Graph Layout Algorithms. In *Graph Drawing and Network Visualization (GD'23)*, volume 14466 of *LNCS*, pages 251–252. Springer, 2023. doi:10.31219/osf.io/yftju.
- 4 Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring Network Structure, Dynamics, and Function using NetworkX. In *Proceedings of the 7th Python in Science Conference*, pages 11–15, 2008.
- 5 Gavin J. Mooney, Helen C. Purchase, Michael Wybrow, and Stephen G. Kobourov. The Multi-Dimensional Landscape of Graph Drawing Metrics. In *2024 IEEE 17th Pacific Visualization Conference (PacificVis)*, pages 122–131. IEEE, 2024. doi:10.1109/PacificVis60374.2024.00022.
- 6 Helen C. Purchase. Metrics for Graph Drawing Aesthetics. *Journal of Visual Languages & Computing*, 13(5):501–516, 2002. doi:10.1006/jvlc.2002.0232.
- 7 Martyn Taylor and Peter Rodgers. Applying Graphical Design Techniques to Graph Visualisation. In *Ninth International Conference on Information Visualisation (IV'05)*, pages 651–656, London, England, 2005. IEEE. doi:10.1109/IV.2005.19.

AdMaTile: Visualizing Event-Based Adjacency Matrices in a Multiple-Coordinated-Views System

Nikolaus-Mathias Herl  

TU Wien, Austria

Velitchko Filipov  

TU Wien, Austria

Abstract

Conventional dynamic networks represent network changes via a discrete sequence of timeslices, which usually entails loss of information on fine-grained dynamics. Recently, event-based networks emerged as an approach to model this temporal (event-based) information more precisely. Adjacency-matrix-based visualizations of temporal networks are under-investigated in related literature and present a promising research direction for network visualization. Our approach *AdMaTile* (**A**djacency **M**atrix and **T**imeline **E**xplorer) is designed to visualize event-based networks using multiple matrix views, timelines, difference maps, and staged transitions.

2012 ACM Subject Classification Human-centered computing → Graph drawings; Human-centered computing → Visual analytics

Keywords and phrases Event-based, Temporal Graphs, Adjacency Matrix, Network Visualization

Digital Object Identifier 10.4230/LIPIcs.GD.2024.46

Category Poster Abstract

Supplementary Material *Software (Source Code)*: <https://bitbucket.org/NikiHerl/admatile>

Acknowledgements This work was supported by the FWF SANE project [10.55776/I6635].

1 Introduction

Dynamic graphs/networks extend the already versatile graph model with explicitly time-varying nodes, links, and/or attributes. Traditional dynamic network visualization [4] models graph dynamics as a sequence of states (i.e. timeslices) resulting in large memory requirements or loss of temporal information through aggregation. Event-based networks [7, 10] (i.e. temporal graphs) emerged as a solution to preserve these fine-grained changes, by recording the timestamps of individual graph events, i.e. (dis)appearances of nodes, links, or attributes. Current approaches to visualizing dynamic networks mostly utilize node-link diagrams to represent the graph’s structure [4] and visualize the temporal information using juxtaposition, animation, and space-time cubes [1, 3, 10, 2]. Juxtaposition encodes the temporal dimension as discrete timeslices side by side, obscuring fine temporal details that may be crucial, such as in contact tracing networks. Animation, on the other hand, a more natural way to encode time, avoids such issues but has limitations in tasks involving a comparison of distant timeslices [5]. The event-based *space-time cube* ($2D + t$) [10, 2] has been used for node-link diagrams/animations that capture finer details, particularly when many changes occur. Space-time-cubes have also been projected down to $2D$ [6], depicting node trajectories and guiding to interesting intervals (“when”) or structures in the graph (“where”). Our research contributes to temporal graph drawing with an adjacency-matrix-based visualization of temporal networks, addressing some drawbacks of existing methods. Adjacency matrices provide an alternative representation of (dynamic) graphs, which, unlike node-link diagrams, doesn’t suffer from (link-)overplotting as a graph’s size and density grow. By integrating



© Nikolaus-Mathias Herl and Velitchko Filipov;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 46; pp. 46:1–46:3

Leibniz International Proceedings in Informatics



LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

multiple matrix views with timelines, animation [9], difference maps [8], and animated staged transitions, our approach mitigates perceptual issues associated with animation and the information loss with juxtaposition.

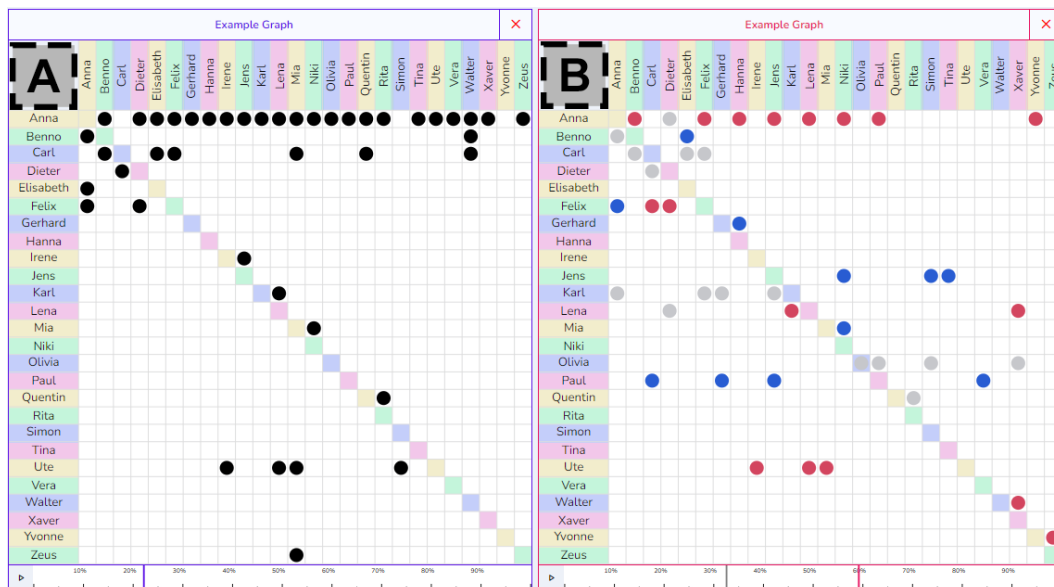


Figure 1 *AdMaTile* displaying two juxtaposed views of the same temporal network. **(A)** Shows the state of the graph at a specific point in time. **(B)** A difference map between the primary selected time (red cursor on timeline) and the right-clicked time (grey cursor).

2 AdMaTile

Our approach, *AdMaTile*, available at admatile.web.app, parses dynamic (event-based) graphs encoded in the GEXF file format. These are visualized in interactive views that combine adjacency matrices, timelines, difference maps, and animated staged transitions. Multiple graphs and multiple views of each graph can be displayed simultaneously enabling a comparison of patterns between specific moments in time or between graphs using juxtaposition (small-multiples) approach (see Figure 1). The evolution of the graph can be observed through animation. Left-clicking the timeline and brushing enables analyst-driven navigation of the temporal dimension (see Figure 1-A).

Right-clicking on the timeline evokes one of two views: (i) a staged transition between the selected and the right-clicked point in time; or (ii) a difference map of the two timestamps. In a staged transition, first “removed” links are highlighted red and shrink away, then “added” links are highlighted blue and grow to full dots, indicating the emergence of new relationships. Difference maps display the same information instantly, with links either colored so as to emphasize changes (see Figure 1-B), or emphasizing unchanged links. The instant behavior allows brushing the timeline (holding the right-click) to search and identify particularly different (or similar) states of the graph in time.

To aid the identification of edges’ source and target nodes, the node labels and their respective diagonal cells are colored in a cyclical color scheme. Furthermore, hovering over matrix cells displays a crosshair and highlights the source and target nodes.

References

- 1 Daniel Archambault, Helen Purchase, and Bruno Pinaud. Animation, Small Multiples, and the Effect of Mental Map Preservation in Dynamic Graphs . *IEEE Transactions on Visualization and Computer Graphics*, 2011. doi:10.1109/TVCG.2010.78.
- 2 Alessio Arleo, Silvia Miksch, and Daniel Archambault. Event-based Dynamic Graph Drawing without the Agonizing Pain. *Computer Graphics Forum*, 41(6):226–244, 2022. doi:10.1111/cgf.14615.
- 3 Benjamin Bach, Emmanuel Pietriga, and Jean-Daniel Fekete. Visualizing Dynamic Networks with Matrix Cubes. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 877–886. Association for Computing Machinery, 2014. doi:10.1145/2556288.2557010.
- 4 Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. A Taxonomy and Survey of Dynamic Graph Visualization. *Computer Graphics Forum*, 36(1):133–159, 2017. doi:10.1111/cgf.12791.
- 5 Velitchko Filipov, Alessio Arleo, Markus Bögl, and Silvia Miksch. On Network Structural and Temporal Encodings: A Space and Time Odyssey. *IEEE Transactions on Visualization and Computer Graphics*, 2023. doi:10.1109/TVCG.2023.3310019.
- 6 Velitchko Filipov, Davide Ceneda, Daniel Archambault, and Alessio Arleo. TimeLighting: Guidance-Enhanced Exploration of 2D Projections of Temporal Graphs. In *Graph Drawing and Network Visualization (GD '23)*, pages 231–245. Springer, 2024. doi:10.1007/978-3-031-49272-3_16.
- 7 Petter Holme and Jari Saramäki. Temporal Networks. *Physics Reports*, 519(3):97–125, 2012. doi:10.1007/978-3-642-36461-7.
- 8 Sébastien Rufiange and Michael J McGuffin. DiffAni: Visualizing Dynamic Graphs with a Hybrid of Difference Maps and Animation . *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2556–2565, 2013. doi:10.1109/TVCG.2013.149.
- 9 Sébastien Rufiange and Guy Melançon. Animatrix: A Matrix-based Visualization of Software Evolution. In *2014 second IEEE working conference on software visualization*, pages 137–146, 2014. doi:10.1109/VISSOFT.2014.30.
- 10 Paolo Simonetto, Daniel Archambault, and Stephen Kobourov. Event-based Dynamic Graph Visualisation. *IEEE Transactions on Visualization and Computer Graphics*, 26(7):2373–2386, 2018. doi:10.1109/TVCG.2018.2886901.

Strict Upward Planar Grid Drawings of Binary Trees with Minimal Area

Maarten Löffler 

Utrecht University, The Netherlands

Abstract

Given a rooted binary tree T and a tuple (w, h) , we wish to test whether there exists a strict upward drawing of T on a $w \times h$ grid; that is, a planar grid drawing with straight-line edges where every vertex has a strictly lower y -coordinate than its parent.

Biedl and Mondal [2] prove that this problem is NP-hard for general trees; their construction crucially uses nodes of very high degree. Akatiya et al [1] prove that the problem is also NP-hard for binary trees with a fixed combinatorial embedding; their construction crucially relies on the fixed embedding. Both pose the question for binary trees and a free embedding as an open problem.

Here, we show that this problem is also NP-hard.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases Upward drawings, grid drawings, minimal area

Digital Object Identifier 10.4230/LIPIcs.GD.2024.47

Category Poster Abstract

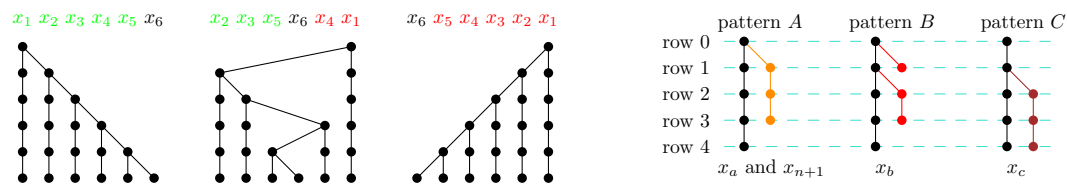
1 Result

► **Theorem 1.** *Testing whether a rooted binary tree T admits a strict upward planar embedding on a $w \times h$ grid is NP-complete.*

We reduce from *monotone not-all-equal-3SAT*, which is NP-hard by Schaefer’s dichotomy theorem [3]. In this problem, we are given a n variables and m clauses (triples of variables, since negative literals do not occur), and we need to find a variable assignment such that all variables in each clause are neither all **true** nor all **false**.

Given a 3SAT formula, we set $w = n + 4$ and $h = n + 4m + 1$ and construct a tree T as follows.

Variables



■ **Figure 1** (left) Three embeddings of the permutation gadget. (right) The three clause patterns.

Variables are represented by paths in T of length h . In a strict upward drawing, these paths must have one vertex on every row of the grid.

The top $n + 1$ rows contain the **permutation gadget**. In the i th row, we can choose to place the path representing x_i either on the left or on the right, which encodes the **true** and **false** states of x_i . We also add a **dummy variable** x_{n+1} , which has no truth assignment. Refer to Figure 1 (left).



© Maarten Löffler;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 47; pp. 47:1–47:3

Leibniz International Proceedings in Informatics

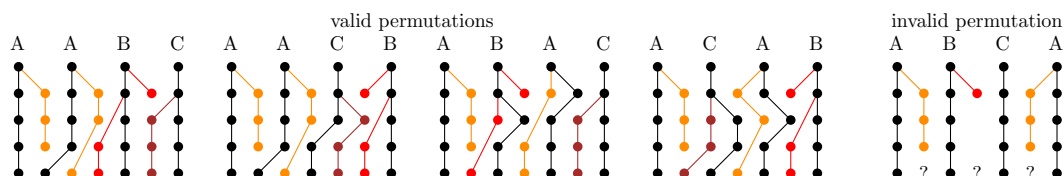


LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

47:2 Strict Upward Planar Grid Drawings of Binary Trees with Minimal Area

The vertices in the $n + 1$ st row, and thus the paths hanging from them, have 2^n possible permutations. Specifically, these permutations have all **true** variables to the left with increasing indices, and all **false** variables to the right with decreasing indices; these groups are separated by x_{n+1} .

Clauses



■ **Figure 2** The clause gadget can be embedded if not both outermost paths use pattern *A*.

Each clause is represented by four rows, which we index 1—4; we also refer to the last row of the previous gadget as row 0. A clause (x_a, x_b, x_c) with $a < b < c$ is encoded using the four paths for x_a , x_b , x_c , and x_{n+1} . Each of these has 3 additional vertices in one or two subtrees, for 12 additional vertices in total. The total width of the construction is $n + 4$; there are three “empty columns”, giving 12 empty spots in these four rows. Our gadget will allow us to fill exactly these spots if and only if the clause is satisfied. We use 3 different patterns for attaching the additional vertices, where we use the same pattern for both x_a and x_{n+1} . Refer to Figure 1 (right).

► **Lemma 2.** *There is a valid embedding if and only if the two outermost patterns are not both *A*.*

Proof sketch. For the “if” part, refer to Figure 2 for example valid embeddings. For the “only if” part, when both outermost paths use pattern *A*, then both of them must fill rows 1, 2, and 3 of the gadget, which leaves three empty spots in row 4 but only two remaining paths, which is impossible. ◀

Note that the variable assignment in which x_a , x_b , and x_c are all **true** results in the permutation x_a, x_b, x_c, x_{n+1} , and the assignment in which they are all **false** results in the symmetric permutation x_{n+1}, x_c, x_b, x_a , both of which have x_a and x_{n+1} on the outside, and all other variable assignments will not have x_{n+1} on the outside. Therefore, by Lemma 2 we encode exactly the not-all-equal property of a clause.

