

Graph Algorithms: Cuts, Flows, and Network Design

Jason Li^{*1}, Debmalya Panigrahi^{*2}, Laura Sanita^{*3}, and Thatchaphol Saranurak^{*4}

- 1 University of California – Berkeley, US. jmli@alumni.cmu.edu
- 2 Duke University – Durham, US. debmalya.panigrahi@gmail.com
- 3 Università Bocconi – Milan, IT. laura.sanita@unibocconi.it
- 4 University of Michigan – Ann Arbor, US. thsa@umich.edu

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 23422 “Graph Algorithms: Cuts, Flows, and Network Design”. This seminar brought 25 leading researchers in graph algorithms together for a discussion of the recent progress and challenges in two areas: the design of fast algorithm for fundamental flow/cut problems and the design of approximation algorithms for basic network design problems. The seminar included several talks of varying lengths, a panel discussion, and an open problem session. In addition, sufficient time was set aside for research discussions and collaborations.

Seminar October 15–20, 2023 – <https://www.dagstuhl.de/23422>

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Network flows

Keywords and phrases approximation, graph algorithm, maximum flow, minimum cut, network design

Digital Object Identifier 10.4230/DagRep.13.10.76

1 Executive Summary

Jason Li (University of California – Berkeley, US)

Debmalya Panigrahi (Duke University – Durham, US)

Laura Sanita (Università Bocconi – Milan, IT)

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

License  Creative Commons BY 4.0 International license

© Jason Li, Debmalya Panigrahi, Laura Sanita, and Thatchaphol Saranurak

Graph algorithms are among the foundational pillars of algorithm design and combinatorial optimization. In addition to its significance as a theoretical discipline, graph algorithms are also ubiquitous in practice, with applications in essentially every scientific discipline. This has spawned research in many different directions within graph algorithms over the past few decades, and these individual research areas have come to play important roles in the evolution of algorithms research as a whole. Two particularly large and successful subdisciplines are those of *fast algorithms for flows and cuts* and *approximation algorithms for network design*. Many of the algorithmic ideas and techniques that are a standard feature of an algorithmist’s toolkit today trace their origins to groundbreaking research in these two areas spanning problems such as minimum cuts, maximum flows, Steiner trees, and the traveling salesman problem. The last few years, in particular, have been truly outstanding in achieving progress on longstanding questions in both areas. Some of the highlights include the first progress

* Editor / Organizer



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 4.0 International license

Graph Algorithms: Cuts, Flows, and Network Design, *Dagstuhl Reports*, Vol. 13, Issue 10, pp. 76–89

Editors: Jason Li, Debmalya Panigrahi, Laura Sanita, and Thatchaphol Saranurak



DAGSTUHL REPORTS

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

in decades for problems such as the traveling salesman problem (e.g., [20, 19, 17, 28, 34]), graph connectivity augmentation (e.g., [9, 10, 35]), minimum cut (e.g., [24, 11]), vertex connectivity (e.g., [23]), all-pairs minimum cuts (e.g., [3, 2, 5, 4, 6, 25, 26, 1]), and last but not the least, the recent breakthrough achieving an almost linear-time algorithm for maximum flow (and minimum cost flow) [12] (see also [14, 16, 37, 36]).

Traditionally, to a large extent, research in these two subfields has progressed independent of one another: connectivity problems such as minimum cuts and maximum flows are typically polynomial-time solvable and goal is to improve the running time (*efficiency*) of the algorithms; in contrast, network design problems such as Steiner tree and TSP are NP-hard and the goal is to obtain the best *approximation* factor in (any) polynomial time. This has meant that the two areas have focused on different sets of technical tools – the former has developed combinatorial (and more recently, continuous) methods aimed at improving running times, while the latter has focused on polyhedral techniques and the use of mathematical programming for obtaining improved approximations.

