Report from Dagstuhl Seminar 24411

New Tools in Parameterized Complexity: Paths, Cuts, and Decomposition

Fedor V. Fomin[*1], **Dániel Marx**[*2], **Saket Saurabh**[*3], **Roohani Sharma**[*4], **and Madhumita Kundu**[†5]

1 **University of Bergen, NO.** fomin@ii.uib.no
2 **CISPA – Saarbrücken, DE.** marx@cispa.de
3 **The Institute of Mathematical Sciences – Chennai, IN.** saket@imsc.res.in
4 **MPI für Informatik – Saarbrücken, DE.** roohani.sharma90@gmail.com
5 **University of Bergen, NO.** Madhumita.Kundu@uib.no

Abstract

The Dagstuhl Seminar concentrated on the development of new tools arising from the parameterized complexity of cuts, paths, and decompositions in graphs. The last 2 years were very exciting for the area, with a number of breakthroughs.

In FOCS 2021, Korhonen introduced a new method for approximating tree decompositions in graphs. His method, which was deeply rooted in classical graph theory, appeared to be a very handy tool for decomposing graphs, and several STOC/FOCS papers developed this method in various settings. In parallel, a novel perspective on graph decompositions was proposed by Bonnet *et al.* in FOCS 2020. The new theory of twin-width had many exciting consequences, and we were still at the beginning of understanding the real impact of the new decompositions on graph algorithms.

In a series of papers (SODA 2021, STOC 2022, SODA 2023), Kim *et al.* developed beautiful algorithmic methods for handling separators in (undirected, weighted, or directed) graphs by the addition of arcs. The new algorithmic tool was used to resolve a number of long-standing open problems in the area, and it also seemed to pave the road to many more new discoveries.

Reis and Rothvoss (Arxiv 2023) announced a $((\log n)^{O(n)})$ time randomized algorithm to solve integer programs in n variables. This breakthrough had an impact on many problems in parameterized complexity, especially on problems concerning cuts in graphs. Finally, by employing algebraic methods (both new and old), significant progress was made on several problems related to paths, including the classical (k)-disjoint path problems.

This seminar brought together people from the parameterized complexity community, specialists in cuts, flows, and connectivity, and those who had been at the forefront of these new developments. In doing so, it consolidated the results achieved in recent years, discussed future research directions, and further explored the potential applications of the methods and techniques described above.

Seminar October 6–11, 2024 – https://www.dagstuhl.de/24411
2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography; Theory of computation → Design and analysis of algorithms
Keywords and phrases fixed-parameter tractability, intractability, parameterized complexity
Digital Object Identifier 10.4230/DagRep.14.10.1

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

1 Executive Summary

Fedor V. Fomin (University of Bergen, NO)
Dániel Marx (CISPA – Saarbrücken, DE)
Saket Saurabh (The Institute of Mathematical Sciences – Chennai, IN)
Roohani Sharma (MPI für Informatik – Saarbrücken, DE)

Description of the Seminar: Topics and goals

Parameterized Complexity is an alternative approach of handling computational intractability. The main idea of the approach taken by the Parameterized Complexity is to analyze the complexity in finer detail by considering additional problem parameters beyond the input size and expresses the efficiency of the algorithms in terms of these parameters. In this framework, many NP-hard problems have been shown to be (fixed-parameter) tractable (FPT) when certain structural parameters of the inputs are bounded.

In the last three decades, there has been tremendous progress in understanding which problems are fixed-parameter tractable and which problems are not (under standard complexity assumptions). For all these years the central vision of the Parameterized Complexity has been to provide the algorithmic and complexity-theoretic toolkit for studying multivariate algorithmics in different disciplines and subfields of Computer Science. To achieve this vision several algorithmic and complexity theoretic tools such as *polynomial time preprocessing, aka, kernelization*, *color-coding*, *graph-decompositions*, *parameterized integer programming*, *iterative compression*, or *lower bounds methods based on assumptions stronger than* P≠NP have been developed. These tools are *universal* as they not only helped in the development of the core of Parameterized Complexity but also led to its success in other subfields of Computer Science such as Approximation Algorithms, Computational Social Choice, Computational Geometry, problems solvable in P (polynomial time) to name a few.

In the last few years several decade old open problems in Parameterized Complexity have been resolved. These have resulted in several new algorithmic tools for the core of Parameterized Complexity. These include tools such as iterative and local improvement methods for graph decomposition, methods arising from extremal combinatorics and graph theory, flow augmentation, faster algorithms for solving integer programs on n variables, and new algebraic methods. A natural question is to extend these tools in different directions and explore the limits and applicability of these new tools. Thus,

> *the main objective of the Dagstuhl Seminar was to initiate the discussion on extension, limits and applicability of newly developed tools arising from paths, cuts, and decomposition.*

One of the seminar's central goals was to facilitate a fruitful dialogue between researchers working at the core of Parameterized Complexity and those from Mathematical Programming, Computational Linear Algebra, Graph Theory, and Combinatorics, who had contributed to recent advances in parameterized algorithms. The Dagstuhl event enabled participants to explore possibilities for developing new tools and techniques that emerged from this collaboration.

Next, the seminar presented a few concrete examples of newly developed tools and techniques from various domains of Parameterized Complexity, which formed the focal points of discussion.

- ***Width Parameters: Treewidth and Twinwidth.***
- ***Tools Based on Extremal Combinatorics: Paths and Rainbow Matching.***
- ***Cut Based Tool: Flow Augmentation.***
- ***Mathematical Programming.***
- ***Algebraic Methods.***

Related Seminars

This Dagstuhl Seminar could be considered as a continuation of the Dagstuhl Seminar series on parameterized algorithms and complexity. The previous seminars in this series are New Horizons in Parameterized Complexity (seminar 19041), Randomization in Parameterized Complexity (seminar 17041), Optimality and Tight Results in Parameterized Complexity (seminar 14451), Data Reduction and Problem Kernels (seminar 12241), Parameterized complexity and approximation algorithms (seminar 09511), Structure Theory and FPT Algorithmics for Graphs, Digraphs and Hypergraphs (seminar 07281), Exact Algorithms and Fixed-Parameter Tractability (seminar 05301), Fixed Parameter Algorithms (seminar 03311), and Parameterized Complexity (seminar 01311).

Organization of the Seminar

During the five-day seminar, 48 researchers from theoretical computer science, mathematical optimization, and operations research convened. Attendees ranged from senior scientists to postdoctoral scholars and advanced doctoral candidates, creating a rich environment for both mentorship and innovation.

