Report from Dagstuhl Seminar 22062

Computation and Reconfiguration in Low-Dimensional Topological Spaces

Maike Buchin[*1], Anna Lubiw[*2], Arnaud de Mesmay[*3], Saul Schleimer[*4], and Florestan Brunck[†5]

1 **Faculty of Computer Science, Ruhr-Universität Bochum, DE.** maike.buchin@rub.de
2 **Cheriton School of Computer Science, University of Waterloo, CA.** alubiw@waterloo.ca
3 **LIGM, CNRS, Univ. Gustave Eiffel, ESIEE Paris, Marne-la-Vallée, FR.** arnaud.de-mesmay@univ-eiffel.fr
4 **Mathematics Institute, University of Warwick, Coventry, GB.** s.schleimer@warwick.ac.uk
5 **IST Austria – Klosterneuburg, AT.** florestan.brunck@ist.ac.at

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 22062 "Computation and Reconfiguration in Low-Dimensional Topological Spaces". The seminar consisted of a small collection of introductory talks, an open problem session, and then the seminar participants worked in small groups on problems on reconfiguration within the context of objects as diverse as elimination trees, morphings, curves on surfaces, translation surfaces and Delaunay triangulations.

Seminar February 6–11, 2022 – http://www.dagstuhl.de/22062

2012 ACM Subject Classification Human-centered computing → Graph drawings; Theory of computation → Computational geometry; Mathematics of computing → Geometric topology

Keywords and phrases Geometric Topology, Computational Geometry, Graph Drawing, Reconfiguration, Dagstuhl Seminar

Digital Object Identifier 10.4230/DagRep.12.2.17

1 Executive Summary

Maike Buchin (Faculty of Computer Science, Ruhr-Universität Bochum, maike.buchin@rub.de)
Anna Lubiw (Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada, alubiw@waterloo.ca)
Arnaud de Mesmay (LIGM, CNRS, Univ. Gustave Eiffel, ESIEE Paris, F-77454 Marne-la-Vallée, France, arnaud.de-mesmay@univ-eiffel.fr)
Saul Schleimer (Mathematics Institute, University of Warwick, Coventry CV4 7AL, United Kingdom, s.schleimer@warwick.ac.uk)

This seminar was proposed as a followup to the Dagstuhl Seminars 17072: "Applications of Topology to the Analysis of 1-Dimensional Objects" and 19352: "Computation in Low-Dimensional Geometry and Topology". The goal of these seminars was to bring together

[*] Editor / Organizer
[†] Editorial Assistant / Collector

researchers from different communities who are working on low-dimensional topological spaces (curves, embedded graphs, knots, surfaces, three-manifolds), in order to foster collaborations and synergies. Indeed, while the mathematical study of these objects has a rich and old history, the study of their algorithmic properties is still in its infancy, and new questions and problems keep coming from theoretical computer science and more applied fields, yielding a fresh and renewed perspective on computation in topological spaces.

The success of previous seminars demonstrated that research in low-dimensional topology is very active and fruitful, and also that there was a strong demand for a new seminar gathering researchers from the various involved communities, namely geometric topology and knot theory, computational geometry and topology, all the way to graph drawing and trajectory analysis.

For this iteration we placed a particular emphasis on topics related to geometric and topological **reconfiguration**: How can one structure be changed into another? How far apart are two structures? Such questions lie at the heart of various geometric problems such as computing the Fréchet distance as a way to quantify curve similarity, or morphing between two versions of a common graph. In many cases, the combinatorics and the geometry of a reconfiguration space also emerged as important objects of study: examples include associahedra, the flip graphs of triangulations, and the curve complexes in geometric topology.

The seminar started with four overview talks given by researchers in geometric topology, computational geometry, topological dynamics, and graph drawing to motivate and propose open problems that would fit the diverse backgrounds of participants and the specific focus on reconfiguration chosen for this year's workshop. This was followed by an open problem session where we gathered fifteen open problems, some of which were circulated in advance of the meeting. The remainder of the week was spent actively working on solving these problems in small groups.

The Covid pandemic prevented many participants from attending the seminar physically, and the entirety of the seminar took place in a hybrid setting, with most working groups featuring both online and physical participants. In order to coordinate the progress, we used Coauthor, a tool designed for by Erik Demaine (MIT), which greatly facilitated the collaborations, and also allowed participants to have a record of the work when the seminar concluded. We also held two daily progress report meetings, allowing people to share progress and allow people to switch groups. In addition to the traditional hike, a virtual social meeting was held on Gather.town to foster interactions between the online and the physical participants.

We now briefly describe the problems that have been worked on, with a more in-depth survey of the problems and the progress being done being featured farther down in this Dagstuhl Report. Some more open problems that have been proposed but not worked on are also listed at the end of the document.

Two groups worked on questions pertaining to reconfiguring curves in the plane and on surfaces. The group 4.1 investigated problems inspired by nonograms, where one aims at introducing switches at intersections of curves in the plane to remove so-called popular faces. The group 4.5 looked at the reconfiguration graph obtained under the action of local moves on minimal closed (multi-)curves on surfaces, and whether such multi-curves could be realized as the set of geodesics of some hyperbolic metric on the surface.

A different flavor of surfaces was studied by the group 4.4, who investigated how square-tiled surfaces could be transformed under the action of shears of cylinder blocks.

The working group 4.2 studied the longstanding problem of the computational complexity of evaluation the rotation distance between elimination trees in graphs. A different flip graph, namely the one of order-k Delaunay triangulations was the topic of study of group 4.7.

Finally, two groups worked on motion of discrete objects in different contexts. The group 4.3 initiated a generalization of the classical theory of morphings of planar graph when one allows the morph to go through a third dimension. The group 4.6 investigated Turning machines, which is a simple model of molecular robot aiming to fold into specific shapes.

All in all, the seminar fostered a highly collaborative research environment by allowing researchers from very diverse backgrounds to work together on precise problems. While the hybrid setting proved to be a significant challenge, the quality of the equipment at Dagstuhl and the online tools that were used provided a practical way for all the participants to interact and to make progress on problems related to reconfiguration in geometric and topological settings.

2 Table of Contents

3 Overview of Talks

3.1 Playing Puzzles on Square-Tiled Surfaces

Hugo Parlier (University of Luxembourg – Esch-sur-Alzette, Luxembourg, hugo.parlier@uni.lu)

Joint work of Hugo Parlier, Paul Turner, Mario Gutierrez, Reyna Juarez, Mark Bell, Lionel Pournin

This talk will be about a project aiming to illustrate geometry through puzzles. The puzzles are played on (square-tiled) surfaces, and have natural configuration graphs with a geometry of their own. These graphs are reminiscent of combinatorial graphs used in the study of moduli spaces of surfaces which can be visualized in similar ways.

The puzzles were created together with Paul Turner, and brought to life together with Mario Gutierrez and Reyna Juarez. The pictures of moduli spaces were created with Mark Bell and Lionel Pournin.

References

1 Parlier, Hugo; Pournin, Lionel, Modular flip-graphs of one-holed surfaces. Eur. J. Comb. 67, 158-173 (2018).

3.2 Flip Graphs and Polytopes

Jean Cardinal (Université Libre de Bruxelles – Bruxelles, Belgique, cardinaljean@gmail.com)

We present various families of flip graphs that are skeletons of polytopes. We begin with the ubiquitous Associahedron, whose skeleton is the flip graph of triangulations of a convex n-gon and the rotation graph of binary trees with $n-1$ leaves. We then introduce graph associahedra, an elegant generalization of associahedra whose skeleton is the rotation graph of so-called elimination trees on a given simple connected graph G. Next, we review well-known properties of the flip graph of acyclic orientations of a graph G, the skeleton of the graphical zonotope of G. We then proceed to show that graphical zonotopes and graph associahedra have a common generalization called hypergraphic polytopes, whose skeletons are flip graphs of acyclic orientations of a given hypergraph.

For each family of flip graphs, we mention old and new results on flip distance and Hamiltonicity properties, emphasizing the computational aspects: How hard is it to compute the flip distance between two given objects? Does there exist an efficient Gray code for listing these objects, one flip at a time?

References

1 Alexander Postnikov: Permutohedra, associahedra, and beyond. International Mathematics Research Notices, 6:1026–1106, 2009.
2 Thibault Manneville and Vincent Pilaud: graph properties of graph associahedra. Séminaire Lotharingien de Combinatoire 73 (2015), Article B73d
3 Marcelo Aguiar and Federico Ardila: Hopf monoids and generalized permutahedra, arXiv:1709.07504, 2017.
4 Carolina Benedetti, Nantel Bergeron, John Machacek: Hypergraphic polytopes: combinatorial properties and antipode. arXiv:1712.08848, 2017.

5 Jean Cardinal, Stefan Langerman, Pablo Pérez-Lantero: On the diameter of tree associahedra. Electron. J. Comb. 25(4): P4.18, 2018.
6 Prosenjit Bose, Jean Cardinal, John Iacono, Grigorios Koumoutsos, Stefan Langerman: Competitive online search trees on trees. SODA 2020: 1878-1891.
7 Jean Cardinal, Lionel Pournin, Mario Valencia-Pabon: Bounds on the diameter of graph associahedra. LAGOS 2021: 239-247. (arXiv:2106.16130).
8 Jean Cardinal, Arturo I. Merino, Torsten Mütze: Combinatorial generation via permutation languages. IV. Elimination trees. CoRR abs/2106.16204 (2021). Extended abstract in SODA 2022.

3.3 Geometry of Large Genus Flat Surfaces and Open Problems on Square-Tiled Surfaces

Élise Goujard (University of Bordeaux – Bordeaux, France, elise.goujard@u-bordeaux.fr)

Joint work of Delecroix, Vincent; Zograf, Peter; Zorich, Anton
Main reference Vincent Delecroix, Elise Goujard, Peter Zograf, Anton Zorich: "Large genus asymptotic geometry of random square-tiled surfaces and of random multicurves", arXiv, CoRR, Vol. abs. 2007.04740, 2020.
URL http://dx.doi.org/10.48550/ARXIV.2007.04740

In this talk we present some results (joint work with Vincent Delecroix, Peter Zograf and Anton Zorich) on the geometry of large genus surfaces and related problems on square-tiled surfaces. The study of the $SL(2,\mathbb{R})$-dynamics and the geometry of the moduli space of translation surfaces allows to prove equidistribution of square-tiled surfaces with fixed combinatorics in the strata and uncorrelation between horizontal and vertical combinatorics [2], as well as large genus asympstotics for the distribution of cylinders for instance [1]. Some results about the statistics of random square-tiled surfaces with no constraints on the singularities (especially for half-translation square-tiled surfaces) are still open.

References

1 V. Delecroix, é. Goujard, P. Zograf, A. Zorich, *Large genus asymptotic geometry of random square-tiled surfaces and of random multicurves* , `arXiv:2007.04740`.
2 V. Delecroix, é. Goujard, P. Zograf, A. Zorich, *Enumeration of meanders and Masur-Veech volumes.* Forum Math. Pi 8 (2020), e4, 80 pp.

3.4 Morphing Graph Drawings

Fabrizio Fratti (Roma Tre University – Rome, Italy, frati@dia.uniroma3.it)

A morph between two geometric shapes is a continuous transformation of one shape into the other. Morphs are useful in many areas of computer science, including Computer Graphics, Animation, and Modeling. This talk surveys known results and algorithms on morphing graph drawings.

The first part of the talk is devoted to morphing algorithms for planar straight-line drawings. Contraction-based algorithms [1, 2], coefficient-interpolation-based algorithms [4], and one-coefficient-at-a-time algorithms [7] are described. The running time and the resolution of the described morphing algorithms are discussed.

The second part of the talk deals with other graph drawing styles. In particular, morphs between non-planar graph drawings [3], three-dimensional morphs [5], and upward morphs [6] are discussed

References

1. S.S. Cairns, *Deformations of plane rectilinear complexes*, "The American Mathematical Monthly", volume 51, 5, 247–252, Mathematical Association of America, 1944.
2. Soroush Alamdari and Patrizio Angelini and Fidel Barrera-Cruz and Timothy M. Chan and Giordano Da Lozzo and Giuseppe Di Battista and Fabrizio Frati and Penny Haxell and Anna Lubiw and Maurizio Patrignani and Vincenzo Roselli and Sahil Singla and Bryan T. Wilkinson, *How to Morph Planar Graph Drawings*, SIAM J. Comput., volume 46, 2, 824–852, 2017.
3. Patrizio Angelini and Michael A. Bekos and Fabrizio Montecchiani and Maximilian Pfister, *On Morphing 1-Planar Drawings*, 47th International Workshop on Graph-Theoretic Concepts in Computer Science (WG), Lecture Notes in Computer Science, 12911, 270–282, Springer, 2021.
4. Michael S. Floater and Craig Gotsman, *How to morph tilings injectively*, Journal of Computational and Applied Mathematics, volume 101, 1, 117-129, 1999, doi:10.1016/S0377-0427(98)00202-7.
5. Elena Arseneva and Prosenjit Bose and Pilar Cano and Anthony D'Angelo and Vida Dujmovic and Fabrizio Frati and Stefan Langerman and Alessandra Tappini, *Pole Dancing: 3D Morphs for Tree Drawings*, J. Graph Algorithms Appl., volume 23, 3, 579–602, 2019, `https://doi.org/10.7155/jgaa.00503`.
6. Giordano Da Lozzo and Giuseppe Di Battista and Fabrizio Frati and Maurizio Patrignani and Vincenzo Roselli, *Upward Planar Morphs*, Algorithmica, volume 82, 10, 2985–3017, 2020, `https://doi.org/10.1007/s00453-020-00714-6`, doi:10.1007/s00453-020-00714-6.
7. Jeff Erickson and Patrick Lin, *Planar and Toroidal Morphs Made Easier*, 29th International Symposium on Graph Drawing and Network Visualization (GD), Lecture Notes in Computer Science, Springer, 2021.

4 Working Groups

4.1 Reconfiguring Popular Faces

Florestan Brunck (Institute of Science and Technology, Austria, florestan.brunck@ist.ac.at)
Hsien-Chih Chang (Dartmouth College, USA, hsien-chih.chang@dartmouth.edu)
Maarten Löffler (Utrecht University, the Netherlands, m.loffler@uu.nl)
Tim Ophelders (Utrecht University & TU Eindhoven, the Netherlands, t.a.e.ophelders@uu.nl)
Lena Schlipf (Universität Tübingen, Germany, lena.schlipf@uni-tuebingen.de)

Let $\mathcal{A}$ be a set of curves which lie inside the area bounded by a closed curve F, called the *frame*. All curves in $\mathcal{A}$ are either *closed* or they are *open* with a start and end point on F. We refer to $\mathcal{A}$ as a *curve arrangement*, see Figure 1a. We consider only *simple* arrangements, where no three curves meet in a point, and all intersections are transversal crossings (no tangencies). The arrangement $\mathcal{A}$ can be seen as an embedded multigraph whose vertices are crossings between curves and whose edges are *curve segments*. $\mathcal{A}$ subdivides the region bounded by F into *faces*. We call a face *popular* when it is incident to multiple curve segments belonging to the same curve in $\mathcal{A}$ (see Figures 1b-c).

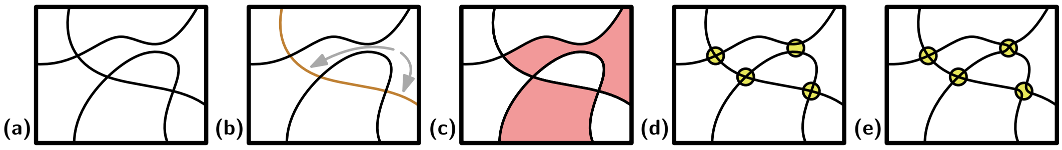


Figure 1 **(a)** An arrangement of curves inside a frame. **(b)** The red curve is incident to the top right face in two disconnected segments, making the face *popular.* **(c)** All popular faces are highlighted. **(d)** A set of switches. **(e)** A possible reconfiguration after which no more faces are popular.

▶ **Proposition 1.** *We do not like popular faces.*

Now, let a *switch* be a local area in which we are allowed to reroute the curves of $\mathcal{A}$ (see Figures 1d-e).

▶ **Problem 4.1.** *Given a curve arrangement and a set of switches, can we reconfigure the curves so as to remove all, or as many as possible, popular faces? And can we minimize the number of switch operations?*

Once we are given a set of switches, the question above is essentially combinatorial. Now suppose the switches are not given in advance, but we allow curves to be reconfigured whenever they are "sufficiently close" to each other.

▶ **Observation 1.** *If all intersections are switches, then it is possible to remove all popular faces.*

Proof. Simply set every switch to any non-intersecting state. Then each curve bounds a single face on each side, and thus no face is popular. ◀

▶ **Problem 4.2.** *Is there a reasonable way to geometrically determine a set of switches? And how does this influence the complexity of Problem 4.1?*

Finally, suppose that our curves have *fixed* parts and *flexible* parts. That is, each curve is a smooth concatenation of pieces, each of which is either fixed or flexible. Fixed pieces may never be altered. Flexible pieces may be changed.

▶ **Problem 4.3.** *Given a curve arrangement with fixed and flexible pieces, can we reconfigure the flexible parts of the curves so as to remove all popular faces? Can we stay as close to the original arrangement as possible?*

Motivation & Background

Our question is motivated by the problem of generating *curved nonograms*. Nonograms, also known as *Japanese puzzles*, *paint-by-numbers*, or *griddlers*, are a popular puzzle type where one is given an empty grid and a set of *clues* on which grid cells need to be colored. A clue consists of a sequence of numbers specifying the numbers of consecutive filled cells in a row or column. A solved nonogram typically results in a picture (see Figure 2 (a)). There is quite some work in the literature on the difficulty of solving nonograms [1].