In recent years, however, this distinction between the two subfields has started to blur, and the two areas have started to move closer to one another. This is for two main reasons:

- (a) *Recent progress in foundational questions in each area has crucially relied on structural insights from the other area.* For example, one of the main new ingredients in the recent breakthrough results in approximation algorithms for the traveling salesman problem (e.g., [20, 17]) is a better understanding of the structure of near-minimum cuts (e.g., [7]) in an undirected graph. Or, recent work in the all-pairs minimum cuts problem [1] that advances the state of the art for this problem after 60 years crucially makes use of Steiner tree approximations [27]. Or, cut matching games originally devised for sparsest cut approximations [22] have led to fast expander decompositions (e.g., [30]) that, in turn, play a crucial role in recent progress in deterministic minimum cut algorithms [24].
- (b) *There is growing interest in understanding approximation-efficiency tradeoffs in graph algorithms.* Graph sparsification (e.g., [8, 15, 32, 33]) has emerged as a standard tool that “compresses” an arbitrary graph into a sparse subgraph (called the *sparsifier*) while approximately preserving the values of all cuts in the graph. This naturally leads to an approximation-efficiency tradeoff by running existing algorithms on the sparsifier rather than on the input graph. But, beyond the black box use of sparsifiers, efficiency at the expense of mild approximation has been employed as a technical tool to breach longstanding running time barrier in recent years, and has often eventually led to faster exact algorithms as well. A famous example is the maximum flow problem in undirected graphs, where nearly-linear time approximation algorithms were designed in the last decade [21, 31, 29] and has eventually resulted in the very recent breakthrough achieving an almost-linear time exact algorithm [12]. Another recent example is the all-pairs minimum cuts problem for which the first paper to breach the 60-year old running time bound of Gomory and Hu [18] incurred a mild approximation [25], but this has now led to a faster exact algorithm as well [1]. Finally, understanding the efficiency-approximation tradeoff is an important goal for NP-hard network design problems such as Steiner tree [27] and Steiner forest [13], and this, in turn, has implications for minimum cut problems [1].

The goal of this seminar was to bring the leading researchers from these two communities of fast flow/cut algorithms and approximation algorithms for network design together for an exchange of ideas and knowledge, and a discussion of the major technical challenges in each research area.

Seminar Structure and Participants

The seminar brought together 25 researchers from the two communities highlighted above, roughly equally split between the two areas. There was also a mix of senior and junior participants, ranging from senior members of the community to current PhD students and postdoctoral researchers. In terms of gender balance, around 20% of the attendees were female. (The organizers had originally planned for a more equitable balance, but there were several late retractions, primarily due to geopolitical reasons, that affected the gender ratio.)

There were 17 scheduled talks, divided into long (60 minutes) and short (30 minutes) presentations. There was also an open problem session and a panel discussion on the future directions for the community. The schedule left plenty of time for collaboration and free discussion among the participants.

Outcomes

The main objective of the seminar was to provide a forum for the exchange of ideas between the research communities of fast flow/cut algorithms and approximation algorithms for network design. These are adjoining areas where researchers have a working knowledge of, and appreciation for, each other's work. As expected, the seminar led to cohesive interactions and meaningful discussions. Individual research talks and afternoon breaks created concrete opportunities for learning about recent progress in each other's areas and fostering collaborations. The open problem session highlighted the major research challenges in the two areas, which is particularly beneficial for junior members of the community who attended the seminar. The panel discussion allowed the participants to reflect on and discuss higher-level questions about the research directions that the communities should pursue in the near future. Overall, we believe that the seminar played an important role in community building, research collaborations, and in shaping the two research areas for the foreseeable future.

The organizers would like to thank the Scientific Directorate and the administration of the Dagstuhl Center for their amazing support in the organization of the Dagstuhl Seminar, and for supporting this important research area.

References

- 1 Amir Abboud, Robert Krauthgamer, Jason Li, Debmalya Panigrahi, Thatchaphol Saranurak, and Ohad Trabelsi. Gomory-Hu tree in subcubic time. *CoRR*, abs/2111.04958, 2021.
- 2 Amir Abboud, Robert Krauthgamer, and Ohad Trabelsi. Cut-equivalent trees are optimal for min-cut queries. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020*. IEEE Computer Society, 2020.
- 3 Amir Abboud, Robert Krauthgamer, and Ohad Trabelsi. New algorithms and lower bounds for all-pairs max-flow in undirected graphs. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5–8, 2020*, pages 48–61. SIAM, 2020.
- 4 Amir Abboud, Robert Krauthgamer, and Ohad Trabelsi. APMF < APSP? Gomory-Hu tree for unweighted graphs in almost-quadratic time. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7–10, 2022*, pages 1135–1146. IEEE, 2021.
- 5 Amir Abboud, Robert Krauthgamer, and Ohad Trabelsi. Subcubic algorithms for Gomory-Hu tree in unweighted graphs. In Samir Khuller and Virginia Vassilevska Williams, editors,

- STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21–25, 2021*, pages 1725–1737. ACM, 2021.
- 6 Amir Abboud, Robert Krauthgamer, and Ohad Trabelsi. Friendly cut sparsifiers and faster gomory-hu trees. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9–12, 2022*, pages 3630–3649. SIAM, 2022.
 - 7 András A Benczúr. A representation of cuts within $6/5$ times the edge connectivity with applications. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 92–102. IEEE, 1995.
 - 8 András A. Benczúr and David R. Karger. Randomized approximation schemes for cuts and flows in capacitated graphs. *SIAM J. Comput.*, 44(2):290–319, 2015.
 - 9 Jaroslav Byrka, Fabrizio Grandoni, and Afrouz Jabal Ameli. Breaching the 2-approximation barrier for connectivity augmentation: a reduction to Steiner tree. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 815–825, 2020.
 - 10 Federica Cecchetto, Vera Traub, and Rico Zenklusen. *Bridging the Gap between Tree and Connectivity Augmentation: Unified and Stronger Approaches*, page 370–383. Association for Computing Machinery, New York, NY, USA, 2021.
 - 11 Ruoxu Cen, Jason Li, Danupon Nanongkai, Debmalya Panigrahi, Thatchaphol Saranurak, and Kent Quanrud. Minimum cuts in directed graphs via partial sparsification. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7–10, 2022*, pages 1147–1158. IEEE, 2021.
 - 12 Li Chen, Rasmus Kyng, Yang P. Liu, Richard Peng, Maximilian Probst Gutenberg, and Sushant Sachdeva. Maximum flow and minimum-cost flow in almost-linear time. *CoRR*, abs/2203.00671, 2022.
 - 13 Richard Cole, Ramesh Hariharan, Moshe Lewenstein, and Ely Porat. A faster implementation of the goemans-williamson clustering algorithm. In S. Rao Kosaraju, editor, *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7–9, 2001, Washington, DC, USA*, pages 17–25. ACM/SIAM, 2001.
 - 14 Sally Dong, Yu Gao, Gramoz Goranci, Yin Tat Lee, Richard Peng, Sushant Sachdeva, and Guanghao Ye. Nested dissection meets ipms: Planar min-cost flow in nearly-linear time. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9–12, 2022*, pages 124–153. SIAM, 2022.
 - 15 Wai Shing Fung, Ramesh Hariharan, Nicholas J. A. Harvey, and Debmalya Panigrahi. A general framework for graph sparsification. *SIAM J. Comput.*, 48(4):1196–1223, 2019.
 - 16 Yu Gao, Jason Li, Danupon Nanongkai, Richard Peng, Thatchaphol Saranurak, and Sorrachai Yingchareonthawornchai. Deterministic graph cuts in subquadratic time: Sparse, balanced, and k -vertex. *arXiv preprint arXiv:1910.07950*, 2019.
 - 17 Shayan Oveis Gharan, Amin Saberi, and Mohit Singh. A randomized rounding approach to the traveling salesman problem. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22–25, 2011*, pages 550–559. IEEE Computer Society, 2011.
 - 18 Ralph E Gomory and Tien Chung Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570, 1961.
 - 19 Anna R. Karlin, Nathan Klein, and Shayan Oveis Gharan. An improved approximation algorithm for TSP in the half integral case. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22–26, 2020*, pages 28–39. ACM, 2020.