The seminar featured 22 presentations of varying lengths. Six keynote speakers–Dániel Marx, Michał Pilipczuk, Euiwoong Lee, Tuukka Korhonen, Jie Xue, and Daniel Lokshtanov–delivered 60-minute talks that provided overviews of state-of-the-art methods and showcased recent breakthroughs in their respective areas. The remaining slots were filled with shorter, 30-minute talks covering various topics.

At the beginning of the week, open problem sessions encouraged participants to share challenges and spark collaborative research. The schedule also included ample free time, which attendees used for productive discussions and joint work, fostering new ideas and potential research partnerships.

Outcome

Organizers and participants regarded the seminar as a great success. It successfully brought together the relevant research communities, facilitated the sharing of state-of-the-art results, and enabled discussions on the major challenges in the field. The talks were not only excellent but also highly stimulating, prompting active engagement in working groups during afternoons and evenings. Particularly noteworthy was the participation of younger researchers (postdocs and PhD students), who integrated seamlessly and contributed to the seminar's collegial and productive atmosphere.

2 Table of Contents

3 Overview of Talks

3.1 Approximating Small Sparse Cuts

Aditya Anand (University of Michigan – Ann Arbor, US), Thatchaphol Saranurak (University of Michigan – Ann Arbor, US)

Joint work of Aditya Anand, Jason Li, Thatchaphol Saranurak
Main reference Aditya Anand, Euiwoong Lee, Jason Li, Thatchaphol Saranurak: "Approximating Small Sparse Cuts", CoRR, Vol. abs/2403.08983, 2024.
URL https://doi.org/10.48550/ARXIV.2403.08983

We study polynomial-time approximation algorithms for (edge/vertex) Sparsest Cut and Small Set Expansion in terms of k, the number of edges or vertices cut in the optimal solution. Our main results are $O(poly \log k)$-approximation algorithms for various versions in this setting.

Our techniques involve an extension of the notion of sample sets (Feige and Mahdian STOC'06), originally developed for small balanced cuts, to sparse cuts in general. We then show how to combine this notion of sample sets with two algorithms, one based on an existing framework of LP rounding and another new algorithm based on the cut-matching game, to get such approximation algorithms. Our cut-matching game algorithm can be viewed as a local version of the cut-matching game by Khandekar, Khot, Orecchia and Vishnoi and certifies an expansion of every vertex set of size s in $O(\log s)$ rounds. These techniques may be of independent interest.

As corollaries of our results, we also obtain an O(logopt)-approximation for min-max graph partitioning, where opt is the min-max value of the optimal cut, and improve the bound on the size of multicut mimicking networks computable in polynomial time.

3.2 Unbreakable Decomposition in Close-to-Linear Time

Aditya Anand (University of Michigan – Ann Arbor, US), Euiwoong Lee (University of Michigan – Ann Arbor, US), Thatchaphol Saranurak (University of Michigan – Ann Arbor, US)

Joint work of Aditya Anand, Euiwoong Lee, Jason Li, Yaowei Long, Thatchaphol Saranurak
Main reference Aditya Anand, Euiwoong Lee, Jason Li, Yaowei Long, Thatchaphol Saranurak: "Unbreakable Decomposition in Close-to-Linear Time", in Proc. of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025, pp. 1464–1493, SIAM, 2025.
URL https://doi.org/10.1137/1.9781611978322.46

Unbreakable decomposition, introduced by Cygan et al. (SICOMP'19) and Cygan et al. (TALG'20), has proven to be one of the most powerful tools for parameterized graph cut problems in recent years. Unfortunately, all known constructions require at least $\Omega_k(mn^2)$ time, given an undirected graph with n vertices, m edges, and cut-size parameter k. In this work, we show the first close-to-linear time parameterized algorithm that computes an unbreakable decomposition. More precisely, for any $0 < \epsilon \leq 1$, our algorithm runs in time $2^{O(\frac{k}{\epsilon}\log(k/\epsilon))}m^{1+\epsilon}$ and computes a $(O(k/\epsilon), k)$ unbreakable tree decomposition of the input graph, where each bag has adhesion at most $O(k/\epsilon)$. This immediately opens up possibilities for obtaining close-to-linear time algorithms for numerous problems whose only known solution is based on unbreakable decomposition.

3.3 Planar Min-Sum Disjoint Paths in Subexponential FPT time

Matthias Bentert (University of Bergen, NO), Fedor V. Fomin (University of Bergen, NO), Petr A. Golovach (University of Bergen, NO)

In the Min-Sum Disjoint Paths problem, we are given an edge-weighted n-vertex graph and k terminal pairs. The task is to connect all terminal pairs by pairwise internally vertex-disjoint paths of minimum total length (or decide that there is no set of pairwise disjoint paths). We show that Min-Sum Disjoint Paths on (directed or undirected) planar input graphs can be solved in $2^{O(\sqrt{\ell}\log^{O(1)}(\ell))}n^{O(1)}$ time, where ℓ is the number of edges in an optimal solution. We complement our main result with an ETH-based lower bound excluding $2^{o(\sqrt{n})}$-time algorithms for undirected and unweighted planar graphs even in the special case where we ask whether all terminal pairs can be connected by shortest paths between the respective endpoints.

3.4 Twin-width – Part 2

Édouard Bonnet (ENS – Lyon, FR)

We survey some algorithmic applications of twin-width.

3.5 Algorithms for 2-connected network design and flexible Steiner trees with a constant number of terminals

Joseph Cheriyan (University of Waterloo, CA)

Joint work of Joseph Cheriyan, Ishan Bansal, Logan Grout, Sharat Ibrahimpur
Main reference Ishan Bansal, Joe Cheriyan, Logan Grout, Sharat Ibrahimpur: "Algorithms for 2-Connected Network Design and Flexible Steiner Trees with a Constant Number of Terminals", in Proc. of the Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2023, September 11-13, 2023, Atlanta, Georgia, USA, LIPIcs, Vol. 275, pp. 14:1–14:14, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.
URL https://doi.org/10.4230/LIPICS.APPROX/RANDOM.2023.14

The k-Steiner-2NCS problem is as follows: Given a constant k, and an undirected connected graph $G = (V, E)$, non-negative costs c on the edges, and a partition of V into a set of terminals, T, and a set of non-terminals (or, Steiner nodes), $V - T$, where $|T| = k$, our algorithmic goal is to find a min-cost two-node connected subgraph that contains the terminals.

We present a randomized polynomial-time algorithm for the unweighted problem, and a randomized FPTAS for the weighted problem.

We obtain similar results for the k-Steiner-2ECS problem, where the input is the same, and the algorithmic goal is to find a min-cost two-edge connected subgraph that contains the terminals.