Van de Kerkhof et al. introduced *curved* nonograms, a variant in which the puzzle is no longer played on a grid but on any arrangement of curves [4] (see Figure 2b). In curved nonograms, a clue specifies the numbers of filled faces of the arrangement in the sequence of faces that are incident to a common curve on one side. Van de Kerkhof et al. focus on

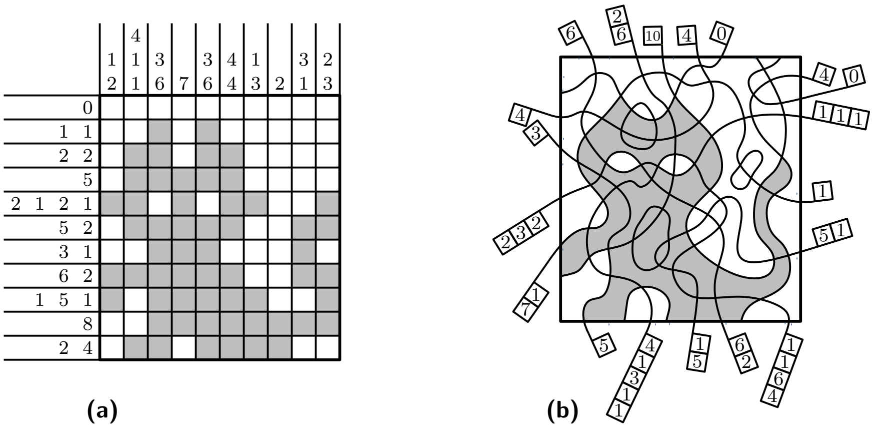


Figure 2 Two nonogram puzzles in solved state. **(a)** A classic nonogram. **(b)** A curved nonogram.

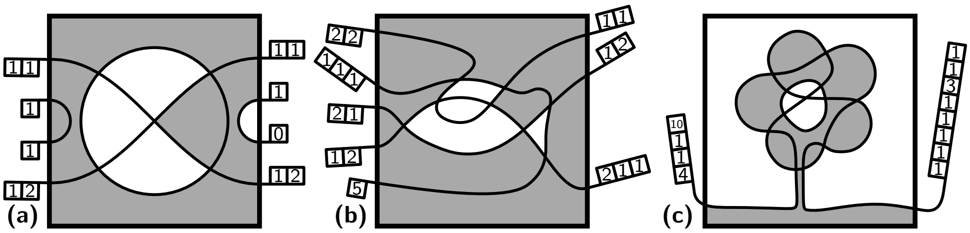


Figure 3 Three types of curved nonograms of increasing complexity [4], shown with solutions. **(a)** *Basic* puzzles have no popular faces. **(b)** *Advanced* puzzles may have popular faces, but no self-intersections. **(c)** *Expert* puzzles have self-intersecting curves. We can observe closed curves (without clues) in **(a)** and **(c)**.

heuristics to automatically generate such puzzles from a desired solution picture by extending curve segments to a complete curve arrangement. Van de Kerkhof et al. observe that curved nonograms come in different flavours of increasing complexity – not in terms of how hard it is to *solve* a puzzle, but how hard it is to understand the rules (see Figure 3). They state that it would be of interest to generate puzzles of a specific complexity level; their generators are currently not able to do so other than by trial and error.

- *Basic* nonograms are puzzles in which each clue corresponds to a sequence of unique faces. The analogy with clues in classical nonograms is straightforward.
- *Advanced* nonograms may have clues that correspond to a sequence of faces in which some faces may appear multiple times because the face is incident to the same curve (on the same side) multiple times. When such a face is filled, it is also counted multiple times; in particular, it is no longer true that the sum of the numbers in a clue is equal to the total number of filled faces incident to the curve. This makes the rules harder to understand, and thus advanced nonograms are only suitable for more experienced puzzle freaks.
- *Expert* nonograms may have clues in which a single face is incident to the same curve on *both* sides. They are even more confusing than advanced nonograms.

It is not hard to see that expert puzzles correspond exactly to arrangements with self-intersecting curves. The difference between basic and advanced puzzles is more subtle; it corresponds exactly to the presence of *popular faces* in the arrangement.

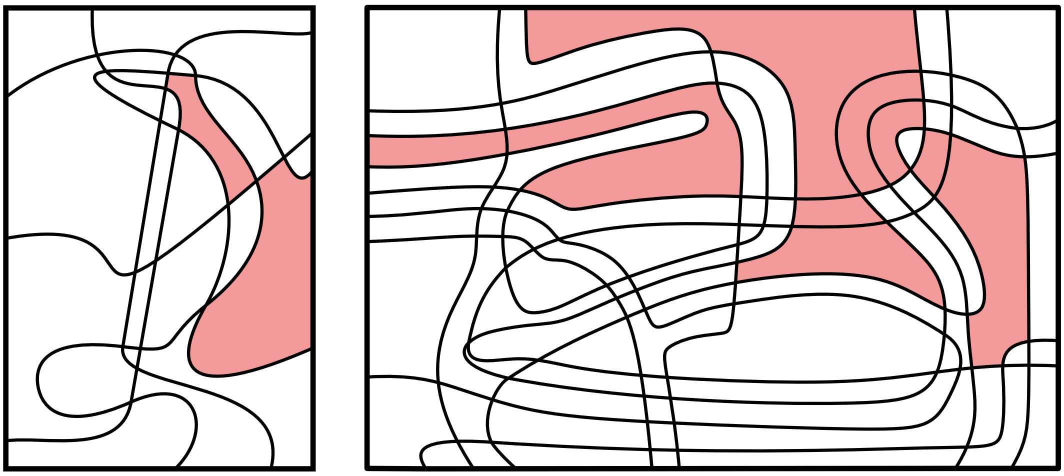


Figure 4 Some examples of real puzzles (without the clues) with all popular faces highlighted.

One possibility to generate nonograms of a specific complexity would be to take an existing generator and modify the output. Recently, de Nooijer [2] investigated how this might be done by *inserting* a new curve into the arrangement; see also [3]. Problems 1-3 explore a different approach, geared towards the generation of *basic* puzzles. Of course, another interesting question is what can be done to generate *advanced* puzzles.

▶ **Problem 4.4.** *How do Problems 1-3 change when the goal is only to remove all self-intersections?*

Results

In this section, we present NP-hardness results and bring a negative answer to both Problem 4.1 and its self-intersection variant (Problem 4.4). We begin by examining the self-intersection problem and derive Theorem 1, from which we later reduce our original question about the removal of popular faces, as well as subsequent extensions concerning the minimal number of switch operations required. The organisation of our results is summarized in Fig. 5.

Base Results.

▶ **Theorem 1.** *Given a curve arrangement and a prescribed set of switches, it is NP-hard to decide whether it is possible to configure the switches such that the resulting arrangement has no self-intersections.*

The proof of Theorem 1 is ultimately a reduction from 3-SAT, but involves an intermediate reduction through a naturally related problem, which we call the *permuter problem*.

The Permuter Problem and its Reduction from 3-SAT. Consider the straight-line-edges drawing of the complete bipartite graph $K_{k,k}$ in the plane, with bi-partition I and O (referred as its *inputs* and *outputs*) such that I and O are evenly distributed on opposite sides of a rectangle R. Fixing the same linear order on the vertices of I and O, the *k-permuter* $\Pi_{k,\sigma}$ is the matching of $K_{k,k}$ associated to the permutation σ of $[n]$. An instance of the permuter problem consists of a finite collection $\{\Pi_{k_i}, \sigma_i\}_{i\in[n]}$ of k_i-permuters, realised in the plane using an associated collection $\{R_i\}_{i\in[n]}$ of rectangles, together with a finite collection of paths $\{P_i\}_{i\in[m]}$, such that:

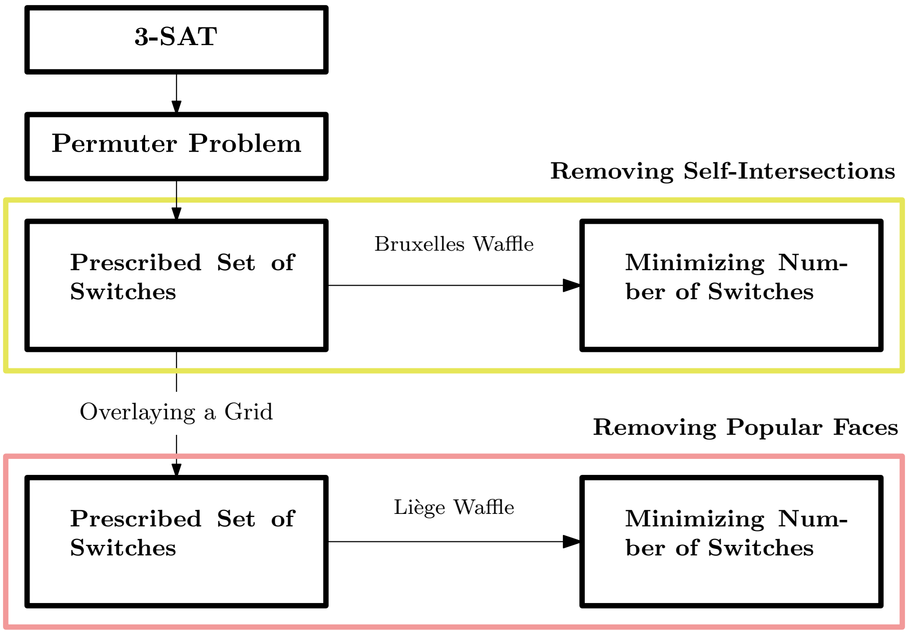


Figure 5 The flow of successive reductions underlying our NP-hardness proofs.

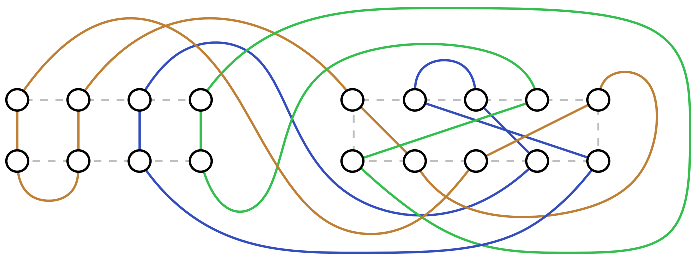


Figure 6 An instance of the k-permuter problem with a 4-permuter and a 5-permuter, assigned with the permutations $\sigma_1 = \mathrm{Id}$ and $\sigma_2 = (12534)$. There are 3 resulting closed loops: one simple (green) and two self-intersecting (blue and brown).

- Every element of $\{P_i\}_{i\in[m]}$ has both its endpoints in one of the k-permuters (possibly the same, and possibly both inputs/outputs).
- Every input and output of every permuter is connected to a unique element of $\{P_i\}_{i\in[m]}$.
- For all $i \in [m]$, for all $j \in [n]$, $P_i \cap R_j = \emptyset$.

By construction, every path of $\{P_i\}_{i\in[m]}$ belongs to a unique closed loop. The *Permuter problem* then consists in deciding whether or not there exists a choice of n permutations $\sigma_1, \sigma_2, \ldots, \sigma_n$ of $[k_1], [k_2], \ldots, [k_n]$ such that each resulting loop is *simple* (see Fig. 6).

▶ **Claim 1.** *The Permuter problem is NP-hard.*

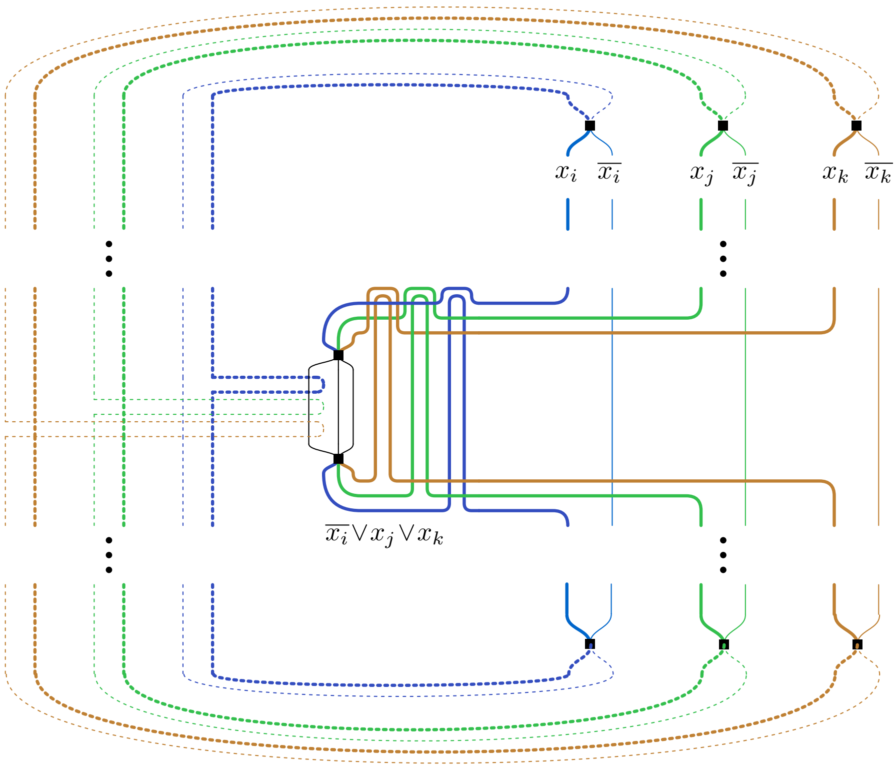


Figure 7 A single 3-clause in the reduction from 3-SAT to the Permuter problem.

Proof. The proof is by reduction from 3-SAT. We shall use two types of gadgets: 2-permuters for variable assignment and 3-permuters for clause verification (see Fig. 7). For simplicity, each k-permuter is depicted by a black box on the diagram, where the value of k is made clear by the number of incoming/outgoing paths. Each different colour in the figure indicates a different variable. The thick or thin dashed lines on the top, bottom and middle-left part of the diagram indicate respectively the false and true literals of each variable. While the thick and think solid lines in the middle-right section of the diagram indicate respectively the true or false assignment of each variable. Given a Boolean formula with n variables, we construct $2n$ non-crossing semi-circular arcs. We replicate this construction twice to form the top and bottom parts of the diagram. In the middle, we show a single clause gadget, involving two 3-permuters. To simulate the two logical OR of the clause, we proceed as follows: if the corresponding 3-clause involves the variables x_i, x_j and x_k, we select the wire corresponding to their desired truth value in the clause (i.e. thick for x_i, thin for $\bar{x}_i$) and "drag" them towards the gadget to intersect the same single path chosen among the three paths linking the output of the first 3-permuter to the inputs of the second. By construction, there does not exist a valid permutation assignment to the two 3-permuters which avoids all possible self-intersections with the three black paths if and only if x_i, x_j and x_k all have the wrong truth assignment. Furthermore, to the right of the 3-clause gadget, we have weaved the incoming paths of the first and the outgoing paths of the second in such a way that, if the composition of the two 3-permuters were not the identity, at least one of the resulting closed loop would self-intersect. Thus each such pair of 3-permuters cannot "cheat" and has to compose to the identity. As a consequence, for each of the variables involved, the composition of its two 2-permuters must also be the identity. By construction, there are then exactly two ways of ensuring this is the case: either both 2-permuters are the identity

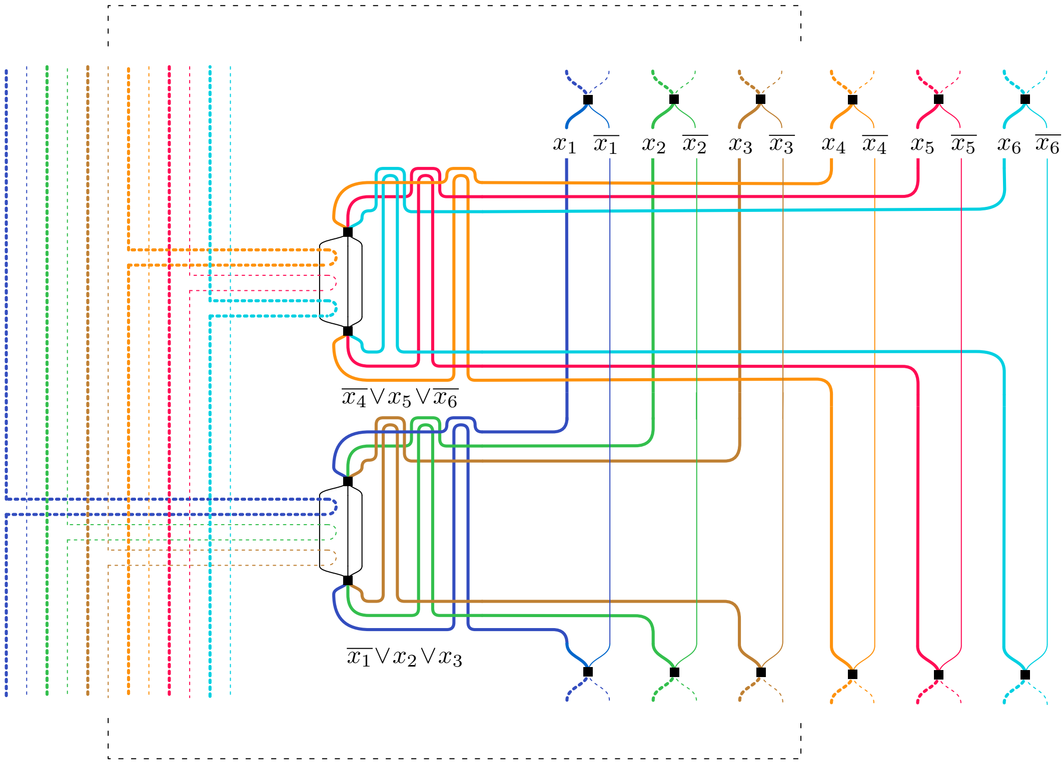


Figure 8 The instance of the Permuter problem created for the Boolean formula $(\overline{x_1} \vee x_2 \vee x_3) \wedge (\overline{x_4} \vee x_5 \vee \overline{x_6})$.

themselves (setting the variable to be true), or both of them correspond to the transposition (12) swapping the inputs (setting the variable to be false). Fig. 8 shows the instance created for the Boolean formula $(\overline{x_1} \vee x_2 \vee x_3) \wedge (\overline{x_4} \vee x_5 \vee \overline{x_6})$. ◀

The Permuter Problem Reduces to the Self-Intersection Problem.

