Report from Dagstuhl Seminar 22461

# Dynamic Graph Algorithms

**Aaron Bernstein[*1], Shiri Chechik[*2], Sebastian Forster[*3], Tsvi Kopelowitz[*4], Yasamin Nazari[†5], and Nicole Wein[†6]**

1 **Rutgers University – New Brunswick, US.** aaron.bernstein@rutgers.edu
2 **Tel Aviv University, IL.** shiri.chechik@gmail.com
3 **Universität Salzburg, AT.** forster@cs.sbg.ac.at
4 **Bar-Ilan University – Ramat Gan, IL.** kopelot@gmail.com
5 **Universität Salzburg, AT.** ynazari@cs.sbg.ac.at
6 **Rutgers University – Piscataway, US.** nwein@mit.edu

**Abstract**

This report documents the program and the outcomes of Dagstuhl Seminar 22461 "Dynamic Graph Algorithms", which took place from November 13 to November 18, 2022.

The field of dynamic graph algorithms studies algorithms for processing graphs that are changing over time. Formally, the goal is to process an interleaved sequence of update and query operations, where an update operation changes the input graph (e.g. inserts/deletes an edge), while the query operation is problem-specific and asks for some information about the current graph – for example, an s-t path, or a minimum spanning tree. The field has evolved rapidly over the past decade, and this Dagstuhl Seminar brought together leading researchers in dynamic algorithms and related areas of graph algorithms.

**Seminar** November 13–18, 2022 – http://www.dagstuhl.de/22461
**2012 ACM Subject Classification** Theory of computation → Dynamic graph algorithms
**Keywords and phrases** dynamic graphs, graph algorithms
**Digital Object Identifier** 10.4230/DagRep.12.11.45

## 1 Executive Summary

*Sebastian Forster (Universität Salzburg, AT)*
*Aaron Bernstein (Rutgers University – New Brunswick, US)*
*Shiri Chechik (Tel Aviv University, IL)*
*Tsvi Kopelowitz (Bar-Ilan University – Ramat Gan, IL)*



The field of dynamic graph algorithms has evolved rapidly over the past decade. New techniques, new problems, new lower bounds, and new approaches have yielded an extremely fruitful research environment. This seminar provided a venue for the community to establish the main challenges that remain and to actively shape the direction of the field going forward.

The seminar brought together the leading researchers and "rising stars" of the field as well as experts in "neighboring" areas such as distributed algorithms, parallel algorithms, streaming algorithms, online algorithms, approximation algorithms, data structures, fine-grained and parameterized complexity, and optimization. Many participants were also actively researching algorithms engineering for dynamic graph problems, which added interesting perspectives on the prevalent theory-practice gap and fundamental methodological challenges.

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


Dynamic Graph Algorithms, *Dagstuhl Reports*, Vol. 12, Issue 11, pp. 45–65
Editors: Aaron Bernstein, Shiri Chechik, Sebastian Forster, Tsvi Kopelowitz, Yasamin Nazari, and Nicole Wein
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Several participants gave talks that were highlighting "cutting-edge" advances in the field, including results that very recently appeared in top theory venues such as STOC, FOCS and SODA (and in some cases won the best paper award). Some participants explored connections to other related areas in algorithms research such as distributed and streaming algorithms or differential privacy. Many of the talks also included suggestions on future directions and highlighted the main challenges in the area.

In two open problem sessions, the participants explicitly identified several central open problems that continue to resist progress. We hope that the resulting list of open problems will be a valuable resource for future research in this field.

During and after the sessions, attendees participated in vibrant discussions. Such interactions enhanced the overall experience and made a clear distinction to a traditional conference-like format. In addition to the research activities, there were many social activities (such as board games, poker, music night, hiking, and ping pong) that made the workshop also a great networking opportunity, in particular for the relatively large fraction of participants who were first-time visitors to Schloss Dagstuhl.

## Acknowledgments

The organizers would like to thank Yasamin Nazari and Nicole Wein for helping edit this report, Monika Henzinger for helping develop the concept of this seminar, and the Dagstuhl team for their first-rate support.

## 2 Table of Contents

# 3 Overview of Talks

## 3.1 Dynamic Distributed Subgraph Finding

*Keren Censor-Hillel (Technion – Haifa, IL)*



This talk is a discussion on exciting recent progress in distributed subgraph finding (static and dynamic) and describing some of the many intriguing open questions.

## 3.2 Stronger 3-SUM Lower Bounds for Approximate Distance Oracles via Additive Combinatorics

*Nick Fischer (MPI für Informatik – Saarbrücken, DE)*


Joint work of Amir Abboud, Karl Bringmann, Nick Fischer
Main reference Amir Abboud, Karl Bringmann, Nick Fischer: "Stronger 3-SUM Lower Bounds for Approximate Distance Oracles via Additive Combinatorics", CoRR, Vol. abs/2211.07058, 2022.
URL https://doi.org/10.48550/arXiv.2211.07058

In this work we prove conditional lower bounds against approximate distance oracles in static and dynamic settings. The seminal Thorup-Zwick distance oracles achieve stretch $2k \pm O(1)$ after preprocessing a graph in $O(mn^{1/k})$ time. For the same stretch, and assuming the query time is $n^{o(1)}$, Abboud, Bringmann, Khoury and Zamir (STOC '22) proved an $\Omega(m^{1+\frac{1}{12.7552 \cdot k}})$ lower bound on the preprocessing time; we improve it to $\Omega(m^{1+1/2k})$ which is only a factor 2 away from the upper bound. Additionally, we obtain tight bounds for stretch $3 - \epsilon$ and higher lower bounds for dynamic shortest paths.