Our methods build on results by Bjorklund, Husfeldt, and Taslaman (SODA 2012) that give a randomized polynomial-time algorithm for the unweighted k-Steiner-cycle problem; this problem has the same inputs as the unweighted k-Steiner-2NCS problem, and the algorithmic goal is to find a min-cost simple cycle that contains the terminals.

3.6 Topological methods for graph algorithms: (multi-)cuts on surface-embedded graphs

Éric Colin de Verdière (Gustave Eiffel University – Marne-la-Vallée, FR)

Minimum cut and multicut problems for general graphs are well-studied. In this talk, we survey algorithms and lower bounds for these problems when restricting to graphs that are either planar or embeddable on a fixed surface. For such classes of graphs, topological techniques, developed originally in computational geometry/topology with very different motivations in mind, turn out to be very useful.

We first aim to provide an executive summary of the computational topology routines that are the most useful for cut problems for surface-embedded graphs. We then discuss near-linear time algorithms for minimum cut on a fixed surface, which outperform the existing generic (polynomial-time) combinatorial algorithms. We then turn to the multicut problem, presenting an algorithm that is fixed-parameter tractable parameterized by the genus and the number of terminals, and an almost matching lower bound. There is actually a refined complexity analysis depending also on the demand pattern. Finally, we present an algorithm for the multicut problem that returns an $(1+\varepsilon)$-approximation and that is fixed parameter tractable in the genus, the number of terminals, and ε.

All these results rely on topological methods: The subgraph of the dual of the input graph, made of the edges dual to a multicut, has nice properties, which can be exploited using classical tools from algebraic topology such as homotopy, homology, and covering spaces.

3.7 Efficient Approximation of Fractional Hypertree Width

Daniel Lokshtanov (University of California – Santa Barbara, US), Saket Saurabh (The Institute of Mathematical Sciences – Chennai, IN), Vaishali Surianarayanan (University of California – Santa Barbara, US), Jie Xue (New York University – Shanghai, CN)

Joint work of Daniel Lokshtanov, Viktoriia Korchemna, Saket Saurabh, Vaishali Surianarayanan, Jie Xue
Main reference Viktoriia Korchemna, Daniel Lokshtanov, Saket Saurabh, Vaishali Surianarayanan, Jie Xue: "Efficient Approximation of Fractional Hypertree Width", CoRR, Vol. abs/2409.20172, 2024.
URL https://doi.org/10.48550/ARXIV.2409.20172

We give two new approximation algorithms to compute the *fractional hypertree width* of an input hypergraph. The first algorithm takes as input n-vertex m-edge hypergraph H of fractional hypertree width at most ω, runs in polynomial time and produces a tree decomposition of H of fractional hypertree width $\mathcal{O}(\omega \log n \log \omega)$, i.e., it is an $\mathcal{O}(\log n \log \omega)$-approximation algorithm. As an immediate corollary this yields polynomial time $\mathcal{O}(\log^2 n \log \omega)$-approximation algorithms for (generalized) hypertree width as well. To the best of our knowledge our algorithm is the first non-trivial polynomial-time approximation algorithm for fractional hypertree width and (generalized) hypertree width, as opposed to algorithms that run in polynomial time only when ω is considered a constant. For hypergraphs with the *bounded intersection property* (i.e. hypergraphs where every pair of hyperedges have at most η vertices in common) the algorithm outputs a hypertree decomposition with fractional hypertree width $\mathcal{O}(\eta\omega^2 \log \omega)$ and generalized hypertree width $\mathcal{O}(\eta\omega^2 \log \omega(\log \eta + \log \omega))$. This ratio is comparable with the recent algorithm of Lanzinger and Razgon [STACS 2024], which produces a hypertree decomposition with generalized hypertree width $\mathcal{O}(\omega^2(\omega + \eta))$, but uses time (at least) exponential in η and ω.

The second algorithm runs in time $n^{\omega} m^{\mathcal{O}(1)}$ and produces a tree decomposition of H of fractional hypertree width $\mathcal{O}(\omega \log^2 \omega)$. This significantly improves over the $(n+m)^{\mathcal{O}(\omega^3)}$ time algorithm of Marx [ACM TALG 2010], which produces a tree decomposition of fractional hypertree width $\mathcal{O}(\omega^3)$, both in terms of running time and the approximation ratio.

Our main technical contribution, and the key insight behind both algorithms, is a variant of the classic Menger's Theorem for clique separators in graphs: For every graph G, vertex sets A and B, family $\mathcal{F}$ of cliques in G, and positive rational f, either there exists a sub-family of $\mathcal{O}(f \cdot \log^2 n)$ cliques in $\mathcal{F}$ whose union separates A from B, or there exist $f \cdot \log |\mathcal{F}|$ paths from A to B such that no clique in $\mathcal{F}$ intersects more than $\log |\mathcal{F}|$ paths.

3.8 Twin-width I

Eun Jung Kim (KAIST – Daejeon, KR & CNRS – Paris, FR)

Twin-width is a relatively new notion introduced in 2020 by Bonnet, Kim, Thomassé and Watrigant. Since then, it is prove to demonstrate rich structure and be a useful tool for understanding graph classes and logic, designing algorithms, etc. In this talk, we present the notion, grid theorem for twin-width and how to use it for proving certain graph classes have bounded twin-width as well as other use cases of twin-width.

3.9 Longest Path parameterized by Maximum Independent Set

Fedor V. Fomin (University of Bergen, NO)

Question 1. What is the complexity of Longested Path parameterized by $\alpha(G)$ (the largest size of an indpendent set). We know that the problem is **FPT** in undirected graphs and either returns a path of length at least k or establishes that $\alpha(G) \geq k$. However, what is the complexity of the problem in directed graphs? Is it even in **XP**?

Additionally, what about the **promise version**, where an independent set of size $\alpha(G)$ is provided as input?

Question 2. (Gallai-Milgram Theorem) Let G be a directed graph. The vertices of G can be covered by at most $\alpha(G)$ disjoint paths.

Additionally, there exist at least $k - t$ edge-disjoint paths if the graph satisfies certain conditions.