▶ **Claim 2.** *The problem of configuring a given set of switches to avoid self-intersections reduces from the Permuter problem.*

Proof. The proof of the claim proceeds as follows: we first construct self-intersection gadgets to simulate 2-permuters, take note of the fact that S_n is generated by transpositions and show how to use 2-permuters to construct general k-permuters. The construction for 2-permuters is presented on Fig. 9: the inputs and outputs are connected by two curves weaved into a double coil structure with two intersections, one of which is a switch. The 3 resulting configurations are shown on the figure; only the leftmost two are free of self-intersections.

To simulate a general k-permuter, we introduce the gadget described on Fig. 10. We begin with a k by k square and evenly distribute k inputs and outputs on its top and bottom edges, respectively. For all $i \in [n]$, the top i-th input is connected to the bottom $(n-i)$-th output by the path of slope -1 which gets reflected into a path of slope 1 upon meeting the left edge of the square. We then insert a total of $\frac{k(k-1)}{2}$ 2-permuter: one at every site where two paths intersect inside the square. While many configurations of this gadget are redundant and yield the same permutation, a short inductive argument shows that it is indeed able to simulate any permutation on k elements. The base case is simply

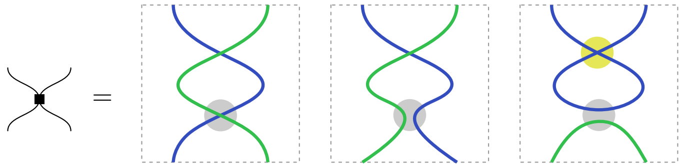


Figure 9 Constructing a 2-permuter using two "doubly-coiled" curves. The grey disk indicates the only switch available, the yellow disk highlights the self-intersection forbidding the third possibility given by the switch.

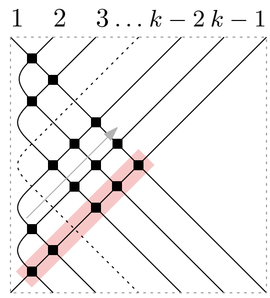


Figure 10 Constructing a k-permuter, using $\frac{k(k-1)}{2}$ 2-permuters.

the 2-permuter we previously described. Assume then by induction that the version of our gadget with $k-1$ inputs and outputs can successfully simulate all permutations of $[k-1]$. Note that any permutation σ of $[k]$ can be written as the composition of a permutation of $[k-1]$ followed by the insertion of the element k into one of the k positions before or after one of the $k-1$ permuted elements. It is thus enough to show that the addition of the last "row" of k 2-permuters (highlighted in pink) can simulate this last insertion step. Labelling the $(k-1)$ 2-permuters of the $(k-1)$-th row according to the direction indicated on Fig. 10, we insert the element k in position i (before the i-th element) by setting all the 2-permuters from positions i to k to swap their inputs, and the remaining i 2-permuters to the identity. This effectively shifts all the elements with positions greater than i by 1 (the permuters reroute their corresponding wires to the segment of slope 1 instead of -1) as we sequentially shift the element k to the left i times (the permuters successively let the k-th input path "slide" on the last path of slope -1). ◀

Extensions

We now extend the result from Theorem 1 in several ways, to prove that the problem remains hard when we wish to minimize the number of switch operations, or when the goal is to remove all popular faces rather than self-intersections.

The idea is always to locally alter the reduction in a way that does not affect its global properties.

Minimizing the number of switches. In the previous section, we had a prescribed set of switches; indeed, this is necessary since if we are allowed to switch everywhere, then we can always remove all self-intersections by Observation 1.

However, now consider the scenario in which we wish to minimize the number of switch operations, or, in the decision version, we wish to test for a given k whether there exists a sequence of k switch operations such that the resulting arrangement has no self-intersections. In this scenario, we may or may not have a prescribed set of switches.

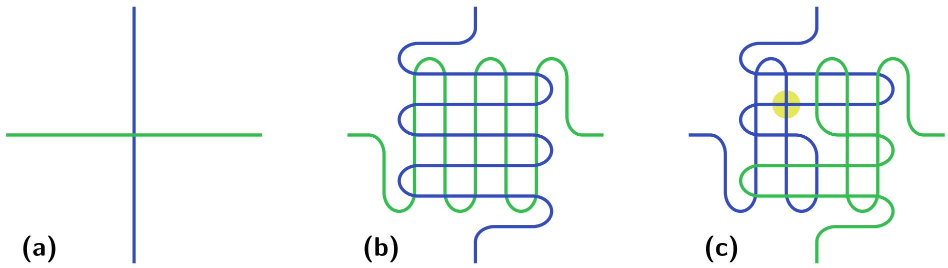


Figure 11 The Bruxelles waffle. (a) A crossing between two different curves. (b) We adapt both of the curves so they intersect c^2 times. (c) Any sequence of local switches to emulate a global switch must be of length at least c or lead to a self-intersection.

The idea is to emulate the construction from Section 4.1, but to replace every self-intersection in the construction which is *not* a switch by a *waffle gadget*. Such a gadget is built in such a way that even if *every* intersection in the gadget is a switch, the number of switches required to change its global state is more than a parameter c. If we then choose $c > k$, the result follows since essentially we are never allowed to switch these gadgets.

The Bruxelles Waffle

In this section, we describe the *Bruxelles waffle* (in contrast to the *Liège waffle* which we describe in Section 4.1). The construction is illustrated in Figure 11.

Here we describe the construction in words.

▶ **Lemma 2.** *In a Bruxelles waffle, any sequence of fewer than c switches must result in an arrangement with either the same combinatorial structure as the original, or at least one self-intersection.*

Proof. Assume that after fewer than c switches, no self-intersections remain. We need to show that opposite terminals lie on the same strand. Assume for a contradiction that opposite terminals do not lie on the same strand. Then the left terminal lies on the same strand as either the top or the bottom terminal. Without loss of generality (by rotational symmetry of the gadget) assume that the left terminal lies on the same strand as the bottom terminal.

Because there are c rows, at least one row has not undergone any switch, and the horizontal path in that row has a single color. Because there are no self-intersections, all strands crossing the row vertically have a color different from the horizontal path of the row. Removing the row splits the gadget into a part above and a part below the row, and the left and right endpoints of the horizontal path connect to different such parts. The strand containing the horizontal path of the row cannot be closed up without crossing the row, so the strand connects a terminal below and a terminal above the row. That is, it connects the bottom or left terminal to the top or right terminal. This contradicts our assumption that the bottom and left terminals lie on the same strand. ◀

We conclude:

▶ **Theorem 3.** *Given a curve arrangement and an integer k, it is NP-hard to decide whether there exists a sequence of k switches such that the resulting arrangement has no self-intersections.*

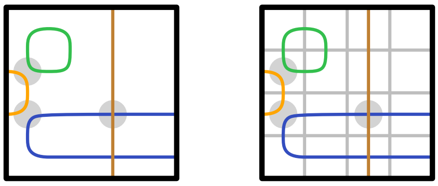


Figure 12 Overlaying a grid. (a) None of the popular faces are the result of a self-intersection. (b) After superimposing a fine enough grid whose non-empty cells are of type (1) or (2), any popular face is now necessarily caused either by a self-intersection or a switch incident to two strands of the same colour.

Removing popular faces. Next, we consider the problem of removing all popular faces rather than only the self-intersections (that is, the aim is to produce a *basic* nonogram rather than an *advanced* one).

The idea is to globally overlay the construction from Section 4.1 with a sufficiently fine grid of perpendicular lines, in which none of the intersections with these new lines are switches. We construct our grid in such a way that each non-empty grid cell either has:

(1) A single arc traversing it and connecting distinct sides of the cell.

(2) Two arcs crossing exactly once (or meeting at a switch) and connecting opposite sides of the cell.

Additionally, we note that in our reduction of the self-intersection problem from the Permuter problem, configurations with no self-intersections have the added property that no switches involve two strands that belong to a single curve. To see this, it is enough to look at the design of 2-permuters: if both strands involved in their unique switch belonged to the same curve, then the upper crossing of their double-coil design would be a self-intersection. Because of this property, it is clear that once the grid is overlaid, all remaining popular faces are due to self-intersections (see Fig. 12). Therefore, in order to remove all popular faces we need to remove all self-intersections, and this is also sufficient.

We conclude:

▶ **Theorem 4.** *Given a curve arrangement and a prescribed set of switches, it is NP-hard to decide whether it is possible to configure the switches such that the resulting arrangement has no popular faces.*

Removing popular faces with a minimum number of switches. Finally, we consider the setting where every intersection is an allowed switch. In this setting we wish to remove all popular faces using a minimum number of switches. The idea is similar to that in Section 4.1, but we will need to use a different gadget that ensures we cannot perform any switches (since having self-intersections does not necessarily imply there are no popular faces). To this end, we introduce the *Liège waffle*.

The Liège Waffle

In this section, we describe the *Liège waffle*. It is illustrated in Figure 13.

Essentially, we overlay c new closed curves on the two (crossing) terminal strands, such that each of the new curves is incident to each of the four unbounded faces, and that the disk bounded by the curve contains the crossing between the two terminal strands.

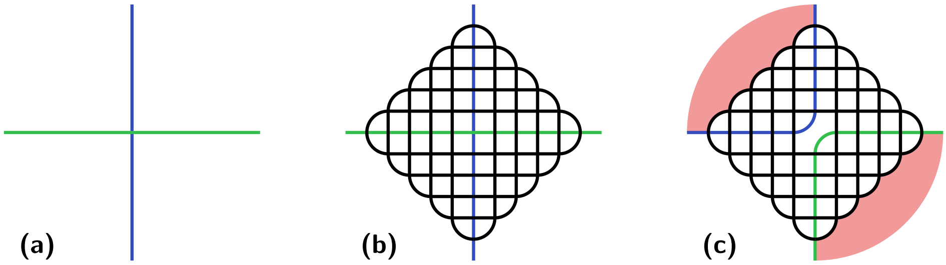


Figure 13 The Liège waffle. (a) A crossing between two different curves. (b) We insert a sequence of c new curves. (c) Any sequence of local switches to emulate a global switch must be of length at least c or lead to a popular face.

▶ **Lemma 5.** *In a Liège waffle, any sequence of fewer than c switches must result in an arrangement with either the same combinatorial structure as the original, or at least one popular face.*

Proof. If we change the global connectivity, then one of the four unbounded faces will globally have two strands of the same curve; say the top left face (Figure 13 (c)). In order for this face to *not* be popular, these two strands must be consecutive along the curve. Initially, all intersections incident to the face are crossing, so they must all be uncrossed. There are $c + 1$ such intersections, so we need at least $c + 1$ switches. ◀

We conclude:

▶ **Theorem 6.** *Given a curve arrangement and an integer k, it is NP-hard to decide whether there exists a sequence of k switches such that the resulting arrangement has no popular faces.*

References

1. Yen-Chi Chen and Shun-Shii Lin. A fast nonogram solver that won the TAAI 2017 and ICGA 2018 tournaments. *ICGA Journal*, 41(1):2–14, 2019.
2. Phoebe de Nooijer. Resolving popular faces in curve arrangements. Master's thesis, Utrecht University, 2022.
3. Phoebe de Nooijer, Soeren Nickel, Alexandra Weinberger, Zuzana Masárová, Tamara Mchedlidze, Maarten Löffler, and Günter Rote. Removing popular faces in curve arrangements by inserting one more curve. In *Proc. 38th European Workshop on Computational Geometry*, 2022.
4. Mees van de Kerkhof, Tim de Jong, Raphael Parment, Maarten Löffler, Amir Vaxman, and Marc J. van Kreveld. Design and automated generation of Japanese picture puzzles. *Comput. Graph. Forum*, 38(2):343–353, 2019.

4.2 Rotation Distance between Elimination Tree

Maike Buchin ((Ruhr-Universität Bochum, Netherlands, maike.buchin@rub.de)
Jean Cardinal (ULB – Brussels, Belgium, jean.cardinal@ulb.be)
Linda Kleist (TU Braunschweig, Germany, kleist@ibr.cs.tu-bs.de)
Boris Klemz (Universität Würzburg, Germany, boris.klemz@uni-wuerzburg.de)
Anna Lubiw (University of Waterloo, Canada, alubiw@uwaterloo.ca)
Torsten Mütze (University of Warwick – Coventry, UK, torsten.mutze@warwick.ac.uk)
Alexander Neuhaus (Ruhr-Universität Bochum, Germany,alexander.neuhaus-b7x@rub.de)
Lionel Pournin (Université Paris Nord – Villetaneuse, France, lionel.pournin@univ-paris13.fr)

Problem description

Given a simple connected graph $G = (V, E)$, an *elimination tree* on G is obtained by selecting a root $r \in V$, and defining the subtrees of r as elimination trees on the connected components of $G \setminus \{r\}$. (Note that $G \setminus \{r\}$ can be connected, in which case r has a single child in the elimination tree.) There is a surjective mapping from the set of permutations S_n to elimination trees on an n-vertex graph: define r as the vertex of the current subgraph that has minimum index in the permutation [21]. Elimination trees are natural generalizations of binary search trees, which are obtained by letting G be a path on n vertices [38, 33].

Just like in binary search trees, one can define a *rotation* operation on elimination trees. Two elimination trees differ by a rotation if there exist two permutations that generate them, and which differ by a single adjacent transposition [50]. The rotation graph between elimination trees of a graph G is the skeleton of a polytope known as the *graph associahedron* of G [43, 44]. We consider the following computational problem:
Input: A simple connected graph G, two elimination trees T_1, T_2 on G, and an integer k.
Question: Can T_1 be transformed into T_2 using at most k rotations?

What is the complexity of this problem? Is it NP-hard?

The motivation here is the longstanding open question of the complexity of computing the rotation distance between two binary trees, a fixed-parameter version of the above problem in which G is an n-vertex path.

Directions

Complexity of tree-depth. The tree-depth $\mathrm{td}(G)$ of a graph G is the minimum height of an elimination tree on G. Maybe the following result and its proof (see Pothen [56]) could be of some use in a hardness proof for the rotation distance.

▶ **Theorem 1.** *Deciding the tree-depth of a graph G is* NP*-hard.*

Proof. We reduce from the *complete balanced subgraph* problem in a bipartite graph. In this NP-hard problem, we wish to find the largest induced $K_{t,t}$ in a given bipartite graph G. Let us denote the maximum size t by $\mathrm{CBS}(G)$. We claim that for the complement $\bar{G}$ of a bipartite graph G,

$$\mathrm{td}(\bar{G}) = |V(G)| - -\mathrm{CBS}(G).$$

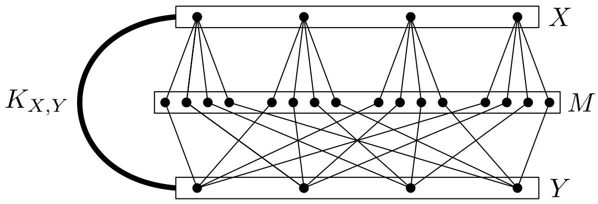


Figure 14 Our gadget corresponding to an edge $\{x, y\}$.

Hardness follows from the claim. The claim can be proved by observing that elimination trees on the complement of a bipartite graph have a special structure: only one vertex has more than one children, and this vertex has exactly two children. Hence elimination trees have an "inverted Y" shape. The shortest branch has length CBS(G). ◀

▶ **Conjecture 1.** *Rotation distance between elimination trees on a graph G is* NP*-hard, even if G is the complement of a bipartite graph.*

Edge gadget. In an attempt to prove NP-completeness, we considered the design of a gadget that allows to implement a binary choice.

Suppose we want to reduce from vertex cover. We need an edge gadget, such that the edge is covered when at least one of its endpoint is selected. The idea is to build a gadget graph G which has two sets of vertices X and Y and two elimination trees T_1 and T_2 for this graph, such that on any geodesic between T_1 and T_2, all vertices in X are removed first (this is interpreted as "one endpoint of the edge is selected in the vertex cover"), *or* (this is a logical or, not an exclusive or) all vertices in Y are removed first (this is interpreted as "the other endpoint of the edge is selected in the vertex cover"). We need to design an edge gadget so that not removing X or Y first is very bad, and removing X or Y or $X + Y$ first is equally good and optimal, and any other order of removing them is not optimal.

Let us define the gadget $\hat{G}_e$ for an edge e as follows, see also fig. 14: $\hat{G}_e$ is a graph on two independent sets X and Y, each composed of m/k vertices, and a third independent set $M = \{1, 2, \ldots, m\}$. In what follows, k is a function of m, to be decided later (maybe something like $\sqrt{m}$). The first vertex of X is connected to the first k vertices of M, the second to the next k vertices, and so on. The same goes for Y. Finally, we add all edges of the complete bipartite graph between X and Y, see fig. 14.

Now define two elimination trees T_1 and T_2 on this gadget. T_1 first eliminates all vertices of M in the order $1, 2, \ldots, m$, then vertices of X. Then all vertices of Y are siblings in the tree. In T_2, we first eliminate vertices of M in order $m, m - 1, \ldots, 1$, then Y, then X.

▶ **Lemma 2.** *The rotation distance between T_1 and T_2 is $O(m^2/k)$.*

Proof. Move all vertices of X up. This costs (roughly)

$$|X| \cdot |M| = \frac{n}{k} \cdot m$$

rotations. Now we are left with m/k connected components, each a star of $k + 1$ vertices. Now reorder the vertices of each star in $2k$ rotations each, costing

$$\frac{m}{k} \cdot 2k$$

rotations overall. Now push X back down, past all vertices of M, in again $\frac{m}{k} \cdot m$ rotations, taking care that the vertices of M are in reverse order (as in T_2). Finally, all vertices of X need to move down past all vertices of Y, costing $|X| \cdot |Y| = m^2/k^2$ rotations. The total is

$$2\frac{m}{k} \cdot m + \frac{m}{k} \cdot 2k + m^2/k^2 = 2\frac{m^2}{k} + o(m^2/k)$$ ◀

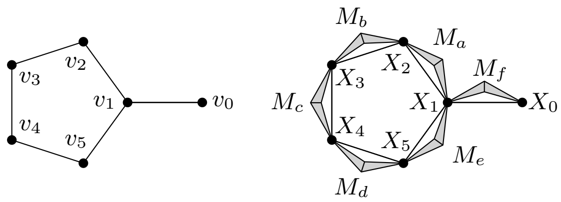


Figure 15 Combining edge gadgets: graph.