## 3.3 Deterministic Incremental APSP with Polylogarithmic Update Time and Stretch

*Sebastian Forster (Universität Salzburg, AT)*


Joint work of Sebastian Forster, Yasamin Nazari, Maximilian Probst Gutenberg
Main reference Sebastian Forster, Yasamin Nazari, Maximilian Probst Gutenberg: "Deterministic Incremental APSP with Polylogarithmic Update Time and Stretch", CoRR, Vol. abs/2211.04217, 2022.
URL https://doi.org/10.48550/arXiv.2211.04217

We provide the first deterministic data structure that given a weighted undirected graph undergoing edge insertions, processes each update with polylogarithmic amortized update time and answers queries for the distance between any pair of vertices in the current graph with a polylogarithmic approximation in $O(\log \log n)$ time.

Prior to this work, no data structure was known for partially dynamic graphs, i.e., graphs undergoing either edge insertions or deletions, with less than $n^{o(1)}$ update time except for dense graphs, even when allowing randomization against oblivious adversaries or considering only single-source distances.

## 3.4 Incremental Approximate Maximum Flow in $m^{1/2+o(1)}$pdate time

*Gramoz Goranci (ETH Zürich, CH)*


**Joint work of** Gramoz Goranci, Monika Henzinger
**Main reference** Gramoz Goranci, Monika Henzinger: "Incremental Approximate Maximum Flow in $m^{1/2+o(1)}$ update time", CoRR, Vol. abs/2211.09606, 2022.
**URL** https://doi.org/10.48550/arXiv.2211.09606

We show a $(1+\epsilon)$-approximation algorithm for maintaining maximum $s$-$t$ flow under $m$ edge insertions in $m^{1/2+o(1)}\epsilon^{-1/2}$ amortized update time for directed, unweighted graphs. This constitutes the first sublinear dynamic maximum flow algorithm in general sparse graphs with arbitrarily good approximation guarantee.

## 3.5 Fully Dynamic Graph Algorithms in Practice: (Some) Lessons Learned

*Kathrin Hanauer (Universität Wien, AT)*


**Joint work of** Kathrin Hanauer, Monika Henzinger, Christian Schulz, Leonhard Paul Sidl
**Main reference** Kathrin Hanauer, Monika Henzinger, Christian Schulz: "Fully Dynamic Single-Source Reachability in Practice: An Experimental Study", in Proc. of the Symposium on Algorithm Engineering and Experiments, ALENEX 2020, Salt Lake City, UT, USA, January 5-6, 2020, pp. 106–119, SIAM, 2020.
**URL** https://doi.org/10.1137/1.9781611976007.9

Fully dynamic graph algorithms have received growing attention in experimental work recently, though still by far not as much as they have in theory. In this talk, we will consider different approaches to maintain reachability information in a graph as it undergoes a series of insertions and deletions and analyze their behavior in practice on different types of fully dynamic instances.

The study of fully dynamic algorithms in practice is complicated by a lack of real-world, "real-dynamic" instances that are available publicly. In this talk will take a closer look at how the characteristics of an update sequence can influence the behavior of dynamic algorithms for reachability and subgraph counting and also review different approaches that are currently in use to overcome the shortage in fully dynamic instances in practice.

The talk concludes with a summary of lessons learned when engineering specifically dynamic graph algorithms.

## 3.6 Strongly polynomial dynamic algorithms for minimum-weight cycle and related problems

*Adam Karczmarz (University of Warsaw, PL & and IDEAS NCBR – Warsaw, PL)*


**Main reference** Adam Karczmarz: "Fully Dynamic Algorithms for Minimum Weight Cycle and Related Problems", in Proc. of the 48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference), LIPIcs, Vol. 198, pp. 83:1–83:20, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021.
**URL** https://doi.org/10.4230/LIPIcs.ICALP.2021.83

A relatively small portion of the known dynamic algorithms for shortest paths and related problems have strongly polynomial update bounds. This means, roughly speaking, that in most cases the update bounds either do not hold for real-weighted graphs or depend on the magnitude of the graph's weights. One notable exception is the fully dynamic APSP algorithm of Demetrescu and Italiano [J.ACM'04].

In this talk, we will consider maintaining negative/minimum-weight/minimum-mean cycles in dynamic real-weighted digraphs. The best-known static strongly polynomial algorithms for these classical problems run in $O(nm)$ time. For some of these problems, non-trivial strongly polynomial update bounds can be obtained. For others, we will try to identify some challenges.

## 3.7 Dynamic Algorithms for Packing-Covering LPs via Multiplicative Weight Updates

*Peter Kiss (University of Warwick – Coventry, GB)*


**Joint work of** Sayan Bhattacharya, Peter Kiss, Thatchaphol Saranurak
**Main reference** Sayan Bhattacharya, Peter Kiss, Thatchaphol Saranurak: "Dynamic Algorithms for Packing-Covering LPs via Multiplicative Weight Updates", CoRR, Vol. abs/2207.07519, 2022.
**URL** https://doi.org/10.48550/arXiv.2207.07519

In the dynamic linear program (LP) problem, we are given an LP undergoing updates and we need to maintain an approximately optimal solution. Recently, significant attention (e.g., [Gupta et al. STOC'17; Arar et al. ICALP'18, Wajc STOC'20]) has been devoted to the study of special cases of dynamic packing and covering LPs, such as the dynamic fractional matching and set cover problems. But until now, there is no non-trivial dynamic algorithm for general packing and covering LPs. In this work, we settle the complexity of dynamic packing and covering LPs, up to a polylogarithmic factor in update time. More precisely, in the partially dynamic setting (where updates can either only relax or only restrict the feasible region), we give near-optimal deterministic $\epsilon$-approximation algorithms with polylogarithmic amortized update time. Then, we show that both partially dynamic updates and amortized update time are necessary; without any of these conditions, the trivial algorithm that recomputes the solution from scratch after every update is essentially the best possible, assuming SETH. To obtain our results, we initiate a systematic study of the multiplicative weights update (MWU) method in the dynamic setting. As by-products of our techniques, we also obtain the first online $(1+\epsilon)$-competitive algorithms for both covering and packing LPs with polylogarithmic recourse, and the first streaming algorithms for covering and packing LPs with linear space and polylogarithmic passes.