3.10 Steiner Tree Parameterized by Multiway Cut and Even Less

Bart Jansen (TU Eindhoven, NL)

Joint work of Bart M. P. Jansen, Celine Swennenhuis
Main reference Bart M. P. Jansen, Céline M. F. Swennenhuis: "Steiner Tree Parameterized by Multiway Cut and Even Less", in Proc. of the 32nd Annual European Symposium on Algorithms, ESA 2024, September 2-4, 2024, Royal Holloway, London, United Kingdom, LIPIcs, Vol. 308, pp. 76:1–76:16, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024.
URL https://doi.org/10.4230/LIPICS.ESA.2024.76

In the Steiner Tree problem we are given an undirected edge-weighted graph as input, along with a set K of vertices called *terminals*. The task is to output a minimum-weight connected subgraph that spans all the terminals. The famous Dreyfus-Wagner algorithm running in $3^{|K|}\mathsf{poly}(n)$ time shows that the problem is fixed-parameter tractable parameterized by the number of terminals. We present fixed-parameter tractable algorithms for Steiner Tree using structurally smaller parameterizations.

Our first result concerns the parameterization by a multiway cut S of the terminals, which is a vertex set S (possibly containing terminals) such that each connected component of $G - S$ contains at most one terminal. We show that Steiner Tree can be solved in $2^{O(|S| \log |S|)}\mathsf{poly}(n)$ time and polynomial space, where S is a minimum multiway cut for K. The algorithm is based on the insight that, after guessing how an optimal Steiner tree interacts with a multiway cut S, computing a minimum-cost solution of this type can be formulated as minimum-cost bipartite matching.

Our second result concerns a new hybrid parameterization called *K-free treewidth* that simultaneously refines the number of terminals $|K|$ and the treewidth of the input graph. By utilizing recent work on $\mathcal{H}$-Treewidth in order to find a corresponding decomposition of the graph, we give an algorithm that solves Steiner Tree in time $2^{O(k)}\mathsf{poly}(n)$, where k denotes the K-free treewidth of the input graph. To obtain this running time, we show how the *rank-based* approach for solving Steiner Tree parameterized by treewidth can be extended to work in the setting of K-free treewidth, by exploiting existing algorithms parameterized by $|K|$ to compute the table entries of leaf bags of a tree K-free decomposition.

3.11 A Subexponential Time Algorithm for Makespan Scheduling of Unit Jobs with Precedence Constraints

Jesper Nederlof (Utrecht University, NL)

Joint work of Jesper Nederlof, Céline M. F. Swennenhuis, Karol Wegrzycki
Main reference Jesper Nederlof, Céline M. F. Swennenhuis, Karol Wegrzycki: "A Subexponential Time Algorithm for Makespan Scheduling of Unit Jobs with Precedence Constraints", CoRR, Vol. abs/2312.03495, 2023.
URL https://doi.org/10.48550/ARXIV.2312.03495

In a classical scheduling problem, we are given a set of n jobs of unit length with precedence constraints, and the goal is to find a schedule of these jobs on m identical machines that minimizes the makespan. In standard 3-field notation, it is denoted as $Pm|\text{prec}, p_j = 1|C_{\max}$.

For $m = 2$ machines, the problem can be solved in polynomial time. Settling the complexity for any constant $m \geq 3$ is a longstanding open question in the field, asked by Lenstra and Rinnooy Kan [OR 1978] in the late 70s and prominently featured in the textbook of Garey and Johnson. Since then, the problem has been thoroughly investigated, but nontrivial solutions had been found only in special cases or relaxed settings. For example, despite the possibility of the problem being polynomially solvable in the exact setting, just the existence of an approximation-scheme is widely regarded as a major open problem (see the survey of Bansal [MAPS 2017]), but so far, only superpolynomial approximations are known.

In this paper, we make the first progress on the exact complexity of $Pm|\text{prec}, p_j = 1|C_{\max}$. We present an algorithm that runs in $2^{O(\sqrt{n}\log n)}$ time for $m = O(1)$. Before our work, only a $2^{O(n)}$ time exact algorithm was known by Held and Karp [ACM 1961].

3.12 The Parameter Report: An Orientation Guide for Data-Driven Parameterization

Christian Komusiewicz (Friedrich-Schiller-Universität Jena, DE)

Joint work of Christian Komusiewicz, Nils Morawietz, Frank Sommer, Luca Staus

A strength of parameterized algorithmics is that each problem can be parameterized by an essentially inexhaustible set of parameters. Usually, the choice of the considered parameter is informed by the theoretical relations between parameters with the general goal of achieving FPT-algorithms for smaller and smaller parameters. However, the FPT-algorithms for smaller parameters usually have higher running times and it is not clear whether the decrease in the parameter value or the increase in the running time bound dominates in real-world data. Any answer to this question requires knowledge on typical parameter values. To provide a data-driven guideline for parameterized complexity studies of graph problems, we present the first comprehensive comparison of parameter values for a set of benchmark graphs originating from real-world applications. Our study covers degree-related parameters, such as maximum degree or degeneracy, neighborhood-based parameters such as neighborhood diversity and modular-width, modulator-based parameters such as vertex cover number and feedback vertex set number, and the treewidth of the graphs. Our implementation and full experimental data are openly available.

3.13 Linear-Time Algorithms for k-Edge-Connected Components, k-Lean Tree Decompositions, and More

Tuukka Korhonen (University of Copenhagen, DK)

We present $k^{O(k^2)}m$ time algorithms for various problems about decomposing a given graph by edge cuts or vertex separators of size less than k into parts that are “well-connected” with respect to cuts or separators of size less than k; here, m is the total number of vertices and edges of the graph. As an application of our results, we obtain for every fixed k a linear-time algorithm for computing the k-edge-connected components of a given graph, solving a long-standing open problem. More generally, we obtain a $k^{O(k^2)}m$ time algorithm for computing a k-Gomory-Hu tree of a given graph, which is a structure representing all pairwise minimum cuts of size less than k.

Our main technical result, from which the other results follow, is a $k^{O(k^2)}m$ time algorithm for computing a k-lean tree decomposition of a given graph. This is a tree decomposition with adhesion size less than k that captures the existence of separators of size less than k between subsets of its bags. A k-lean tree decomposition is also an unbreakable tree decomposition with optimal unbreakability parameters for the adhesion size bound k.

As further applications, we obtain $k^{O(k^2)}m$ time algorithms for k-vertex connectivity and for k-Gomory-Hu tree for element connectivity. All of our algorithms are deterministic.

Our techniques are inspired by the tenth paper of the Graph Minors series of Robertson and Seymour and by Bodlaender's parameterized linear-time algorithm for computing treewidth.