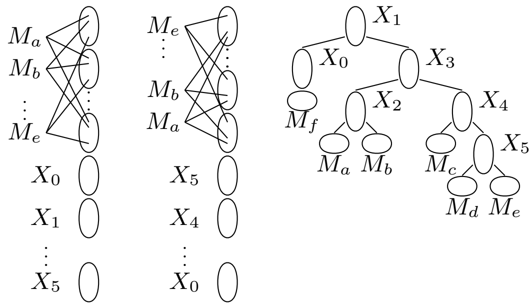


Figure 16 Combining edge gadgets: elimination tree.

It remains unclear whether copies of this gadget can be combined in a reduction; see Figures 16 and 16 for an example.

Reduction from token swapping. An alternative base problem would be the following (see Aichholzer et al. [55]). The input is a tree on n nodes, and two configurations (initial and final) where n distinct tokens are situated at the nodes. The goal is to get the tokens from their initial positions to their final positions (i.e. to effect the given permutation of tokens). The reconfiguration operation is a *swap* that exchanges the two tokens at the endpoints of an edge. In other words, we have the permutation group generated by the transpositions determined by the edges of the tree. The question is: Can you get from the initial to the final configuration with at most k swaps?

What are the similarities?

- working in a tree
- the reconfiguration graph is regular (from any configuration, there are $n-1$ possible "flips")
- the NP-hardness proof for token swapping on a tree uses ideas like the ones being suggested above: an "obvious" solution involves lots of flips and the only way to reduce the number of flips is to use a carefully structured solution that controls the movement.

Token swapping on a path is the same as rotation distance for elimination sequences of a clique. So they have an easy common case. The proof that token swapping on a tree is NP-complete is really long and hard [55].

Other open problems. The rotation distance problem can be specialized into possibly easier problems by restricting the types of possible input graphs G. The following cases might be relevant:

G is a star. The rotation distance problem is in P(Lionel and Jean).

G is a path. This is the famously open problem of computing the rotation distance between two binary search trees.

G is a broom. Can this be proved polynomial-time equivalent to the case of a path?

G is a tree. Same question as for brooms.

References

1 Wolk, E. The comparability graph of a tree. *Proceedings Of The American Mathematical Society.* **13** pp. 789-795 (1962)
2 Golumbic, M. Trivially perfect graphs. *Discrete Mathematics.* **24**, 105-107 (1978)
3 Yan, J., Chen, J. & Chang, G. Quasi-threshold Graphs. *Discrete Applied Mathematics.* **69**, 247-255 (1996)
4 Drange, P., Fomin, F., Pilipczuk, M. & Villanger, Y. Exploring the Subexponential Complexity of Completion Problems. *ACM Transactions On Computation Theory.* **7**, 14:1-14:38 (2015)
5 Nešetřil, J. & Mendez, P. On Low Tree-Depth Decompositions. *Graphs Comb..* **31**, 1941-1963 (2015)
6 Berendsohn, B. & Kozma, L. Splay trees on trees. *CoRR.* **abs/2010.00931** (2020)
7 Chapoton, F., Fomin, S. & Zelevinsky, A. Polytopal Realizations of Generalized Associahedra. *Canadian Mathematical Bulletin.* **45** pp. 537-566 (2002)
8 Del Pia, A. & Michini, C. On the diameter of lattice polytopes. *Discrete & Computational Geometry.* **55** pp. 681-687 (2016)
9 Deza, A., Manoussakis, G. & Onn, S. Primitive zonotopes. *Discrete & Computational Geometry.* **60** pp. 27-39 (2018)
10 Deza, A. & Pournin, L. Improved bounds on the diameter of lattice polytopes. *Acta Mathematica Hungarica.* **154** pp. 457-469 (2018)
11 Dotsenko, V., Shadrin, S. & Vallette, B. Toric varieties of Loday's associahedra and noncommutative cohomological field theories. *Journal Of Topology.* **12** pp. 463-535 (2019)
12 Kalai, G. & Kleitman, D. A quasi-polynomial bound for the diameter of graphs of polyhedra. *Bulletin Of The American Mathematical Society.* **26** pp. 315-316 (1992)
13 Klee, V. & Walkup, D. The d-step conjecture for polyhedra of dimension d<6. *Acta Mathematica.* **117** pp. 53-78 (1967)
14 Mizera, S. Combinatorics and topology of Kawai-Lewellen-Tye relations. *Journal Of High Energy Physics.* pp. 2017:97 (2017)
15 Parlier, H. & Pournin, L. Flip-graph moduli spaces of filling surfaces. *Journal Of The European Mathematical Society.* **19** pp. 2697-2737 (2017)
16 Santos, F. A counterexample to the Hirsch conjecture. *Annals Of Mathematics.* **176** pp. 383-412 (2012)
17 Semple, C. & Steel, M. Phylogenetics. (Oxford University Press,2003)
18 Sleator, D., Tarjan, R. & Thurston, W. Rotation distance, triangulations, and hyperbolic geometry. *Journal Of The American Mathematical Society.* **1** pp. 647-681 (1988)
19 Davis, M., Januszkiewicz, T. & Scott, R. Fundamental groups of blow-ups. *Advances In Mathematics.* **177**, 115-179 (2003)
20 Loday, J. Realization of the Stasheff polytope. *Archiv Der Mathematik.* **83**, 267-278 (2004,9)
21 Cardinal, J., Merino, A. & Mütze, T. Combinatorial generation via permutation languages. IV. Elimination trees. *CoRR.* **abs/2106.16204** (2021), https://arxiv.org/abs/2106.16204
22 Cardinal, J., Langerman, S. & Pérez-Lantero, P. On the Diameter of Tree Associahedra. *Electr. J. Comb..* **25**, P4.18 (2018)
23 Ceballos, C., Manneville, T., Pilaud, V. & Pournin, L. Diameters and geodesic properties of generalizations of the associahedron. *27th International Conference On Formal Power Series And Algebraic Combinatorics (FPSAC 2015).* pp. 345-356 (2015,7)
24 Korte, B., Lovász, L. & Schrader, R. Greedoids. (Springer, Berlin,1991)
25 Björner, A. & Ziegler, G. Introduction to greedoids. *Matroid Applications.* (1992)
26 Stasheff, J. Homotopy Associativity of H-Spaces. I. *Transactions Of The American Mathematical Society.* **108**, 275-292 (1963)

27 Tamari, D. Monoïdes préordonnés et chaînes de Malcev. (1951), Thèse de Mathématiques, Paris

28 Lee, C. The Associahedron and Triangulations of the n-gon. *European Journal Of Combinatorics.* **10**, 551-560 (1989)

29 Aspvall, B. & Heggernes, P. Finding minimum height elimination trees for interval graphs in polynomial time. *BIT Numerical Mathematics.* **34**, 484-509 (1994,12)

30 Deogun, J., Kloks, T., Kratsch, D. & Müller, H. On the vertex ranking problem for trapezoid, circular-arc and other graphs. *Discrete Applied Mathematics.* **98**, 39-63 (1999)

31 Ceballos, C., Santos, F. & Ziegler, G. Many non-equivalent realizations of the associahedron. *Combinatorica.* **35**, 513-551 (2015,10)

32 Pournin, L. The asymptotic diameter of cyclohedra. *Israel Journal Of Mathematics.* **219**, 609-635 (2017,4)

33 Pournin, L. The diameter of associahedra. *Advances In Mathematics.* **259** pp. 13-42 (2014)

34 Forcey, S., Lauve, A. & Sottile, F. New Hopf Structures on Binary Trees. *Discrete Mathematics & Theoretical Computer Science.* (2009,1)

35 Carr, M. & Devadoss, S. Coxeter complexes and graph-associahedra. *Topology And Its Applications.* **153**, 2155-2168 (2006)

36 Jordan, C. Sur les assemblages de lignes. *Journal Für Die Reine Und Angewandte Mathematik.* **70** pp. 185-190 (1869)

37 Demaine, E., Harmon, D., Iacono, J. & Pătraşcu, M. undefined.

38 Sleator, D. & Tarjan, R. Self-Adjusting Binary Search Trees. *J. ACM.* **32**, 652-686 (1985)

39 Nešetřil, J. & Mendez, P. Sparsity: Graphs, Structures, and Algorithms. (Springer,2012)

40 Katchalski, M., McCuaig, W. & Seager, S. Ordered colourings. *Discrete Mathematics.* **142**, 141 – 154 (1995)

41 Even, G. & Smorodinsky, S. Hitting sets online and unique-max coloring. *Discrete Applied Mathematics.* **178** pp. 71-82 (2014)

42 Postnikov, A., Reiner, V. & Williams, L. Faces of generalized permutohedra. *Documenta Mathematica.* **13** pp. 207-273 (2008)

43 Postnikov, A. Permutohedra, Associahedra, and Beyond. *International Mathematics Research Notices.* **2009**, 1026-1106 (2009)

44 Manneville, T. & Pilaud, V. Graph properties of graph associahedra. *Séminaire Lotharingien De Combinatoire.* **B73d** (2015)

45 Bixby, R., Cunningham, W. & Topkis, D. The Partial Order of a Polymatroid Extreme Point. *Mathematics Of Operations Research.* **10**, 367-378 (1985)

46 Diestel, R. Graph Theory. (Springer,2005)

47 Farber, M. & Jamison, R. Convexity in Graphs and Hypergraphs. *SIAM Journal On Algebraic Discrete Methods.* **7**, 433-444 (1986)

48 Bodlaender, H., Deogun, J., Jansen, K., Kloks, T., Kratsch, D., Müller, H. & Tuza, Z. Rankings of Graphs. *SIAM J. Discrete Math..* **11**, 168-181 (1998)

49 Bodlaender, H., Gilbert, J., Hafsteinsson, H. & Kloks, T. Approximating Treewidth, Pathwidth, Frontsize, and Shortest Elimination Tree. *J. Algorithms.* **18**, 238-255 (1995)

50 Bose, P., Cardinal, J., Iacono, J., Koumoutsos, G. & Langerman, S. Competitive Online Search Trees on Trees. *Proceedings Of The Annual ACM-SIAM Symposium On Discrete Algorithms (SODA20).* (2020)

51 Schäffer, A. Optimal node ranking of trees in linear time. *Information Processing Letters.* **33**, 91-96 (1989)

52 Llewellyn, D., Tovey, C. & Trick, M. Local optimization on graphs. *Discrete Applied Mathematics.* **23**, 157-178 (1989)

53 Mozes, S., Onak, K. & Weimann, O. Finding an optimal tree searching strategy in linear time. *Proceedings Of The Nineteenth Annual ACM-SIAM Symposium On Discrete Algorithms, SODA 2008, San Francisco, California, USA, January 20-22, 2008.* pp. 1096-1105 (2008)
54 Onak, K. & Parys, P. Generalization of Binary Search: Searching in Trees and Forest-Like Partial Orders. *47th Annual IEEE Symposium On Foundations Of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings.* pp. 379-388 (2006)
55 Aichholzer, O., Demaine, E., Korman, M., Lynch, J., Lubiw, A., Masárová, Z., Rudoy, M., Williams, V. & Wein, N. Hardness of Token Swapping on Trees. *CoRR.* **abs/2103.06707** (2021), https://arxiv.org/abs/2103.06707
56 Pothen, A. The complexity of optimal elimination trees., Tech. Report CS-88-13, Pennsylvania State University
57 Karp, A. Bit Reversal on Uniprocessors. *SIAM Review.* **38**, 1-26 (1996)
58 Wilber, R. Lower Bounds for Accessing Binary Search Trees with Rotations. *SIAM J. Comput..* **18**, 56-67 (1989)
59 Demaine, E., Harmon, D., Iacono, J., Kane, D. & Patrascu, M. The geometry of binary search trees. *Proceedings Of The Twentieth Annual ACM-SIAM Symposium On Discrete Algorithms, SODA 2009, New York, NY, USA, January 4-6, 2009.* pp. 496-505 (2009)
60 Devadoss, S. A realization of graph associahedra. *Discrete Mathematics.* **309**, 271-276 (2009)
61 Dehornoy, P. On the rotation distance between binary trees. *Advances In Mathematics.* **223**, 1316-1355 (2010)
62 Emamjomeh-Zadeh, E., Kempe, D. & Singhal, V. Deterministic and probabilistic binary search in graphs. *Proceedings Of The 48th Annual ACM SIGACT Symposium On Theory Of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016.* pp. 519-532 (2016)

4.3 Morphing Planar Graph Drawings Through 3D

Kevin Buchin (Technische Universität Dortmund, Germany, k.a.buchin@tue.nl)
Will Evans (University of British Columbia, Canada, will@cs.ubc.ca)
Fabrizio Frati (Roma Tre University, Italy, fabrizio.frati@uniroma3.it)
Irina Kostitsyna (TU Eindhoven, Netherlands, i.kostitsyna@tue.nl)
Maarten Löffler (Utrecht University, Netherlands, m.loffler@uu.nl)
Tim Ophelders (Utrecht University & TU Eindhoven, Netherlands, t.a.e.ophelders@uu.nl)
Carola Wenk (Tulane University, U.S.A., cwenk@tulane.edu)
Alexander Wolff (Universität Würzburg, Germany)

A *morph* is a continuous transformation between two given drawings of the same graph. A morph is required to preserve specific topological and geometric properties of the input drawings. For example, if the input drawings are planar and straight-line, the morph is required to preserve such properties throughout the transformation. A morphing problem often assumes that the input drawings are "topologically equivalent", that is, they have the same "topological structure". For example, if the input drawings are planar, they are required to have the same rotation system (i.e., the same clockwise order of the edges incident to each vertex) and the same walk bounding the outer face; this condition is obviously necessary (and it turns out, also sufficient) for a planarity-preserving morph to exist between the given drawings. A *linear* morph is a morph in which vertices move along straight-line segments, from their initial to their final position, at uniform speed. A *piecewise-linear* morph consists of a sequence of linear morphs, each of which is called a *step* (see Figure 17).

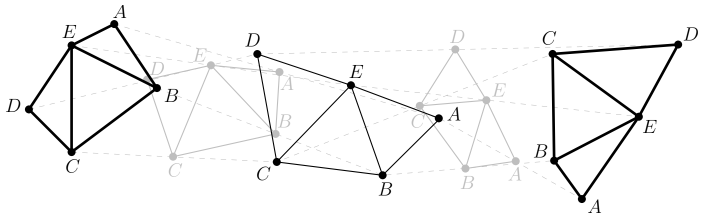


Figure 17 A piece-wise linear morph consisting of two steps.

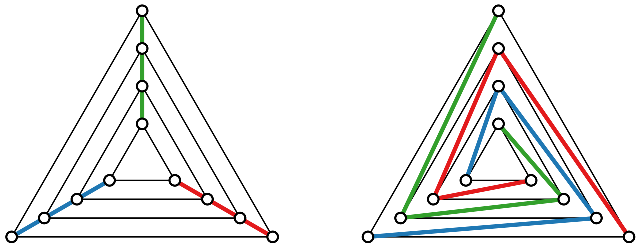


Figure 18 Two planar straight-line drawings of the same n-vertex planar graph that require $\Omega(n)$ steps to be morphed into each other (in 2D).

We focused on the design of algorithms and bounds for morphing two-dimensional graph drawings in the three-dimensional space by means of few morphing steps. It is well-known that, given any two (topologically-equivalent) planar straight-line drawings of the same n-vertex planar graph, there exists a morph with $O(n)$ steps that transforms one drawing into the other one. In the plane, this bound is worst-case optimal [1].

Is it possible to reduce the number of steps by allowing the morph to exploit a third dimension? This question was first posed and studied by Arseneva et al. [3]. They proved that, given any two planar straight-line drawings of the same n-vertex tree, there exists a crossing-free morph with $O(\log n)$ steps that transforms one drawing into the other one. Whether a crossing-free morph with $o(n)$ steps exists for any two planar straight-line drawings of the same n-vertex planar graph is the main question that we addressed.

A Lower Bound

We soon realized that a major challenge is how to construct a three-dimensional morph with $o(n)$ steps between the two planar straight-line drawings that provide the lower bound for two-dimensional morphs, shown in Figure 18. In fact, we worked towards a proof of the following conjecture.

▶ **Conjecture 2.** *Every three-dimensional crossing-free morph between the planar straight-line drawings shown in Figure 18 requires $\Omega(n)$ steps, where n is the number of vertices of the graph.*

We came up with the following tentative approach for a proof of Conjecture 2. Consider two triangles T_1 and T_2 lying on horizontal planes in 3D, where T_1 is above T_2. For $i = 1, 2$, let a_i, b_i, and c_i be the vertices of T_i. Suppose that a_1 is connected to a_2, b_1 is connected to b_2, and c_1 is connected to c_2 by means of three strings that spiral around each other $\Omega(n)$ times. This configuration can be reached with a single morphing step from the drawing in the right part of Figure 18: T_1 and T_2 are the outermost and innermost triangles, while the strings represent the colored paths. Then $\Omega(n)$ morphing steps seem to be necessary to "despiralize" the colored paths, that is, to morph the described geometric object to a configuration in which the colored paths are vertical. This configuration can be reached with a single morphing step from the drawing in the left part of Figure 18. Similarly to Alamdari et al. [1], our strategy to prove the claimed lower bound consists of considering a concept of "winding number", which is a measure of the described spiralization, and to prove that each morphing step changes the winding number only by a constant.

Conclusions

Although effective for tree drawings, the use of a third dimension does not seem to be helpful for morphing planar straight-line drawings with a sub-linear number of steps. On the other hand, we devised an approach that allows us to construct a morph between any two (possibly topologically non-equivalent) drawings of the same n-vertex planar graph with $O(n)$ steps. Such a morph does not exist when restricting to two dimensions. Immediate future work includes the formalization of this algorithm and the lower bound described above.