## 3.8 Balanced Allocations: The Heavy Case With Deletions

*William Kuszmaul (MIT – Cambridge, US)*


**Joint work of** Nikhil Bansal, William Kuszmaul
**Main reference** Nikhil Bansal, William Kuszmaul: "Balanced Allocations: The Heavily Loaded Case with Deletions", CoRR, Vol. abs/2205.06558, 2022.
**URL** https://doi.org/10.48550/arXiv.2205.06558

In the 2-choice allocation problem, $m$ balls are placed into $n$ bins, and each ball must choose between two random bins $i, j \in [n]$ that it has been assigned to. It has been known for more than two decades, that if each ball follows the GREEDY strategy (i.e., always pick the less-full bin), then the maximum load will be $m/n + O(\log \log n)$ with high probability in $n$ (and $m/n + O(\log m)$ with high probability in $m$). It has remained an open question whether the same bounds hold in the *dynamic* version of the same game, where balls are inserted/deleted with no more than $m$ balls present at a time.

We show that, somewhat surprisingly, these bounds do not hold in the dynamic setting: already on 4 bins, there exists a sequence of insertions/deletions that cause the GREEDY strategy to incur a maximum load of $m/4 + \Omega(\sqrt{m})$ with probability $\Omega(1)$, This raises the question of whether *any* 2-choice allocation strategy can offer a strong bound in the dynamic setting. Our second result answers this question in the affirmative: we present a new strategy, called MODULATEDGREEDY, that guarantees a maximum load of $m/n + O(\log m)$, at any given moment, with high probability in $m$.

## 3.9 Optimal Decremental Connectivity in Non-Sparse Graphs

*Jakub Łącki (Google – New York, US)*


**Joint work of** Anders Aamand, Adam Karczmarz, Jakub Łącki, Nikos Parotsidis, Peter M. R. Rasmussen, Mikkel Thorup
**Main reference** Anders Aamand, Adam Karczmarz, Jakub Lacki, Nikos Parotsidis, Peter M. R. Rasmussen, Mikkel Thorup: "Optimal Decremental Connectivity in Non-Sparse Graphs", CoRR, Vol. abs/2111.09376, 2021.
**URL** https://arxiv.org/abs/2111.09376

We show an algorithm for decremental maintenance of connected components and 2-edge connected components , which handles any sequence of edge deletions in $O(m + n\text{polylog}n)$ time and answers queries in constant time. This talk focuses on three ideas behind this result: a new sparse connectivity certificate, which can be updated dynamically, a new way of using the XOR-trick, which allows one detect small cuts, and a self-check technique, which allows us to obtain a Las Vegas randomized algorithm based on a Monte Carlo data structure.

## 3.10 Online Routing and Network Design with Predictions

*Nicole Megow (Universität Bremen, DE)*


**Joint work of** Nicole Megow, Giulia Bernardini, Alexander Lindermayr, Alberto Marchetti-Spaccamela, Leen Stougie, Michelle Sweering

Online optimization refers to solving problems where an initially unknown input is revealed incrementally, and irrevocable decisions must be made not knowing future requests. The assumption of not having any prior knowledge about future requests seems overly pessimistic. Given the success of machine-learning methods and data-driven applications, one may expect to have access to predictions about future requests. However, simply trusting them might lead to very poor solutions as these predictions come with no quality guarantee. In this talk we present recent developments in the young line of research that integrates such error-prone predictions into algorithm design to break through worst case barriers. We discuss algorithmic challenges with a focus on online routing and network design and present algorithms with performance guarantees depending on a novel error metric.

## 3.11 Deterministic Fully Dynamic Distance Approximation

*Yasamin Nazari (Universität Salzburg, AT)*


**Joint work of** Jan van den Brand, Sebastian Forster, Yasamin Nazari
**Main reference** Jan van den Brand, Sebastian Forster, Yasamin Nazari: "Fast Deterministic Fully Dynamic Distance Approximation", CoRR, Vol. abs/2111.03361, 2021.
**URL** https://arxiv.org/abs/2111.03361

The first part of the talk focuses on our deterministic fully dynamic algorithms for computing approximate distances in a graph. Specifically, we are given an unweighted and undirected graph $G = (V, E)$ undergoing edge insertions and deletions, and a parameter $0 < \epsilon \leq 1$, and our goal is to maintain $(1 + \epsilon)$-approximate distances between a single pair (st distance), a single source to all nodes (SSSP), or all pairs (APSP). We discuss how combinatorial tools such as emulators can be combined with algebraic data structures to obtain deterministic algorithms with improved worst-case guarantees for these problems.

The second part of the talk focuses on future directions for obtaining improved fully dynamic algorithms for weighted or directed graphs. We explore possible candidate combinatorial structures that could be used and the challenges in maintaining them in the fully dynamic settings.

## 3.12 Scalable dynamic graph processing with low latency: insights and challenges

*Nikos Parotsidis (Google Research – Zürich, CH)*



In this talk we discuss insights from the development of a scalable system for processing dynamic graph algorithms with low latency inside Google. We discuss applications, requirements, and challenges that arise in such a real-world system. The three challenges that we discuss are 1) how to maintain a solution that does not change very drastically during the execution of the algorithm, 2) how to process a graph in a distributed and dynamic fashion; which is mandated by the scale of the data, and 3) how to process a large volume of concurrent updates, each within low latency. These challenges naturally lead us in exploring new models (and evaluating the suitability of existing models) for tackling them.