Full construction

Our full construction (refer to accompanying poster for an example) uses straight paths without additional vertices for each variable that is not part of a clause. The correctness relies the ability of these paths to not influence the gadget.

► **Lemma 3.** *The presence or absence of any vertical paths separating the four subtrees in a clause gadget has no influence on the satisfiability of the gadget.*

We also need the clause gadgets to be independent; that is, additional vertices in one clause gadget should not fill empty spots in different gadgets.

► **Lemma 4.** *The entire construction can be embedded if and only if each clause can be individually satisfied.*

2 Future Work

Our construction relies critically on the strictness of the drawings. What is the complexity of finding **non-strict** upward planar embeddings of trees on a given grid?

References

- 1 Hugo Akitaya, Maarten Löffler, and Irene Parada. How to fit a tree in a box. *Graphs and Combinatorics*, 2022.
- 2 Therese Biedl and Debajyoti Mondal. On upward drawings of trees on a given grid. In Fabrizio Frati and Kwan-Liu Ma, editors, *Graph Drawing and Network Visualization*, pages 318–325, Cham, 2018. Springer International Publishing.
- 3 Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, STOC '78, pages 216–226, New York, NY, USA, 1978. Association for Computing Machinery. doi:10.1145/800133.804350.

Determining Sugiyama Topology with Model Order

Sören Domrös  

Department of Computer Science, Kiel University, Germany

Reinhard von Hanxleden  

Department of Computer Science, Kiel University, Germany

Abstract

Traditional implementations of the Sugiyama algorithm optimize aesthetic criteria such as the number of backward edges, edge length, or edge crossings. If we, however, utilize the model order, as provided e. g. by a textual graph input file, we can determine the topology of a Sugiyama layout in a one-pass algorithm while controlling the secondary notation and with it the intention expressed by the underlying model, which typically cannot be captured by layout algorithms.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis

Keywords and phrases Automatic Layout, Model Order, Layered Layout

Digital Object Identifier 10.4230/LIPIcs.GD.2024.48

Category Poster Abstract

1 Introduction

Jünger and Mutzel [4] pointed out that in “many cases, such as the drawing of flowcharts, the input data can be expected to determine the choice” of edge reversal when explaining the cycle breaking step of the Sugiyama [6] algorithm.

We here propose to go even further and to use the “input data“, also referred to as *model order* [1, 2], to determine the whole topology of a layered layout.

2 Efficient Solutions to Layered Layout Topology

The *Sugiyama* or *layered* algorithm is divided into five phases: the *topological phases* cycle breaking, layer assignment, and crossing minimization and the *geometrical phases* node placement and edge routing. Domrös et al. [1] presented an approach to partially solve the topological phases in linear time, but requiring intermediate processors that sorted the nodes before each step. They argued that a manually written input model, e. g. a program in a domain-specific language, carries intention and secondary notation that can be transferred into the layout, which can typically not be captured by static layout algorithms [5]. We here showcase under which assumptions this is possible and show obviously not optimal (ONO) layouts that might be created by this approach. Directly controlling the layout by the input model increases control over the layout without introducing additional layout constraints. It also is naturally embedded in the typical editing workflow for programming.

We assume a directed ordered graph $G = (V, E)$ with an ordered set of nodes V and $|V| = n$ and ordered set of edges E with $|E| = e$ that corresponds to an input model with nodes ordered breadth-first based on their desired position and edges ordered based on their desired destination, as visualized in Figure 1.

The *cycle breaking step* can be trivially solved in $\mathcal{O}(e)$ by iterating over all edges and reversing all edges for which the source node has a higher model order than the target node [1], as seen in Figure 1a at the edges between n5 and n2. Note that this solution might not find the minimal feedback arc set, but assuming that the model order of the input model is intended this would be acceptable.



© Sören Domrös and Reinhard von Hanxleden;
licensed under Creative Commons License CC-BY 4.0

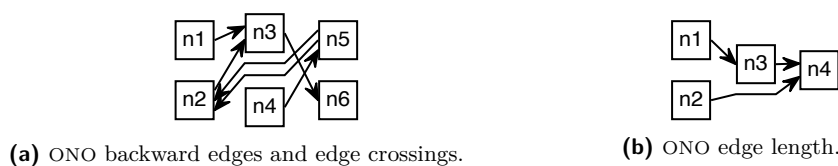
32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 48; pp. 48:1–48:3

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** Two example drawings that determine topology from model order. Figure 1a is not optimal regarding backward edges and edges crossings. Figure 1b is not optimal regarding edge length. However, both layouts reflect model order.

The *layer assignment step* assigns the first node to the first layer and continues iterating over all nodes and assigns them to the current layer until a node occurs that is connected to an already placed node, which creates a new layer, as seen in Figure 1b. By marking each target node of all already handled nodes as a node that needs to be in a new layer and assuming the breadth-first order of nodes, we can limit the runtime of the layer assignment to $\mathcal{O}(n + e)$ without considering dummy nodes. Additionally, we need to mark potential positions for long edge dummy nodes that need to be introduced if an edge spans multiple layers to create a proper layered graph. By considering the sources of a dummy node and the nodes in the previous layer, the dummy nodes can be directly assigned to their correct position in a layer without an additional sorting step. This improves the layer assignment post-processing proposed by Domrös et al. [1], which requires a sorting step. Again, this layer assignment may not minimize the edge length but instead expresses the model order.

Crossing minimization can be skipped entirely since the nodes are already in the intended order and the edge order corresponds to the order of the nodes in a sensible model. The order of outgoing edges on a node and the position of the dummy nodes introduced for long edges already determine the order of nodes, dummy nodes, and edges and hence the whole topology. Since this, however, limits the position of dummy nodes and can only statically assign dummy nodes to a specific position relative on real nodes, sorting, as proposed by Domrös et al. [1], or greedily switching nodes and edge anchor points as proposed by Eades and Kelly [3] might still be beneficial, which would increase the runtime complexity.

3 Obviously not Optimal Input Models

When comparing dummy nodes to real nodes that are part of the input model when deciding for potential routes during layer assignment, one has to statically decide whether a dummy node is below or above a so-called *dangling source node* that has no connection to a previous layer. E. g., in Figure 1a dummy nodes are placed above real nodes and as a result n4 is below the edges from n5 to n2. This means, we cannot control this by reordering the input model.

Another concern are edges that require an additional edge dummy in the same layer to route the edge around its node, which we also refer to as *feedback edges*. Here, similar to the dummy nodes compared to dangling source nodes, we cannot statically decide whether the feedback edge dummy should be above or below a node without creating unnecessary edge crossings that cannot be prevented by controlling the input model.

Finally, the approach presented here, while very fast, is probably only sensible if the model order of the input model carries indeed intention, as it does not aim to optimize standard aesthetic criteria such as crossing minimization. If only the order of nodes is intentional, or only the order of edges, one might prefer to fall back to approaches proposed by Domrös et al. [1, 2] that enforce node or edge order selectively, use them only as a tie-breaker, or utilize different partial ordering groups to only consider model order that corresponds to intention.

References

- 1 Sören Domrös, Max Riepe, and Reinhard von Hanxleden. Model order in Sugiyama layouts. In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 3: IVAPP*, pages 77–88. INSTICC, SciTePress, 2023. doi:10.5220/0011656700003417.
- 2 Sören Domrös and Reinhard von Hanxleden. Diagram control and model order for sugiyama layouts. In *Diagrammatic Representation and Inference, 14th International Conference, DIAGRAMS '24*. To be published, 2024.
- 3 Peter Eades and David Kelly. Heuristics for reducing crossings in 2-layered networks. *Ars Combinatoria*, 21:89–98, 1986.
- 4 Michael Jünger and Petra Mutzel, editors. *Graph Drawing Software*. Springer, 2004. doi:10.1007/978-3-642-18638-7.
- 5 Marian Petre. Why looking isn't always seeing: Readership skills and graphical programming. *Communications of the ACM*, 38(6):33–44, June 1995. doi:10.1145/203241.203251.
- 6 Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiro Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man and Cybernetics*, 11(2):109–125, February 1981. doi:10.1109/TSMC.1981.4308636.

Introducing Fairness in Graph Visualization

Seok-Hee Hong  

University of Sydney, Australia

Giuseppe Liotta  

University of Perugia, Italy

Fabrizio Montecchiani  

University of Perugia, Italy

Martin Nöllenburg  

TU Wien, Austria

Tommaso Piselli  

University of Perugia, Italy

Abstract

Information visualization tools are an essential component of many data-driven decision-making systems that rely on human feedback. The aim of this paper is to propose a novel research direction focused on *fair visualizations of graphs*.

2012 ACM Subject Classification Human-centered computing → Graph drawings

Keywords and phrases Network visualization, Fairness, Stress minimization

Digital Object Identifier 10.4230/LIPIcs.GD.2024.49

Category Poster Abstract

Related Version *Full Version*: <https://doi.org/10.2312/mlvis.20241124>

Supplementary Material

Software (Source Code): <https://github.com/tommaso-piselli/fairness-MLVis/tree/main>

archived at [swh:1:dir:378c9ba331d5a22b515264c6de24c45f6a40c065](https://swh.io/1/dir/378c9ba331d5a22b515264c6de24c45f6a40c065)

1 Motivation and Contribution

In a recent survey focused on bias in machine learning, Mehrabi et al. [7] define fairness as the absence of any prejudice or favoritism toward an individual or a group based on their inherent or acquired characteristics.

Information visualization tools are an essential component of many data-driven decision-making systems that rely on human feedback. Although there is a substantial body of literature on fairness in artificial intelligence and related fields, fairness issues in information visualization have been surprisingly overlooked. The aim of this paper is to propose a novel research direction on this topic, focused on *fair visualizations of graphs*.

Borrowing an example from [3], imagine two competing parties, the reds and the blues. Also, suppose we are given a visualization of the graph modeling the relationships among the parties' members. Using recent layout algorithms, we can optimize a desired set of quality criteria (see, e.g., [1]), hence producing a readable and effective layout of our graph. However, the global optimization process underlying our layout algorithm will not provide any guarantee that the readability of the visualization “around” red vertices will be of the same quality as for blue vertices. In fact, while nearly every graph drawing algorithm optimizes global metrics for the computed layout and can readily incorporate local constraints, only a few algorithms are capable of handling more general constraints at the subgroup level [2, 5]. In contrast, a fair visualization should ensure that no party is favored in terms of readability.



© Seok-Hee Hong, Giuseppe Liotta, Fabrizio Montecchiani, Martin Nöllenburg, and Tommaso Piselli; licensed under Creative Commons License CC-BY 4.0

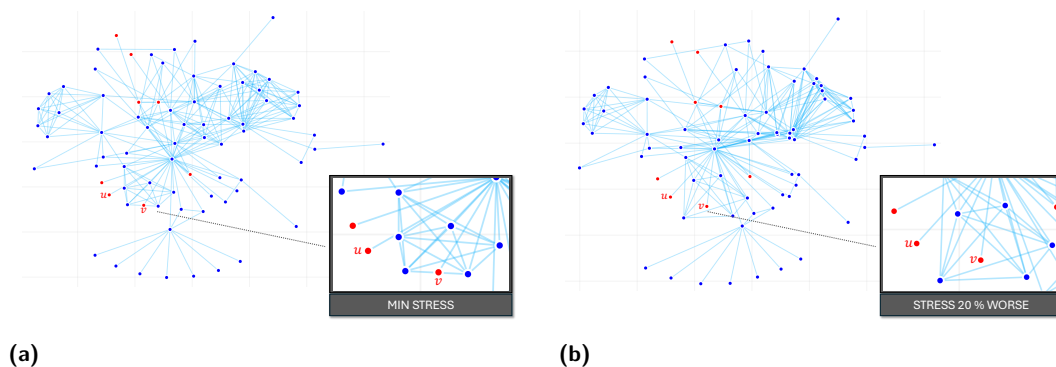
32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 49; pp. 49:1–49:3

Leibniz International Proceedings in Informatics



LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** Two straight-line drawings of the same graph. (a) is obtained by optimizing the stress function, while (b) is obtained starting from (a) and by subsequently optimizing the fairness function without worsening the stress by more than 20%. One can observe (zoomed windows) the increase on the readability around the two red vertices u and v when optimizing fairness (the red vertices are fewer than the blue ones). In particular, in (a), the edge incident to u overlaps with a blue vertex, while v overlaps with an edge between two blue vertices. Both ambiguities are resolved in (b).

This means that the potential visual complexity of the representation is equally distributed between the two sets, which becomes especially challenging when the cardinalities of the two sets are unbalanced. Although a fair drawing might be suboptimal in terms of overall readability, it offers greater insight to end users by balancing readability between the two vertex groups. Figure 1 illustrates an example of the impact of fairness on the readability of a straight-line drawing with several blue vertices and few red vertices.

Our results are as follows.

- We provide a conceptual contribution by formalizing the notion of fair straight-line graph drawings, based on the concept of stress, a well-known and widely adopted quality function (see, e.g., [4]). Clearly, the concept of fair straight-line drawings can be transferred to other quality criteria, as well as to other graph drawing paradigms.
- We present empirical results concerning the price of fairness to be paid in terms of additional stress with respect to stress-minimal (but potentially unfair) solutions. To this aim, we implement a gradient-descent based algorithm that can optimize multiple drawing criteria. Our investigation reveals that multi-objective functions that optimize fairness and stress together can output straight-line drawings with good fairness at the expenses of a relatively small increment of global stress.

Due to space limitations, we present below our fairness model for straight-line drawings, and we point the reader to [6] for an extended abstract of the paper.

2 Fairness of Straight-line Drawings

Let $G = (V, E)$ be a graph and let Γ be a straight-line drawing of G . For a pair of vertices $u, v \in V$, let $\delta(u, v)$ be the length of any shortest path in G between u and v . Also, let $\|\Gamma(u) - \Gamma(v)\|_2$ be the Euclidean distance of u and v in Γ . Moreover, let $\omega : V \times V \rightarrow \mathbb{Q}$ be a weighting function. The *stress* of Γ is defined as follows:

$$\text{stress}(\Gamma) = \sum_{u, v \in V} \omega(u, v) (\|\Gamma(u) - \Gamma(v)\|_2 - \delta(u, v))^2.$$

Assume now that the vertex set V of G is the union of two non-empty disjoint subgroups of vertices, that is, $V = V_R \cup V_B$ (with $V_R \neq \emptyset$ and $V_B \neq \emptyset$); vertices in V_R (V_B) are called *red* (*blue*). Thus, let $G = (V_R \cup V_B, E)$ be a graph and let Γ be a straight-line drawing of G . To convey the notion of fairness in Γ , we can refine the concept of stress by either focusing exclusively on the red vertices or on the blue vertices.

$$\text{stress}_R(\Gamma) = \sum_{u \in V_R, v \in V} \omega(u, v) (\|\Gamma(u) - \Gamma(v)\|_2 - \delta(u, v))^2$$



$$\text{stress}_B(\Gamma) = \sum_{u \in V_B, v \in V} \omega(u, v) (\|\Gamma(u) - \Gamma(v)\|_2 - \delta(u, v))^2$$

Ideally, Γ should not be unfair to any of the two sets of vertices, that is, the difference between $\text{stress}_R(\Gamma)$ and $\text{stress}_B(\Gamma)$ normalized by their cardinalities should be as close to zero as possible. More formally, we conveniently define the *unfairness* $\lambda(\Gamma)$ of Γ , whose minimization leads to a fair drawing: $\lambda(\Gamma) = \left(\frac{\text{stress}_R(\Gamma)}{|V_R|} - \frac{\text{stress}_B(\Gamma)}{|V_B|} \right)^2$.