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, Bryan T. Wilkinson, *How to Morph Planar Graph Drawings*, SIAM J. Comput. 46(2):824–852, 2017.

2 Patrizio Angelini, Michael A. Bekos, Fabrizio Montecchiani, Maximilian Pfister, *On Morphing 1-Planar Drawings*, 47th International Workshop on Graph-Theoretic Concepts in Computer Science (WG '21): 270–282, 2021.

3 Elena Arseneva, Prosenjit Bose, Pilar Cano, Anthony D'Angelo, Vida Dujmović, Fabrizio Frati, Stefan Langerman, Alessandra Tappini, *Pole Dancing: 3D Morphs for Tree Drawings*, J. Graph Algorithms Appl. 23(3):579–602, 2019.

4 Giordano Da Lozzo, Giuseppe Di Battista, Fabrizio Frati, Maurizio Patrignani, Vincenzo Roselli, *Upward Planar Morphs*, Algorithmica 82(10): 2985–3017, 2020.

4.4 Statistics of Square Tiled Surfaces and the shearing block grame

Vincent Delecroix (Université de Bordeaux, France, vincent.delecroix@u-bordeaux.fr)
Elise Goujard (Université de Bordeaux, France, elise.goujard@u-bordeaux.fr)
Luke Jeffreys (University of Bristol, UK, luke.jeffreys@bristol.ac.uk)
Hugo Parlier (University of Luxembourg, Luxembourg, hugo.parlier@uni.lu)
Saul Schleimer (University of Warwick, UK, s.schleimer@warwick.ac.uk)

This working group focuses on some problems on square-tiled surfaces, as the shearing block game.

Discussed problems

▶ **Definition.** *A **square-tiled surface** is an oriented compact connected surface obtained by gluing a finite number of isometric squares along parallel sides by translation (right $\leftrightarrow$ left, up $\leftrightarrow$ down).*

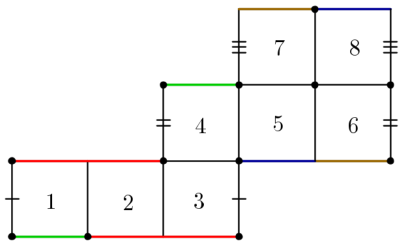


Figure 19 A square-tiled surface.

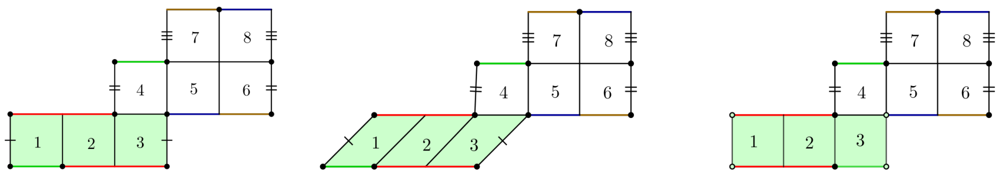


Figure 20 A shear on the bottom left horizontal cylinder.

A cylinder on a square-tiled surface (more generally a translation surface) is a maximal collection of regular parallel closed geodesics. Cylinders are bounded by conical singularities of the induced flat metric on the square-tiled surface. On the picture above, the surface has 3 horizontal cylinders, and 2 vertical ones.

The "shearing block game" was suggested by Saul Schleimer and inspired by the talk of Hugo Parlier on playing puzzles on square-tiled surfaces.

The game goes as follows: Starting from one pattern, is it possible to get to another given pattern by a series of shears on cylinder blocks?

The layout problem

This game raises a first question: can every square-tiled surface be laid out in the plane?

More precisely, a square-tiled surface made of n squares is conveniently encoded by a pair of permutations $(r, u) \in S_n \times S_n$ which record the gluing pattern: we glue to the right of the square labelled i the square $r(i)$ and above the square labelled i the square $u(i)$ It is also convenient to think of a square-tiled surface $(r, u) \in S_n \times S_n$ as an oriented 4-valent graph $G(r, u)$ where we have oriented edges for each pair $(i, r(i))$ and $(i, u(i))$ for $i \in \{1, 2, \ldots, n\}$

Given a square-tiled surface given by two permutations, how to draw it? We would ideally want to produce a layout in the plane such that

i. the layout is made of unit squares centered on $\mathbb{Z}^2$;
ii. all touching squares in the plane should be glued the same way in the surface; and
iii. the layout is connected.

We will call such a layout a hard layout. If we relax condition ii. by cutting slits into the layout so that some adjacent squares are not adjacent in the surface then we will call such a layout a soft layout. Some possible variations on the layout are:

- soft layout (allowing slits) – It is equivalent to the existence of a spanning tree of $G(r, u)$ which is isomorphic to a subgraph of $\mathbb{Z}^2$ where the r edges are horizontal and the u edges are vertical (see the right of Figure 21);

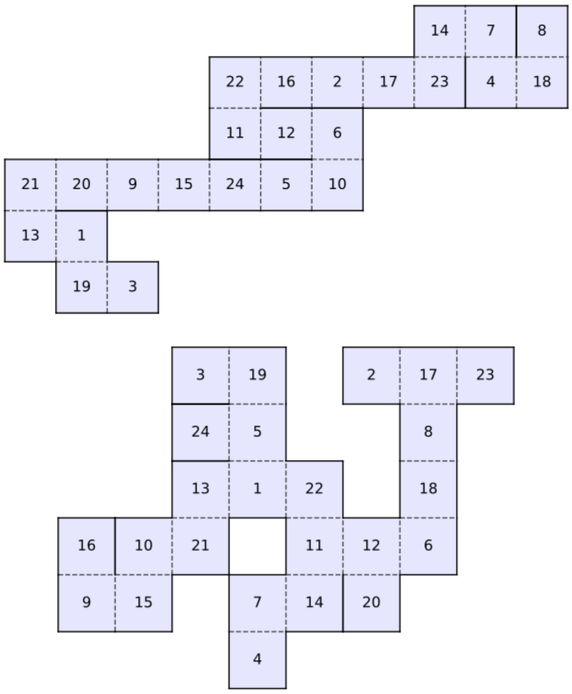


Figure 21 A snake layout (on the left, the extremities of the snake are the squares 3 and 8) , and a soft layout (on the right) of the surface

$$\begin{aligned} r &= (1, 22, 16, 2, 17, 23, 13)(3, 19)(4, 18)(5, 10, 21, 20, 9, 15, 24)(6, 11, 12)(7, 14) \\ u &= (1, 5, 19)(2, 23, 14, 11, 22, 4, 7, 20, 12, 15, 10, 6, 18, 8, 17, 9, 16)(13, 21). \end{aligned}$$

- hard layout (do not allow slits) – It is equivalent to the existence of a connected subgraph of $G(r, u)$ which is isomorphic to an induced subgraph of $\mathbb{Z}^2$ where the r edges are horizontal and the u edges are vertical (see the right of Figure 23);
- snake layout – Impose that the spanning tree is a path (in particular, the graph $G(r, u)$ admits a non-oriented Hamiltonian path [4]) (see the left of Figure 21);
- staircase layout (only move east or north) – In terms of permutations we want an enumeration $x_1, \ldots, x_n$ of $\{1, 2, \ldots, n\}$ so that two $x_{i+1} = r(x_i)$ or $x_{i+1} = u(x_i)$ It is equivalent to an oriented Hamiltonian path in the graph.

Here is an example lying in $\mathcal{H}(2, 2)^{odd}$ of a surface with no staircase nor snake layout (see Figure 22). The structure of the digraph associated to this square-tiled surface makes it clear that no staircase layout would be possible.

Some surfaces admit a snake layout but no staircase layout, see Figure 23.

There exist examples of square-tiled surfaces (r, u) with no soft layout. For instance the following square-tiled-surface made of 25 squares has no soft layout:

$$\begin{aligned} r &= (1, 6, 11, 16, 21)(2, 3, 4, 5)(7, 8, 9, 10)(12, 13, 14, 15)(17, 18, 19, 20)(22, 23, 24, 25) \\ u &= (1, 2)(6, 7)(11, 12)(16, 17)(21, 22) \end{aligned}$$

It can be used as a "gadget" to be plugged in other surfaces but it is not a "generic" gadget as it imposes too many fixed points.

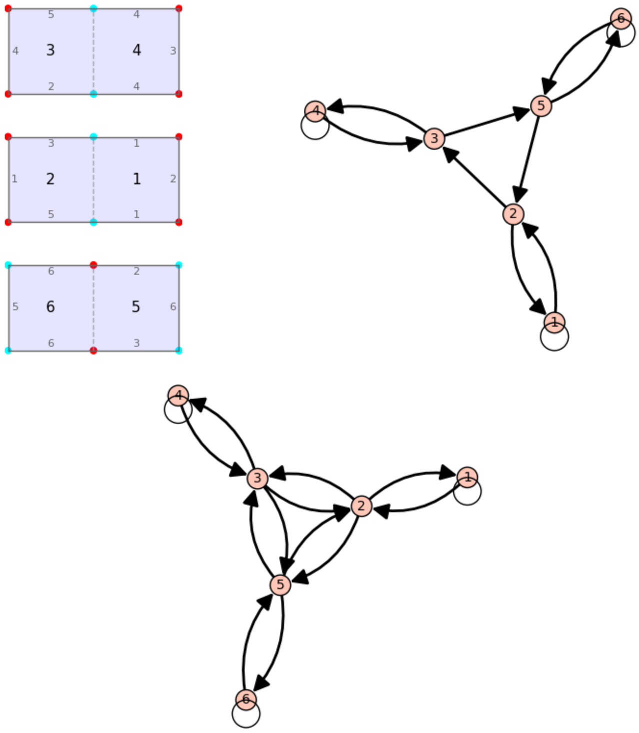


Figure 22 The square-tiled surface and its associated digraphs (staircase and snake) for the permutations

$$
\begin{aligned}
r &= (1,2)(3,4)(5,6)\\
u &= (1)(2,3,5)(4)(6).
\end{aligned}
$$

Luke and Vincent coded up some methods that search for and, if one exists, then draws a staircase/snake/soft layout representation for a square-tiled surface. Luke's method makes use of the `hamiltonian_path` method available inside Sage to find snake and staircase layouts, while Vincent's method for finding soft and snake layouts frames the problem as an integer linear programming problem and utilises Sage's `MixedIntegerLinearProgram` method. The code and some examples can be found in the files `https://coauthor.csail.mit.edu/file/ZwgDps62nDPygpfBW` and `https://raw.githubusercontent.com/videlec/Dagstuhl2022/master/Problem8-SquareTiledSurfaces/layout.py`.

The structure that exists in the digraphs of the counter examples suggests a choice of "gadget" to include in higher complexity square-tiled surfaces.

▶ **Definition.** *Let call a bridge a pair of edges e, e' such that*

- *e goes from u to v*
- *e' goes from v to u*
- *any path from u to v has to go through e*
- *any path from v to u has to go through e'.*

▶ **Conjecture 1.** *Consider the following quotient graph. Its vertices are the vertices adjacent to bridges in the initial graph and we put an edge between u and v if either*

- *there is a bridge between u and v*

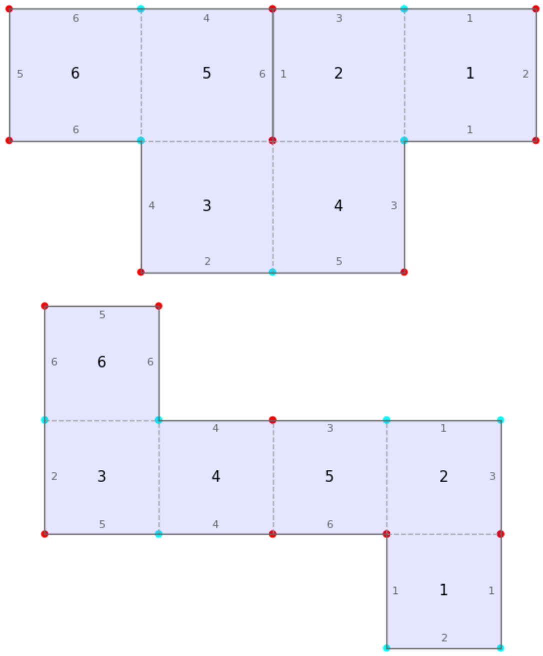


Figure 23 Snake layouts for the surfaces

$$\begin{array}{llll} r & = (1,2)(3,4)(5,6) & r & = (1)(2,3,4,5)(6) \\ u & = (1)(2,3,5,4)(6) & u & = (1,2)(3,6,5)(4). \end{array}$$

Both surfaces do not admit staircase layouts (although the layout on the right would be an east-south variation of a staircase layout).

- *there is an oriented path between u and v that does not go through any bridge*

Then a Hamiltonian path of the initial graph induces a Hamiltonian path on this quotient graph.

The general idea was that these bridge structures that appear in the counter-example can be used to build many more. However, one might expect that bridges are unlikely to be very common in general. In any case, we expect that having a Hamiltonian path in the digraph is not likely to be very common.

The experiments of this section lead to the following questions and conjectures:

Fix n and choose (uniformly at random) a pair of permutations r and u from the symmetric group S_n. You could also add conditions, say on the number of fixed points of the commutator $rur^{-1}u^{-1}$ forcing the surface to be more flat/lower genus. This gives a "generic" square-tiled surface $S = S(r, u)$.

▶ **Question 1.** *As $n \to \infty$ does a random pair $(r, u) \in S_n \times S_n$ have a soft layout, a staircase layout, or (stronger still) a staircase layout starting from any of its squares?*

▶ **Conjecture 2.** *In that regime, the probability that a square-tiled surface has a soft/snake layout tends to 1, whereas the probability that a square-tiled surfaces has a positive staircase layout tends to 0.*

The intuition behind these conjectures is that generically r and u have very few short cycles, and these short cycles tend to be separated. This means that it is very hard to build a "gadget" (as the example with 25 squares). Here are some ideas for producing a (soft) snake layout of generic S: We find all of the horizontal cylinders, sort them by length, layout the biggest cylinder, attach all of the medium and small cylinders to it (underneath) and then perform some kind of snake layout with the remaining long cylinders. If this works, it may help promoting from the soft layout to the hard.

Also, generically the big cycle of say r is too big, so too many cycles (if u) should meet it, making it impossible to get a positive staircase layout.

▶ **Conjecture 3.** *Deciding whether a given pair $(r, u) \in S_n \times S_n$ has a soft/hard/staircase-/snake layout is NP-complete.*

▶ **Question 2.** *For a random square-tiled surface and a random spanning tree of that square-tiled surface, what is the area of the obtained (possibly overlapping) layout? What is the diameter of the layout? What is the diameter of the spanning tree? What proportion of the area corresponds to the overlapping part of the layout?*

The shearing block game

The first question raised for the shearing block game was the classification of the orbits for the shearing block moves. Obviously it does not change the total number of squares, nor the stratum (characterized by the order of conical singularities of the flat metric) the surface belongs to. One can check that it also doesn't change the connected component of this stratum.

▶ **Conjecture 4.** *The following invariants classify orbits of the shearing block game:*
- *connected component of stratum (Kontsevich-Zorich)*
- *number of squares n*

This conjecture was checked by computer experiments (see code at `https://github.com/videlec/Dagstuhl2022`) in the following cases:
- genus 2:
  - $\mathcal{H}(2)^{hyp}$ up to 30 squares
  - $\mathcal{H}(1,1)^{hyp}$ up to 18 squares
- genus 3:
  - $\mathcal{H}(4)^{hyp}$ up to 12 squares
  - $\mathcal{H}(4)^{odd}$ up to 10 squares
  - $\mathcal{H}(3,1)^{c}$ up to 10 squares
  - $\mathcal{H}(2,2)^{hyp}$ up to 10 squares
  - $\mathcal{H}(2,2)^{odd}$ up to 10 squares
  - $\mathcal{H}(2,1,1)^{c}$ up to 10 squares
  - $\mathcal{H}(1,1,1,1)^{c}$ up to 10 squares

If the conjecture is correct, the size of each graph (connecting the different configurations of a game) is computable in the following way. For each stratum component $\mathcal{C}$ there exists a quasi-modular form whose nth coefficient is the number of square-tiled surfaces in $\mathcal{C}$ made of n squares.

Experimental data for the diameter could also be obtained from surface-dynamics, but first, let us define formally the graph associated to the game. An edge could represent several possible moves:

1) one shear in one cylinder ($1/\ell$ fractional Dehn twist where l is the perimeter of the cylinder)
2) any number of shears in one cylinder
3) any number of shears in any number of (parallel) cylinders
4) 90 degree rotation

Numerical experiments give the following sizes and diameters for the graphs with respect to the choices 1) or 4) for the edges:

n	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
size	3	9	27	45	90	135	201	297	405	525	693	918	1062	1395	1620	2043	2295	2835	3120	3915	4158	5085	5337
diam.	2	4	13	12	15	13	18	17	19	18	19	19	21	20	21	21	22	21	23	22	24	24	23

Furthermore, we can consider various metrics on these graphs. The initial metric considered was the natural metric for "slow moves": an edge representing a single shear of size one (right or left) in a single cylinder as length 1. There are various suggestions of other costs:

- $\log(k+1)$ for a shear of size k in a single cylinder,
- $\log(k+1)$ for a shear of size k on a "stack" horizontal cylinders,
- $\log(k+1)$ for a $k$fold shear in any collection of horizontal cylinders
- 1 for any horizontal shear in any collection of horizontal cylinders...

As we change the metric, the diameter of the graph changes. This raises the following question:

▶ **Question 3.** *Is there a (natural) definition of a "single move" and a corresponding metric on the graph that allows to connect two vertices very quickly? Is it possible that two "large" moves suffice?*

Some leads considered to solve the conjecture were the following:

- one-cylinder surfaces are easy to play with (in one direction), and we know from [3] how many such surfaces we have in a given stratum.
- the $\mathrm{SL}(2,\mathbb{Z})$ orbit closures are connected via the shearing block moves.