## 3.13 A Blackbox Reduction for Adaptive Adversaries using Differential Privacy

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


**Main reference** Amos Beimel, Haim Kaplan, Yishay Mansour, Kobbi Nissim, Thatchaphol Saranurak, Uri Stemmer: "Dynamic Algorithms Against an Adaptive Adversary: Generic Constructions and Lower Bounds", CoRR, Vol. abs/2111.03980, 2021.
**URL** https://arxiv.org/abs/2111.03980

This talk is a tutorial on how to use differential privacy to obtain a black-box reduction that can transform any dynamic algorithm for any estimation problem that works against an oblivious adversary to another algorithm against an adaptive adversary.

## 3.14 Recent Results in Engineering Dynamic Graph Algorithms

*Christian Schulz (Universität Heidelberg, DE)*



In recent years, significant advances have been made in the design and analysis of fully dynamic algorithms. However, these theoretical results have received very little attention from the practical perspective. Few of the algorithms are implemented and tested on real datasets, and their practical potential is far from understood. In this talk, we give a brief overview of results in engineering dynamic graph algorithms that we achieved recently. To this end, we give a high level overview of dynamic algorithms and their performance for (hyper) graph ($b$-)matching, independent sets, edge-orientation, reachability as well as $k$-center clustering and minimum cuts.

## 3.15 Queuing Safely for Elevator Systems Amidst a Pandemic

*Clifford Stein (Columbia University, US)*


**Joint work of** Sai Mali Ananthanarayanan, Charles C. Branas, Adam Elmachtoub, Clifford Stein, Yeqing Zhou
**Main reference** Sai Mali Ananthanarayanan, Charles C. Branas, Adam Elmachtoub, Clifford Stein, Yeqing Zhou: "Queuing Safely for Elevator Systems Amidst a Pandemic" (December 21, 2020). Production and Operations Management, 00 1– 18. Available at SSRN.
**URL** https://doi.org/10.1111/poms.13686

The requirement of social distancing during the COVID-19 pandemic has presented significant challenges for high-rise buildings, which heavily rely on elevators for vertical transportation. In particular, the need for social distancing has reduced elevator capacity typically by at least two-thirds or as much as over 90% the normal amount. This reduction is a serious concern, as reduced elevator capacities cause large queues to build up in lobbies, which makes social distancing difficult and results in large wait times. The objective of this study is to safely manage the elevator queues by proposing simple, technology-free interventions that drastically reduce the waiting time and length of lobby queues. We use mathematical modeling, epidemiological expertise, and simulation to design and evaluate our interventions. The key idea is to explicitly or implicitly group passengers that are going to the same floor into the same elevator as much as possible. In the Cohorting intervention, we attempt to find passengers going to the same floor as the first person in the queue. In the Queue Splitting intervention, we create a different queue for different groups of floors. Based on simulation and analytical findings, Cohorting and Queue Splitting can significantly reduce queue length and wait time, while also maintaining safety from viral transmission in otherwise crowded elevators, building lobbies, and entrances. These interventions are generally accessible for many buildings since they do not require programming the elevators, and rely on only using signage and/or a queue manager to guide passengers.

## 3.16 Dynamic Graph Sketching: To Infinity And Beyond

*David Tench (Rutgers University – Piscataway, US)*


**Joint work of** David Tench, Evan West, Victor Zhang, Michael A. Bender, Abiyaz Chowdhury, J. Ahmed Dellas, Martin Farach-Colton, Tyler Seip, Kenny Zhang
**Main reference** David Tench, Evan West, Victor Zhang, Michael A. Bender, Abiyaz Chowdhury, J. Ahmed Dellas, Martin Farach-Colton, Tyler Seip, Kenny Zhang: "GraphZeppelin: Storage-Friendly Sketching for Connected Components on Dynamic Graph Streams", in Proc. of the SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 – 17, 2022, pp. 325–339, ACM, 2022.
**URL** https://doi.org/10.1145/3514221.3526146

Existing graph stream processing systems must store the graph explicitly in RAM which limits the scale of graphs they can process. The graph semi-streaming literature offers algorithms which avoid this limitation via linear sketching data structures that use small (sublinear) space, but these algorithms have not seen use in practice to date.This talk explores what is needed to make graph sketching algorithms practically useful, and as a case study present a sketching algorithm for connected components and a corresponding high-performance implementation. Finally, we give an overview of the many open problems in this area, focusing on improving query performance of graph sketching algorithms.

## 3.17 Dynamic Matching with Better-than-2 Approximation in Polylogarithmic Update Time

*David Wajc (Stanford University, US)*


**Joint work of** Sayan Bhattacharya, Peter Kiss, Thatchaphol Saranurak, David Wajc
**Main reference** Sayan Bhattacharya, Peter Kiss, Thatchaphol Saranurak, David Wajc: "Dynamic Matching with Better-than-2 Approximation in Polylogarithmic Update Time", CoRR, Vol. abs/2207.07438, 2022.
**URL** https://doi.org/10.48550/arXiv.2207.07438

We present dynamic algorithms with polylog update time for the value version of the dynamic matching problem with approximation ratio strictly better than 2. Specifically, we obtain a $1 + 1/\sqrt{2} + \epsilon \approx 1.707 + \epsilon$ approximation in bipartite graphs and a $1.973 + \epsilon$ approximation in general graphs.

## 3.18 Dynamic Distance Oracles in Planar Graphs

*Oren Weimann (University of Haifa, IL)*



A distance oracle is a data structure for answering distance queries on a graph. While on general graphs efficient distance oracles must settle for approximate answers, on planar graphs recent progress has lead to exact oracles with almost linear space and polylogarithmic query time (i.e. almost optimal). However, in the dynamic setting (when the underlying graph is subject to updates) there has been no significant progress in recent years. The state of the art is an exact oracle from more than 20 years ago that given a planar graph supports both updates and queries in $\tilde{O}(n^{2/3})$ time. On the lower-bound side, conditioned on the APSP hypothesis, in any dynamic exact distance oracle (in fact, even in the offline setting) either the update or the query must take $\Omega(n^{1/2})$ time, leaving an intriguing gap. For approximate distances, the currently fastest oracle requires $\tilde{O}(n^{1/2})$ time for both updates and queries (and there is no known lower bound), and in the offline setting there is an almost optimal solution with polylogarithmic time for both updates and queries. In this talk we will describe these upper and lower bounds (for exact distances), the tight connections in planar graphs between distance oracles and maximum-flow (or minimum-cut) oracles, concrete open problems, and possible directions for solving them.