References

- 1 Abu Reyan Ahmed, Felice De Luca, Sabin Devkota, Stephen G. Kobourov, and Mingwei Li. Multicriteria scalable graph drawing via stochastic gradient descent, *(SGD)²*. *IEEE Trans. Vis. Comput. Graph.*, 28(6):2388–2399, 2022. doi:10.1109/TVCG.2022.3155564.
- 2 Tim Dwyer. Scalable, versatile and simple constrained graph layout. *Comput. Graph. Forum*, 28(3):991–998, 2009. doi:10.1111/j.1467-8659.2009.01449.x.
- 3 Vincent Froese, Leon Kellerhals, and Rolf Niedermeier. Modification-fair cluster editing. In *AAAI 2022*, pages 6631–6638. AAAI Press, 2022. doi:10.1609/AAAI.V36I6.20617.
- 4 Emden R. Gansner, Yehuda Koren, and Stephen C. North. Graph drawing by stress majorization. In János Pach, editor, *GD 2004*, volume 3383 of *Lecture Notes in Computer Science*, pages 239–250. Springer, 2004. doi:10.1007/978-3-540-31843-9_25.
- 5 Jane Hoffswell, Alan Borning, and Jeffrey Heer. Setcola: High-level constraints for graph layout. *Comput. Graph. Forum*, 37(3):537–548, 2018. doi:10.1111/cgf.13440.
- 6 Seok-Hee Hong, Giuseppe Liotta, Fabrizio Montecchiani, Martin Nöllenburg, and Tommaso Piselli. Introducing Fairness in Graph Visualization via Gradient Descent. In Daniel Archambault, Ian Nabney, and Jaakko Peltonen, editors, *Machine Learning Methods in Visualisation for Big Data*. The Eurographics Association, 2024. doi:10.2312/mlvis.20241124.
- 7 Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6):115:1–115:35, 2022. doi:10.1145/3457607.

Level Planarity Is More Difficult Than We Thought

Simon D. Fink  



Algorithms and Complexity Group, Technische Universität Wien, Austria

Matthias Pfretzschner  

Faculty of Computer Science and Mathematics, Universität Passau, Germany

Ignaz Rutter  

Faculty of Computer Science and Mathematics, Universität Passau, Germany

Peter Stumpf  

Charles University, Prague, Czech Republic

Abstract

We consider three simple quadratic-time algorithms for LEVEL PLANARITY and give a level-planar instance that they either falsely classify as negative or for which they output a non-planar drawing.

2012 ACM Subject Classification Human-centered computing → Graph drawings

Keywords and phrases level planarity, 2-SAT, simple algorithm, counterexample

Digital Object Identifier 10.4230/LIPIcs.GD.2024.50

Category Poster Abstract

Related Version *Full Version including Poster*: <https://arxiv.org/abs/2409.01727>

Funding *Simon D. Fink*: Vienna Science and Technology Fund (WWTF) [10.47379/ICT22029].

Peter Stumpf: Czech Science Foundation grant GAČR 23-04949X.

1 Introduction

Given a graph $G = (V, E)$ and a *level assignment* $\ell : V(G) \rightarrow \mathbb{N}$, the problem LEVEL PLANARITY asks for a crossing-free drawing of G where vertices have their prescribed level as y -coordinate and all edges are y -monotone. When initially considering the problem in 1988, Di Battista and Nardelli [1] gave a linear-time algorithm for the restricted case where the graph is a hierarchy, i.e., only one vertex has no neighbors on a lower level. A subsequent attempt to extend this algorithm to the general case [7] was shown to be incomplete [10]. Jünger et al. finally gave the first linear-time algorithm for testing [11] and embedding [8, 9] level graphs around the turn of the millennium. Because this algorithm is quite involved, slower but simpler algorithms were developed by Randerath et al. [12], Healy and Kuusik [6], as well as Harrigan and Healy [5] in the decade thereafter. All these algorithms consider the pairwise ordering of vertices on the same level, greedily fixing an order for a (certain) pair and then checking for further orders implied by this. If the process terminates without finding a contradiction, we obtain a total vertex order for each level and thereby a level planar embedding. In the following, we give a level-planar counterexample that each known variant of this algorithm either incorrectly classifies as negative instance or correctly identifies as positive instance but outputs a drawing that is not planar. To the best of our knowledge, this leaves no correct *simple* embedding algorithm for level graphs. In particular, we are not aware of *any* correct implementation for embedding level-planar graphs.

Randerath et al. use an explicit 2-SAT formulation for the pairwise orders of vertices on the same level. Due to known gaps in the proof of Randerath et al., Brückner et al. [2, 3] showed this characterization via a 2-SAT formula is equivalent to the Hanani-Tutte-style characterization of LEVEL PLANARITY [4]. Thereby, our counterexample only breaks the proof of correctness as well as the embedder by Randerath et al., while their 2-SAT formulation still yields a correct *test* for LEVEL PLANARITY via this indirect proof [2, 3].



© Simon D. Fink, Matthias Pfretzschner, Ignaz Rutter, and Peter Stumpf;
licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 50; pp. 50:1–50:3

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

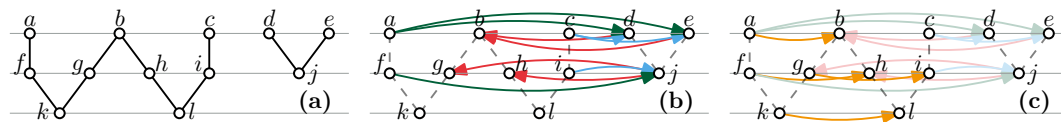
2 Randerath et al.

The algorithm by Randerath et al. [12] works as follows. First, edges spanning multiple levels are subdivided such that subsequently edges only occur between adjacent levels, resulting in a *proper* level graph. The planarity of the resulting graph is then tested using a 2-SAT formula. The formula contains a variable $(a < b)$ for every pair a, b of vertices that appear on the same level, encoding the relative order of these two vertices. For every pair of edges $uv, xy \in E$ with $\ell(u) = \ell(x) = \ell(v) + 1 = \ell(y) + 1$ with $u \neq x, v \neq y$ it adds the 2-SAT constraint $(u < x) \Leftrightarrow (v < y)$. Combining this with the constraints for antisymmetry $((a < b) \Leftrightarrow \neg(b < a))$ and transitivity $((a < b) \wedge (b < c) \Rightarrow (a < c))$ necessary for finding total orders yields a 3-SAT formula. However, Randerath et al. [12] show that omitting the transitivity constraints yields an *equisatisfiable* 2-SAT formula. To prove this equivalence, they show that the 2-SAT formula can be used to compute a level-planar embedding of the input graph. They greedily pick and assign equivalence classes of the formula in arbitrary order, but prioritize transitive closures where possible. Figure 1 shows a counterexample where the algorithm gives a false-negative answer when assigning classes in the shown order.

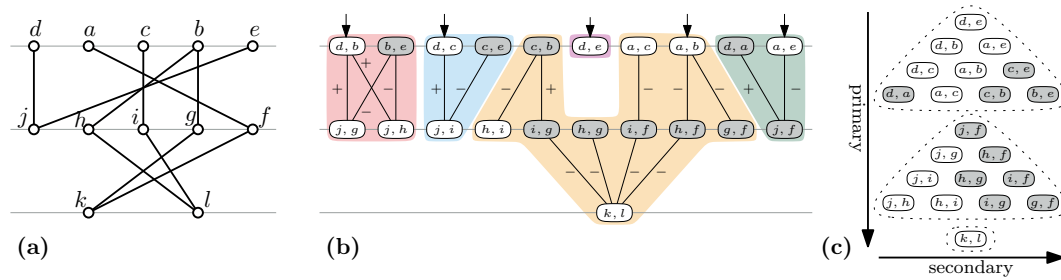
3 Healy and Kuusik & Harrigan and Healy

The algorithms by Healy and Kuusik [6] as well as the one by Harrigan and Healy [5] uses a similar concept. Instead of working with equivalence classes of a 2-SAT formula, they work with connected components of the closely related *vertex exchange graph* (ve-graph). This graph contains one vertex for every ordered pair of vertices that appear on the same level. Two vertices of the ve-graph are adjacent if they correspond to a pair of independent edges between the same levels. Starting with an arbitrary drawing \mathcal{L} of the input graph, the edges of the ve-graph are first labeled with $+$ or $-$, depending on whether the corresponding edges cross in \mathcal{L} . Subsequently, a DFS is used to test the ve-graph for odd-labeled cycles, which corresponds to a contradiction within a 2-SAT equivalence class. The two algorithms now differ slightly in how they continue to construct an embedding. Similar to Randerath et al., Healy and Kuusik [6] fix the orders of vertex pairs (i.e., whether all pairs of a connected ve-graph component are swapped or not) in an arbitrary order, also performing the transitive closure if possible. Thus, the processing order from Figure 1 also breaks this approach.

The later Harrigan and Healy approach [5] is slightly more involved. During the DFS traversal, they already change the relative order of some vertex pairs compared to the initial drawing \mathcal{L} [5, Algorithm 1]. Subsequently, the ve-graph is traversed again in a specific order and, for some vertices of \mathcal{L} , the chosen vertex order is flipped [5, Algorithm 2]. Using choices as shown in Figure 2, this does not yield a planar embedding even for a positive instance.



■ **Figure 1** (a) A level-planar graph G . (b) The green, blue, and red 2-SAT equivalence classes can be greedily assigned in this order. Subsequently, transitive closure forces $a < b$ as well as $i < g$, but the planarity constraints force $a < b \leftrightarrow f < h \leftrightarrow k < l \leftrightarrow g < i$ (c), yielding a contradiction.



■ **Figure 2** (a) An initial drawing \mathcal{L} for Figure 1a. (b) The corresponding labeled ve-graph. Arrows mark the chosen DFS entry points, pairs marked as swapped by Algorithm 1 are shown in gray. (c) The processing order for the vertices of the ve-graph in Algorithm 2.

References

- 1 Giuseppe Di Battista and Enrico Nardelli. Hierarchies and planarity theory. *IEEE Transactions on Systems, Man and Cybernetics*, 18(6):1035–1046, 1988. doi:10.1109/21.23105.
- 2 Guido Brückner, Ignaz Rutter, and Peter Stumpf. Level planarity: Transitivity vs. even crossings. In Therese Biedl and Andreas Kerren, editors, *Proceedings of the 26th International Symposium on Graph Drawing and Network Visualization (GD '18)*, volume 11282 of *Lecture Notes in Computer Science*, pages 39–52. Springer, 2018. doi:10.1007/978-3-030-04414-5_3.
- 3 Guido Brückner, Ignaz Rutter, and Peter Stumpf. Level-planarity: Transitivity vs. even crossings. *Electronic Journal of Combinatorics*, 29(4), 2022. doi:10.37236/10814.
- 4 Radoslav Fulek, Michael J. Pelsmajer, Marcus Schaefer, and Daniel Štefankovič. Hanani-Tutte, monotone drawings, and level-planarity. In János Pach, editor, *Thirty Essays on Geometric Graph Theory*, pages 263–287. Springer, 2013. doi:10.1007/978-1-4614-0110-0_14.
- 5 Martin Harrigan and Patrick Healy. Practical level planarity testing and layout with embedding constraints. In Seok-Hee Hong, Takao Nishizeki, and Wu Quan, editors, *Proceedings of the 15th International Symposium on Graph Drawing (GD '07)*, volume 4875 of *Lecture Notes in Computer Science*, pages 62–68. Springer, 2007. doi:10.1007/978-3-540-77537-9_9.
- 6 Patrick Healy and Ago Kuusik. Algorithms for multi-level graph planarity testing and layout. *Theoretical Computer Science*, 320(2-3):331–344, 2004. doi:10.1016/J.TCS.2004.02.033.
- 7 Lenwood S. Heath and Sriram V. Pemmaraju. Recognizing leveled-planar dags in linear time. In Franz-Josef Brandenburg, editor, *Proceedings of the 3rd International Symposium on Graph Drawing (GD '95)*, volume 1027 of *Lecture Notes in Computer Science*, pages 300–311. Springer, 1995. doi:10.1007/BFB0021813.
- 8 Michael Jünger and Sebastian Leipert. Level planar embedding in linear time. In Jan Kratochvíl, editor, *Proceedings of the 7th International Symposium on Graph Drawing (GD '99)*, volume 1731 of *Lecture Notes in Computer Science*, pages 72–81. Springer, 1999. doi:10.1007/3-540-46648-7_7.
- 9 Michael Jünger and Sebastian Leipert. Level planar embedding in linear time. *Journal of Graph Algorithms and Applications*, 6(1):67–113, 2002. doi:10.7155/JGAA.00045.
- 10 Michael Jünger, Sebastian Leipert, and Petra Mutzel. Pitfalls of using pq-trees in automatic graph drawing. In Giuseppe Di Battista, editor, *Proceedings of the 5th International Symposium on Graph Drawing (GD '97)*, volume 1353 of *Lecture Notes in Computer Science*, pages 193–204. Springer, 1997. doi:10.1007/3-540-63938-1_62.
- 11 Michael Jünger, Sebastian Leipert, and Petra Mutzel. Level planarity testing in linear time. In Sue Whitesides, editor, *Proceedings of the 6th International Symposium on Graph Drawing (GD '98)*, volume 1547 of *Lecture Notes in Computer Science*, pages 224–237. Springer, 1998. doi:10.1007/3-540-37623-2_17.
- 12 Bert Randerath, Ewald Speckenmeyer, Endre Boros, Peter L. Hammer, Alexander Kogan, Kazuhisa Makino, Bruno Simeone, and Ondrej Cepek. A satisfiability formulation of problems on level graphs. *Electronic Notes Discrete Mathematics*, 9:269–277, 2001. doi:10.1016/S1571-0653(04)00327-0.

Evolutionary Algorithms for One-Sided Bipartite Crossing Minimisation

Jakob Baumann  

University of Passau, Germany

Ignaz Rutter  

University of Passau, Germany

Dirk Sudholt  

University of Passau, Germany

Abstract

Evolutionary algorithms (EAs) are universal solvers inspired by principles of natural evolution. In many applications, EAs produce astonishingly good solutions. To complement recent theoretical advances in the analysis of EAs on graph drawing [1], we contribute a fundamental empirical study.

We consider the so-called ONE-SIDED BIPARTITE CROSSING MINIMISATION (OBCM): given two layers of a bipartite graph and a fixed horizontal order of vertices on the first layer, the task is to order the vertices on the second layer to minimise the number of edge crossings. We empirically analyse the performance of simple EAs for OBCM and compare different mutation operators on the underlying permutation ordering problem: exchanging two elements (*exchange*), swapping adjacent elements (*swap*) and jumping an element to a new position (*jump*). EAs using jumps easily outperform all deterministic algorithms in terms of solution quality after a reasonable number of generations. We also design variations of the best-performing EAs to reduce the execution time for each generation. The improved EAs can obtain the same solution quality as before and run up to 100 times faster.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Mutation Operator, Layered Graphs, Crossing Minimisation

Digital Object Identifier 10.4230/LIPIcs.GD.2024.51

Category Poster Abstract

Related Version *ArXiv Version*: <https://arxiv.org/abs/2409.15312>

1 Empirical Performance Comparison

We study the mutations swap, exchange, and jump on a simplistic $(1+1)$ -type EA [7], on ONE-SIDED BIPARTITE CROSSING MINIMISATION [3]. The $(1+1)$ -EAs (Swap-EA, Exchange-EA, Jump-EA) start with a random permutation and apply the corresponding operator k times, following a Poisson distribution with $\lambda = 1$. We also consider the randomised local search (RLS), where we set $k = 1$ constant. We compare the EAs to four state-of-the-art algorithms: The Barycenter and Median algorithms [3], Nagamochi’s algorithm [6], and a heuristic known as Sifting [5]. Nagamochi’s algorithm gives the best theoretical approximation ratio, but its performance was never empirically evaluated; a gap we aim to close with this work. We note that there are other well-performing algorithms, for which evaluations are readily available. We believe that the chosen subset is sufficient for this comparison.