- 20 Anna R. Karlin, Nathan Klein, and Shayan Oveis Gharan. A (slightly) improved approximation algorithm for metric TSP. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21–25, 2021*, pages 32–45. ACM, 2021.
- 21 Jonathan A. Kelner, Yin Tat Lee, Lorenzo Orecchia, and Aaron Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5–7, 2014*, pages 217–226, 2014.
- 22 Rohit Khandekar, Satish Rao, and Umesh V. Vazirani. Graph partitioning using single commodity flows. *J. ACM*, 56(4):19:1–19:15, 2009.
- 23 Jason Li, Danupon Nanongkai, Debmalya Panigrahi, Thatchaphol Saranurak, and Sorrachai Yingchareonthawornchai. Vertex connectivity in poly-logarithmic max-flows. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21–25, 2021*, pages 317–329. ACM, 2021.
- 24 Jason Li and Debmalya Panigrahi. Deterministic min-cut in poly-logarithmic max-flows. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16–19, 2020*, pages 85–92. IEEE, 2020.
- 25 Jason Li and Debmalya Panigrahi. Approximate gomory-hu tree is faster than $n - 1$ max-flows. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21–25, 2021*, pages 1738–1748. ACM, 2021.
- 26 Jason Li, Debmalya Panigrahi, and Thatchaphol Saranurak. A nearly optimal all-pairs min-cuts algorithm in simple graphs. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7–10, 2022*, pages 1124–1134. IEEE, 2021.
- 27 Kurt Mehlhorn. A faster approximation algorithm for the Steiner problem in graphs. *Information Processing Letters*, 27(3):125–128, 1988.
- 28 Tobias Mömke and Ola Svensson. Approximating graphic TSP by matchings. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22–25, 2011*, pages 560–569. IEEE Computer Society, 2011.
- 29 Richard Peng. Approximate undirected maximum flows in $O(\text{mpolylog}(n))$ time. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10–12, 2016*, pages 1862–1867, 2016.
- 30 Thatchaphol Saranurak and Di Wang. Expander decomposition and pruning: Faster, stronger, and simpler. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6–9, 2019*, pages 2616–2635. SIAM, 2019.
- 31 Jonah Sherman. Nearly maximum flows in nearly linear time. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26–29 October, 2013, Berkeley, CA, USA*, pages 263–269. IEEE Computer Society, 2013.
- 32 Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM J. Comput.*, 40(6):1913–1926, 2011.
- 33 Daniel A. Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM J. Comput.*, 40(4):981–1025, 2011.
- 34 Vera Traub and Jens Vygen. Approaching $3/2$ for the s - t -path TSP. *J. ACM*, 66(2):14:1–14:17, 2019.

- 35 Vera Traub and Rico Zenklusen. Local search for weighted tree augmentation and steiner tree. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 – 12, 2022*, pages 3253–3272. SIAM, 2022.
- 36 Jan van den Brand, Yin Tat Lee, Yang P. Liu, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang. Minimum cost flows, mdps, and ℓ_1 -regression in nearly linear time for dense instances. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21–25, 2021*, pages 859–869. ACM, 2021.
- 37 Jan van den Brand, Yin Tat Lee, Danupon Nanongkai, Richard Peng, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang. Bipartite matching in nearly-linear time on moderately dense graphs. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16–19, 2020*, pages 919–930. IEEE, 2020.

2 Table of Contents

Executive Summary

Jason Li, Debmalya Panigrahi, Laura Sanita, and Thatchaphol Saranurak 76

Overview of Talks

Fast Algorithms via Dynamic-Oracle Matroids <i>Joakim Blikstad</i>	83
The girth problem and its variants in network design <i>Greg Bodwin</i>	83
Differentially Private Densest Subgraph <i>Michael Dinitz</i>	84
On Dynamic Graph Approximations: The case of j -Trees <i>Gramoz Goranci</i>	84
Approximation Algorithms for 2-Connectivity <i>Fabrizio Grandoni</i>	84
Polylogarithmic Universal Steiner Trees and Strong Sparse Partition Hierarchies <i>Ellis Hershkowitz</i>	85
All-Pairs Minimum Cuts in Almost-Linear Time <i>Jason Li</i>	85
Recent Advances on Maximum Flows <i>Yang P. Liu</i>	85
Fair Division of Indivisible Goods and Graph Algorithms <i>Kurt Mehlhorn</i>	86
Hopsets and Algorithmic Applications <i>Yasamin Nazari</i>	86
Algorithms for Coloring Tournaments <i>Alantha Newman</i>	86
Quotient sparsification for submodular functions <i>Kent Quanrud</i>	87
Using Isolating Mincuts for Fast Graph Algorithms: A tutorial <i>Thatchaphol Saranurak</i>	87
Decremental Bipartite Matching <i>Aaron Sidford</i>	87
Approximation Algorithms for Connectivity Augmentation Problems <i>Vera Traub</i>	88
Faster Deterministic Vertex Connectivity Algorithms <i>Sorrachai Yingchareonthawornchai</i>	88
Participants	89

3 Overview of Talks

3.1 Fast Algorithms via Dynamic-Oracle Matroids

Joakim Blikstad (KTH Royal Institute of Technology – Stockholm, SE)