## 3.19 Optimal resizable arrays

*Uri Zwick (Tel Aviv University, IL)*


**Joint work of** Robert E. Tarjan, Uri Zwick
**Main reference** Robert E. Tarjan, Uri Zwick: "Optimal resizable arrays", CoRR, Vol. abs/2211.11009, 2022.
**URL** https://doi.org/10.48550/arXiv.2211.11009

A *resizable array* is an array that can *grow* and *shrink* by the addition or removal of items from its end, or both its ends, while still supporting constant-time *access* to each item stored in the array given its *index*. Since the size of an array, i.e., the number of items

in it, varies over time, space-efficient maintenance of a resizable array requires dynamic memory management. A standard doubling technique allows the maintenance of an array of size $N$ using only $O(N)$ space, with $O(1)$ amortized time, or even $O(1)$ worst-case time, per operation. Sitarski and Brodnik et al. describe much better solutions that maintain a resizable array of size $N$ using only $N + O(\sqrt{N})$ space, still with $O(1)$ time per operation. Brodnik et al. give a simple proof that this is best possible.

We distinguish between the space needed for *storing* a resizable array, and accessing its items, and the *temporary* space that may be needed while growing or shrinking the array. For every integer $r \geq 2$, we show that $N + O(N^{1/r})$ space is sufficient for storing and accessing an array of size $N$, if $N + O(N^{1-1/r})$ space can be used briefly during grow and shrink operations. Accessing an item by index takes $O(1)$ worst-case time while grow and shrink operations take $O(r)$ amortized time. Using an exact analysis of a *growth game*, we show that for any data structure from a wide class of data structures that uses only $N + O(N^{1/r})$ space to store the array, the amortized cost of grow is $\Omega(r)$, even if only grow and access operations are allowed. The time for grow and shrink operations cannot be made worst-case, unless $r = 2$.

# 4 Open problems

## 4.1 Reducing weighted matching to unweighted matching

*Aaron Bernstein (Rutgers University – New Brunswick, US)*



In dynamic unweighted matching, the goal is to maintain an (approximate) maximum cardinality matching in a graph that is changing over time. There is an extensive literature on this problem, with many different state-of-the-art results; these achieve different approximation-ratio/update-time tradeoffs, and also vary on secondary parameters (e.g. worst-case vs. amortized, adaptive vs. oblivious adversary, fully dynamic vs. decremental vs. incremental).

For almost all of these trade-offs, the state-of-the-art for maximum *weighted* matching lags far behind that for unweighted matching. One could try to adapt each unweighted algorithm separately to the weighted case, but it would be nice to have a single all-purpose tool.

There has been some partial progress in this direction. In 2017, Stubbs and Vassilevska Williams showed how to transform any algorithm for unweighted matching into one for weighted matching, but at the cost of a $(1/2 - \epsilon)$ approximation [2]. In 2021, Bernstein, Dudeja, and Langley reduced the approximation overhead to $2/3 - -\epsilon$ in non-bipartite graphs and $1 - \epsilon$ in bipartite graphs [1]. That is, in bipartite graphs, this paper shows a black-box conversion from *any* algorithm for unweighted matching to one for weighted matching that is essentially as good: the approximation guarantee reduces by $1 - \epsilon$, while the update time increases by $\log(W)$. Moreover, this transformation preserves all secondary parameters (worst-case, deterministic, etc.); it also works in multiple other models (e.g. streaming, MPC).

The above algorithm gives us a blueprint for the kind of result we would like, but it only works in bipartite graphs. (In non-bipartite graphs the approximation overhead is 3/2, which in matching is a big drawback.) Is it possible to get a similar transformation in non-bipartite graphs?

### Open Problem

Is there a black-box transformation that converts any algorithm for dynamic unweighted matching in *non-bipartite* graphs into one for dynamic weighted matching, while only reducing the approximation guarantee by $(1-\epsilon)$, and only increasing the update time by $polylog(nW)$?

### References

1 Aaron Bernstein, Aditi Dudeja, and Zachary Langley. A framework for dynamic matching in weighted graphs. In *STOC*, pages 668–681. ACM, 2021.
2 Daniel Stubbs and Virginia Vassilevska Williams. Metatheorems for dynamic weighted matching. In *ITCS*, pages 58:1–58:14, 2017.

## 4.2 Communication Complexity of Max-Flow

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



Suppose the edges of a graph $G$ (with $n$ vertices and $m$ edges) are partitioned between two parties Alice and Bob. They wish to solve the $(s,t)$-max-flow problem on their graph with as few bits of communication as possible (in any number of interactive rounds).

### Open Problem

It is an open problem to settle this communication complexity of max-flow. Specifically, are there near linear in $n$ communication protocols, or can we show higher lower bounds? Another direction is looking at the round-communication tradeoff (relevant for streaming, distributed, etc.) of max-flow and related problems.

### Some known "results":

- Trivial algorithm $\tilde{O}(m)$: Alice sends all her edges to Bob.
- Interior point method $\tilde{O}(n\sqrt{n})$: One can simulate the $O(\sqrt{n})$-round interior point method for max-flow in $O(n)$ communication each rounds. *(not verified, but should work)*
- $\Omega(n \log n)$ lower bound.