We performed tests on three different instances, similar to [2]. Due to space limitations, and as there are no significant differences, we present only the results for *random instances*, with $n = 100$ vertices on both layers, where we added each edge with a fixed probability p . Note that we also considered differently sized layers and increasing density p ; the behaviour of RLS/EAs was basically the same, while the other algorithms were slightly affected, most of which was also covered in [2]. We computed the optimum solution using an ILP [4], which can solve instances of size up to $n \approx 190$. EAs and RLS perform a preprocessing step of computing the cross table [2], which takes $\Theta(nm)$ steps.



© Jakob Baumann, Ignaz Rutter, and Dirk Sudholt;
licensed under Creative Commons License CC-BY 4.0

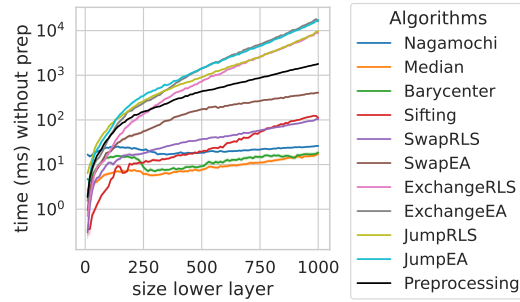
32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 51; pp. 51:1–51:3

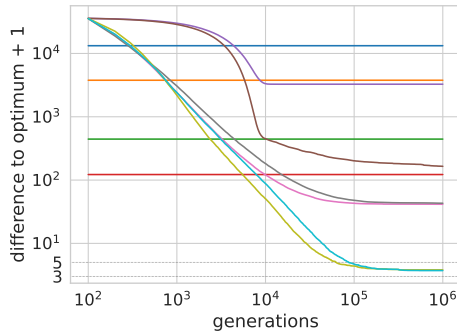
Leibniz International Proceedings in Informatics



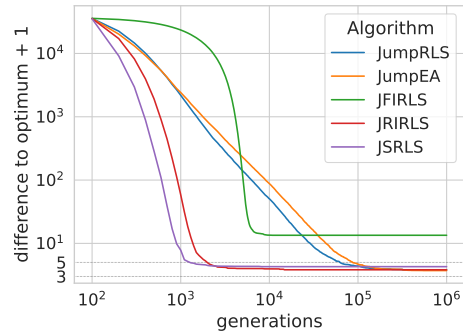
LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



(a) Preprocessing is needed for Sifting and EAs/RLS.



(b) Classical algorithms and simple EAs.



(c) Jump variants.

■ **Figure 1** (a) Wall-clock times averaged over the same 100 random instances with increasing n . Evolutionary algorithms were stopped when no improvement was found throughout $n^{1.5}$ subsequent generations. The costs for initialising the crossing matrix for the EAs and the Sifting algorithm are subtracted and shown separately. (b) & (c) Difference between the final evolved crossing number (for EAs) or the returned crossing number (for deterministic algorithms) and the optimal crossing number plotted over generations for classical algorithms and evolutionary algorithms. The plots show averages taken over a suite of instances.

Previous theoretical work [1] suggests that jump is the most effective mutation operator, which we confirm empirically. When given enough time, Jump-RLS/EA almost find nearly-optimal solutions, see Figure 1(b). We verify with statistical significance (using the Wilcoxon rank sum test [8]) that swaps are worse than exchange, which are in turn worse than jumps on the tested instances. The jump-operator also clearly outperforms all other state-of-the-art algorithms when given enough time.

While the jump-algorithms show the best performance, their running times are amongst the highest, see Figure 1. We improve the convergence-speed of Jump-RLS by not performing jumps at random, but by scanning for *acceptable jumps* (i.e. not increasing the crossings number), which does not increase the expected running time asymptotically. We propose three different strategies to make a choice among the acceptable moves found by the algorithm: Performing the first acceptable jump (*JFIRLS*), scanning all jumps and selecting an acceptable one uniformly at random (*JRIRLS*), and choosing the best jump (*JSRSL*). We tested the three algorithms on the same datasets. We verified with statistical significance that the *JFIRLS* is worse than the other two variants, which show roughly the same performance, see Figure 1(c). The *JRIRLS* and the *JSRSL* converge up to 100 times faster than a normal Jump-RLS or Jump-EA on these instances, which coincides with a factor of n .

References

- 1 Jakob Baumann, Ignaz Rutter, and Dirk Sudholt. Evolutionary computation meets graph drawing: Runtime analysis for crossing minimisation on layered graph drawings. In Xiaodong Li and Julia Handl, editors, *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2024, Melbourne, VIC, Australia, July 14-18, 2024*. ACM, 2024. To appear. doi:10.1145/3638529.3654105.
- 2 Camil Demetrescu and Irene Finocchi. Removing cycles for minimizing crossings. *ACM J. Exp. Algorithmics*, 6:2–es, 2001. doi:10.1145/945394.945396.
- 3 Peter Eades and Nicholas C. Wormald. Edge crossings in drawings of bipartite graphs. *Algorithmica*, 11(4):379–403, 1994. doi:10.1007/BF01187020.
- 4 Michael Jünger and Petra Mutzel. 2-layer straightline crossing minimization: Performance of exact and heuristic algorithms. *J. Graph Algorithms Appl.*, 1(1):1–25, 1997. doi:10.7155/jgaa.00001.
- 5 Christian Matuszewski, Robby Schönfeld, and Paul Molitor. Using sifting for k -layer straightline crossing minimization. In *Graph Drawing, 7th International Symposium, GD'99, Střirín Castle, Czech Republic, September 1999, Proceedings*, pages 217–224. Springer, Springer, 1999. doi:10.1007/3-540-46648-7_22.
- 6 Hiroshi Nagamochi. An improved bound on the one-sided minimum crossing number in two-layered drawings. *Discrete & Computational Geometry*, 33:569–591, 2005. doi:10.1007/s00454-005-1168-0.
- 7 Pietro S. Oliveto, Jun He, and Xin Yao. Time complexity of evolutionary algorithms for combinatorial optimization: A decade of results. *International Journal of Automation and Computing*, 4(3):281–293, 2007. doi:10.1007/S11633-007-0281-3.
- 8 Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

Polygonally Anchored Graph Drawing

Alvin Chiu 

University of California, Irvine, CA, USA

Ahmed Eldawy 

University of California, Riverside, CA, USA

Michael T. Goodrich 

University of California, Irvine, CA, USA

Abstract

We investigate force-directed graph drawing techniques under the constraint that some nodes must be anchored to stay within a given polygonal region associated with it (i.e. some positional information is known). The low energy layouts produced by such algorithms may reveal geographic information about nodes with no such knowledge a priori. Some applications of graph drawing with partial positional information include location-based social networks and rail networks, where the geographical locations need not be precise.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases polygonal anchors, force-directed

Digital Object Identifier 10.4230/LIPIcs.GD.2024.52

Category Poster Abstract

Funding *Ahmed Eldawy*: Supported by NSF fund IIS-2046236.

Michael T. Goodrich: Supported by NSF grant 2212129.

1 Introduction

We explore polygonally anchored graph drawing, where some nodes may have positional information in the form of a polygonal region. In particular, we use the standard force-directed graph layout algorithm by Fruchterman-Reingold [4], modified to restrict nodes to their associated “anchor” region (if specified). The low energy layouts produced by such algorithms may reveal geographic information about nodes with no such knowledge a priori. Some applications of graph drawing with partial positional information include location-based social networks and rail networks. Work by social scientists supports the idea that one’s social network (of family, friends, coworkers, etc.) is one of the best predictors for the region they identify with [1]. And in rail networks, stations and connections are often associated with the area they bring service to, so their placement in a map may be ambiguous [7, 2].

Related work has considered anchored graph drawing, where the input graph is assumed to have full positional information that must be respected in some way [6]. Other works have used centroidal forces as we do to keep a node inside a given region [8].

We investigate three different metrics of forces used to constrain a node to its associated region. We call this region its *anchor*, and a node with an anchor is called a *vessel*. In the Fruchterman-Reingold algorithm, repulsive forces between nodes and attractive forces between adjacent nodes are applied iteratively until the global “temperature” of the system decays to 0, a quantity that controls the amount of displacement [5]. Our modification is to introduce an additional force from the vessel nodes to their anchors that is applied in each of these iterations. In order to ensure that the vessel remains tethered to its anchor region, we apply a displacement force towards the region (where exactly is determined by the metric), multiplied by a sufficiently large constant to act as if it were an “infinite force”.



© Alvin Chiu, Ahmed Eldawy, and Michael T. Goodrich;
licensed under Creative Commons License CC-BY 4.0

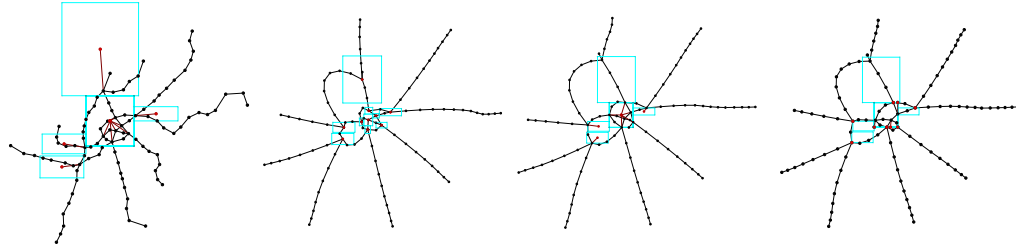
32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 52; pp. 52:1–52:3

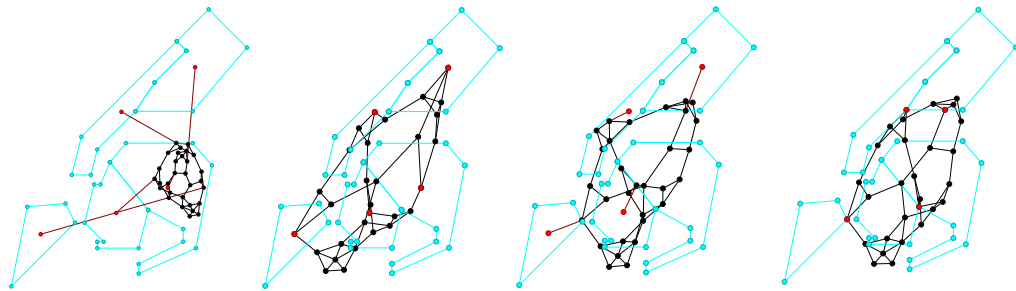
Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



(a) Input Graph. (b) Centroidal Force. (c) Inside-Out Force. (d) Closest Point Force.



■ **Figure 1** The black nodes and edges denote the original input graph, while the teal denotes the anchor regions. The anchoring forces are shown with a red edge connecting the vessel to the last point to pull it (the red node may overlap the vessel). The top image shows an input graph of the Vienna subway map with real geographical locations, where the 10 stations with intersections/transfers are anchored geographically. The bottom image shows a social network in Queens, one of the 5 boroughs of NYC. 5 nodes are then anchored to one of the 5 boroughs.

In the first metric that we call the *centroidal force*, every point in the anchor region applies an displacement force (the displacement vector from the vessel to the point) on the vessel node. We show later that this metric is identical to applying one singular force from the anchor’s centroid, scaled up by the anchor’s area. Hence, this force degenerates into the case where the anchors are simply points, which has been explored in prior work [3].

The second metric, *inside-out force*, is similar to the first in that we pull a vessel node inside to the anchor’s centroid, but we only apply this force if the vessel is outside the anchor region (and so zero force is applied if inside).

Finally, the third metric is the *closest point force* that also applies zero force if the vessel node is already inside the anchor region, but otherwise applies a force from the closest point in the anchor to the vessel.

We now show that applying a force from all points in a region is indeed equivalent to applying one singular force from the region’s centroid, scaled by the region’s area. Let $v = (v_x, v_y)$ be a vessel point attached to an anchor point (x, y) . Then the force vector F we apply on the vessel is $F = C \cdot \langle x - v_x, y - v_y \rangle$ for some large constant C .

Let P be the region defined by an anchor with area A . Its centroid will be given by the point $c = (c_x, c_y)$, where

$$c_x = \frac{\iint_P x \, dx \, dy}{\iint_P dx \, dy} = \frac{\iint_P x \, dx \, dy}{A}, c_y = \frac{\iint_P y \, dx \, dy}{\iint_P dx \, dy} = \frac{\iint_P y \, dx \, dy}{A}$$

If every point (x, y) in an anchor region P applies a force $C \cdot \langle x - v_x, y - v_y \rangle$ on the vessel point, then the total force applied will be

$$F_x = \iint_P C \cdot (x - v_x) dx dy = C \cdot \iint_P x dx dy - C v_x \cdot \iint_P dx dy = CA(c_x - v_x)$$

$$F_x = \iint_P C \cdot (x - v_x) dx dy = C \cdot \iint_P x dx dy - C v_x \cdot \iint_P dx dy = CA(c_x - v_x)$$

Notice that this force is equal to just one force being applied from the centroid c , scaled up by the anchor's area A .

References

- 1 Zack W. Almquist and Carter T. Butts. Predicting regional self-identification from spatial network models. *Geographical Analysis*, 47(1):50–72, 2015. doi:10.1111/gean.12045.
- 2 Ulrik Brandes and Dorothea Wagner. Using graph layout to visualize train interconnection data. In Sue H. Whitesides, editor, *Graph Drawing*, pages 44–56, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. doi:10.1007/3-540-37623-2_4.
- 3 Giordano Da Lozzo, Marco Di Bartolomeo, Maurizio Patrignani, Giuseppe Di Battista, Davide Cannone, and Sergio Tortora. Drawing georeferenced graphs - combining graph drawing and geographic data. In *IVAPP 2015 - Proceedings of the 6th International Conference on Information Visualization Theory and Applications, Berlin, Germany, 11-14 March, 2015*, pages 109–116. SciTePress, 2015. doi:10.5220/0005266601090116.
- 4 Thomas M. J. Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991. doi:10.1002/spe.4380211102.
- 5 Stephen G Kobourov. Spring embedders and force directed graph drawing algorithms. *arXiv preprint arXiv:1201.3011*, 2012. arXiv:1201.3011.
- 6 Kelly Lyons, Henk Meijer, and David Rappaport. Algorithms for cluster busting in anchored graph drawing. *Journal of Graph Algorithms and Applications*, 2(1):1–24, 1998. doi:10.7155/JGAA.00004.
- 7 Alex Wolff. Drawing subway maps: A survey. *Informatik Forsch. Entw*, 22:23–44, 2007. doi:10.1007/s00450-007-0036-y.
- 8 Hsiang-Yun Wu, Martin Nöllenburg, and Ivan Viola. Multi-level area balancing of clustered graphs. *IEEE Transactions on Visualization and Computer Graphics*, 28(7):2682–2696, 2022. doi:10.1109/TVCG.2020.3038154.