This raises the following question:

▶ **Question 4.** *How do* $\mathrm{SL}(2,\mathbb{Z})$ *orbit closures lie inside of the graph? How do they "approach" each other?*

Here are further questions concerning this game played on random surfaces:

▶ **Question 5.** *Consider a random square-tiled surface S in the same setting as before.*

- *What is the size of the component (of the shearing block game) containing a generic S? (Conjecture: $(N!)^2/N^{\log(n)}$)*
- *What is the diameter of the component? Can we navigate?*
- *Suppose that we perform random moves to S to get the sequence of surfaces (S_k) with associated permutations r_k and u_k Note that the cycle type of the commutator $r_k u_k R_k U_k$ is fixed. However, we can still ask: does r_k converge to the generic permutation as k tends to infinity?*
- *How quickly does the random walk (on the component) mix?*

An application – if the graph is connected, is this a good way to sample square-tiled surfaces?

References

1 Schloss Dagstuhl – Editorial Office,*The dagrep-v2018 class*. Schloss Dagstuhl, Germany, 2011.
2 John Q. Open; Joan R. Access, *Seminar Sample*, Dagstuhl Reports, 1:1, 1–8, 2011.
3 V. Delecroix, E. Goujard, P. Zograf, A. Zorich, *Contribution of one-cylinder square-tiled surfaces to Masur-Veech volumes. With an appendix by Philip Engel.* Astérisque No. 415, Quelques aspects de la théorie des systèmes dynamiques: un hommage à Jean-Christophe Yoccoz. I (2020), 223–274. `https://arxiv.org/abs/1903.10904`
4 `https://www.cs.utoronto.ca/~gbrunet/project2.pdf`

4.5 The graph of minimal configurations of a multicurve

Mark Bell (Hampshire, UK)
Benjamin Burton (The University of Queensland – Brisbane, Australia, bab@maths.uq.edu.au)
Erin Moriarty Wolf Chambers (St. Louis University, US, erin.chambers@slu.edu)
Vincent Delecroix (University of Bordeaux, France, vincent.delecroix@labri.fr)
Niloufar Fuladi (University Paris-Est – Marne-la-Vallée, France, niloufar.fuladi@aol.com)
Luke Jeffreys (University of Bristol, UK, luke.jeffreys@bristol.ac.uk)
Francis Lazarus (CNRS – Grenoble, France, Francis.Lazarus@grenoble-inp.fr)
Arnaud de Mesmay (University Paris-Est – Marne-la-Vallée, France, ademesmay@gmail.com)
Hugo Parlier (University of Luxembourg, Luxembourg, hugo.parlier@uni.lu)
Saul Schleimer (University of Warwick – Coventry, UK, s.schleimer@warwick.ac.uk)
Éric Colin de Verdière (CNRS & LIGM – Marne-la-Vallée, France, ericcolindeverdiere@u-pem.fr)
Hsien-Chih Chang (Dartmouth College – Hanover, US, hsien-chih.chang@dartmouth.edu)
Jean Cardinal (ULB – Brussels, Belgium, jean.cardinal@ulb.be)

This working group focused on the graph of minimal configurations of a multicurve.

Discussed Problem

Consider a set of curves (a.k.a a *multicurve*) Γ on a surface S. Here, we consider curves up to continuous deformations (free homotopy), so that a curve is actually a conjugacy class in $\pi_1(S)$. We shall only consider **primitive** curves, that is curves that are not proper powers of other curves, moreover pairwise distinct in Γ. A **configuration** of Γ is a choice of a representative of each curve in Γ so that all their intersections on S are transverse and there is no triple points. A configuration is considered up to isotopy. A configuration is **minimal** if its number of double points is minimal among all possible configurations of Γ.

It is known (see e.g. [10]) that any configuration of Γ can be brought to a minimal one by elementary moves, a.k.a. shadows of Reidemeister moves. In fact, its was shown that any two configurations of Γ are related by a *monotonic* sequence of moves, where the number double points changes monotonically [3, 13, 2]. In particular, any two minimal configurations of Γ are related by a sequence of 3-3 moves. The **configuration graph** of Γ has for vertex set the set of minimal configurations of Γ and two vertices are connected by an edge if they are related by a single 3-3 move. The previous remark ensures that this graph is connected. It follows from Hass and Scott [11] that this graph is finite. There are two natural questions related to this graph:

- What is its size?
- What is its diameter?

Following Souto and Vo [14], we can reduce to the case where Γ is *filling*, i.e. such that each of its configurations cuts S into topological disks and annuli bounded on one side by a boundary component of S. Indeed, after putting Γ into minimal configuration [2], one can construct a subsurface $S(\Gamma) \subset S$ where Γ is filling. $S(\Gamma)$ is obtained by replacing every component C of the complement $S \setminus \Gamma$ that is not a disk by a set of annuli, with one annulus per boundary component of C that is not a boundary component of S. (Formally, one needs to replace C by its metric completion in order to speak of its boundary components). We call such an annulus a **connecting annulus**. The fact that all minimal configurations are related by 3-3 moves implies that the topology of $S(\Gamma)$ is independent of the minimal configuration used to cut S. Now, two multicurves Γ and Γ' have the same type in S (see below for a definition) if and only if there is a homeomorphism $\varphi : S(\Gamma) \to S(\Gamma')$ sending Γ to Γ' (up to homotopy) such that φ can be extended to a self-homeomorphism of S. This last condition can be checked easily as follows: Consider the graph G_Γ whose vertices are the connecting annuli in $S(\Gamma)$ and whose edges correspond to connecting annuli bounding a same component of $S \setminus S(\Gamma)$. We moreover mark every vertex of G_Γ with the topological type of the incident component in $S \setminus S(\Gamma)$ (its genus and its total number of boundary components). Then φ extends to a self-homeomorphism of S if an only if it induces an isomorphism between the marked graphs G_Γ and $G_{\Gamma'}$.

We can thus assume that Γ is filling. In this case any configuration determines a combinatorial map in the obvious way: the vertex and edges of this map are simply given by the arrangement of the curves in the configuration, and the faces of the map correspond to the complementary disks and annuli (one per boundary component of S). These maps are 4-regular as we assume that there is no triple point in a configuration. We can replace the minimal configurations in the configuration graph by their associated combinatorial maps. Note that applying a self-homeomorphism of S to a configuration does not change (the isomorphism class of) its associated combinatorial map. Also note that the configuration graph is entirely determined by *any* of its configuration maps, from which we can recover all the other configurations by 3-3 moves. Say that two multicurves have the same **type** if they are in the same orbit of the mapping class group. More formally, this means that there is a 1-1 correspondence between the two sets of free homotopy classes of curves in the two multicurves that is induced by some self-homeomorphism of S. Hence, two filling multicurves have the same type if and only if they have the same configuration graph.

Recently, Souto and Vo [14] described a polynomial time algorithm to detect when two curves have the same type. Using normal coordinates with respect to a fixed triangulation, their algorithm enumerates a polynomial number of candidate mapping classes that must contain a mapping class sending one curve to the other one in case they indeed have the same type. It remains to check whether any candidate mapping class relates the input curves to decide if they have the same type. Our study of the graph of minimal configurations is motivated by an alternative approach to the algorithm of Souto and Vo.

Looking for hyperbolic configurations

Suppose that our surface S is equipped with some Riemannian metric. Freedman et al. [8] proved that shortening a given curve as much as possible for this metric puts the curve in minimal configuration[1]. Conversely, Neumann-Coto [12] proved that every minimal configuration of a multicurve is in the configuration of shortest geodesics for *some* Riemaniann metric. It thus seems natural to encode a configuration by a metric for which it is a minimizer. Hyperbolic metrics provide an interesting subset of metrics. We say that a configuration of a multicurve is **hyperbolic stretchable** or, more simply, **hyperbolic**, if it can be realized by homotopic geodesics for some hyperbolic metric. Hass and Scott [11] gave counterexamples to the fact that every multicurve configuration is hyperbolic. It follows that the graph of hyperbolic configurations is in general smaller that the whole graph of configurations. The consideration of hyperbolic configurations raises several questions.

[1] In full generality, one should allow the minimal configurations to have crossings with multiplicity larger than two. A crossing with multiplicity k should count for $\binom{k}{2}$ simple crossings.

1. One already serves as a conclusion in [4]: Given a multicurve, is there an algorithm to (construct or) detect configurations that are hyperbolic?
2. Is the graph of hyperbolic configurations of a multicurve connected?
3. How big is the graph of hyperbolic configurations compared to the whole graph of configurations? What is its diameter?

The answer to the second question should be positive: simply interpolate hyperbolic metrics associated to two configurations in the Teichmüller space of S. The interpolation path should be generic in the sense that singular metrics (for which the multicurve has crossings of multiplicity larger than two) along this path should be isolated and should allow for only one non-regular crossing of multiplicity exactly three (this requires an argument to claim that this is indeed the generic situation.). Then, the regular configurations along this path are related by 3-3 moves, showing the connectivity of the graph of hyperbolic configurations.

We have not studied the third question very much. What should be clear is that the whole configuration graph can be very big with respect to the number of crossings. As an indication one may consider the number $\Omega(2^{n^2/5})$ of pseudoline arrangements [5], where the number of crossings is quadratic with respect to the number n of pseudolines.

We now turn to the first question. Unfortunately, determining whether a configuration is hyperbolic stretchable seems quite hard. We actually prove.

▶ **Proposition 2.** *Given a combinatorial map representing a configuration of a multicurve on a surface, it is $\exists\mathbb{R}$-hard to decide if the configuration is hyperbolic.*

Recall that a problem is $\exists\mathbb{R}$-hard if the Existential Theory of the Reals (ETR) many-one reduces to it in polynomial-time, and that the problems in ETR are emptiness of semi-algebraic systems.

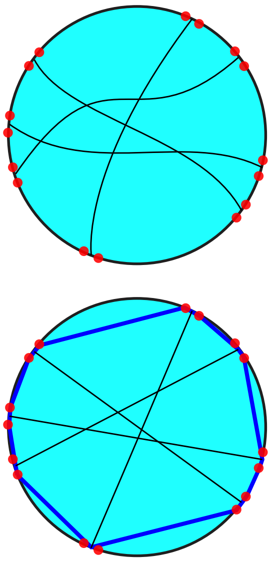
Proof. The proof goes by showing that the stretchability of a pseudoline arrangement in the plane reduces in polynomial time to the hyperbolic stretchability of a multicurve configuration. The proposition then follows from Mnëv's universality theorem implying that the problem of stretchability of pseudoline arrangements is polynomially equivalent to ETR. See [7] or a quick overview in [6] for precise statements. Recall that a pseudoline arrangement is a collection of simple arcs in a disk such that the arc endpoints are on the disk boundary and such that any two arcs intersect exactly once and transversely.

We give two reductions, transforming an arrangement of n pseudolines into a configuration of n curves either on a sphere with $4n$ punctures or on a closed surface of genus $2n$.

Reduction to a configuration on a punctured sphere. Consider an instance I_1 of stretchability of an arrangement of n pseudolines. We build an instance I_2 of hyperbolic stretchability of a multicurve configuration such that I_1 is positive iff I_2 is.

Let D be the disk containing the instance I_1. We introduce $4n$ punctures on the boundary of D in pairs; each pair surrounds one of the $2n$ endpoints of the n pseudolines as on the next figure. We then take a copy D' of D and attach them together: Think of D as the Northern hemisphere, D' as the Southern hemisphere, the boundary of the disks to be the equator. The punctures coming from the disks are identified so that there are $4n$ punctures in total. Each pseudoline is now a closed curve, passing once in the Northern and once in the Southern hemispheres. The reduction takes trivially polynomial time.

Assume that I_1 is positive. We have an arrangement of lines in the Euclidean disk. We put the $4n$ punctures on the boundary of the disk, and then consider the $4n$-gon whose vertices are the punctures (so we cut off some small pieces of the disk). Now we view this



$4n$-gon as an ideal (hyperbolic) polygon induced by the Klein model of the disk. We take a copy of this ideal polygon and attach them side by side. This gives a hyperbolic metric on the $4n$-punctured sphere where the arrangement of the closed curves in I_2 is realized. (This is because, when we cut off the disk into a $4n$-gon, no intersection between pseudolines has been removed.) We then relax these closed curves to geodesics in their respective homotopy classes. If the two punctures surrounding an endpoint of a pseudoline were close enough together, this doesn't change the combinatorial arrangement of the curves, because the initial pseudoline arrangement was in general position. So I_2 is positive. (We know that we need doubly exponential precision for the location of the punctures [9], but we don't care: after all we just consider decision problems!)

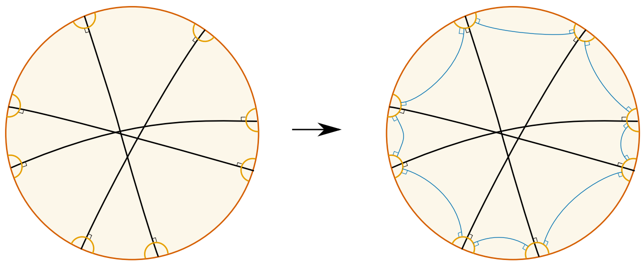
Conversely, assume that I_2 is positive. Consider the corresponding hyperbolic surface with cusps, S. In S, consider a topological disk D that contains I_1 as a topological arrangement. Consider the Klein model K of the universal cover of S. Lift D (and the pieces of the lines inside D) in K. The picture in K is a topological disk $\tilde{D}$ and line segments $\tilde{D}$ whose combinatorial arrangement is exactly the input to I_1. Extend these lines to straight line segments with endpoints on the boundary of $\tilde{D}$; because, in $\tilde{D}$, every pair of segments already cross (since I_1 is a pseudoline arrangement), this extension does not create new crossings. In other words, if we now view $\tilde{D}$ as a Euclidean disk, our arrangement is now combinatorially equivalent to I_1. So I_1 is positive.

Reduction to a configuration on a closed surface. As before we start with an instance I_1 of an arrangement of n pseudolines $\ell_1, \dots, \ell_n$ in a disk D. As before, we introduce $4n$ punctures on the boundary ∂D of D surrounding the $2n$ pseudoline endpoints. This divides ∂D into $4n$ arcs, where one out of every two contains an endpoint. We denote by α_i the arcs that contains an enpoint and by β_i the remaining ones. We then take a copy D' of D and glue their arcs α_i via the identity map. This results in a sphere with $2n$ boundaries. We cap off each of these boundaries with a one-holed torus to obtain a configuration of n closed curves on a closed surface of genus $2n$. This ends the construction of our instance I_2 for hyperbolic stretchability[2]. The construction trivially takes polynomial time.

Assume that I_1 is positive. We have an arrangement of L lines in the Euclidean disk. Considering the Klein model of the disk this provides an isomorphic arrangement of hyperbolic lines that we still denote by $\ell_1, \ldots, \ell_n$. Let α_i be a collection of geodesics as follows.

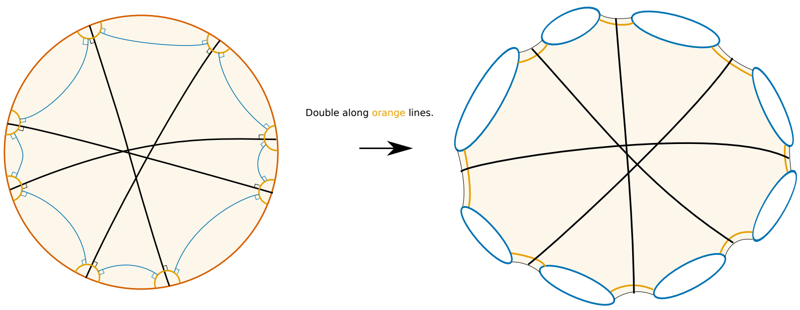
1. If $i \neq j$ then α_i is disjoint from α_j (even at infinity – they do not share an ideal point).
2. α_i and ℓ_i meet in a single point.
3. α_i cuts a half plane H_i off of D; the intersection of H_i with the line arrangement L is exactly one end of ℓ_i.

Let β_i be the common perpendicular to α_i and α_{i+1}.



Claim: β_i is disjoint from L.

We now cut D along the α_i and β_i to obtain a hyperbolic right-angled $4n$-gon. We double this $4n$-gon and glue the two copies along the α_i. We obtain a hyperbolic sphere with $2n$ geodesic boundaries as shown below.

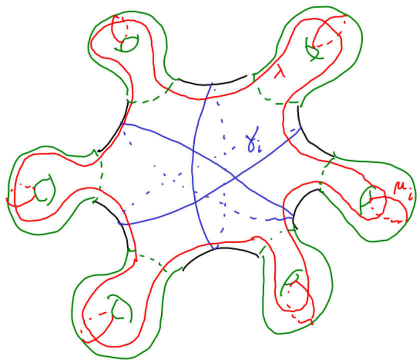


We finally cap off the $2n$ geodesic boundaries with hyperbolic one holed tori. We denote by S the resulting hyperbolic surface. Let γ_i be the union of ℓ_i and its copy; this is a closed geodesic on S. Clearly, S and the γ_i certify that I_2 is positive.

The reverse implication goes exactly as in the previous reduction on a punctured sphere. This ends the proof of the proposition.

We can slightly strengthen the proposition by restricting the hardness result to filling configurations. In order to do so we just need to add $2n+1$ curves to the γ_i in the previous construction. One of these curves, λ, goes through all the handles as on the next figure. And the $2n$ other curves, μ_i, cut open each handle. The μ_i are disjoint from the γ_i and will thus be so in any minimal configuration. This allows us to recover the extra $2n+1$ curves without the need for a marking. ◀

[2] An alternative construction consists of taking four copies of D, gluing (cyclically) the jth and $(j+1)$th copies along the α_i or along the β_i according to the parity of j. The gives a surface of genus $2n-1$ with a configuration of $2n$ (?) curves.



In search of a canonical configuration

In view of the hardness result in Proposition 2, the restriction of the configuration graph to hyperbolic configurations might not be so beneficial from the computational viewpoint. There is however a surprising benefit to consider hyperbolic configurations. They contain a preferred, i.e. canonical, configuration (at the expense of allowing degenerate configurations with crossings of multiplicity higher than two)!