License © Creative Commons BY 4.0 International license
© Joakim Blikstad

We initiate the study of matroid problems in a new oracle model called dynamic oracle. Our algorithms in this model lead to new bounds for some classic problems, and a “unified” algorithm whose performance matches previous results developed in various papers. We also show a lower bound that answers some open problems from a few decades ago. We show an algorithm with $\tilde{O}(n + r\sqrt{r})$ dynamic-rank-query and time complexities for the matroid union problem. This implies an improvement over the traditional rank-query complexity for matroid union. As an interesting special case, it is the first algorithm which, in sufficiently dense graphs, achieves nearly linear time $\tilde{O}(m + n\sqrt{n})$ for the problem of finding k disjoint spanning trees in a graph. We also show simple super-linear ($\Omega(n \log n)$) query lower bounds for matroid intersection in our dynamic-rank-oracle and the traditional independence-query models; the latter improves the previous $\log 2(3)n - o(n)$ bound.

3.2 The girth problem and its variants in network design

Greg Bodwin (University of Michigan – Ann Arbor, US)

License © Creative Commons BY 4.0 International license
© Greg Bodwin

The girth problem as a central open question in extremal combinatorics, which asks to determine the maximum possible number of edges in an n -node graph whose girth (shortest cycle length) is larger than k . In 1993, a seminal work of Althöfer, Das, Dobkin, Joseph, and Soares showed that determining the size/stretch tradeoff available to graph spanners is equivalent to settling the girth problem. This set in motion a line of research that seeks to understand sparse graph structures by analyzing their forbidden patterns, and using these patterns to invoke ideas from extremal combinatorics.

This talk will survey some successes of the method, including other objects that can also be reduced to the girth problem (distance oracles, fault-tolerant spanners), and related problems that capture other objects in network design, such as the weighted girth problem (light spanners), the bipartite girth problem (distance preservers, reachability preservers), the forbidden biclique problem (directed distance preservers), and the bridge girth problem (reachability preservers, flow-cut gaps). We will survey the common technical threads in these arguments, and overview the many open problems that remain.

3.3 Differentially Private Densest Subgraph

Michael Dinitz (Johns Hopkins University – Baltimore, US)

License  Creative Commons BY 4.0 International license
© Michael Dinitz

Joint work of Satyen Kale, Silvio Lattanzi, and Sergei Vassilvitskii
Main reference Michael Dinitz, Satyen Kale, Silvio Lattanzi, Sergei Vassilvitskii: “Improved Differentially Private Densest Subgraph: Local and Purely Additive”, CoRR, Vol. abs/2308.10316, 2023.
URL <https://doi.org/10.48550/ARXIV.2308.10316>

We study the Densest Subgraph problem under the additional constraint of differential privacy. In the LEDP (local edge differential privacy) model, introduced recently by Dhulipala et al. [FOCS 2022], we give an (ϵ, δ) -differentially private algorithm with no multiplicative loss: the loss is purely additive. This is in contrast to every previous private algorithm for densest subgraph (local or centralized), all of which incur some multiplicative loss as well as some additive loss. Moreover, our additive loss matches the best-known previous additive loss (in any version of differential privacy) when $1/\delta$ is at least polynomial in n , and in the centralized setting we can strengthen our result to provide better than the best-known additive loss. Additionally, we give a different algorithm that is ϵ -differentially private in the LEDP model which achieves a multiplicative ratio arbitrarily close to 2, along with an additional additive factor. This improves over the previous multiplicative 4-approximation in the LEDP model. Finally, we conclude with extensions of our techniques to both the node-weighted and the directed versions of the problem.

3.4 On Dynamic Graph Approximations: The case of j-Trees

Gramoz Goranci (Universität Wien, AT)

License  Creative Commons BY 4.0 International license
© Gramoz Goranci


Joint work of Gramoz Goranci, Li Chen, Monika Henzinger, Richard Peng, Thatchaphol Saranurak

Approximating graphs by j-trees is a powerful algorithmic paradigm that has proven effective in significantly speeding up cut-based optimization problems, approximate maximum flows, and exact minimum cost-flow computations.

In this talk, I will explain how to dynamically maintain j-trees and discuss some of the implications of this result.