String Graph with Cop Number 4

Stephane Durocher ✉

Department of Computer Science, University of Manitoba, Winnipeg, Canada

Myroslav Kryven ✉ 

Department of Computer Science, University of Manitoba, Winnipeg, Canada

Maarten Löffler ✉

Department of Information and Computing Sciences, Utrecht University, The Netherlands

Abstract

Cops and Robbers is a well-studied pursuit-evasion game in which a set of cops seeks to catch a robber in a graph G , where cops and the robber move along edges of G . The *cop number* of G is the minimum number of cops that is sufficient to catch the robber. The game of Cops and Robbers has been well-studied on *beyond-planar graphs* (that is, graphs that can be drawn with only few crossings) [1, 4] as well as *intersection graphs* (that is, graphs where the vertices represent geometric objects, and an edge exists between two vertices if the corresponding objects intersect). We consider a well-known subclass of intersection graphs called *string graphs* where the objects are curves. So far no string graph with cop number larger than three was known. We construct the first string graph with cop number four, which improves the previous bound and answers an open question by Gavenčiak et al. [5].

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis

Keywords and phrases point set embedding, upward planar path embedding, dynamic programming

Digital Object Identifier 10.4230/LIPIcs.GD.2024.53

Category Poster Abstract

The game of cops and robbers

One of the most common and well-studied pursuit-evasion problems is the game of *Cops and Robbers* on graphs, which was formalized by Quilliot [8] and Nowakowski and Winkler [7] in the 1980s; see also the recent book by Bonato and Nowakowski [2]. The game is played in a graph by two players: the robber player and the cop player. The common assumption that we also adopt here is that each player has full information about the graph and the other player's moves. The game consists of rounds (or steps) on a given graph. In the initial round, the cop player selects starting vertices for a set of cops, and then the robber player selects a starting vertex for a robber. In the subsequent rounds, the players alternate turns; during the cops' turn, the cop player may move some of the cops to adjacent vertices. Similarly, the robber player may move the robber to an adjacent vertex during the robber's turn. The cop player wins if the robber and any of the cops are simultaneously on the same vertex; otherwise, when the game continues indefinitely, the robber player wins. If a single cop suffices to catch the robber in a graph G , even when the robber plays adversarially, then G is a *cop-win graph*; otherwise, G is a *robber-win graph*. The minimum number of cops necessary to catch the robber in G , denoted $c(G)$, is called the *cop number* of G .

String graphs

The class of *string graphs* is the class of intersection graphs of *strings* where each string is a bounded curve in the plane, i.e., a continuous image of the interval $[0, 1]$ into \mathbb{R}^2 . One of the most notable subclasses of string graphs is the class of *k -string graphs*, which are the graphs admitting a string representation in which every two curves intersect in at most k points. It is known that k -string graphs is a strict subclass of $k + 1$ -string graphs for all $k \geq 1$ and this inclusions is strict [3, 6].



© Stephane Durocher, Myroslav Kryven, and Maarten Löffler;
licensed under Creative Commons License CC-BY 4.0

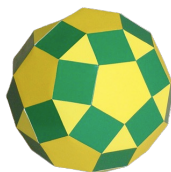
32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 53; pp. 53:1–53:3

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** Rhombicosidodecahedron.

The *girth* of a graph is the length of a shortest cycle in the graph.

► **Lemma 1** ([1]). *A graph with min degree δ and girth at least five has cop number at least δ .*

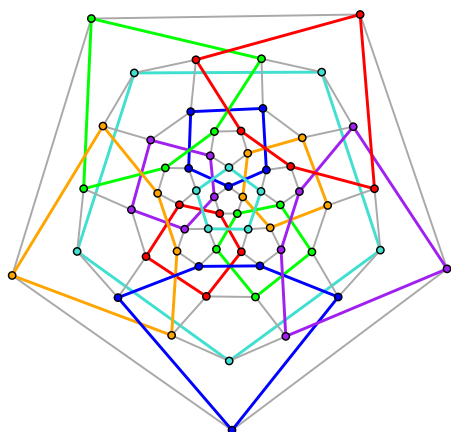
In the following we construct a 1-string graph with minimum degree four and girth five. And thus, as implied by Lemma 1, its cop number is at least four. This answers an open question by Gavenčiak et al. [5] as well as an open question on Math Stack Exchange [9].

1-string graph with cop number 4

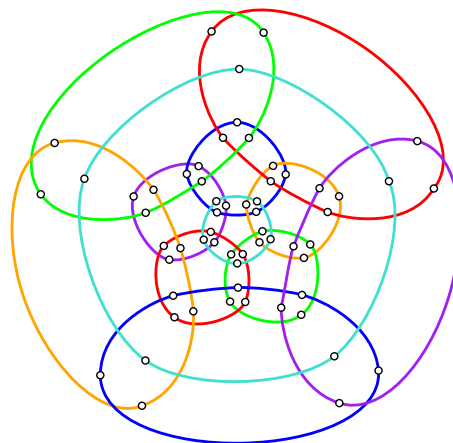
► **Theorem 2.** *There is a 1-string graph G with minimum degree four and girth at least five.*

Proof. Our construction is based on an Archimedean solid Rhombicosidodecahedron R ; see Figure 1. We consider a planar drawing of R and subdivide each edge incident to a triangular face with a *subdivision* vertex, let the resulting graph be R' ; see Figure 2b. To distinguish the vertices of R in R' from the subdivision vertices (marked with dots in Figure 2b), we call the vertices of R the *original* vertices (the crossings in Figure 2b).

To construct our string graph G we cover the edges of R' with the strings so that the strings cross *internally* (that is, on each string s participating in the crossing, the crossing is between two other intersections on s) at the original vertices and the endpoints of the strings intersect at the subdivision vertices. More precisely, we partition the edges of R' into twelve *colored* cycles; see colored cycles in Figure 2b. Each such cycle consists of five strings. Each string is between two consecutive subdivision vertices of a cycle and two consecutive strings in the cycle intersect at their endpoints. Figure 2a shows a straight-line embedding of G .



(a) Straight-line drawing of the graph G .



(b) Planar drawing of the graph R' and the string representation of the graph G .

■ **Figure 2** The string graph G and its string representation.


It is easy to check that every string s intersects with exactly four other strings: two strings intersect s at its endpoints and two strings cross s internally.

Now let us reassure that there are no cycles of length three or four in G . Notice that such a cycle, must have its edges from at most four different colored cycles in Figure 2b. By simply checking every set of at most four colored cycles, we can see that there is no cycle of length three or four. ◀

References

- 1 M. Aigner and M. Fromme. A game of cops and robbers. *Discrete Applied Mathematics*, 8(1):1–12, 1984. doi:10.1016/0166-218X(84)90073-8.
- 2 A. Bonato and R.J. Nowakowski. *The Game of Cops and Robbers on Graphs*. Student mathematical library. American Mathematical Society, 2011.
- 3 Petr Chmel and Vít Jelínek. String Graphs with Precise Number of Intersections. In *31st International Symposium on Graph Drawing and Network Visualization*, pages 78–92, Berlin, Heidelberg, 2024. Springer-Verlag. doi:10.1007/978-3-031-49272-3_6.
- 4 Stephane Durocher, Shahin Kamali, Myroslav Kryven, Fengyi Liu, Amirhossein Mashghdoust, Avery Miller, Pouria Zamani Nezhad, Ikaro Penha Costa, and Timothy Zapp. Cops and robbers on 1-planar graphs. In Michael A. Bekos and Markus Chimani, editors, *Graph Drawing and Network Visualization (GD)*, volume 14466 of *LNCS*, pages 3–17. Springer, 2023. doi:10.1007/978-3-031-49275-4_1.
- 5 Tomáš Gavenčiak, Przemysław Gordinowicz, Vít Jelínek, Pavel Klavík, and Jan Kratochvíl. Cops and robbers on intersection graphs. *European Journal of Combinatorics*, 72:45–69, 2018. doi:10.1016/j.ejc.2018.04.009.
- 6 Jan Kratochvíl and Jiří Matoušek. Intersection graphs of segments. *Journal of Combinatorial Theory, Series B*, 62(2):289–315, 1994. doi:10.1006/jctb.1994.1071.
- 7 Richard J. Nowakowski and Peter Winkler. Vertex-to-vertex pursuit in a graph. *Discrete Mathematics*, 43:235–239, 1983. doi:10.1016/0012-365X(83)90160-7.
- 8 Alain Quilliot. *Jeux et pointes fixes sur les graphes*. PhD thesis, Université de Paris VI, 1978.
- 9 Anonymus User and Misha Lavrov. String graph with girth 5 and min degree 4. Math Stack Exchange, 2017. URL: <https://math.stackexchange.com/questions/2241951/string-graph-with-girth-5-and-min-degree-4>.

Approximating the Crossing Number of Dense Graphs

Oriol Solé Pi  

Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract

We present a deterministic $n^{2+o(1)}$ -time algorithm that approximates the crossing number of any graph G of order n up to an additive error of $o(n^4)$, as well as a randomized polynomial-time algorithm that constructs a drawing of G with $\text{cr}(G) + o(n^4)$ crossings. These results imply a $(1 + o(1))$ -approximation algorithm for the crossing number of dense graphs. Our work builds on the machinery used by Fox, Pach and Súk [10], who obtained similar results for the rectilinear crossing number.

2012 ACM Subject Classification Mathematics of computing → Graph theory; Mathematics of computing → Graph algorithms; Mathematics of computing → Approximation algorithms

Keywords and phrases Crossing numbers, Approximation algorithms, Geometric graph theory

Digital Object Identifier 10.4230/LIPIcs.GD.2024.54

Category Poster Abstract

Related Version *Full Version:* <https://arxiv.org/abs/2401.00665>

1 Preliminaries and previous work

Let G be a finite simple graph. The *crossing number* of G , denoted by $\text{cr}(G)$, is the minimum number of crossing points between edges, where the minimum is taken over all drawings of G on the plane. A *straight-line drawing* of G is a drawing such that each edge is represented by a segment joining the corresponding endpoints. The *rectilinear crossing number* of G , $\overline{\text{cr}}(G)$, is the least number of crossings amongst all straight-line drawings of G . Clearly, $\text{cr}(G) \leq \overline{\text{cr}}(G)$, and it is known that there are graphs for which the inequality is strict [5]. The crossing number and the rectilinear crossing number have been studied extensively, and we refer the reader to the comprehensive monograph of Schaefer [17] for a review of the existing literature and several interesting questions.

Computing the crossing number is known to be NP-complete [11], while determining the rectilinear crossing numbers is complete for the existential theory of reals [4], and hence NP-hard. For any fixed k , there is a linear time algorithm that decides whether $\text{cr}(G) \leq k$. In contrast, it is NP-hard to determine if $\overline{\text{cr}}(G) \leq k$ holds. A considerable amount of work has been put into developing approximation algorithms for both $\text{cr}(G)$ and $\overline{\text{cr}}(G)$. A graph drawing technique of Bhatt and Leighton [3] and the approximation algorithm for optimal balanced cuts of Arora et al. [2] can be used to find, in polynomial time, a straight-line drawing of any bounded degree n -vertex graph G with no more than $O(\log^4 n(n + \text{cr}(G)))$ crossings. It wasn't until several years later that Chuzhoy [6], using the edge planarization method from [8], found a polynomial-time $O(n^{9/10})$ -approximation algorithm for $\text{cr}(G)$ for bounded degree graphs (by this, we mean a multiplicative approximation). Building on this method further, Kawarabayashi and Sidiropoulos [12, 13] improved the approximation ratio to $O(n^{1/2})$, and then Mahabadi and Tan [7] found a randomized $O(n^{1/2-\delta})$ -approximation algorithm, where $\delta > 0$ is a constant.



© Oriol Solé Pi;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 54; pp. 54:1–54:3

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

It is well known that if an n -vertex graph G is dense (i.e., it has $\Omega(n^2)$ edges) then both $\text{cr}(G)$ and $\overline{\text{cr}}(G)$ are of order $\Omega(n^4)$ (this follows from the celebrated crossing lemma of Ajtai et al. [1] and Leighton [14]). Fox, Pach and Suk [10] presented an algorithm that constructs a straight-line drawing of G with $\overline{\text{cr}}(G) + o(n^4)$ crossings. If G is dense then this algorithm produces a drawing with $(1 + o(1))\overline{\text{cr}}(G)$ crossings.

2 Our results

We have obtained a close analog of the result from [10] for the crossing number.

► **Theorem 1.** *There exists a deterministic $n^{2+o(1)}$ -time algorithm that for any given n -vertex graph G approximates $\text{cr}(G)$ up to an additive error of $O(n^4/(\log \log n)^\delta)$. Furthermore, there is a randomized polynomial-time algorithm that, with probability $1 - o(1)$, computes a drawing of G with $\text{cr}(G) + O(n^4/(\log \log n)^\delta)$ crossings. Here, δ denotes an absolute positive constant.*

The approximation part of the algorithm follows the same strategy as the one for rectilinear crossing numbers:

1. We are given an n -vertex graph $G = (V, E)$ as input.
2. Set $\varepsilon = (\log \log n)^{-\frac{1}{2c}}$ for some suitable absolute constant c and find an equitable Frieze-Kannan ε -regular partition $\mathcal{P} = \{V_1, V_2, \dots, V_k\}$ of G using the algorithm in [9], where $k \leq O(2^{\sqrt{\log \log n}})$. This takes $n^{2+o(1)}$ time.
3. Construct the edge weighted graph G/\mathcal{P} which has a vertex for each V_i and where the edge between V_i and V_j has weight equal to the number of edges between these two sets. Then, compute the crossing number of G/\mathcal{P} (a crossing between edges of weights w_1 and w_2 has weight $w_1 w_2$) by brute force and output this quantity. This can be done in $n^{o(1)}$ time.

3 Overview of the proof of correctness

The main novel ingredient which makes it possible to prove the correctness of the above algorithm is the following bound on the difference between the crossing numbers of two graphs on the same vertex set in terms of their distance in the *(labeled) cut metric*. The definition of this metric can be found, for example, in [15].

► **Theorem 2.** *Let G_1 and G_2 be graphs with the same vertex set V . If $d_\square(G_1, G_2) \geq n^{-4}$, then $|\text{cr}(G_1) - \text{cr}(G_2)| \leq C d_\square(G_1, G_2)^{1/4} n^4$, where C is an absolute constant.*

Proof sketch. Start with a drawing \mathcal{D} of G_1 which attains $\text{cr}(G_1)$; we will use \mathcal{D} as a blueprint to construct a drawing of G_2 with few crossings. After adding a node at each crossing of \mathcal{D} , we arrive at a planar map. By carefully and repeatedly applying a planar cycle separator theorem due to Miller [16], along with some packing and covering arguments, for any $t \in (0, 1)$ it is possible to subdivide the plane into $r = O(1/t^2)$ connected regions which contain no more than $\lceil t^2 n \rceil$ vertices of G and satisfy the following key property: Any vertex of G inside the region and any point on its boundary can be connected by a curve that has no more than tn^2 intersection points with the edges of the drawing that have no endpoint in that same region.