Suppose that S is a (closed, connected, oriented) surface. Suppose that γ is a (simple?) closed curve in S. For any hyperbolic metric σ on S, we can define $\ell_\gamma(\sigma)$ to be the length of the geodesic representative of γ with respect to σ. Thus ℓ_γ is a function from Teichmuller space to $\mathbb{R}$. Wolpert proves that ℓ_γ is convex with respect to the Weil-Petersson metric [15] and Bestvina et al. with respect to some well chosen Fenchel-Nielsen coordinates [1].

Suppose now that (γ_i) is a filling collection of curves. Then there is a *unique* σ that minimises the sum $\sum_i \ell_{\gamma_i}$. The intersection pattern of the geodesic representatives of the γ_i, with respect to σ can therefore be considered a canonical form for the family γ_i. Note however that there might be points with more than double intersections in this canonical form.

We conclude with many questions on this minimising metric.

1. Is the minimising metric algebraic? How do the degree/height of the algebraic numbers depend on the complexity of the given curves (γ_i)?
2. Given a hyperbolic metric σ, can we find a collection of curves (γ_i) having σ as a minimiser? Written this way the answer is no (countable versus uncountable). We hence have two questions: – Is any hyperbolic metric σ the minimizer of some *weighted* collection of curves? Of simple curves? Of two simple curves? – What are the hyperbolic metrics σ which are minimizers of a collection of curves?
3. Is there such a minimiser in the flat setting? Note that the length function is convex in the $\mathrm{SL}(2, \mathbb{R})$ directions. However, it is hard to believe that it is convex along any linear deformation (in period coordinates).
4. What happens in the case of the torus (hyperbolic or flat)?
5. Is there a way to get a nice bound on the diameter of the configuration graph using this approach? A discrete algorithm to find the moves?
6. Perhaps Whitehead's algorithm is relevant here? We can use Whitehead to find, in polynomial time, an optimal cut system (cutting the surface into a connected planar surface). See the exposition of Berge; he describes how to use max-flow-min-cut to search for the Whitehead automorphisms.
7. There is a combinatorial proof of Nielsen realisation, I believe due to Hensel, Osajda, and Przytycki.

References

1 Mladen Bestvina, Ken Bromberg, Koji Fujiwara, and Juan Souto. Shearing coordinates and convexity of length functions on teichmüller space. *American journal of Mathematics*, pages 1449–1476, 2013.

2 Hsien-Chih Chang and Arnaud de Mesmay. Tightening curves on surfaces monotonically with applications. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 747–766. SIAM, 2020.

3 Maurits de Graaf and Alexander Schrijver. Making curves minimally crossing by Reidemeister moves. *Journal of Combinatorial Theory, Series B*, 70(1):134–156, 1997.

4 Vincent Despré and Francis Lazarus. Computing the geometric intersection number of curves. *Journal of the ACM (JACM)*, 66(6):1–49, 2019.

5 Adrian Dumitrescu and Ritankar Mandal. New lower bounds for the number of pseudoline arrangements. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 410–425. SIAM, 2019.

6 Jeff Erickson. Optimal curve straightening is $\exists\mathbb{R}$-complete.

7 Stefan Felsner and Jacob E Goodman. Pseudoline arrangements. *Handbook of Discrete and Computational Geometry*, 3, 2017.

8 Michael Freedman, Joel Hass, and Peter Scott. Closed geodesics on surfaces. *Bull. London Math. Soc.*, 14:385–391, 1982.

9 Jacob E Goodman, Richard Pollack, and Bernd Sturmfels. Coordinate representation of order types requires exponential storage. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 405–410, 1989.

10 Joel Hass and Peter Scott. Intersections of curves on surfaces. *Israel Journal of Mathematics*, 51(1-2):90–120, 1985.

11 Joel Hass and Peter Scott. Configurations of curves and geodesics on surfaces. *Geometry and Topology Monographs*, 2:201–213, 1999.

12 Max Neumann-Coto. A characterization of shortest geodesics on surfaces. *Algebraic and Geometric Topology*, 1:349–368, 2001.

13 J.M. Paterson. A combinatorial algorithm for immersed loops in surfaces. *Topology and its Applications*, 123(2):205–234, 2002.

14 Juan Souto and Thi Hanh Vo. Deciding when two curves are of the same type.

15 Scott A. Wolpert. Convexity of geodesic-length functions: a reprise.

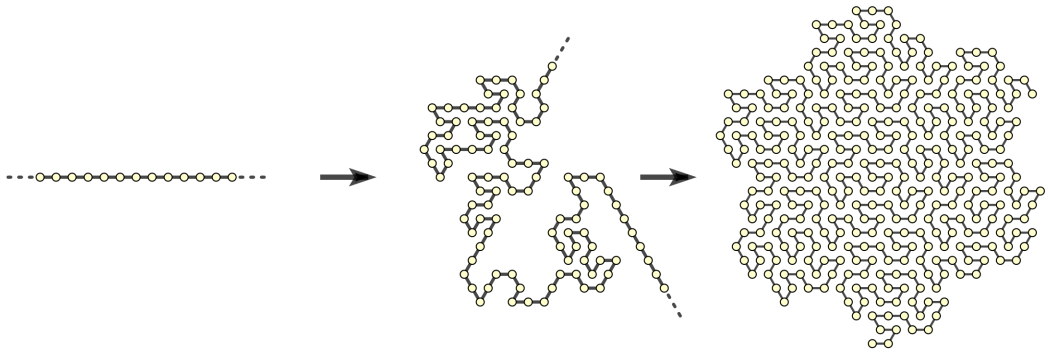


Figure 24 A turning machine folding into a Gosper curve.

4.6 Turning Machines

Kevin Buchin (Technische Universität Dortmund, Germany, k.a.buchin@tue.nl)
Jean Cardinal (Université Libre de Bruxelles, Belgium, jean.cardinal@ulb.be)
Will Evans (University of British Columbia, Canada, will@cs.ubc.ca)
Irina Kostitsyna (TU Eindhoven, the Netherlands, i.kostitsyna@tue.nl)
Alexander Wolff (Universität Würzburg, Germany)

The *turning machine* [1] is a very simple model of a molecular robot whose task is to fold into a desired shape. Such a machine consists of a number of molecular computing units (called *monomers*) that together form a chain. Depending on their states, the monomers can rotate, thus bending the chain and changing its global structure (see Figure 24). On his website [2], Woods gives an informal description of the model and has example videos where turning machines fold into certain shapes. During the workshop, we discussed how to assign initial states to the monomers such that they can then rotate and bring the chain into a prescribed shape.

Model. Consider a triangular grid $G_\triangle = (V_\triangle, E_\triangle)$. A turning machine T is a chain of monomers in $G_\triangle$. For simplicity, we introduce a system of coordinates on $G_\triangle$ (see Figure 25). Initially, the monomers of a turning machine form a horizontal chain $m_1, m_2, \dots, m_n$ that starts from the origin and extends East, that is, monomer m_i occupies grid point $(i-1, 0)$. We orient the edges of the chain from left to right. For $i \in \{1, 2, \dots, n-1\}$, monomer m_i has state s_i, which is an integer value that indicates how many times the edge leaving m_i needs to rotate.

A turning machine folds by the means of local rotations of monomers. In one step, a monomer m_i with a non-zero state s_i can rotate its outgoing edge by 60° – counterclockwise if $s_i > 0$ and clockwise if $s_i < 0$. This results in the suffix of the chain translating by a unit vector in the triangular grid. We require the chain to not self-intersect, so not all monomers can rotate at any moment in time. In other words, some monomers can be *blocked*. When a monomer rotates, the absolute value of its state decreases by one.

For example, the second monomer from the left in Figure 25(c) can rotate. As a result, the state of this monomer decreases from 3 to 2 and the subchain to the right of the monomer translates by the unit vector shown in the figure.

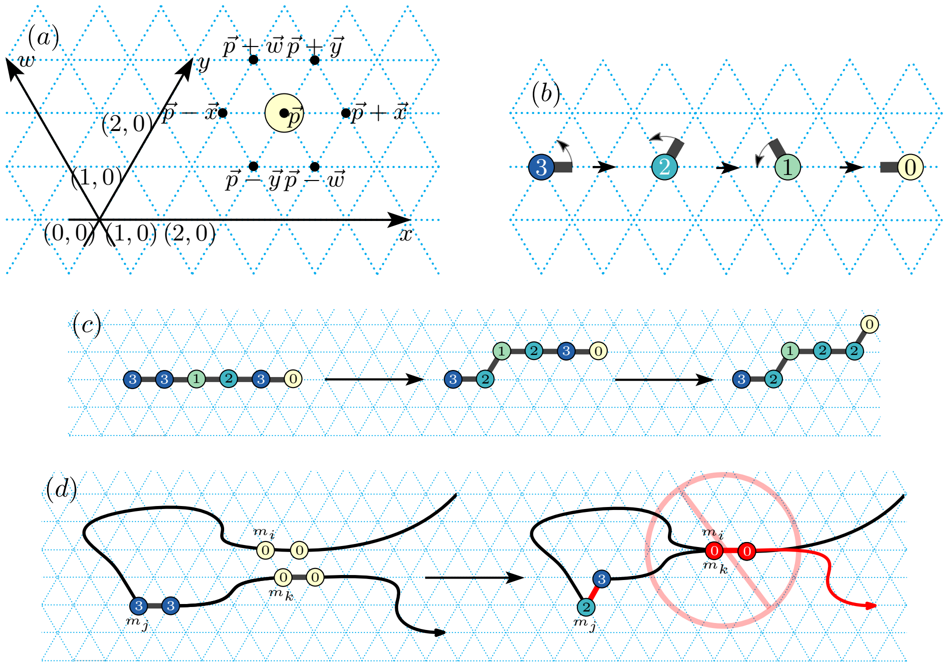


Figure 25 Figure taken from [1]. Turning machine model. (a) Triangular grid conventions. A configuration showing a single monomer on the triangular grid $G_\triangle$, along with axes x, y and w. (b) Local movement (rotation): A monomer in state 3 pointing to the east undergoes three turning rule applications finishing in state 0 with no more applicable rules. Locally, the monomer exhibits a rotation motion. (c) Global movement (translation): Left: Example initial configuration. Middle: A movement rule applied to the second monomer from the left. It decrements its state to 2 and translates all subsequent monomers by $+\vec{w}$. Right: Another state-3 monomer turns. (d) Blocking: Monomers are not permitted to make movements that would result in a self-intersecting configuration. The monomer m_j is *blocked*: if it were to move, then m_k would move to overlap with m_i.

A turning machine evolves as a continuous-time Markov chain with rotation rules applied asynchronously. This modeling assumption can be used to analyze the running time of the folding process. However, for the purposes of this report, we are only concerned with the order in which the rotation rules are applied. At any moment in time, any monomer with a non-zero state that is not blocked can be the next one to rotate. Therefore, when analyzing the folding process, we assume that the sequence of monomers to which the rules are applied is given by an adversary.

We say that a turning machine successfully folds if eventually all monomers have state 0. Some Turning Machines can evolve into a state where not all monomers have state 0 but no monomer can rotate without the chain self-intersecting. We call these states *permanently blocked*.

Open problems. Kostitsyna et al. [1] have studied the problem of shape formation by a turning machine. Given a desired target shape $S \subset V_\triangle$, specified by a connected subset of nodes of the triangular grid, we consider the problem of creating a turning machine that folds into S. Thus, the goal of the shape formation problem is to assign initial states to the monomers such that the turning machine *always* folds into a desired target shape S (up to a translation / rotation). The paper provides initial results on which classes of shapes can be folded by Turning Machines. It remains open to expand this class of shapes, or prove negative results.

We approached the problem of shape formation from a different angle. Given a turning machine T, we can easily identify the target shape that T encodes. Observe that the differences of the values of initial states of the monomers of T uniquely determine the angles of the chain in the folded state. However, it is challenging to decide whether a given turning machine T always reaches its final folded state, or even if T sometimes reaches its final folded state (i.e., there exists a sequence of rotation rule applications to the monomers that leads to T successfully folding). We know the answer to the question *Do all Turning Machines fold sometimes?*, which is no, there exist Turning Machines that can never be folded. We also know that the question whether a turning machine folds is decidable.

▶ **Observation 2.** *The state space of a turning machine with $n+1$ monomers is bounded by $W^n \cdot \text{poly}(n)$, where $W = \max_i\{s_i\} - \min_j\{s_j\}$ is the size of the range of the initial states of the monomers. Hence we can decide whether a turning machine always folds by exploring the (exponential-size) state space.*

We conjecture that both of the questions above are NP-hard. During the seminar we have made partial progress on the way to prove these conjectures.

▶ **Conjecture 3.** *Given a turning machine T, it is NP-hard to decide whether T always folds.*

▶ **Conjecture 4.** *Given a turning machine T, it is NP-hard to decide whether T sometimes folds.*

We also considered new tools that may help us to reason about Turning Machines. Below, we provide some observations that we have proven during the seminar. We finish the report with some open questions. We say that a turning machine is *non-negative* if all its monomers have non-negative states. Given two non-negative Turning Machines T and T', we say that T' *dominates* T if, for each $i \in \{1, 2, \ldots, n-1\}$, it holds that $s'_i \geq s_i$, where s_i and s'_i are the initial states of monomers m_i and m'_i of T and T', respectively. Here's our first tool.

▶ **Proposition 3.** *Let T and T' be two non-negative Turning Machines such that T' dominates T. If T' always folds, then T always folds as well.*

Note that the same argument does not apply if we have negative initial states. Rotating a monomer with a negative state (that is, rotating clockwise) in the dominated turning machine corresponds to rotating all the remaining monomers into the counterclockwise direction of the dominating turning machine. We can, however, modify the domination definition as follows.

▶ **Definition 1.** We say that a turning machine T' *dominates* a turning machine T if, for each $i \in \{1, 2, \ldots, n-1\}$, $s'_i \geq s_i$ if $s_i > 0$ and $s'_i \leq s_i$ if $s_i < 0$ (if $s_i = 0$ then s'_i can take any value).

One of the questions we would like to explore further is whether we can obtain a result similar to Proposition 3 for this new definition of domination. Can the property of domination help us expand the class of Turning Machines for which we can determine whether they always fold, sometimes fold, or never fold? Another question we leave for further investigation is whether there are other "domination" properties that can be formulated for sub- or super-chains that can help answer folding questions.

Conclusions. Kostitsyna et al. [1] initiated the study of the turning machines, a simple model for a molecular folding robot. The Dagstuhl Seminar provided us with a great opportunity to expand the range of research questions that can be studied within this model. We have obtained some preliminary results.

References

1 Irina Kostitsyna, Cai Wood, and Damien Woods, *Turning machines: a simple algorithmic model for molecular robotics*, Natural Computing (2022). doi: 10.1007/s11047-022-09880-8
2 Damien Woods, *Turning machines! New paper on reconfigurable robotics*, 2020. `https://dna.hamilton.ie/2020-09-16-turning-machines.html`. Accessed 2022-04-09.

4.7 Flips in Higher Order Delaunay trianlgulations

Anne Driemel (Universität Bonn, Germany, driemel@cs.uni.bonn.de)
Lena Schlipf (Universität Tübingen, Germany, schlipf@informatik.uni.tuebingen.de)
Rodrigo I. Silveira (Universitat Politècnica de Catalunya, Barcelona, rodrigo.silveira@upc.edu)
Jonathan Spreer (University of Sydney, Austrialia, jonathan.spreer@sydney.edu.au)

In this working group we consider a class of triangulations of points in $\mathbb{R}^2$ known as *higher Order Delaunay triangulations* (HODTs) [2]. Given a set of points S, and a parameter k, an *order-k (Delaunay) triangle* is a triangle whose circumcircle contains at most k points from S. An *Order-k (Delaunay) triangulation* is one where all triangles are order-k triangles.

The main topic studied is the flip graph of order-k triangulations. A basic aspect to understand is its connectivity. It is known that it is always connected for $k = 0, 1, 2$, and that it may be not connected for $k \geq 3$, even for points in convex position. Recently, a bit more was understood. In particular, in [1], the following was shown. Let $G(\mathcal{T}_k(S))$ be the flip graph of order-k triangulations of a point set S. Firstly, it was shown that for any k there exists a point set S in convex position where $G(\mathcal{T}_k(S))$ is disconnected. Moreover, $k-1$ flips are sometimes necessary in order to transform an order-k triangulation of S into another. Secondly, for any order-k triangulation of a point set in convex position there exists some other order-k triangulation at distance at most $k-1$ in $G(\mathcal{T}_{2k-2}(S))$. Finally, it was also shown that in case $k = 2, 3, 4, 5$, for any order-k triangulation of a point set in general position there exists an order-k triangulation at distance at most $k-1$ in $G(\mathcal{T}_{2k-2}(S))$. These results imply that the diameter of the flip graph is $O(kn)$.

Discussed Problems

With the long-term goal of understanding the flip graph of order-k HODTs, we focused on some simpler and more concrete questions.

In relation to computation, an interesting question is how fast one can compute the flip distance between two triangulations. Here there are two natural variants of the question: (i) one can go through any triangulation; (ii) you can only go through order-k triangulations. We did not make much progress on this front.

A more combinatorial aspect has to do with understanding the structure of fixed edges, those present in *any* order-k Delaunay triangulation. It is know that already for $k = 2$, the subdivision given by the set of fixed edges can produce polygons that contain holes. During the workshop we have found examples, also for $k = 2$, where holes can be nested. An example with several (non-nested) components for $k = 2$ was also found, but it contains many co-circular points. It is unclear whether the same can be achieved if points are in general position.

Conclusions

We have made some progress in understanding the structure of fixed edges for $k = 2$, but are still far from understanding its flip graph. Results on the flip graph can have direct application to algorithms, either exact or fixed-parameter tractable. For instance, a practical fast algorithm to compute flip distance for $k = 2$ would be very interesting.

References

1 Elena Arseneva, Prosenjit Bose, Pilar Cano, and Rodrigo I. Silveira. Flips in higher order Delaunay triangulations. In Yoshiharu Kohayakawa and Flávio Keidi Miyazawa, editors, *LATIN 2020: Theoretical Informatics – 14th Latin American Symposium, São Paulo, Brazil, January 5-8, 2021, Proceedings*, volume 12118 of *Lecture Notes in Computer Science*, pages 223–234. Springer, 2020.