3.5 Approximation Algorithms for 2-Connectivity

Fabrizio Grandoni (SUPSI – Lugano, CH)

License  Creative Commons BY 4.0 International license
© Fabrizio Grandoni

Given an undirected graph G , the 2-edge-connected spanning subgraph problem is to compute a subgraph S of G with the minimum possible number of edges which is 2-edge-connected, i.e., removing any edge from S leaves a connected graph (spanning all the nodes). The 2-vertex-connected spanning subgraph problem is defined similarly w.r.t. 2-vertex-connectivity. In this talk I will illustrate some recent progress on approximation algorithms for these two problems.

3.6 Polylogarithmic Universal Steiner Trees and Strong Sparse Partition Hierarchies

Ellis Hershkowitz (Brown University – Providence, US)

License © Creative Commons BY 4.0 International license
© Ellis Hershkowitz

Joint work of Ellis Hershkowitz, Costas Busch, Da Qi Chen, Arnold Filtser, Daniel Hathcock, Rajmohan Rajaraman

An alpha-approximate universal Steiner tree (UST) of a graph G is a spanning tree T such that, for any vertex terminal subset S , the minimal subtree of T connecting S is within an alpha factor of the cost of the cheapest Steiner tree in G connecting S . Alpha-approximate USTs immediately give alpha-approximations for well-studied variants of Steiner tree such as online or oblivious Steiner tree. Sub-linear-approximate USTs are known but neither the existence of nor poly-time algorithms for computing poly-logarithmic-approximate USTs were previously known.

In this talk, I will discuss the first construction of poly-logarithmic USTs. The result is based on new constructions of poly-logarithmic-quality graph hierarchies called strong sparse partitions which may be interesting in their own right. Roughly, strong sparse partitions provide deterministic guarantees on how often balls of particular radii are cut.

3.7 All-Pairs Minimum Cuts in Almost-Linear Time

Jason Li (University of California – Berkeley, US)

License © Creative Commons BY 4.0 International license
© Jason Li

Joint work of Jason Li, Amir Abboud, Robert Krauthgamer, Debmalya Panigrahi, Thatchaphol Saranurak, Ohad Trabelsi

Main reference Amir Abboud, Robert Krauthgamer, Jason Li, Debmalya Panigrahi, Thatchaphol Saranurak, Ohad Trabelsi: “Gomory-Hu Tree in Subcubic Time”, CoRR, Vol. abs/2111.04958, 2021.

URL <https://arxiv.org/abs/2111.04958>

We present recent progress on the problem of computing all-pairs minimum cuts, and more generally the Gomory-Hu tree of a graph. In particular, we obtain the first running time improvement since Gomory and Hu’s original algorithm in 1961, as well as a subsequent improvement to almost-linear time, resolving the complexity of this problem. We discuss important tools that paved the way for the discovery of the algorithm, most notably the isolating cuts problem and a reduction to single-source minimum cut.

3.8 Recent Advances on Maximum Flows

Yang P. Liu (Institute for Advanced Study – Princeton, US)

License © Creative Commons BY 4.0 International license
© Yang P. Liu

We discuss extensions of the recent almost-linear-time maximum flow and mincost flow algorithm to dynamic and deterministic settings. Joint work with Jan van den Brand, Li Chen, Rasmus Kyng, Richard Peng, Maximilian Probst Gutenberg, Sushant Sachdeva, and Aaron Sidford.

3.9 Fair Division of Indivisible Goods and Graph Algorithms

Kurt Mehlhorn (MPI für Informatik – Saarbrücken, DE)


License  Creative Commons BY 4.0 International license
© Kurt Mehlhorn

A set of indivisible goods is to be allocated to a group of agents, e.g, a car, a computer, a tooth-brush. Each agent has a valuation over sets of goods. There are only two restrictions on a valuation. The value of the empty set is zero and more is better. We want to allocate the goods in a fair manner. Three notions of fairness have emerged: envy-freeness, fair share, and maximum Nash-value. We discuss exact and approximate existence and complexity.

Some of the algorithms have a strong connection to matchings.

3.10 Hopsets and Algorithmic Applications

Yasamin Nazari (VU Amsterdam, NL)

License  Creative Commons BY 4.0 International license
© Yasamin Nazari

Given a weighted graph G , a hopset of hopbound β and stretch $(1 + \epsilon)$ is a set of edges such that for any pair of nodes u and v in G , there is a path in $G \cup H$ of at most β hops whose length is within a $(1 + \epsilon)$ factor of the distance between u and v in G . Hopsets have recently found many applications in fast distance computation in various computational models such as dynamic, parallel and distributed models. This talk gives an introduction to hopsets for undirected graphs and their algorithmic applications in these settings. We conclude with open problems on applications of directed hopsets.