3.14 Parameterized Inapproximability Hypothesis under Exponential Time Hypothesis

Bingkai Lin (Nanjing University, CN)

Joint work of Bingkai Lin, Venkatesan Guruswami, Xuandi Ren, Yican Sun, Kewen Wu
Main reference Venkatesan Guruswami, Bingkai Lin, Xuandi Ren, Yican Sun, Kewen Wu: "Parameterized Inapproximability Hypothesis under Exponential Time Hypothesis", in Proc. of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024, pp. 24–35, ACM, 2024.
URL https://doi.org/10.1145/3618260.3649771

The Parameterized Inapproximability Hypothesis (PIH) asserts that no fixed parameter tractable (FPT) algorithm can distinguish a satisfiable CSP instance, parameterized by the number of variables, from one where every assignment fails to satisfy an ε fraction of constraints for some absolute constant $\varepsilon > 0$. PIH plays the role of the PCP theorem in parameterized complexity. However, PIH has only been established under Gap-ETH, a very strong assumption with an inherent gap.

In this work, we prove PIH under the Exponential Time Hypothesis (ETH). This is the first proof of PIH from a gap-free assumption. Our proof is self-contained and elementary. We identify an ETH-hard CSP whose variables take vector values, and constraints are either linear or of a special parallel structure. Both kinds of constraints can be checked with constant soundness via a "parallel PCP of proximity" based on the Walsh-Hadamard code.

3.15 Feedback Vertex Set on Planar Graphs

Daniel Lokshtanov (University of California – Santa Barbara, US)

Given an undirected graph $G = (V, E)$, the **Feedback Vertex Set (FVS)** problem asks for a minimum-size subset $S \subseteq V$ such that the subgraph induced by $V \setminus S$ is acyclic. The unweighted Feedback Vertex Set problem is known to admit an EPTAS on planar graphs, whereas the weighted version only has a PTAS. A natural question is whether the problem admits an EPTAS even in the weighted case.

3.16 Cuts, Paths, and Decompositions

Dániel Marx (CISPA – Saarbrücken, DE)

We overview the many different types of cut, path, and decomposition problems in the field of parameterized algorithms and sketch some connections between the different topics.

3.17 Minimum Isolating Cuts for Fast Graph Algorithms: A tutorial

Thatchaphol Saranurak (University of Michigan – Ann Arbor, US)

I give a gentle tutorial on using the minimum isolating cuts for fast graph algorithms, as well as, give a survey of its applications.

3.18 Parameterized Streaming

Ramanujan Sridharan (University of Warwick – Coventry, GB)

Joint work of Ramanujan Sridharan, Daniel Lokshtanov, Pranabendu Misra, Fahad Panolan, M. S. Ramanujan, Saket Saurabh, Meirav Zehavi
Main reference Daniel Lokshtanov, Pranabendu Misra, Fahad Panolan, M. S. Ramanujan, Saket Saurabh, Meirav Zehavi: "Meta-theorems for Parameterized Streaming Algorithms‡", in Proc. of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024, pp. 712–739, SIAM, 2024.
URL https://doi.org/10.1137/1.9781611977912.28

The streaming model has been studied from the point of parameterized complexity in the last decade by several researchers. However, the applicability of the streaming model to central problems in parameterized complexity has remained somewhat limited due to simple $\Omega(n)$-space lower bounds for many problems. In other words, the $k^{O(1)}(\log n)^{O(1)}$ space requirement of the parameterized streaming model is too restrictive. This has motivated the study of parameterized semi-streaming algorithms.

In this talk, we will discuss some recent developments in this direction.

3.19 Planar Disjoint Shortest Paths is Fixed-Parameter Tractable

Giannos Stamoulis (University of Warsaw, PL)

Joint work of Michał Pilipczuk, Giannos Stamoulis, Michał Włodarczyk

In the Disjoint Shortest Paths problem one is given a graph G and a set $T = \{(s1_, t_1), ..., (s_k, t_k)\}$ of k vertex pairs. The question is whether there exist vertex-disjoint paths $P_1, ..., P_k$ in G so that each P_i is a shortest path between s_i and t_i. While the problem is known to be W[1]-hard in general, we show that it is fixed-parameter tractable on planar graphs with positive edge weights. Specifically, we propose an algorithm for Planar Disjoint Shortest Paths with running time $2^{O(k \log k)} n^{O(1)}$. Notably, our parameter dependency is better than state-of-the-art $2^{O(k^2)}$ for the Planar Disjoint Paths problem, where the sought paths are not required to be shortest paths.

3.20 Parameterized Approximation for Capacitated d-Hitting Set with Hard Capacities

Vaishali Surianarayanan (University of California – Santa Barbara, US), Daniel Lokshtanov (University of California – Santa Barbara, US), Saket Saurabh (The Institute of Mathematical Sciences – Chennai, IN), Jie Xue (New York University – Shanghai, CN)

Joint work of Vaishali Surianarayanan, Abhishek Sahu, Daniel Lokshtanov, Saket Saurabh, Jie Xue

In the Capacitated d-Hitting Set problem input is a universe U equipped with a capacity function $\mathsf{cap} : U \to \mathbb{N}$, and a collection $\mathcal{A}$ of subsets of U, each of size at most d. The task is to find a minimum size subset S of U and an assignment $\phi : \mathcal{A} \to S$ such that, for every set $A \in \mathcal{A}$ we have $\phi(A) \in A$ and for every $x \in U$ we have $|\phi^{-1}(x)| \leq \mathsf{cap}(x)$. When $d = 2$ the problem is known under the name Capacitated Vertex Cover. In Weighted Capacitated d-Hitting Set each element of U has a positive integer weight and the goal is to find a capacitated hitting set of minimum weight.

In this paper we initiate the study of parameterized (approximation) algorithms for Capacitated d-Hitting Set. Capacitated d-Hitting Set is a well studied problem and is known to admit a d-approximation algorithm and no $(d - \epsilon)$-approximation under UGC for any $\epsilon > 0$. Further, unweighted Capacitated d-Hitting Set for $d \geq 3$ is W[1]-hard parameterized by solution size. Our main result is a parameterized approximation algorithm that runs in time $\left(\frac{k}{\epsilon}\right)^k 2^{k^{O(kd)}}(|U| + |\mathcal{A}|)^{O(1)}$ and either concludes that there is no solution of size at most k or outputs a solution S of size at most $4/3 \cdot k$ and weight at most $2 + \epsilon$ times the minimum weight of a solution whose size is at most k. We also complement our algorithmic results with hardness results.

3.21 Parameterized Complexity in Explainable AI

Stefan Szeider (TU Wien, AT)

As machine learning (ML) models become increasingly complex, interpretability and explainability in ML-based decisions, referred to as eXplainable AI (XAI) has become an important objective. In this talk, we introduce two computational questions that arise in the context of XAI and discuss possible parameterizations.

1. Finding a small symbolic ML model that best represents given data.
2. Generating a concise explanation for a prediction from an existing symbolic model.