2 Joachim Gudmundsson, Mikael Hammar, and Marc J. van Kreveld. Higher order Delaunay triangulations. *Comput. Geom.*, 23(1):85–98, 2002.

5 Open Problems

5.1 Representing Graphs by Polygon Contact in 3D

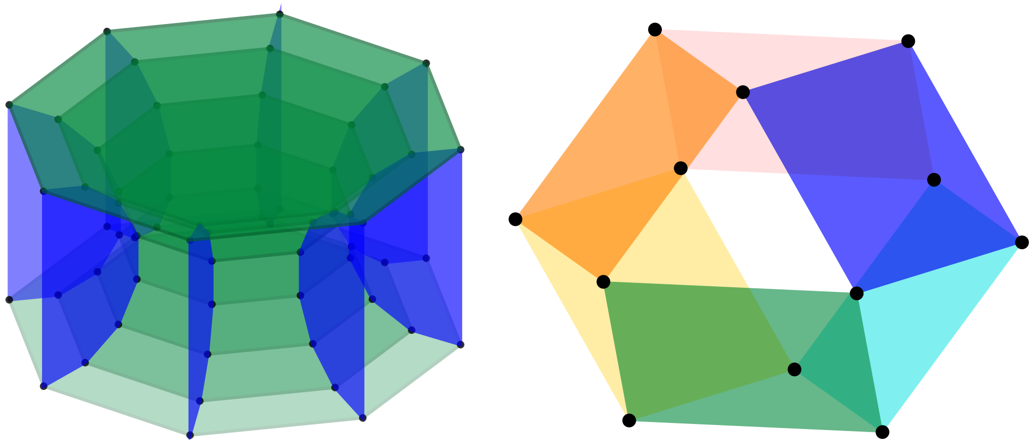
Will Evans (University of British Columbia, Canada, will@cs.ubc.ca)
Alexander Wolff (Universität Würzburg, Germany)

We are interested in representing graphs as contact graphs of *convex* polygons in 3D. Adjacency is represented by vertex contacts. Any two polygons must either be disjoint or they can share one vertex; see Figure 26. With others, we [1] showed that any graph admits such a contact representation, but for K_n we need volume $O(n^4 n!)$. Let $\mathrm{vol}(n, \Delta)$ be the maximum volume needed for representing any graph with n vertices and vertex degree at most Δ. We know that $\mathrm{vol}(n, 3) = O(n^3)$. The proof was quite tricky, and it seemed to depend a lot on the fact that triangles behave much more nicely than polygons of larger degree. So what is $\mathrm{vol}(n, 4)$? What about lower bounds for $\mathrm{vol}(n, \Delta)$?

We are also interested in how restrictions on other measures of the graph impact the volume required for its contact representation. In particular, limiting the rectilinear or book thickness of the graph, its k-planarity, or its clique number seem to have the potential to impact this volume.

References

1 W. Evans, P. Rzążewski, N. Saeedi, C.-S. Shin, and A. Wolff. Representing graphs and hypergraphs by touching polygons in 3D. In D. Archambault and C. D. Tóth, editors, *Proc. 27th Int. Symp. Graph Drawing & Network Vis. (GD'19)*, volume 11904 of *LNCS*, pages 18–32. Springer, 2019. `http://arxiv.org/abs/1908.08273`. URL: `https://doi.org/10.1007/978-3-030-35802-0_2`.



(a) the complete bipartite graph $K_{8,8}$. **(b)** the square of the 6-cycle, C_6^2.

Figure 26 Examples of contact representations of graphs using polygons in $\mathbb{R}^3$.

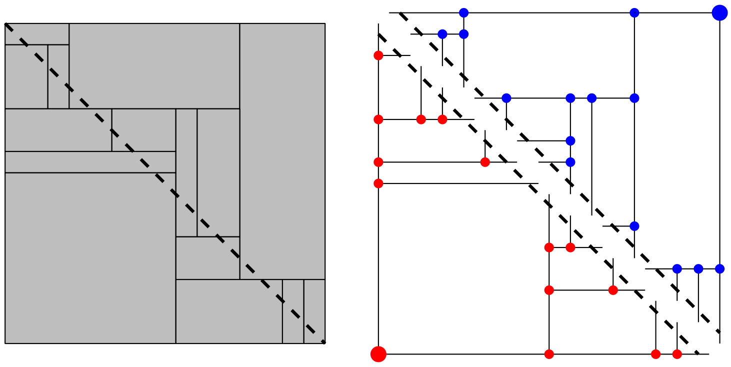


Figure 27 Diagonal rectangulation (left) and the corresponding pair of binary trees (right).

5.2 Diameter of rectangulation flip graph

Torsten Mütze (University of Warwick, UK, torsten.mutze@warwick.ac.uk)

A *diagonal rectangulation* is a partition of the unit square into finitely many interior-disjoint rectangles, such that no four rectangles meet in a point, and every rectangle intersects the SE-NW-diagonal, drawn dashed in Figure 27. The number of diagonal rectangulations with n rectangles is given by the Baxter numbers, and they are in bijection to $\{2\underline{41}3, 3\underline{14}2\}$-avoiding permutations, and to pairs of binary trees rooted in the NE and SW corners [2].

We equip the set of all diagonal rectangulations with n rectangles with a *flip* operation that either reverses the orientation of two rectangles whose union is a rectangle, or that rotates one of the arms of a T-join by 90°, which creates a flip graph G_n; see Figure 28. In this figure, the first type of flips induces the red edges, and the second type of flips induces the blue edges. These flips on rectangulations correspond to rotations in the aforementioned binary trees.

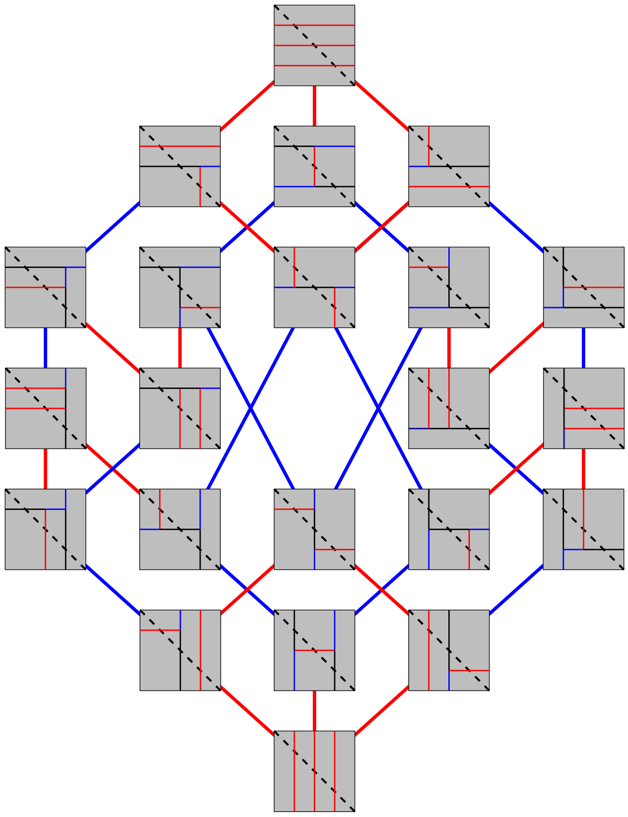


Figure 28 Flip graph G_n of diagonal rectangulations with $n = 4$ rectangles.

The graph G_n is known to be the cover graph of a lattice, and the skeleton of a polytope [3], very much analogous to the Tamari lattice and the associahedron for triangulations. In this project we are interested in the diameter of G_n, analogous to the famous diameter question for the associahedron [4].

Question: What is the diameter of G_n?

This problem was mentioned by Jean Cardinal at SoCG 2018, and so far only relatively little is known. Ackerman et al. [1] proved an upper bound of $11n$.

References

1 Ackerman, E., Allen, M. M., Barequet, G., Löffler, M., Mermelstein, J., Souvaine, D. L., and Tóth, C. D. The flip diameter of rectangulations and convex subdivisions. *Discrete Math. Theor. Comput. Sci. 18*, 3 (2016), Paper No. 4, 17.
2 Felsner, S., Fusy, E., Noy, M., and Orden, D. Bijections for Baxter families and related objects. *J. Combin. Theory Ser. A 118*, 3 (2011), 993–1020.
3 Law, S., and Reading, N. The Hopf algebra of diagonal rectangulations. *J. Combin. Theory Ser. A 119*, 3 (2012), 788–824.
4 Pournin, L. The diameter of associahedra. *Adv. Math. 259* (2014), 13–42.

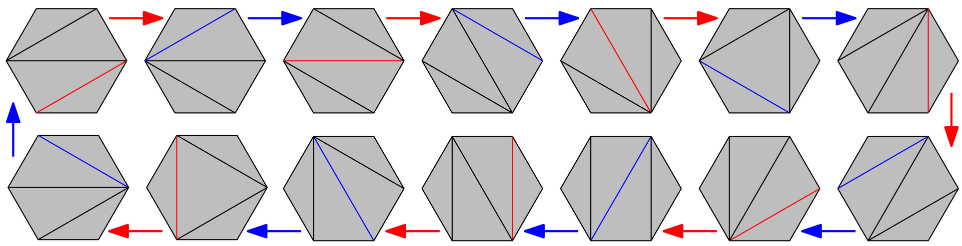


Figure 29 Gray code for all 14 triangulations of the 6-gon.

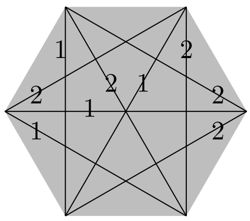


Figure 30 Number of appearances of each edge along the Gray code from Figure 29.

5.3 Balanced triangulation Gray codes

Torsten Mütze (University of Warwick, UK, torsten.mutze@warwick.ac.uk)

We consider all triangulations of a convex n-gon. Two of them differ in a *flip* if they agree in all but 2 triangles. The corresponding flip graph, which has as nodes all triangulations, with edges connecting triangulations that differ in a flip, is the famous associahedron. It is well-known [3, 4] that for any $n \geq 3$, the associahedron has a Hamilton cycle. In other words, there is a Gray code for triangulations, i.e., we can list them cyclically so that any two consecutive triangulations differ in a flip; see Figure 29.

In a recent paper [2], we consider cycles in the associahedron that are *balanced*, i.e., if we count the number of times that each of the $\binom{n}{2} - n$ possible inner edges of the triangulation appears along the cycle, then these counts differ by at most ± 1; see Figure 30. Clearly, if an edge appears k times along the cycle, it also has to disappear k times.

Question: Is there a balanced Gray code for triangulations for every $n \geq 3$?

This problem is analogous to the problem of constructing balanced Gray codes for all 2^n binary strings of length n, where one bit is flipped in each step, and each of the n bits should be flipped the same number of times (up to ± 1) [1].

References

1 Bhat, G. S., and Savage, C. D. Balanced Gray codes. *Electron. J. Combin. 3*, 1 (1996), Paper 25, 11 pp.
2 Felsner, S., Kleist, L., Mütze, T., and Sering, L. Rainbow cycles in flip graphs. *SIAM J. Discrete Math. 34*, 1 (2020), 1–39.
3 Hurtado, F., and Noy, M. Graph of triangulations of a convex polygon and tree of triangulations. *Comput. Geom. 13*, 3 (1999), 179–188.
4 Lucas, J. M. The rotation graph of binary trees is Hamiltonian. *J. Algorithms 8*, 4 (1987), 503–535.

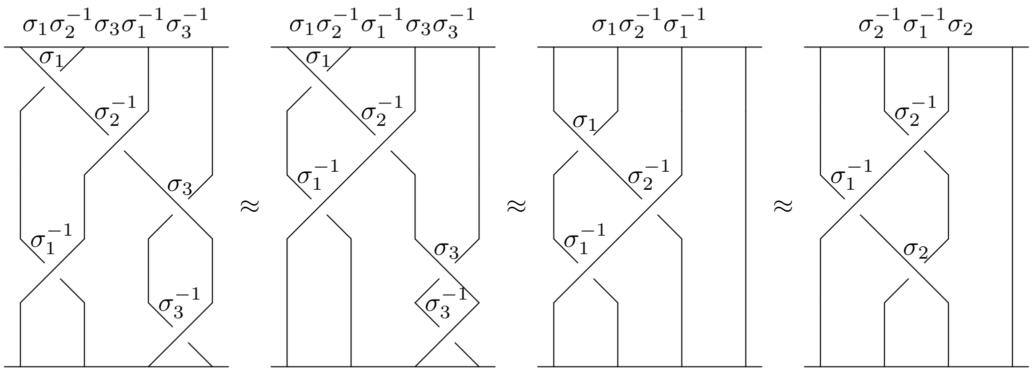


Figure 31 The braid words $\sigma_1\sigma_2^{-1}\sigma_3\sigma_1^{-1}\sigma_3^{-1}$ and $\sigma_2^{-1}\sigma_1^{-1}\sigma_2$ are isotopic.

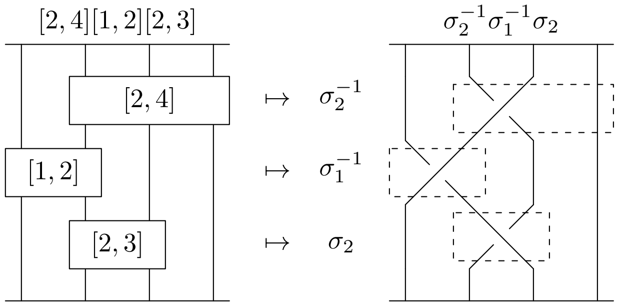


Figure 32 The braid word $\sigma_1\sigma_2^{-1}\sigma_3\sigma_1^{-1}\sigma_3^{-1}$ matches the pattern $[2,4][1,2][2,3]$ because the isotopic braid word $\sigma_2^{-1}\sigma_1^{-1}\sigma_2$ matches it directly.

5.4 Pattern-Matching on Braids

Tim Ophelders (Utrecht University & TU Eindhoven, The Netherlands, t.a.e.ophelders@uu.nl)

The *braid group* on n strands has generators $\sigma_1, \ldots, \sigma_{n-1}$ and the following relations.

- $\sigma_i\sigma_j = \sigma_j\sigma_i$ for all $i+1 < j$.
- $\sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1}$.

Let $\Sigma_{i,j} := \{\sigma_i, \ldots, \sigma_{j-1}\} \cup \{\sigma_i^{-1}, \ldots, \sigma_{j-1}^{-1}\}$ and $\Sigma_n := \Sigma_{1,n}$. A *braid word* on n strands is a sequence of elements of Σ_n. We say that two braid words are *isotopic* if the corresponding elements of the braid group are equal. See Figure 31 for an example and a geometric interpretation of the braid group.

Let $I_n := \{[i,j] \mid i,j \in \{1,\ldots,n\}\}$ be the set of intervals with integer endpoints between 1 and n. A *pattern* on n strands is a sequence of elements of $\Sigma_n \sqcup I_n$.

A braid word *directly matches* a pattern p if it can be obtained from p by substituting each element $[i,j]$ by some sequence of elements of $\Sigma_{i,j}$ (different occurrences of $[i,j]$ may be substituted by different such sequences). A braid word *matches* a pattern p if it is isotopic to a braid word that directly matches p, see Figure 32.

Question 1. Given a braid b and a pattern p, is testing whether b matches p decidable?

Call a pattern *pure* if all of its elements are drawn from I_n, i.e. none are drawn from Σ_n.

Subquestion 1.1. Given a braid b and a pure pattern p, is testing whether b matches p decidable?

Participants

- Florestan Brunck
IST Austria –
Klosterneuburg, AT
- Kevin Buchin
TU Eindhoven, NL
- Maike Buchin
Ruhr-Universität Bochum, DE
- Benjamin Burton
The University of Queensland –
Brisbane, AU
- Jean Cardinal
ULB – Brussels, BE
- Éric Colin de Verdière
CNRS & LIGM –
Marne-la-Vallée
- Arnaud de Mesmay
University Paris-Est –
Marne-la-Vallée, FR
- Linda Kleist
TU Braunschweig, DE
- Boris Klemz
Universität Würzburg, DE
- Irina Kostitsyna
TU Eindhoven, NL
- Francis Lazarus
CNRS – Grenoble, FR
- Maarten Löffler
Utrecht University, NL
- Alexander Neuhaus
Ruhr-Universität Bochum, DE
- Tim Ophelders
Utrecht University, NL
- Alexander Wolff
Universität Würzburg, DE

Remote Participants

- Mark Bell
Hampshire, GB
- Hsien-Chih Chang
Dartmouth College –
Hanover, US
- Vincent Delecroix
University of Bordeaux, FR
- Anne Driemel
Universität Bonn, DE
- Nathan Dunfield
University of Illinois –
Urbana Champaign, US
- William Evans
University of British Columbia –
Vancouver, CA
- Brittany Terese Fasy
Montana State University –
Bozeman, US
- Fabrizio Frati
University of Rome III, IT
- Niloufar Fuladi
University Paris-Est –
Marne-la-Vallée, FR
- Elise Goujard
University of Bordeaux, FR
- Luke Jeffreys
University of Bristol, GB
- Anna Lubiw
University of Waterloo, CA
- Torsten Mütze
University of Warwick –
Coventry, GB
- Hugo Parlier
University of Luxembourg, LU
- Lionel Pournin
Université Paris Nord –
Villetaneuse, FR
- Jessica S. Purcell
Monash University –
Clayton, AU
- Saul Schleimer
University of Warwick –
Coventry, GB
- Lena Schlipf
Universität Tübingen, DE
- Eric Sedgwick
DePaul University – Chicago, US
- Rodrigo I. Silveira
UPC Barcelona Tech, ES

- Jonathan Spreer
University of Sydney, AU
- Stephan Tillmann
University of Sydney, AU
- Uli Wagner
IST Austria – Klosterneuburg, AT
- Carola Wenk
Tulane University – New Orleans, US
- Erin Moriarty Wolf Chambers
St. Louis University, US