### Evidence that a $\tilde{O}(n)$ communication protocol might / might not exist:

- The related problem of bipartite matching and its variants (max-cost $b$-matching, vertex capacitated $s,t$-flow, transshipment, negative weight SSSP) admit $\tilde{O}(n)$ communication protocol [1] (based on a straightforward cutting planes approach).
- The $\tilde{O}(n\sqrt{n})$-communication interior point method shows that $O(m)$ is not the answer.
- Experience says that sequential $O(m^\alpha)$ graph algorithms usually translate into $O(n^\alpha)$ communication protocols. However, it is unclear if the sequential $O(m^{1+o(1)})$ time max-flow algorithm [2] would help in the communication setting (it uses $O(m)$ IPM rounds, and relies on complicated data structures).
- The answer can consist of up to $O(m)$ edges. So any $o(m)$ protocol cannot let the parties know the actual edges of the flow (but only a more "compact" representation). This is unlike all the bipartite-matching problems listed above that only use $O(n)$ edges in their optimal flows.

### References

1 Joakim Blikstad, Jan van den Brand, Yuval Efron, Sagnik Mukhopadhyay, and Danupon Nanongkai. Nearly optimal communication and query complexity of bipartite matching. *CoRR*, abs/2208.02526, 2022. in FOCS'22.

2 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. in FOCS'22.


## 4.3 Dynamic Complexity of Low-Stretch Spanning Trees

*Gramoz Goranci (ETH Zürich, CH)*



In this *low-stretch spanning tree* problem, we are given an undirected graph $G$ with $n$ vertices and $m$ edges, and the goal is to compute a spanning tree $T$ of $G$ that minimizes $\text{av-str}_T(G) := (1/m)\sum_{e=(u,v)\in E(G)} \text{dist}_T(u,v)$, referred to as the *average stretch* of $T$. This problem and its variants lie at the core of algorithm design and have found applications in online and approximation algorithms, fast algorithms for computing maximum flows on graphs and solving Laplacian systems, and constructing competitive oblivious routing schemes, among others. It is known that every graph admits a spanning tree of average stretch $O(\log n \log\log n)$ which can be computed in $O(m \log n \log\log n)$ [1]. The stretch guarantee is tight up to a $O(\log\log n)$ factor since any spanning tree of a $n$-vertex grid graph requires $\Omega(\log n)$ average stretch [2].

In the fully dynamic setting, the graph $G$ undergoes an intermixed sequence of edge insertions and deletions, and the goal is to maintain a spanning tree $T$ of $G$ with small average stretch. The first work in this setting [4] gave a dynamic algorithm that supports edge updates in $n^{1/2+o(1)}$ amortized update and ensures that the maintained spanning tree has average stretch $n^{o(1)}$. The update time was subsequently improved to $n^{o(1)}$ while keeping the stretch guarantee the same [3]. Both works make use of randomization and assume an oblivious adversary.

### Open Problems

- Does there exist a fully dynamic algorithm for maintaining a spanning tree with $O(\text{poly}(\log n))$ average stretch in *sub-linear* update time? The question is interesting even in the decremental/deletions-only setting.
- Are there *deterministic* or randomized algorithms that work against *adaptive adversaries* for the dynamic low-stretch spanning tree problem?


### References

1 Ittai Abraham and Ofer Neiman. Using petal-decompositions to build a low stretch spanning tree. *SIAM J. Comput.*, 48(2):227–248, 2019.

2 Noga Alon, Richard M. Karp, David Peleg, and Douglas B. West. A graph-theoretic game and its application to the k-server problem. *SIAM J. Comput.*, 24(1):78–100, 1995.

3 Shiri Chechik and Tianyi Zhang. Dynamic low-stretch spanning trees in subpolynomial time. 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 463–475. SIAM, 2020.

4 Sebastian Forster and Gramoz Goranci. Dynamic low-stretch trees via dynamic low-diameter decompositions. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 377–388. ACM, 2019.

## 4.4 Polylog query time for a dynamic all-pairs problem in plane directed graphs

*Adam Karczmarz (University of Warsaw – Warsaw, PL & and IDEAS NCBR – Warsaw, PL)*



Plane directed graphs allow non-trivial dynamic reachability and distance oracles supporting arbitrary point-to-point queries. For example, one can achieve $\widetilde{O}(\sqrt{n})$ update/query time bound for fully dynamic reachability [2], or incremental distances [1], and $\widetilde{O}(n^{2/3})$ update/query time for fully dynamic distances [3]. For decremental reachability, one can get polylog$(n)$ amortized update time and $\widetilde{O}(\sqrt{n})$ query time [4]. However, to the best my knowledge, no tradeoff with polylog$(n)$ query time and $\widetilde{O}(n^{0.99})$ amortized update time is known for any kind of dynamic all-pairs oracle problem on plane digraphs. Probably the easiest specific problem addressing this should be the following.

### Open Problem

Suppose a plane digraph $G$ is given. Initially, all edges are switched off, and $G$ undergoes edge switch-ons. Design a data structure supporting polylog$(n)$-time arbitrary-pair reachability queries (in the switched on subgraph) and edge switch-ons within $\widetilde{O}(n^{1.99})$ total update time.

### References

1 Debarati Das, Maximilian Probst Gutenberg, and Christian Wulff-Nilsen. A near-optimal offline algorithm for dynamic all-pairs shortest paths in planar digraphs. In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 – 12, 2022*, pages 3482–3495. SIAM, 2022.

2 Krzysztof Diks and Piotr Sankowski. Dynamic plane transitive closure. In *Algorithms – ESA 2007, 15th Annual European Symposium, Eilat, Israel, October 8-10, 2007, Proceedings*, volume 4698 of *Lecture Notes in Computer Science*, pages 594–604. Springer, 2007.

3 Jittat Fakcharoenphol and Satish Rao. Planar graphs, negative weight edges, shortest paths, and near linear time. *J. Comput. Syst. Sci.*, 72(5):868–889, 2006.

4 Adam Karczmarz. Decremental transitive closure and shortest paths for planar digraphs and beyond. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 73–92. SIAM, 2018.