We discuss recent results [1, 2, 3, 4, 5, 6, 7, 8] and examine these questions for various model types, including decision trees (DT), decision sets (DS), decision lists (DL), and binary decision diagrams (BDD), highlighting possible avenues for future research.

References

1 K. Dabrowski, E. Eiben, S. Ordyniak, G. Paesani, and S. Szeider. Learning small decision trees for data of low rank-width. In J. Dy and S. Natarajan, editors, *AAAI'24*, pages 10476–10483. AAAI Press, 2024.

2 E. Eiben, S. Ordyniak, G. Paesani, and S. Szeider. Learning small decision trees with large domain. In E. Elkind, editor, *IJCAI'23*, pages 3184–3192, 2023.

3 H. Gahlawat and M. Zehavi. Learning small decision trees with few outliers: A parameterized perspective. In M. J. Wooldridge, J. G. Dy, and S. Natarajan, editors, *AAAI'24*, pages 12100–12108. AAAI Press, 2024.

4 C. Komusiewicz, P. Kunz, F. Sommer, and M. Sorge. On computing optimal tree ensembles. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *ICML'23*, pages 17364–17374. PMLR, 2023.

5 S. Ordyniak, G. Paesani, M. Rychlicki, and S. Szeider. Explaining decisions in ML models: a parameterized complexity analysis. In *KR'24*, 2024. To appear.

6 S. Ordyniak, G. Paesani, M. Rychlicki, and S. Szeider. A general theoretical framework for learning smallest interpretable models. In J. Dy and S. Natarajan, editors, *AAAI'24*, pages 10662–10669. AAAI Press, 2024.

7 S. Ordyniak, G. Paesani, and S. Szeider. The parameterized complexity of finding concise local explanations. In *IJCAI'23*, pages 3312–3320. ijcai.org, 2023.

8 S. Ordyniak and S. Szeider. Parameterized complexity of small decision tree learning. In *AAAI'21*, pages 6454–6462. AAAI Press, 2021.

3.22 Complexity Framework for Subgraph-Free Graphs & Beyond

Erik Jan van Leeuwen (Utrecht University, NL)

Joint work of Hans Bodlaender, Matthew Johnson, Barnaby Martin, Jelle J. Oostveen, Sukanya Pandey, Daniël Paulusma, Siani Smith, Erik Jan van Leeuwen
Main reference Matthew Johnson, Barnaby Martin, Jelle J. Oostveen, Sukanya Pandey, Daniël Paulusma, Siani Smith, Erik Jan van Leeuwen: "Complexity Framework For Forbidden Subgraphs", CoRR, Vol. abs/2211.12887, 2022.
URL https://doi.org/10.48550/ARXIV.2211.12887

For any finite set $\mathcal{H} = \{H_1, \ldots, H_p\}$ of graphs, a graph is $\mathcal{H}$-subgraph-free if it does not contain any of $H_1, \ldots, H_p$ as a subgraph. We give a new framework that precisely classifies whether problems are "efficiently solvable" or "computationally hard" for $\mathcal{H}$-subgraph-free graphs, depending on $\mathcal{H}$. To illustrate the broad applicability of our framework, we study partitioning, covering and packing problems, network design problems, and width parameter problems. We apply the framework to obtain a dichotomy (depending on $\mathcal{H}$) between polynomial-time solvability and NP-completeness of those problems. For other problems, we obtain a dichotomy between almost-linear-time solvability and having no subquadratic-time algorithm (conditioned on some hardness hypotheses). Along the way, we unify and strengthen known results from the literature. We also discuss recent insights into the complexity on $\mathcal{H}$-subgraph-free graphs of problems that do not fall within the framework.

3.23 Planar Disjoint Paths, Treewidth, and Kernels

Michał Wlodarczyk (University of Warsaw, PL)

Joint work of Michał Wlodarczyk, Meirav Zehavi
Main reference Michal Wlodarczyk, Meirav Zehavi: "Planar Disjoint Paths, Treewidth, and Kernels", in Proc. of the 64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023, pp. 649–662, IEEE, 2023.
URL https://doi.org/10.1109/FOCS57990.2023.00044

In the Planar Disjoint Paths problem, one is given an undirected planar graph with a set of k vertex pairs (s_i, t_i) and the task is to find k pairwise vertex-disjoint paths such that the i-th path connects s_i to t_i. We study the problem through the lens of kernelization, aiming at efficiently reducing the input size in terms of a parameter. We show that Planar Disjoint Paths does not admit a polynomial kernel when parameterized by k unless coNP $\subseteq$ NP/poly, resolving an open problem by [Bodlaender, Thomassé, Yeo, ESA'09]. Moreover, we rule out the existence of a polynomial Turing kernel unless the WK-hierarchy collapses. Our reduction carries over to the setting of edge-disjoint paths, where the kernelization status remained open even in general graphs.

On the positive side, we present a polynomial kernel for Planar Disjoint Paths parameterized by $k+tw$, where tw denotes the treewidth of the input graph. As a consequence of both our results, we rule out the possibility of a polynomial-time (Turing) treewidth reduction to $tw = k^{O(1)}$ under the same assumptions. To the best of our knowledge, this is the first hardness result of this kind. Finally, combining our kernel with the known techniques [Adler, Kolliopoulos, Krause, Lokshtanov, Saurabh, Thilikos, JCTB'17; Schrijver, SICOMP'94] yields an alternative (and arguably simpler) proof that Planar Disjoint Paths can be solved in time $2^{O(k^2)} \cdot n^{O(1)}$, matching the result of [Lokshtanov, Misra, Pilipczuk, Saurabh, Zehavi, STOC'20].

4 Open problems

4.1 Extending 1-Planar Drawings by a Few Vertices

Robert Ganian (TU Wien, AT)

A drawing of a graph G is *1-planar* (or 1-*plane*) if each edge has at most one crossing. We restrict our attention to simple drawings – in particular, the curves representing edges are not allowed to "touch" or cross through vertices which are not their endpoints. A drawing α *extends* a drawing β if $\beta \subseteq \alpha$. The following question was left open in an ICALP paper of Eiben, Ganian, Hamm, Klute and Nöllenburg [3].

Is the following problem FPT w.r.t. k?

Given a graph G, a vertex subset $X \subseteq V(G)$ of size k and a 1-planar drawing $\mathcal{H}$ of the graph $H = G - X$, does there exist a 1-planar drawing of G which extends $\mathcal{H}$?

Essentially, we are given an "almost complete" partial 1-planar drawing of a graph G and want to extend it to a full drawing, where the parameter tells us how many vertices are still missing from the partial drawing. Note that the statement in the paper is slightly more general as it also allows individual edges to be missing from H, but the real difficult/interesting case is the one above.