3.11 Algorithms for Coloring Tournaments

Alantha Newman (Grenoble INP, FR)

License  Creative Commons BY 4.0 International license
© Alantha Newman

Joint work of Alantha Newman, Felix Klingelhoefer

A k -coloring of a tournament is a partition of its vertices into k acyclic sets. Deciding if a tournament is 2-colorable is NP-hard. A natural problem, akin to that of coloring a 3-colorable graph with few colors, is to color a 2-colorable tournament with few colors. This problem does not seem to have been addressed before, although it is a special case of coloring a 2-colorable 3-uniform hypergraph with few colors, which is a well-studied problem with super-constant lower bounds.

We present a new efficient decomposition lemma for tournaments, which we use to design polynomial-time algorithms to color various classes of tournaments with few colors, notably, to color a 2-colorable tournament with ten colors. We also use this lemma to prove equivalence between the problems of coloring 3-colorable tournaments and coloring 3-colorable graphs with constantly many colors. For the classes of tournaments considered, we complement our upper bounds with strengthened lower bounds, painting a comprehensive picture of the algorithmic and complexity aspects of coloring tournaments.

3.12 Quotient sparsification for submodular functions

Kent Quanrud (Purdue University – West Lafayette, US)

License © Creative Commons BY 4.0 International license
© Kent Quanrud

Main reference Kent Quanrud: “Quotient sparsification for submodular functions”, in Proc. of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), Alexandria, VA, USA, pp. 5209–5248, 2024.

URL <https://doi.org/10.1137/1.9781611977912.187>

Graph sparsification has been an important topic with many structural and algorithmic consequences. Recently hypergraph sparsification has come to the fore and has seen exciting progress. In this paper we take a fresh perspective and show that they can be both derived as corollaries of a general theorem on sparsifying matroids and monotone submodular functions.

Quotients of matroids and monotone submodular functions generalize k -cuts in graphs and hypergraphs. We show that a weighted ground set of a monotone submodular function f can be sparsified while approximately preserving the weight of every quotient of f with high probability in randomized polynomial time.

This theorem conceptually unifies cut sparsifiers for undirected graphs [BK15] with other interesting applications. One basic application is to reduce the number of elements in a matroid while preserving the weight of every quotient of the matroid. For hypergraphs, the theorem gives an alternative approach to the hypergraph cut sparsifiers obtained recently in [CKN20], that also preserves all k -cuts. Another application is to reduce the number of points in a set system while preserving the weight of the union of every collection of sets. We also present algorithms that sparsify hypergraphs and set systems in nearly linear time, and sparsify matroids in nearly linear time and queries in the rank oracle model.

3.13 Using Isolating Mincuts for Fast Graph Algorithms: A tutorial

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

License © Creative Commons BY 4.0 International license
© Thatchaphol Saranurak

The Isolating Mincuts algorithm is a new technique recently introduced by [Li and Panigrahi] and [Abboud Krauthgamer Trabelsi]. In the last three years, they found more than ten applications in fast graph algorithms. I will give a gentle tutorial on this technique.

3.14 Decremental Bipartite Matching

Aaron Sidford (Stanford University, US)

License © Creative Commons BY 4.0 International license
© Aaron Sidford

Joint work of Aaron Sidford, Arun Jambulapati, Yujia Jin, Kevin Tian

Main reference Arun Jambulapati, Yujia Jin, Aaron Sidford, Kevin Tian: “Regularized Box-Simplex Games and Dynamic Decremental Bipartite Matching”, CoRR, Vol. abs/2204.12721, 2022.


URL <https://doi.org/10.48550/ARXIV.2204.12721>

Maintaining an approximately maximum matching in a dynamic graph is a fundamental problem in data structures and algorithmic graph theory. In this talk I will discuss recent progress in the special case of decremental bipartite matching, where the only dynamic

updates are deleting edges from an initial bipartite graph. In particular, I will discuss how faster runtimes were obtained by reducing this dynamic problem to solving a sequence of natural convex optimization problems.