## 4.5 Breaking CountMin Sketches Inside a Greedy Outer Loop

*Richard Peng (University of Waterloo, CA)*



Consider the following way of estimating cardinalities of subsets of $[n] = \{1 \dots n\}$:

1. Pick a random permutation $\pi$ of $1, 2, \dots, n$.
2. For a set $S \subseteq [n]$ with size at least $1000 \log n$, store the smallest $100 \log n$ values of

$$\pi(S) = \{\pi(i) : i \in S\},$$

and take their max.

Call this value $Sketch_\pi(S)$.

It can be shown using a reasonably 'standard' use of Chernoff bound that if $|S_1| < 2|S_2|$, then with probability at least $1 - -n^{-3}$ (over the choices of $\pi$), $Sketch_\pi(S_1) < Sketch_\pi(S_2)$. Also, as the sketches have size $O(\log n)$, such a schema gives a low storage method for approximating tracking sizes of sets under mergers. The fun, then happens when one starts to use the output of the data structures to dictate the next merge. That is, consider starting with $n$ sets $S_1 ... S_n$ (of size at least $1000 \log n$, which is easy to enforce by having 'dummy' elements that are in all sets), and after generating an initial random permutation $\pi$, repeatedly perform the following simplification of the min-degree heuristic:

1. For $t = 1 \dots n - 2$
   a. Let i and j be the two remaining sets with the minimum / second minimum sketch values computed w.r.t. $\pi$.
   b. Replace $S_i$, $S_j$ in the collection of sets by their union, $S_i \cup S_j$.

### Open Problem

Exhibit an initial state such that with probability at least 0.1 over the choices of $\pi$, at some iteration of the algorithm, one of $S_i$ and $S_j$ has size more than twice the minimum / second minimum respectively. Alternatively, prove this cannot happen, that is, for any starting configurations of $S_1, S_2, \dots, S_n$, things are happy with probability $> 0.1$ (over choices of $\pi$).

A stronger form of the latter version is showing that in absences of deletions, so just queries and merges, things work.

References:

- Paper that showed approximate min-degree orderings can be solved in almost-linear time by re-introducing randomness to decorrelate intermediate states: `https://arxiv.org/abs/1804.04239`.
- The analysis of randomized Gaussian elimination (which does combine such randomness against adversarial users) by Kyng and Sachdeva: `https://arxiv.org/abs/1605.02353`, and a follow up that defined a 'resparsification game': `https://arxiv.org/abs/1611.06940`.

## 4.6 Dynamic Maximal Matching

*Shay Solomon (Tel Aviv University, IL)*



In the dynamic *maximal matching* (MM) problem, the goal is to maintain a maximal matching in a dynamic $n$-vertex graph that is subject to edge updates. There is a naive deterministic algorithm with a worst-case update time of $O(n)$, and this is the state-of-the-art update time of deterministic algorithms in general graphs, and also of randomized algorithms against an adaptive adversary, even allowing amortization.

It should be noted that in some graph classes, better results are known. In particular, for sparse graphs, there is a deterministic algorithm [2] with a worst-case update time of $O(\sqrt{m})$, where $m$ is the dynamic number of edges in the graph.

This problem provides a prime example of an exponential gap between algorithms that cope against oblivious versus adaptive adversaries. Indeed, allowing randomization against an oblivious adversary, one can achieve a constant amortized update time [3] and a poly-log worst-case update time [1]. We also note that, while the $O(n)$ (or $O(\sqrt{m})$) deterministic bound has resisted any improvement, no lower bound whatsoever is known.

### Open Problem

Is there a deterministic algorithm, or a randomized one against an adaptive adversary, for maintaining MM in $o(n)$ update time in general graphs? Can we push the update time towards a poly-log or even further? Is there any $\omega(1)$ lower bound for such algorithms?

### References

1 Aaron Bernstein, Sebastian Forster, and Monika Henzinger. A deamortization approach for dynamic spanner and dynamic maximal matching. In *30th SODA*, pages 1899–1918. SIAM, 2019.
2 Ofer Neiman and Shay Solomon. Simple deterministic algorithms for fully dynamic maximal matching. In *STOC*, pages 745–754, 2013.
3 Shay Solomon. Fully dynamic maximal matching in constant update time. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 325–334, 2016.

## 4.7 Dynamic Derandomization

*David Wajc (Stanford University, US)*



A common theme in derandomization is the construction of small near-independent probability spaces. The following definition, due to Naor and Naor [1], captures (some) such notions, and generalizes the perhaps more familiar notion of $k$-wise independence.

**Definition.** Let $\mathcal{U}$ be the uniform distribution on $n$ binary variables. A distribution $\mathcal{D}$ on $n$ binary variables is $(\delta, k)$-*dependent* if for any event $A$ that is determined by $k$ or fewer

random variables,

$$\Pr_{\vec{Y}\sim\mathcal{U}}[A] - -\delta \leq \Pr_{\vec{Y}\sim\mathcal{D}}[A] \leq \Pr_{\vec{Y}\sim\mathcal{U}}[A] + \delta.$$

Equivalently, $\mathcal{D}$ satisfies that for any subset of $k$ or fewer indices, $I \subseteq [m]$, $|I| \leq k$,

$$\sum_{\vec{v}\in\{0,1\}^{|I|}} \left| \Pr\left[\bigwedge_{i\in I}(Y_i = v_i)\right] - 2^{-|I|} \right| \leq \delta.$$

Similarly, up to constants, another way of stating the above is that the total variation distance between the distribution of any subset of $k$ variables from the uniform distribution is $O(\delta)$. For example, $k$-wise independent distributions, under which any $k$ variables take on all $2^k$ realizations with probability $2^{-k}$, are precisely $(0,k)$-dependent distributions. The interest in these distributions is due to the following lemma of [1].