Let P_1, P_2, \dots, P_r denote the sets of vertices within each of the r regions of the subdivision. As long as $d_\square(G_1, G_2)$ is small, the number of edges between P_i and P_j will be similar in G_1 and G_2 . For each edge e in G_2 between P_i and P_j ($i \neq j$), choose an edge e' of G_1 between the same sets uniformly at random and route e along e' . Since e and e' might have different endpoints, we need to do some adjustments near the endpoints of e . Because of the

aforementioned property that the regions of the subdivision possess, these adjustments can be carried out without incurring in too many additional crossings. The edges of G_2 that have both endpoints in P_i can be added to the drawing at the end without many complications. One can show that the expected number of crossings in such a drawing of G_2 is no more than $cr(G_1) + Cd_{\square}(G_1, G_2)^{1/4}n^4$. ◀

References

- 1 Miklós Ajtai, Vasek Chvátal, Monty Newborn, and Endre Szemerédi. Crossing-free subgraphs. *North-holland Mathematics Studies*, 60:9–12, 1982.
- 2 Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)*, 56(2):1–37, 2009. doi:10.1145/1502793.1502794.
- 3 Sandeep N Bhatt and Frank Thomson Leighton. A framework for solving VLSI graph layout problems. *Journal of Computer and System Sciences*, 28(2):300–343, 1984. doi:10.1016/0022-0000(84)90071-0.
- 4 Daniel Bienstock. Some provably hard crossing number problems. In *Proceedings of the sixth annual symposium on Computational geometry*, pages 253–260, 1990. doi:10.1145/98524.98581.
- 5 Daniel Bienstock and Nathaniel Dean. Bounds for rectilinear crossing numbers. *Journal of Graph theory*, 17(3):333–348, 1993. doi:10.1002/JGT.3190170308.
- 6 Julia Chuzhoy. An algorithm for the graph crossing number problem. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 303–312, 2011. doi:10.1145/1993636.1993678.
- 7 Julia Chuzhoy, Sepideh Mahabadi, and Zihan Tan. Towards better approximation of graph crossing number. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 73–84. IEEE, 2020. doi:10.1109/FOCS46700.2020.00016.
- 8 Julia Chuzhoy, Yury Makarychev, and Anastasios Sidiropoulos. On graph crossing number and edge planarization. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete algorithms*, pages 1050–1069. SIAM, 2011. doi:10.1137/1.9781611973082.80.
- 9 Domingos Dellamonica, Subrahmanyam Kalyanasundaram, Daniel M Martin, VOJTĚCH RÖDL, and Asaf Shapira. An optimal algorithm for finding frieze–kannan regular partitions. *Combinatorics, Probability and Computing*, 24(2):407–437, 2015. doi:10.1017/S0963548314000200.
- 10 Jacob Fox, János Pach, and Andrew Suk. Approximating the rectilinear crossing number. In *International Symposium on Graph Drawing and Network Visualization*, pages 413–426. Springer, 2016. doi:10.1007/978-3-319-50106-2_32.
- 11 Michael R Garey and David S Johnson. Crossing number is np-complete. *SIAM Journal on Algebraic Discrete Methods*, 4(3):312–316, 1983.
- 12 Ken-ichi Kawarabayashi and Anastasios Sidiropoulos. Polylogarithmic approximation for minimum planarization (almost). In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 779–788. IEEE, 2017. doi:10.1109/FOCS.2017.77.
- 13 Ken-ichi Kawarabayashi and Anastasios Sidiropoulos. Polylogarithmic approximation for euler genus on bounded degree graphs. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 164–175, 2019. doi:10.1145/3313276.3316409.
- 14 Frank Thomson Leighton. *Complexity Issues in VLSI: Optimal Layouts for the Shuffle-Exchange Graph and Other Networks*. MIT Press, Cambridge, MA, USA, 1983.
- 15 László Lovász. *Large networks and graph limits*, volume 60. American Mathematical Soc., 2012.
- 16 Gary L Miller. Finding small simple cycle separators for 2-connected planar graphs. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 376–382, 1984. doi:10.1145/800057.808703.
- 17 Marcus Schaefer. The graph crossing number and its variants: A survey. *The electronic journal of combinatorics*, pages DS21–Apr, 2012.

yFiles – From Data to Meaningful Visualizations

Evmorfia Argyriou  

yWorks GmbH, Tübingen, Germany

Benjamin Niedermann  

yWorks GmbH, Tübingen, Germany

Abstract

Graph visualizations help with complex data analysis but often require expert knowledge to apply and configure advanced algorithms. yFiles, a diagramming SDK, bridges this gap by enabling developers to create interactive visualizations easily. This work demonstrates how yFiles helps transform raw data into accessible graph visualizations.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases diagramming SDK, layout algorithms, interactive graph visualization

Digital Object Identifier 10.4230/LIPIcs.GD.2024.55

Category Software Abstract

1 yFiles – From Data to Graph Visualizations

Graph visualizations help users gain insights from complex data. Depending on the structure of the data, different methods can be applied to place the graph nodes and route the edges. For instance, hierarchical data is well represented with layered drawings, whereas social network data often benefits from force-directed layouts. Research provides powerful algorithms for drawing graphs [1, 4], but their application generally requires expert knowledge. Meanwhile, many users outside the field of graph drawing seek simple, effective tools to visualize their data without the need to understand the technical details. This work shows how yFiles¹, a diagramming SDK, bridges the gap between sophisticated algorithms and the practical needs of users. We demonstrate how this library can transform raw data into interactive graph visualizations accessible to anyone without needing to “reinvent the wheel”.

Challenge 1: From Data to Graph Structures. In typical use cases, users of the diagramming SDK aim to answer questions based on structured data. The first step involves understanding the data structure and identifying specific characteristics. yFiles offers a range of graph analysis algorithms, including k-means or k-core clustering, and centrality algorithms, to help users understand the graph’s structure. This step is crucial for selecting the most appropriate layout algorithm; see Fig. 1 for examples.

Challenge 2: From Graph Structures to Layouts. Once the data structure is understood, users select a specific layout that best fits the graph structure, for instance, a layered [5] or force-based layout [2]. However, most users are not experts in graph drawing and prefer to configure layouts using visual parameters rather than technical ones, such as force strengths. Visual parameters might include node distances, node sizes, edge thickness/bends, labels, or more sophisticated constraints such as group nodes, node ordering, layer assignment, or port constraints. yFiles maps these visual settings onto appropriate algorithms and their parameters, making advanced layout techniques accessible without requiring users to understand the underlying details. For example, the organic layout consists of several force-based algorithms that are applied depending on the graph structure [2, 3].

¹ <https://yfiles.com>



© Evmorfia Argyriou and Benjamin Niedermann;

licensed under Creative Commons License CC-BY 4.0

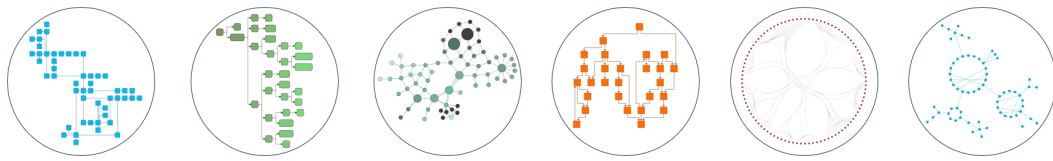
32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 55; pp. 55:1–55:3

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** Examples of layouts offered by yFiles. From left to right: orthogonal, tree, organic (force-based), hierarchic (layered drawing), single cycle, circular layout.

Adapting algorithms from research to complex use cases, regardless of graph size or constraints, is a significant challenge. Among others, yFiles addresses this by allowing the combination of different layout algorithms. For instance, a graph with group nodes might use a force-based method for top-level nodes and a layered layout for the groups' contents. yFiles provides a powerful pipeline mechanism for assembling various layouts using both built-in and user-defined algorithms. The SDK also supports customizing layout algorithms with a wide range of configuration options. Additionally, it offers features such as automatic substructure detection and enhanced visualization of elements such as cycles, stars, and parallel structures. Moreover, yFiles strongly emphasizes generic and common features, such as labeling, group nodes, edge grouping, restricted port locations, and table-like structures, all of which are supported out of the box.

Challenge 3: From Layouts to Network Visualizations. Beyond arranging graph elements, their visual representation – including colors, shapes, and edge thickness – is crucial as these visual aspects encode additional information. yFiles offers a variety of predefined styles for graph elements and also allows users to create custom styles. The SDK integrates seamlessly with various rendering engines, such as SVG or WebGL, depending on the platform.

All layout algorithms can be applied to graphs with animations, helping users understand transitions between different visualizations. This is facilitated by a generic animation framework and morphing algorithms.

Challenge 4: Making the Graph Interactive and Dynamic. Interactivity allows users to explore and edit the graph structure and the underlying data. Common interactions include navigation (zooming and panning), basic modifications, and more complex actions such as collapsing/expanding group nodes and filtering elements. These require layout algorithms that can adapt to changes, which yFiles supports through features such as fixed node subsets and incremental element placement.

Conclusion

yFiles bridges the gap between formal algorithms and the practical needs of users. By providing advanced layout algorithms and integration with existing frameworks, it enables users to create not only visually appealing but also meaningful visualizations of their data.

References

- 1 Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall PTR, USA, 1st edition, 1998.
- 2 Thomas M. J. Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991. doi:10.1002/SPE.4380211102.

- 3 Tomihisa Kamada and Satoru Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15, 1989. doi:10.1016/0020-0190(89)90102-6.
- 4 Michael Kaufmann and Dorothea Wagner, editors. *Drawing graphs: methods and models*. Springer-Verlag, Berlin, Heidelberg, 2001.
- 5 Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiro Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11:109–125, 1981. URL: <https://api.semanticscholar.org/CorpusID:8367756>, doi:10.1109/TSMC.1981.4308636.


The Eclipse Layout Kernel

Maximilian Kasperowski ✉ 

Department of Computer Science, Kiel University, Germany

Sören Domrös ✉ 

Department of Computer Science, Kiel University, Germany

Reinhard von Hanxleden ✉ 

Department of Computer Science, Kiel University, Germany

Abstract

The Eclipse Layout Kernel (ELK) is an open-source framework¹ written in Java, which is transpiled to the JavaScript library `elkjs`². ELK provides extensible and modular algorithms, visibility for diagramming research, and has an active community. The ELK project is both a validation platform for graph drawing algorithm researchers, and a freely available library put in production use to provide automatic layout for academic and commercial applications.

The report by Domrös et al. [3] presents an overview of the available algorithms, the development history, related publications, as well as lessons learned from developing the open-source framework. ELK welcomes new users as well as new contributors.

2012 ACM Subject Classification Software and its engineering → Software libraries and repositories

Keywords and phrases Automatic Layout, Layered Layout, Layout Library

Digital Object Identifier 10.4230/LIPIcs.GD.2024.56

Category Software Abstract

1 ELK's Infrastructure and Applications

ELK provides infrastructure to develop algorithms for arbitrary types of graphs including digraphs, hypergraphs, and compound graphs that may be laid out bottom-up or top-down [6] and allows using different algorithms in different subgraphs. ELK's phase and processor infrastructure supports pre- and post-processors that can be modularly inserted into any algorithm. This allows solving domain-specific layout questions as well as comparing layout strategies without reimplementing whole algorithms.

ELK contains highly configurable³ algorithms for layered layouts, rectangle packing, tree drawing, force and stress layouts, radial layouts, and packing of disconnected components. The flagship algorithm is ELK LAYERED based on the Sugiyama algorithm [11]. It can be configured with over 140 layout options to control details such as spacing for edges, nodes, disconnected components and ports, as well as what strategies for different phases of the layout algorithm. These include model order, node and port constraints, node size constraints, compaction, edge wrapping, label placement, self-loop arrangement, and layout direction.

Notable layout strategies that have been implemented for ELK LAYERED based on published research include Forster constraint resolving during barycenter crossing minimization [4], Brandes and Köpf node placement [1], edge bundling for dataflow diagrams [9] and size-aware and port-aware horizontal node coordinate assignment by Rüegg et al. [8], and port constraints in dataflow diagrams by Spönemann et al. [10]. Moreover, Jabrayilov et al. and Rüegg et al. have worked on the graph layering problem for general directed graphs [5, 7].

¹ <https://eclipse.dev/elk/>

² <https://www.npmjs.com/package/elkjs>

³ <https://eclipse.dev/elk/reference/options.html>



ELK and elkjs are integrated into popular diagramming tools such as mermaid, GLSP, reactflow, Lingua Franca, Sirius, Sprotty, and KIELER, which provide real world layout problems to solve and makes graph drawing research results visible and usable. Particularly elkjs has become very popular⁴ in recent years, as it lets users embed automatic graph visualizations within web applications. elkjs has received about 1700 “stars” on GitHub and has currently over 800 000 weekly downloads.

Much of ELK’s value stems from community interaction. This interaction provides us with valuable insights on how real applications utilize graph drawing. GitHub or gitter⁵ can be used to ask layout questions or report problems. The elklive tool⁶ allows sharing graph configurations in the online editor via a simple link. One key insight of the community interaction is that users do not want complicated solutions but rather control over the layouts.

2 Future Work

Future work focuses on laying the groundwork for visualizing large hierarchical models and exploring how model order [2] can be utilized by layout algorithms.

Top-down layout is an approach to draw nested graphs that serves as an alternative to the bottom-up approach [6]. By drawing nested graphs starting at the root instead of at the leaves we obtain better high-level overviews, especially for large nested graphs.

Model order aims to bring secondary notation and with it intention from a textual model into a diagram. This can determine the whole topology of a layered graph by model order.

References

- 1 U. Brandes and B. Köpf. Fast and simple horizontal coordinate assignment. In *Proc. GD '01*, 2002. doi:10.1007/3-540-45848-4.
- 2 S. Domrös, M. Riepe, and R. von Hanxleden. Model order in Sugiyama layouts. In *Proc. VISIGRAPP 2023 - Volume 3: IVAPP*, 2023. doi:10.5220/0011656700003417.
- 3 S. Domrös, R. von Hanxleden, M. Spönemann, U. Rüegg, and C. D. Schulze. The Eclipse Layout Kernel, 2023. doi:10.48550/arXiv.2311.00533.
- 4 M. Forster. A fast and simple heuristic for constrained two-level crossing reduction. In *Proc. GD '04*, 2005. doi:10.1007/978-3-540-31843-9_22.
- 5 A. Jabrayilov, S. Mallach, P. Mutzel, U. Rüegg, and R. von Hanxleden. Compact layered drawings of general directed graphs. In *Proc. GD '16*, 2016. doi:10.1007/978-3-319-50106-2_16.
- 6 M. Kasperowski and R. von Hanxleden. Top-down drawings of compound graphs, December 2023. doi:10.48550/arXiv.2312.07319.
- 7 U. Rüegg, T. Ehlers, M. Spönemann, and R. von Hanxleden. A generalization of the directed graph layering problem. In *Proc. GD '16*, 2016. doi:10.1007/978-3-319-50106-2_16.
- 8 U. Rüegg, C. D. Schulze, J. J. Carstens, and R. von Hanxleden. Size- and port-aware horizontal node coordinate assignment. In *Proc. GD '15*, 2015. doi:10.1007/978-3-319-27261-0_12.
- 9 U. Rüegg, C. D. Schulze, C. Sprung, N. Wechselberg, and R. von Hanxleden. Edge bundling for dataflow diagrams. Poster at GD '16, 2016.
- 10 M. Spönemann, H. Fuhrmann, R. von Hanxleden, and P. Mutzel. Port constraints in hierarchical layout of data flow diagrams. In *Proc. GD '09*, 2010. doi:10.1007/978-3-642-11805-0.
- 11 K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Trans. Syst. Man. Cybern.*, 1981. doi:10.1109/TSMC.1981.4308636.