The problem was shown to be in XP in a follow-up MFCS paper [3], and the edge-deletion version of the problem – i.e., where we parameterize by how many edges are missing – is known to be FPT [2] (by using decomposition techniques). Several follow-ups to the latter result are known by now [4, 5, 1], but the vertex-deletion case has not been cracked so far.

References

1. S. Bhore, R. Ganian, L. Khazaliya, F. Montecchiani, and M. Nöllenburg. Extending orthogonal planar graph drawings is fixed-parameter tractable. In E. W. Chambers and J. Gudmundsson, editors, *39th International Symposium on Computational Geometry, SoCG 2023, June 12-15, 2023, Dallas, Texas, USA*, volume 258 of *LIPIcs*, pages 18:1–18:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.
2. E. Eiben, R. Ganian, T. Hamm, F. Klute, and M. Nöllenburg. Extending nearly complete 1-planar drawings in polynomial time. In J. Esparza and D. Král', editors, *45th International Symposium on Mathematical Foundations of Computer Science, MFCS 2020, August 24-28, 2020, Prague, Czech Republic*, volume 170 of *LIPIcs*, pages 31:1–31:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020.
3. E. Eiben, R. Ganian, T. Hamm, F. Klute, and M. Nöllenburg. Extending partial 1-planar drawings. In A. Czumaj, A. Dawar, and E. Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 43:1–43:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.
4. R. Ganian, T. Hamm, F. Klute, I. Parada, and B. Vogtenhuber. Crossing-optimal extension of simple drawings. In N. Bansal, E. Merelli, and J. Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 72:1–72:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021.
5. T. Hamm and P. Hliněný. Parameterised partially-predrawn crossing number. In X. Goaoc and M. Kerber, editors, *38th International Symposium on Computational Geometry, SoCG 2022, June 7-10, 2022, Berlin, Germany*, volume 224 of *LIPIcs*, pages 46:1–46:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

4.2 Disjoint Paths Reconfiguration in Planar Graphs

Yusuke Kobayashi (Kyoto University, JP)

Let $G = (V, E)$ be a graph with distinct terminals $s_1, \dots, s_k, t_1, \dots, t_k$. A tuple $\mathcal{P} = (P_1, \dots, P_k)$ of paths is called a *linkage* if they are vertex-disjoint and each P_i connects s_i and t_i for $i = 1, \dots, k$. For two linkages $\mathcal{P} = (P_1, \dots, P_k)$ and $\mathcal{Q} = (Q_1, \dots, Q_k)$, we say that $\mathcal{P}$ is *adjacent* to $\mathcal{Q}$ if there exists $i \in \{1, \dots, k\}$ such that $P_j = Q_j$ for $j \in \{1, \dots, k\} \setminus \{i\}$ and $P_i \neq Q_i$. We say that a sequence $\langle \mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_\ell \rangle$ of linkages is a *reconfiguration sequence* from $\mathcal{P}_1$ to $\mathcal{P}_\ell$ if $\mathcal{P}_i$ and $\mathcal{P}_{i+1}$ are adjacent for $i = 1, \dots, \ell - 1$. If such a sequence exists, we say that $\mathcal{P}_1$ is *reconfigurable* to $\mathcal{P}_\ell$.

In Disjoint Paths Reconfiguration, we are given a graph $G = (V, E)$, distinct terminals $s_1, \dots, s_k, t_1, \dots, t_k$, and two linkages $\mathcal{P}$ and $\mathcal{Q}$. The objective is to determine whether $\mathcal{P}$ is reconfigurable to $\mathcal{Q}$ or not. It is shown in [1] that Disjoint Paths Reconfiguration is PSPACE-complete for $k = 2$. The polynomial solvability for the planar case is open.

For fixed k, is there a polynomial-time algorithm for Disjoint Paths Reconfiguration in planar graphs? The problem is open even for $k = 2$.

References

1 T. Ito, Y. Iwamasa, N. Kakimura, Y. Kobayashi, S.-i. Maezawa, Y. Nozaki, Y. Okamoto, and K. Ozeki. Rerouting Planar Curves and Disjoint Paths. In K. Etessami, U. Feige, and G. Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*, volume 261 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 81:1–81:19, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

4.3 Three-Sets Cut-Uncut

Tuukka Korhonen (University of Copenhagen, DK)

The Three-Sets Cut-Uncut Problem. Given a graph G and three sets of terminal vertices T_1, T_2, and T_3, the goal is to remove a minimum number of edges from G such that in the resulting graph:

1. For each $i \in \{1, 2, 3\}$, every pair of vertices $u, v \in T_i$ remains connected by a path.
2. For every pair of distinct indices $i \neq j$, there is no path between any vertex $u \in T_i$ and any vertex $v \in T_j$.

(Note that it is possible that no solution exists.)

We ask: What is the parameterized complexity of the Three-Sets Cut-Uncut problem on planar graphs when parameterized by $k = |T_1| + |T_2| + |T_3|$? In particular, it remains open whether the problem is polynomial-time solvable even for the case $|T_1| = |T_2| = 1 \text{and} |T_3| = 2$.

For related literature, please see the discussion and references in [1].

References

1 M. Bentert, P. G. Drange, F. V. Fomin, P. A. Golovach, and T. Korhonen. Two-sets cut-uncut on planar graphs. In K. Bringmann, M. Grohe, G. Puppis, and O. Svensson, editors, *51st International Colloquium on Automata, Languages, and Programming, ICALP 2024, July 8-12, 2024, Tallinn, Estonia*, volume 297 of *LIPIcs*, pages 22:1–22:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024.

4.4 Beating $n!$ for Permutation CSPs

Marcin Pilipczuk (University of Warsaw, PL)

We consider a problem where we ask for an existence of a permutation $\pi : [n] \to [n]$ satisfying a number of *constraints*. Each constraint is of the form $(\pi(a) < \pi(b)) \vee (\pi(c) < \pi(d))$ for some $a, b, c, d \in [n]$.

Clearly, this problem can be solved in time roughly $n!$ by brute-force enumeration. Leif Eriksson in his master thesis [1] observed that this can be improved to roughly $((n/2)!)^2$. Does there exist an $2^{o(n \log n)}$-time algorithm?

When the constraints are of the form $\pi(a) < \pi(b)$, a simple dynamic programming algorithm solves the problem in $2^{\mathcal{O}(n)}$ time. To the best of my knowledge, the problem remains open for constraints being an alternative of three comparisons. A simple reduction from $k \times k$ Clique [2] gives an ETH $2^{\Omega(n \log n)}$ lower bound for constraints being an alternative of four comparisons.