3.15 Approximation Algorithms for Connectivity Augmentation Problems

Vera Traub (Universität Bonn, DE)


License  Creative Commons BY 4.0 International license
 © Vera Traub

Augmentation problems are a fundamental class of network design problems. They ask about the cheapest way to increase the (edge-)connectivity of a graph by adding edges among a given set of options. One of the most elementary and intensely studied augmentation problems is the (Weighted) Tree Augmentation Problem. Here, a spanning tree has to be augmented into a 2-edge-connected graph.

Classic techniques for network design yield 2-approximation algorithms for a wide class of augmentation problems. For the Unweighted Tree Augmentation Problem, better-than-2 approximations are known for more than 20 years. However, only recently the first better-than-2 approximations have been found for the more general Unweighted Connectivity Augmentation Problem and Weighted Tree Augmentation Problem. In this talk we will discuss these recent advances.

3.16 Faster Deterministic Vertex Connectivity Algorithms

Sorrachai Yingchareonthawornchai (University of California – Berkeley, US)

License  Creative Commons BY 4.0 International license
 © Sorrachai Yingchareonthawornchai

Joint work of Sorrachai Yingchareonthawornchai, Yonggang Jiang, Chaitanya Nalam, Thatchaphol Saranurak

An n -vertex m -edge graph is k -vertex connected if it cannot be disconnected by deleting less than k vertices. After more than half a century of intensive research, the result by [Li et al. STOC'21] finally gave a randomized algorithm for checking k -connectivity in near-optimal $\widehat{O}(m)$ time where $\widehat{O}(\cdot)$ to hide an $n^{o(1)}$ factor.

Deterministic algorithms, unfortunately, have remained much slower even if we assume a linear-time max-flow algorithm: they either require at least $\Omega(mn)$ time [Even'75; Henzinger Rao and Gabow, FOCS'96; Gabow, FOCS'00] or assume that $k = o(\sqrt{\log n})$ [Saranurak and Yingchareonthawornchai, FOCS'22]. In this talk, I will describe a deterministic algorithm for checking k -vertex connectivity in time proportional to making $\min\{k^2, n\}$ max-flow calls, and, hence, in $\widehat{O}(m \min\{k^2, n\})$ time using the deterministic max-flow algorithm by [Brand et al. FOCS'23]. Our algorithm gives the first almost-linear-time bound for all k where $\sqrt{\log n} \leq k \leq n^{o(1)}$ and subsumes up to a sub-polynomial factor the long-standing state-of-the-art algorithm by [Even'75] which requires $O(n + k^2)$ max-flow calls. For large k , the algorithm runs in $\widehat{O}(mn)$ time, which improves over the state-of-the-art deterministic $\widehat{O}(mn^{1.5})$ -time algorithm [Gabow, FOCS'00]. Our key technique is based on Ramanujan expanders and derandomization of the kernelization technique of [Li et al. STOC'21] for which their kernel construction was randomized.

Participants

- Joakim Blikstad
KTH Royal Institute of
Technology – Stockholm, SE
- Greg Bodwin
University of Michigan –
Ann Arbor, US
- Parinya Chalermsook
Aalto University, FI
- Michael Dinitz
Johns Hopkins University –
Baltimore, US
- Gramoz Goranci
Universität Wien, AT
- Fabrizio Grandoni
SUPSI – Lugano, CH
- Anupam Gupta
Carnegie Mellon University –
Pittsburgh, US
- Ellis Hershkowitz
Brown University –
Providence, US
- Felix Hommelsheim
Universität Bremen, DE
- Alexandra Lassota
MPI für Informatik –
Saarbrücken, DE
- Jason Li
University of California –
Berkeley, US
- Yang P. Liu
Institute for Advanced Study –
Princeton, US
- Kurt Mehlhorn
MPI für Informatik –
Saarbrücken, DE
- Danupon Nanongkai
MPI für Informatik –
Saarbrücken, DE
- Yasamin Nazari
VU Amsterdam, NL
- Alantha Newman
Grenoble INP, FR
- Debmalya Panigrahi
Duke University – Durham, US
- Kent Quanrud
Purdue University –
West Lafayette, US
- Laura Sanita
Università Bocconi – Milan, IT
- Thatchaphol Saranurak
University of Michigan –
Ann Arbor, US
- Aaron Sidford
Stanford University, US
- Joachim Spoerhase
University of Sheffield, GB
- Vera Traub
Universität Bonn, DE
- László A. Végh
London School of Economics &
Political Science, GB
- Sorrachai
Yingchareonthawornchai
University of California –
Berkeley, US