⁴ <https://npmtrends.com/elkjs>

⁵ https://app.gitter.im/#/room/#eclipse_elk:gitter.im

⁶ <https://rtsys.informatik.uni-kiel.de/elklive/index.html>

Immersive Analytics of Graphs in Virtual Reality with GAV-VR

Stefan P. Feyer  

University of Konstanz, Germany

Wilhelm Kerle-Malcharek  

University of Konstanz, Germany

Ying Zhang  

University of Konstanz, Germany

Falk Schreiber  

University of Konstanz, Germany

Monash University, Melbourne, Australia

Karsten Klein¹  

University of Konstanz, Germany

Abstract

The design space for interactive graph visualisation in immersive environments creates opportunities to improve on established solutions in traditional desktop settings. Exploiting this potential requires careful analysis of achievable benefits, required tradeoffs, and disadvantages for particular designs and use-cases. GAV-VR is a modular and user-extensible framework for graph visualisation and analysis in Virtual Reality (VR). It provides the platform to easily create interactive graph visualisations, facilitating both applied graph analysis and evaluation of approaches and methods for visualisation of and interaction with graphs in VR.

2012 ACM Subject Classification Human-centered computing → Visualization systems and tools; Human-centered computing → Visual analytics; Human-centered computing → Virtual reality

Keywords and phrases Networks, Immersive Analytics, Software

Digital Object Identifier 10.4230/LIPIcs.GD.2024.57

Category Software Abstract

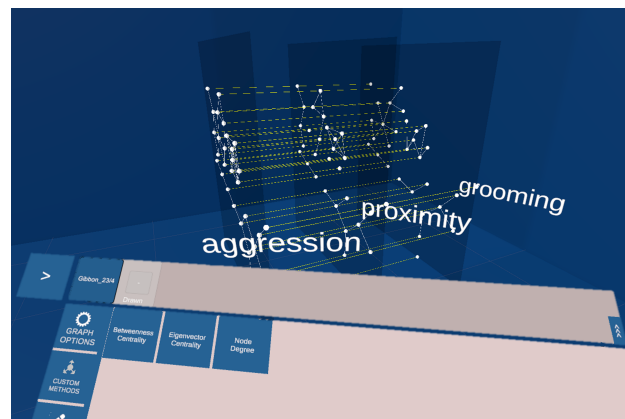
Funding We acknowledge funding by Deutsche Forschungsgemeinschaft (DFG), under Germany's Excellence Strategy – EXC 2117 – 422037984 and DFG project ID 251654672 – TRR 161.

1 Introduction

Interactive 3D graph visualisation, especially in VR, has shown potential in enhancing graph structure understanding and analysis across various applications [3]. The large design space for graph representation and interaction in VR offers significant opportunities for improving analysis processes. Features such as stereoscopic 3D visualisation, user tracking, and direct interaction need to be properly investigated and integrated into efficient designs in order to make use of these opportunities. The effort to implement a VR-based graph visualisation from scratch, but also to re-assemble basic functionalities for each new project, is a large obstacle for quick prototyping and design comparison and distracts researchers and practitioners from their main focus, e.g. graph analysis or evaluation of methods [1].

¹ Corresponding author





■ **Figure 1** Screenshot of a multilayer network visualisation in GAV-VR, with parts of the user interface in the foreground.

We demonstrate *Graph Analysis and Visualisation in VR* (GAV-VR) [2], a framework designed for interactive graph visualisation and analysis in VR. The framework facilitates prototyping, method comparison, and interactive graph analysis. GAV-VR’s modular architecture supports various VR headsets, data formats, and visualisation methods, enabling easy integration of new functionalities.

2 GAV-VR Structure and Use

GAV-VR serves two types of users: *analysts* and *contributors*. Analysts use the system to investigate graphs with a pre-implemented build that requires no coding but includes ready-to-use features for graph analysis in VR. Contributors enhance GAV-VR by developing new features within a modular Unity C# environment. They utilise abstract classes and pre-defined routines without needing a deep understanding of the entire framework.

The architecture of GAV-VR includes a *core* that handles core functionalities and manages modular features, and *modules* that represent the custom content added by contributors. The core provides user interfaces and controls for interaction with graphs, edges, and vertices. Modules can be script-based, implementing new methods and features through C# classes, or file-based, altering object representations through prefab files.

Script-based modules extend functionalities such as file parsing and movement modes, using abstract classes to integrate new features. File-based modules modify visual aspects such as vertex and edge representations, which are automatically integrated into the UI. To integrate modules, contributors either add C# files for script-based modules or place custom prefab files for file-based modules.

In GAV-VR, graphs are visualised as node-link diagrams. The representation of vertices and edges can be customised. The UI is adjustable and supports raycast point-and-click interactions. Users can manipulate graphs directly, e.g. through rotation and translation. Data handling and analysis are managed by the core, which ensures data integrity and validates file formats. Graphs are saved in a simple, human-readable “gavvr” format that captures essential attributes. Performance varies with interactive and non-interactive modes, supporting up to 2,000 objects interactively and 20,000 objects non-interactively, depending on hardware capabilities.

3 Conclusion

The GAV-VR framework offers a robust solution for interactive graph visualisation and analysis within VR environments. By supporting both *analysts* and *contributors*, GAV-VR caters to diverse user needs. The architecture ensures seamless integration of new features while maintaining core functionality. With its support for both script-based and file-based modules, GAV-VR allows extensive customisation, from adding new graph analysis methods to modifying visual representations. Licenced under the GNU Affero General Public License V3, GAV-VR promotes open, non-commercial use, fostering innovation and collaboration in the field of graph visualisation.

References

- 1 Stefan P. Feyer, Bruno Pinaud, Stephen Kobourov, Nicolas Brich, Michael Krone, Andreas Kerren, Michael Behrlich, Falk Schreiber, and Karsten Klein. 2D, 2.5D, or 3D? an exploratory study on multilayer network visualisations in virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):469–479, 2024. doi:10.1109/TVCG.2023.3327402.
- 2 Wilhelm Kerle-Malcharek, Stefan P. Feyer, Falk Schreiber, and Karsten Klein. GAV-VR: an extensible framework for graph analysis and visualisation in virtual reality. In Jean-Marie Normand, editor, *ICAT-EGVE 2023 - International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments*, pages 131–139, Eindhoven, 2023. The Eurographics Association. doi:10.2312/egve.20231321.
- 3 Matthias Kraus, Johannes Fuchs, Björn Sommer, Karsten Klein, Ulrich Engelke, Daniel Keim, and Falk Schreiber. Immersive analytics with abstract 3D visualizations: A survey. In *Computer Graphics Forum*, volume 41(1), pages 201–229. Wiley Online Library, 2022. doi:10.1111/CGF.14430.

Graph Harvester

Julius Deynet ✉

Universität Würzburg, Germany

Tim Hegemann  

Universität Würzburg, Germany

Sebastian Kempf  

Universität Würzburg, Germany

Alexander Wolff  

Universität Würzburg, Germany

Abstract

We present *Graph Harvester*, a website for extracting graphs from illustrations in scientific papers. For every graph that has been extracted, Graph Harvester queries the graph database *House of Graphs*. If the graph is not already present there, the user can upload the graph into the database, possibly after modifying it, and with a reference to the paper that contains the drawing of the graph.

2012 ACM Subject Classification Applied computing → Graphics recognition and interpretation; Mathematics of computing → Graph theory; Information systems → Web applications

Keywords and phrases House of Graphs, Graph recognition, Information extraction

Digital Object Identifier 10.4230/LIPIcs.GD.2024.58

Category Software Abstract

Supplementary Material

Software (Source Code): https://github.com/JuliusDeynet/graph_harvester [5]

archived at `swh:1:dir:f390ce9e8201cb8d2c97848e8fb5170173dcb82b`

Funding *Tim Hegemann*: Supported by BMBF grant 01IS22012C.

1 Graph Harvester

In graph theory, certain graphs are pivotal in numerous publications, serving as key tools in proving or refuting theorems. Hence, having a collection of significant graphs previously used in publications is highly beneficial. This is the idea behind the website *House of Graphs* (HoG) set up by Coolsaet et al. [3, 4]. It explicitly does not aim at making all possible graphs (of a certain size) available, but only “interesting” ones. On the site, one can search graphs using many criteria, draw graphs, and upload new graphs. Still, from a graph drawing perspective, the HoG database is far from being complete. Out of 293 potentially interesting graphs (with at least seven vertices and maximum vertex degree of at least 3) that we extracted from figures in papers presented at GD 2023, the HoG database contained only 57.

Unfortunately, extending the HoG collection is time-consuming. Currently, graphs must be uploaded in one of two formats: either as an adjacency matrix or in graph6 string representation. HoG users, however, may have their graphs represented differently, e.g., as a drawing. Converting a drawing to the required formats by hand is prone to errors for smaller graphs, and practically impossible for larger ones.

To address this problem, we present *Graph Harvester* [6], a website that extracts graphs from drawings of graphs in PDF files. Whereas there has been work on extracting graphs from bitmap images [1], we focus on the simpler problem of extracting graphs from vector data, which is the primary format used in publications nowadays. Graph Harvester works as follows. A user uploads a publication as a PDF file. Then, the website displays each detected



© Julius Deynet, Tim Hegemann, Sebastian Kempf, and Alexander Wolff;
licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 58; pp. 58:1–58:3

Leibniz International Proceedings in Informatics



LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

figure in the file next to a drawing of the extracted graph for review. Furthermore, the website computes a representation in the format accepted by HoG and checks whether the graph is already present in the HoG database. To this end, the following steps are performed.

The initial extraction process is based on modules of the KIETA [7] pipeline. Graphical and textual elements within the PDF file are extracted using PyMuPDF, a Python library for data extraction, analysis, conversion, and manipulation of PDF documents. The graphical elements are clustered to find a cohesive group that contains a figure in vector format (SVG). Then, the corresponding figure caption is determined using a keyword-based approach.

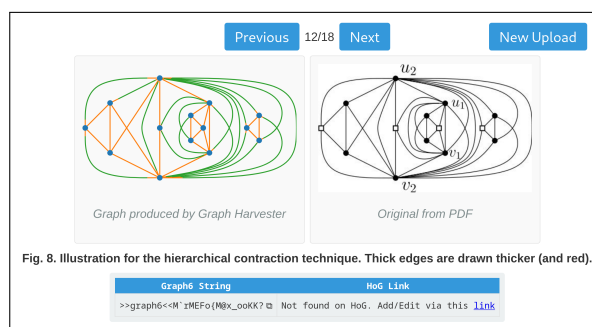
Next, the graphs are extracted from the figures. For this step, four geometric objects are of importance: circles and rectangles, which represent potential vertices; and line segments and curves, which represent potential edges. We call two line segments or Béziers curves *connected* if they share an endpoint (i.e., their endpoints are closer than a given threshold).

Because the PDF format does not support circle objects natively, these are typically represented by cubic Béziers curves. Therefore, any connected sequence of three or more curves forming a cycle, with endpoints at similar distances from their joint barycenter, is considered a circle. The latter condition is used to filter out elliptic shapes, which typically do not represent vertices (but are rather used for highlighting).

Next, the potential vertices are filtered for duplicates and by size. Curves that form cycles but have not been classified as vertices are retained for further consideration, as they may represent edges. Rectangles whose four corners coincide with potential vertices are split into four line segments. Then, edges are handled. Any sequence of connected line segments or Béziers curves is considered an edge candidate if both endpoints match vertex candidates. (Otherwise, they usually represent axes or other annotations.) Edge candidates are subdivided if they contain vertex candidates in their relative interior.

Finally, the graph of all vertex and edge candidates is built and split into connected components. The adjacency matrix is set up separately for each component. Graphs (or components) with fewer than seven vertices or a maximum vertex degree of less than 3 are filtered out, as they are not considered interesting. The remaining matrices are converted to graph6 format and sent to the HoG interface to check whether the corresponding graphs are already present in the HoG database. If they are, their HoG identifiers are reported.

The graph6 representations, HoG identifiers, figure captions, and geometric objects used to represent the graphs are displayed by the Graph Harvester website. The latter are drawn as they were found in the paper. Additionally, a picture of the figure extracted from the original PDF is provided to allow the user to validate Graph Harvester's extraction. If a graph already exists in the HoG database, a link to the HoG entry is shown. Otherwise, users can choose to add the graph to the HoG database, possibly after modifying it with the HoG graph editor. Figure 1 shows a screenshot of the Graph Harvester website.



■ **Figure 1** Output of Graph Harvester for Figure 8(b) of a paper of Bekos et al. [2].

References

- 1 Christopher Auer, Christian Bachmaier, Franz Brandenburg, Andreas Gleißner, and Josef Reislhuber. Optical graph recognition. *J. Graph Algorithms Appl.*, 17(4):541–565, 2013. doi:10.7155/jgaa.00303.
- 2 Michael A. Bekos, Walter Didimo, Giuseppe Liotta, Saeed Mehrabi, and Fabrizio Montecchiani. On RAC drawings of 1-planar graphs. *Theoret. Comput. Sci.*, 689:48–57, 2017. doi:10.1016/j.tcs.2017.05.039.
- 3 Gunnar Brinkmann, Kris Coolsaet, Gauvain Devillez, Jan Goedgebeur, and Hadrien Mélot. House of Graphs. Website, accessed on Sep. 6, 2024. URL: <https://houseofgraphs.org>.
- 4 Kris Coolsaet, Sven D’hondt, and Jan Goedgebeur. House of Graphs 2.0: A database of interesting graphs and more. *Discret. Appl. Math.*, 325:97–107, 2023. doi:10.1016/J.DAM.2022.10.013.
- 5 Julius Deynet, Tim Hegemann, Sebastian Kempf, and Alexander Wolff. Graph Harvester. Software, swhId: `swh:1:dir:f390ce9e8201cb8d2c97848e8fb5170173dcb82b` (visited on 2024-10-14). URL: https://github.com/JuliusDeynet/graph_harvester.
- 6 Julius Deynet, Tim Hegemann, Sebastian Kempf, and Alexander Wolff. Graph Harvester. Website, accessed on Sep. 6, 2024. URL: <https://go.uniwue.de/graph-harvester>.
- 7 Sebastian Kempf, Markus Krug, and Frank Puppe. KIETA: Key-insight extraction from scientific tables. *Appl. Intell.*, 53:9512–9530, 2023. doi:10.1007/s10489-022-03957-8.

CentralityViz: Comprehending Node-Centrality in Large Networks

Garima Jindal  

International Institute of Information Technology Hyderabad (IIITH), India

Kamalakar Karlapalem  

International Institute of Information Technology Hyderabad (IIITH), India

Abstract

CentralityVis is a software tool designed for visualizing large graphs using two community-centric methods: *spiral visualization* and *linear visualization*. Both visualizations are highly scalable, capable of handling networks with hundreds of thousands of nodes and edges. The tool leverages community detection algorithms to group nodes into communities and then orders the nodes of community on centrality in descending order, arranging them in either a spiral or linear layout. CentralityVis provides clear insights into both node and community properties, facilitating the analysis of complex networks. Each visualization method has its strengths: *spiral visualization* is intuitive and resembles traditional node-link diagrams, while *linear visualization* facilitates easy comparison of communities and offers greater scalability in terms of the number of communities that can be represented. To minimize visual clutter, edges are drawn only when needed, ensuring that even large graphs remain clear and comprehensible. CentralityVis is a powerful tool for understanding complex networks, emphasizing both individual nodes and the communities to which they belong.

2012 ACM Subject Classification Human-centered computing → Visualization systems and tools; Human-centered computing → Visual analytics

Keywords and phrases Visual Analytics, Graph Drawing, Community Detection, Node Centrality

Digital Object Identifier 10.4230/LIPIcs.GD.2024.59

Category Software Abstract

Related Version *Full Version*: <https://ieeexplore.ieee.org/abstract/document/10360896>

Supplementary Material

Software (Source code): <https://github.com/Garima17/SpiralVisualization>

archived at `swh:1:dir:0727514f98092fcf77f85c7326cc28d1e17882a9`

Software (source code): <https://github.com/Garima17/Linear-visualization/>

archived at `swh:1:dir:0887f83f27e634a2cac344236f467c98ebb16283`

1 Introduction

CentralityVis is a powerful community-centric network visualization tool specifically designed for undirected, unweighted, and static networks. It has three primary goals: (I) drawing large networks in a compact and intuitive format, (II) visualizing node centrality within network communities, and (III) visualizing both node and community properties.