**Lemma.** *For any $\delta > 0$, a $(\delta, k)$-dependent distribution $\mathcal{D}$ on $n$ binary variables can be constructed using $\log\log n + O(k + \log(\frac{1}{\delta}))$ random bits.*[1] *Moreover, after polytime preprocessing, each random variable can be sampled from $\mathcal{D}$ in $O(k \cdot \log n)$ time.*

The utility of the above for derandomization should be apparent: given a randomized algorithm whose analysis hinges on events determined by few random variables, and allows for small error compared to a fully independent distribution, we can simply try out all random seeds!

In more detail, given an algorithm $\mathcal{A}$ using $n$ random bits, but whose analysis carries through unchanged with $(\delta, k)$-dependent variables, construct a deterministic algorithm $\mathcal{A}'$ that runs a copy of $\mathcal{A}$ for all $2^{\log\log n + O(k+\log(1/\delta))} = \log n \cdot 2^{O(k)+\log(1/\delta)}$ many random seeds for the appropriate distribution. So, for example, for constant $k$ and $\delta$, the obtained deterministic algorithm $\mathcal{A}'$ is only slower than $\mathcal{A}$ by some polynomial additive term (to build $\mathcal{D}$) and a polylogarithmic multiplicative factor (due to trying all seeds for $\mathcal{D}$).

The above approach has found applications in derandomization numerous areas since its introduction. A non-exhaustive list of application areas include parallel computing [8], streaming [7], local computation [6], the Color Coding technique [5], matrix multiplication witnesses [4], and recently in (partially derandomizing) online algorithms [3].

Conspicuously missing from the above list (and its extensions) is the area of dynamic algorithms, where derandomizing via this approach would incur a logarithmic overhead (for sufficiently small $k$ and $\delta$) by running one copy of the random algorithm for each random seed. This suggests the following open question.

## Open Problem

Find an application of $(\delta, k)$-dependent distributions in derandomizing dynamic algorithms, possibly with only a polylogarithmic overhead.

## References

1 Joseph (Seffi) Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SICOMP*, 22(4):838–856, 1993.

2 A Joffe. On a set of almost deterministic $k$-independent random variables. *the Annals of Probability*, 2(1):161–162, 1974.

[1] For $k$-wise independence ($\delta = 0$), the necessary number of random bits blows up to $\lfloor k/2 \rfloor \log n$ [2].

3 Niv Buchbinder, Joseph Seffi Naor, and David Wajc. Lossless online rounding for online bipartite matching (despite its impossibility). In *SODA*, 2023. To appear.

4 Noga Alon and Moni Naor. Derandomization, witnesses for boolean matrix multiplication and construction of perfect hash functions. *Algorithmica*, 16(4):434–449, 1996.

5 Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J.ACM*, 42(4):844–856, 1995.

6 Omer Reingold and Shai Vardi. New techniques and tighter bounds for local computation algorithms. *Journal of Computer and System Sciences*, 82(7):1180–1200, 2016.

7 Joan Feigenbaum, Yuval Ishai, Tal Malkin, Kobbi Nissim, Martin J Strauss, and Rebecca N Wright. Secure multiparty computation of approximations. In *International Colloquium on Automata, Languages, and Programming*, pages 927–938, 2001.

8 Noga Alon. A parallel algorithmic version of the local lemma. *Random Structures & Algorithms*, 2(4):367–378, 1991.

## Participants


- Amir Abboud
Weizmann Institute – . Rehovot, IL
- Aaron Bernstein
Rutgers University – New Brunswick, US
- Sayan Bhattacharya
University of Warwick – Coventry, GB
- Joakim Blikstad
KTH Royal Institute of Technology – Stockholm, SE
- Karl Bringmann
Universität des Saarlandes – Saarbrücken, DE
- Keren Censor-Hillel
Technion – Haifa, IL
- Shiri Chechik
Tel Aviv University, IL
- Keerti Choudhary
Indian Institute of Technology – New Dehli, IN
- Aleksander Christiansen
Technical University of Denmark – Lyngby, DK
- Jeremy Fineman
Georgetown University – Washington, DC, US
- Nick Fischer
MPI für Informatik – Saarbrücken, DE
- Sebastian Forster
Universität Salzburg, AT
- Pawel Gawrychowski
University of Wroclaw, PL
- Gramoz Goranci
ETH Zürich, CH
- Fabrizio Grandoni
SUPSI – Lugano, CH
- Kathrin Hanauer
Universität Wien, AT
- Adam Karczmarz
University of Warsaw, PL & and IDEAS NCBR – Warsaw, PL
- Peter Kiss
University of Warwick – Coventry, GB
- Tsvi Kopelowitz
Bar-Ilan University – Ramat Gan, IL
- William Kuszmaul
MIT – Cambridge, US
- Jakub Lacki
Google – New York, US
- Nicole Megow
Universität Bremen, DE
- Matthias Mnich
TU Hamburg, DE
- Shay Mozes
Reichman University – Herzliya, IL
- Danupon Nanongkai
MPI für Informatik – Saarbrücken, DE
- Yasamin Nazari
Universität Salzburg, AT
- Nikos Parotsidis
Google Research – Zürich, CH
- Richard Peng
University of Waterloo, CA
- Maximilian Probst Gutenberg
ETH Zürich, CH
- Eva Rotenberg
Technical University of Denmark – Lyngby, DK
- Piotr Sankowski
IDEAS NCBR – Warsaw, PL & University of Warsaw, PL
- Thatchaphol Saranurak
University of Michigan – Ann Arbor, US
- Christian Schulz
Universität Heidelberg, DE
- Chris Schwiegelshohn
Aarhus University, DK
- Shay Solomon
Tel Aviv University, IL
- Clifford Stein
Columbia University, US
- David Tench
Rutgers University – Piscataway, US
- Jan van den Brand
Georgia Institute of Technology – Atlanta, US
- Virginia Vassilevska Williams
MIT – Cambridge, US
- David Wajc
Stanford University, US
- Oren Weimann
University of Haifa, IL
- Nicole Wein
Rutgers University – Piscataway, US
- Uri Zwick
Tel Aviv University, IL