References

1 L. Eriksson. Solving temporal CSPs via enumeration and SAT compilation, 2019. Master thesis.

2 D. Lokshtanov, D. Marx, and S. Saurabh. Slightly superexponential parameterized problems. *SIAM J. Comput.*, 47(3):675–702, 2018.

4.5 Better bounds for directed flow-augmentation

Marcin Pilipczuk (University of Warsaw, PL)

Let G be a directed (multi)graph with distinguished vertices $s, t \in V(G)$ and let k be an integer parameter. For an inclusion-wise minimal st-cut $Z \subseteq E(G)$, a set $A \subseteq V(G) \times V(G)$ is *compatible* with Z if Z remains an st-cut in $G + A := (V(G), E(G) \uplus A)$.

The flow-augmentation technique, in its basic form, can be stated as follows: given G, s, t, k as above, one can in FPT time find a family $\mathcal{A}$ of subsets of $V(G) \times V(G)$ such that for every inclusion-wise minimal st-cut Z of size at most k, there exists $A \in \mathcal{A}$ that is compatible with Z and such that, furthermore, Z becomes a *minimum* st-cut in $G + A$. The work [1] provides $\mathcal{A}$ of size bounded by

$$2^{\mathcal{O}(k^4 \log k)} \cdot (\log n)^{\mathcal{O}(k^3)} \leq 2^{\mathcal{O}(k^4 \log k)} n^{o(1)}.$$

The only known lower bound known is an observation that every important separator Z requires a different set A in the family $\mathcal{A}$. As the number of important separators of size at most k can be as large as k-th Catalan number (i.e., $\Omega(4^k k^{-3/2})$), this is also a lower bound on the size of $\mathcal{A}$.

Please provide a better upper or lower bound.

References

1 E. J. Kim, S. Kratsch, M. Pilipczuk, and M. Wahlström. Directed flow-augmentation. In S. Leonardi and A. Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 – 24, 2022*, pages 938–947. ACM, 2022.

4.6 Space efficient Min Cut

Michał Pilipczuk (University of Warsaw, PL)

The following question seems to be fundamental for very low parameterized space complexity classes.

Given an undirected n-vertex graph G, a pair of vertices s and t, and an integer k, is it possible to decide in deterministic space $f(k) + \mathcal{O}(\log n)$, for a computable f, the following question: is there a vertex subset $X \subseteq V(G) \setminus \{s, t\}$ with $|X| \leq k$ such that every s-t path intersects X.

I would conjecture that the answer should be negative, and that establishing it could provide some technique for lower bounds on parameterized space complexity.

4.7 Bisection in Planar Graphs

Saket Saurabh (The Institute of Mathematical Sciences – Chennai, IN)

In the classic **Minimum Bisection** problem, we are given as input a graph G and an integer k. The task is to determine whether there is a partition of $V(G)$ into two parts A and B such that $||A| - |B|| \leq 1$ and there are at most k edges with one endpoint in A and the other in B.

1. Is the problem polynomial time solvable on planar graphs?
2. Does the problem admit a quasi-polynomial time algorithm on planar graphs?
3. Does the problem admit subexponential (in k) time FPT algorithm?

4.8 Skew Multicut on Planar DAGs

Michał Wlodarczyk (University of Warsaw, PL)

In Skew Multicut one is given a (directed) graph G, a set $T \subseteq V(G)$ of terminals, an ordering σ over T, and a budget k. We ask whether there is a set $X \subseteq V(G) \setminus T, |X| \leq k$, such that for each pair $s, t \in T$ with $s <_\sigma t$ there is no (s, t)-path in $G - X$.

Does Skew Multicut admit a polynomial kernel on planar DAGs when parameterized by $k + |T|$?

If yes, can you make the kernel oblivious to σ? That is, the ordering σ is revealed after the compressed instance is returned.

Such an oblivious kernel would be helpful for kernelization of planar DFVS.

Participants

- Akanksha Agrawal
Indian Institute of Techology
Madras, IN
- Aditya Anand
University of Michigan –
Ann Arbor, US
- Matthias Bentert
University of Bergen, NO
- Édouard Bonnet
ENS – Lyon, FR
- Joseph Cheriyan
University of Waterloo, CA
- Éric Colin de Verdière
Gustave Eiffel University –
Marne-la-Vallée, FR
- Eduard Eiben
Royal Holloway, University of
London, GB
- Fedor V. Fomin
University of Bergen, NO
- Robert Ganian
TU Wien, AT
- Petr A. Golovach
University of Bergen, NO
- Bart Jansen
TU Eindhoven, NL
- Eun Jung Kim
KAIST – Daejeon, KR & CNRS –
Paris, FR
- Yusuke Kobayashi
Kyoto University, JP
- Christian Komusiewicz
Friedrich-Schiller-Universität
Jena, DE
- Tuukka Korhonen
University of Copenhagen, DK
- Stefan Kratsch
HU Berlin, DE
- Madhumita Kundu
University of Bergen, NO
- Euiwoong Lee
University of Michigan –
Ann Arbor, US
- Bingkai Lin
Nanjing University, CN
- William Lochet
CNRS – Montpellier, FR
- Daniel Lokshtanov
University of California –
Santa Barbara, US
- Dániel Marx
CISPA – Saarbrücken, DE
- Neeldhara Misra
Indian Institute of Techology –
Madras, IN
- Pranabendu Misra
Chennai Mathematical
Institute, IN
- Jesper Nederlof
Utrecht University, NL
- Daniel Neuen
MPI für Informatik –
Saarbrücken, DE
- Fahad Panolan
University of Leeds, GB
- Marcin Pilipczuk
University of Warsaw, PL
- Michał Pilipczuk
University of Warsaw, PL
- Thatchaphol Saranurak
University of Michigan –
Ann Arbor, US
- Ignasi Sau Valls
LIRMM, Université de
Montpellier, CNRS –
Montpellier, FR
- Saket Saurabh
The Institute of Mathematical
Sciences – Chennai, IN
- Roohani Sharma
MPI für Informatik –
Saarbrücken, DE
- Ramanujan Sridharan
University of Warwick –
Coventry, GB
- Giannos Stamoulis
University of Warsaw, PL
- Vaishali Surianarayanan
University of California –
Santa Barbara, US
- Stefan Szeider
TU Wien, AT
- Erik Jan van Leeuwen
Utrecht University, NL
- Magnus Wahlström
Royal Holloway, University of
London, GB
- Michał Wlodarczyk
University of Warsaw, PL
- Jie Xue
New York University –
Shanghai, CN