CentralityVis helps users comprehend large networks through two interactive visual solutions: (I) Spiral Visualization, published in VIS 2023 [5], can represent networks with up to 10,000 nodes, and (II) Linear Visualization, capable of visualizing up to 50,000 nodes. The tool offers flexibility in selecting the centrality measure, which determines the ordering of nodes within each community in descending order. In Spiral Visualization, these ordered nodes are arranged in a spiral layout, with higher centrality nodes positioned closer to the center, effectively highlighting node rankings within the community. Linear Visualization, on the other hand, organizes nodes in a wrap-around linear format with uniform spacing between



© Garima Jindal and Kamalakar Karlapalem;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 59; pp. 59:1–59:3

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

nodes, facilitating comparisons of node centrality. Additionally, the Linear Visualization aids users in identifying important communities by ranking and arranging them sequentially based on user-selected attributes such as size (number of nodes in a community), edge density (ratio of intra-community connections to total possible connections), or inter-community connections (number of connections to other communities).

The source code for the Spiral visualization and Linear Visualization is available at <https://github.com/Garima17/SpiralVisualization> and <https://github.com/Garima17/Linear-visualization/>.

2 Key Features

Both spiral visualization and linear visualization makes it easier to identify, interpret, and comprehend different network properties, such as (1) identifying the *number of communities* within the network, (2) visualizing and comparing the *sizes* of different communities, (3) visualizing and comparing the *edge-density of communities* (i.e., the ratio of actual links within a community to the total number of possible links), (4) identifying important or *central nodes* [1, 6] within communities, (5) understanding *centrality distribution* within communities, (6) comprehending *connections between communities*, and (7) comprehending *node connections*. The following two videos briefly demonstrate Spiral Visualization <https://youtu.be/cvLdXATHIXY> and Linear Visualization <https://youtu.be/R0mKgpJF3Kw>.

Interactive Elements. CentralityViz supports zooming, panning, and tooltips for detailed exploration, as well as an interlinked view to support the comprehension of different features. Users can filter data for selective exploration. The “*Find Node*” option allows users to search for a node in the visualization based on node ID. Users can also select different centrality measures (i.e., degree [6], closeness [8], betweenness [2], or eigen centrality [10]) based on which they want to visualize the data. The CentralityViz dashboard also provide the option to rank communities based on size, edge-density and number of community connections (as demonstrated in <https://youtu.be/R0mKgpJF3Kw>).

3 Applications

Visualizing large networks [15] is challenging due to visual clutter. In large networks, communities [3] emerge as subsets of nodes that are more connected within themselves compared to the rest of the network. Community detection and analysis of central nodes [6, 2, 10, 8, 1] in communities has diverse applications across multiple fields. In *criminology* [9, 14, 12], it is used to identify criminal or terrorist groups, understand their networks, and identify key players or masterminds behind an attack. In *epidemiology* [11, 4, 13], community analysis can facilitate tracking the spread of diseases, while node centrality analysis within a community can help identify key individuals who may contribute to widespread transmission. In *smart advertising and targeted marketing* [7], identifying influential communities and key individuals can optimize the impact of marketing campaigns. Therefore, detecting and analyzing communities and central nodes is a crucial problem in data science. To the best of our knowledge, no other graph drawing technique effectively visualizes node centrality within communities in large networks [5].

Our software provides substantial value to the graph drawing community by introducing an innovative approach to visualizing and analyzing large networks. Both, spiral and linear visualization are intuitive and can be mastered by users in just 15-20 minutes of training. Our

tool provides insights into both *community and node-centric* properties. It provides a global network overview while allowing users to explore specific details. It enables comprehensive interpretations of node centrality, connectivity patterns within and between communities, and supports informed decision-making based on these insights.

References

- 1 Stephen P Borgatti and Martin G Everett. A graph-theoretic perspective on centrality. *Social networks*, 28(4):466–484, 2006. doi:10.1016/J.SOCNET.2005.11.005.
- 2 Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of mathematical sociology*, 25(2):163–177, 2001.
- 3 Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics reports*, 659:1–44, 2016.
- 4 Nandinee Haq and Z Jane Wang. Community detection from genomic datasets across human cancers. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1147–1150. IEEE, 2016.
- 5 Garima Jindal and Kamalakar Karlapalem. A simple yet useful spiral visualization of large graphs. In *2023 IEEE Visualization and Visual Analytics (VIS)*, pages 171–175. IEEE, 2023. doi:10.1109/VIS54172.2023.00043.
- 6 Warih Maharani, Alfian Akbar Gozali, et al. Degree centrality and eigenvector centrality in twitter. In *2014 8th international conference on telecommunication systems services and applications (TSSA)*, pages 1–5. IEEE, 2014.
- 7 Mohammad Javad Mosadegh and Mehdi Behboudi. Using social network paradigm for developing a conceptual framework in crm. *Australian Journal of Business and Management Research*, 1(4):63, 2011.
- 8 Kazuya Okamoto, Wei Chen, and Xiang-Yang Li. Ranking of closeness centrality for large-scale social networks. In *International workshop on frontiers in algorithmics*, pages 186–195. Springer, 2008. doi:10.1007/978-3-540-69311-6_21.
- 9 Carlos André Reis Pinheiro. Community detection to identify fraud events in telecommunications networks. *SAS SUGI proceedings: customer intelligence*, 2012.
- 10 Britta Ruhnau. Eigenvector-centrality – A node-centrality? *Social networks*, 22(4):357–365, 2000.
- 11 Marcel Salathé and James H Jones. Dynamics and control of diseases in networks with community structure. *PLoS computational biology*, 6(4):e1000736, 2010. doi:10.1371/JOURNAL.PCBI.1000736.
- 12 Hamed Sarvari, Ehab Abozinadah, Alex Mbaziira, and Damon McCoy. Constructing and analyzing criminal networks. In *2014 IEEE security and privacy workshops*, pages 84–91. IEEE, 2014.
- 13 Fumihiko Taya, Joshua de Souza, Nitish V Thakor, and Anastasios Bezerianos. Comparison method for community detection on brain networks from neuroimaging data. *Applied Network Science*, 1:1–20, 2016.
- 14 Todd Waskiewicz. Friend of a friend influence in terrorist social networks. In *Proceedings on the international conference on artificial intelligence (ICAI)*, page 1. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2012.
- 15 Vahan Yoghoudjian, Daniel Archambault, Stephan Diehl, Tim Dwyer, Karsten Klein, Helen C Purchase, and Hsiang-Yun Wu. Exploring the limits of complexity: A survey of empirical studies on graph visualisation. *Visual Informatics*, 2(4):264–282, 2018. doi:10.1016/J.VISINF.2018.12.006.

NodeXL – A few Clicks to Network Insights

Harald Meier   

Social Media Research Foundation, CA, USA

Arber Ceni 

Social Media Research Foundation, CA, USA

Abstract

Network analysis and visualization are crucial for unraveling complex relationships across diverse fields, from social networks to biological systems. NodeXL is a versatile network analysis tool that supports a wide range of network data types and provides seamless access to various social media platforms.

2012 ACM Subject Classification Human-centered computing → Visualization systems and tools; Human-centered computing → Visual analytics; Human-centered computing → Graph drawings

Keywords and phrases Network Analysis, NodeXL, Social Networks

Digital Object Identifier 10.4230/LIPIcs.GD.2024.60

Category Software Abstract

Introduction

Network analysis and visualization are essential for understanding complex relationships across various fields, from social networks to biological systems. As data complexity increases, the need for accessible tools that enable researchers, journalists, and market specialists to derive meaningful insights grows. NodeXL addresses this need by providing a user-friendly platform integrated with Microsoft Excel, allowing users to perform network analysis without programming expertise.

NodeXL is widely used in academic institutions, research labs, newsrooms, and marketing departments. Its accessibility and versatility have made it a popular choice for analyzing community structures in online conversations, particularly on social media platforms. This abstract highlights NodeXL's purpose, key features, and its applications.

Purpose

NodeXL aims to democratize network analysis, making it accessible to users of all skill levels. Whether a novice or an advanced researcher, NodeXL offers comprehensive tools for exploring, visualizing, and analyzing network data. The integration with Microsoft Excel leverages a familiar interface, reducing the learning curve and allowing users to focus on analysis rather than software mechanics.

Main Features

NodeXL offers two versions: NodeXL Basic and NodeXL Pro. NodeXL Basic is free and provides essential tools for network visualization and analysis, including:

- Importing network data from various sources.
- Calculating basic network metrics.
- Group Clustering Algorithms: Clustering to detect communities within networks.
- Visualizing simple networks with customizable layouts.



© Harald Meier and Arber Ceni;

licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

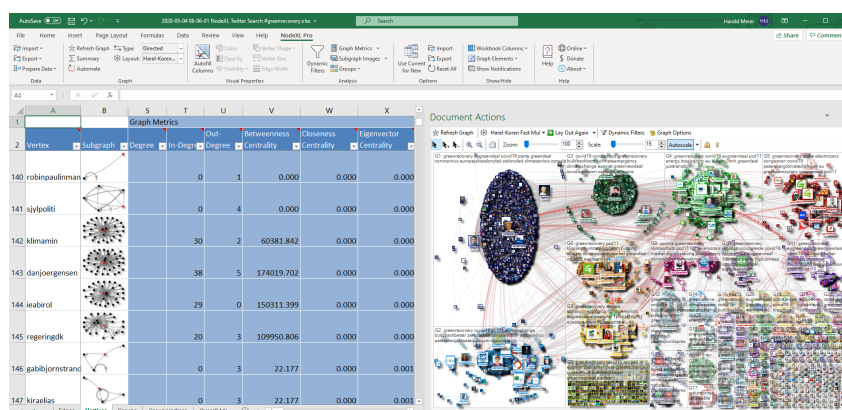
Editors: Stefan Felsner and Karsten Klein; Article No. 60; pp. 60:1–60:2

Leibniz International Proceedings in Informatics



LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

60:2 NodeXL – A few Clicks to Network Insights



■ **Figure 1** NodeXL screenshot.

- NodeXL Pro is a paid version with advanced features for in-depth analysis:
- **Social Media Data Importers:** Seamless integration with platforms like X (formerly Twitter), YouTube, Instagram, Reddit, Wikipedia, and Flickr.
 - **Graph Metrics Calculation:** Includes advanced metrics such as betweenness and eigenvector centrality.
 - **Content and Sentiment Analysis:** Tools for analyzing text and sentiment in social media data.
 - **Time Series Analysis:** Tracks how network structures evolve over time.
 - **Task Automation:** Automates complex analysis tasks, saving time and streamlining workflows.
 - **Export to NodeXL Pro INSIGHTS:** Explore network data sets in a Microsoft Power BI ¹ report template with a huge variety of data visualizations – from simple tables to hashtag clouds, from image grids to scatter plots.

Applicability


NodeXL's versatility makes it suitable for a wide range of applications. In academia, it is used to study social networks and collaboration patterns. In journalism, it visualizes information spread on social media and identifies key influencers. In marketing, it analyzes brand networks and tracks message diffusion. Its ability to handle diverse data types and social media integration makes it particularly valuable for analyzing online conversations, revealing how information flows and communities form.

Conclusion

NodeXL is a powerful yet accessible tool for network analysis and visualization. Its integration with Microsoft Excel, combined with robust features, makes it suitable for users ranging from novices to experienced researchers. NodeXL empowers users across disciplines to explore the complex relationships that define our interconnected world, advancing network science and enabling new discoveries.

¹ <https://powerbi.microsoft.com/>

Knowledge Graph Builder – Constructing a Graph from Arbitrary Text Using an LLM

Andreas Benno Kollegger ✉ 🏠 

Neo4j Inc, Cambridge, UK

Alexander Erdl ✉

Neo4j Inc, Munich, DE

Michael Hunger ✉

Neo4j Inc, Dresden, DE

Abstract

Knowledge graphs improve many information retrieval tasks over structured and unstructured data. However, knowledge graph construction can be challenging even for domain experts. The Knowledge Graph Builder is an application incorporating advanced techniques for deriving a knowledge graph from unstructured data using an LLM.

2012 ACM Subject Classification Information systems → Network data models

Keywords and phrases Knowledge Graph, Lexical Graph

Digital Object Identifier 10.4230/LIPIcs.GD.2024.61

Category Software Abstract

Supplementary Material *Software (Source Code)*: <https://github.com/neo4j-labs/llm-graph-builder>

Acknowledgements We want to thank the engineering team at Neo4j for their pioneering work on knowledge graph construction using an LLM.

1 Knowledge graph builder – Summary

A knowledge graph is a self-descriptive data structure that represents real-world information using the elements of a graph by storing values on both nodes and relationships. Deriving a knowledge graph from arbitrary text produces a graph with two complementary sections: a lexical graph that preserves the structure of the source text and a domain graph that distills the entities and relationships described by the source text. The Knowledge graph builder is an application which constructs a viable knowledge graph from unstructured data.

2 Knowledge graph construction

Knowledge graph construction follows a 3 phase process.

Content decomposition

1. A document node is created to represent the text source.
2. The text content is split into chunks, which are stored as graph nodes.
3. Chunks are connected to each other and the document node.

Content cross-linking

1. Highly similar chunks are connected with a SIMILAR relationship to form a K-Nearest Neighbors graph.
2. Embeddings are computed, stored as properties on the chunk nodes, and backed by a vector index.



© Andreas Benno Kollegger, Alexander Erdl, and Michael Hunger;
licensed under Creative Commons License CC-BY 4.0

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).

Editors: Stefan Felsner and Karsten Klein; Article No. 61; pp. 61:1–61:2

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

61:2 Knowledge Graph Builder

Entity extraction

1. An LLM is used to process each chunk, to identify entities and their inter-relationships.
2. Entities are stored in the graph and connected to the originating chunks.
3. Entity relationships are stored in the graph, creating a rich network of concepts.

The entity extraction can be unbounded, allowing the LLM to extract and classify any entities and relationships it finds, or it can be constrained to conform with a pre-determined collection of entity and relationship types.

3 Knowledge graph model

The resulting knowledge graph is amenable to multiple GraphRAG access patterns.

3.1 Lexical graph

The lexical graph section preserves the implicit structure of the original source text, including the sequential flow of text and also any hierarchies up to the enclosing document itself.

Chunks are part of a document

`(a:Chunk)-[:PART_OF]->(d:Document)`

Chunks form a linked list of sequential text

`(a:Chunk)-[:NEXT_CHUNK]->(b:Chunk)`

Neighborhoods of similar chunks

`(a:Chunk)-[:SIMILAR]->(b:Chunk)`

3.2 Domain graph

Entities are connected to chunk that mentions them

`(a:Chunk)-[:HAS_ENTITY]->(e1)`

The `(e1)` node matches any entities which have been extracted, with corresponding labels determined by the LLM.

Entities related to each other

`(e1)-[r]->(e2)`

The `[r]` relationship will be determined by the LLM.

4 Conclusion

This brief description of the operation of the Knowledge Graph Builder application is just an introduction to the construction of knowledge graphs. More techniques are available to continue elaborating upon and enriching the graph, and connecting to structured data. The broad topic of GraphRAG then leverages the resulting knowledge graph for use with Generative AI applications.