Dagstuhl Seminar 22461
Dynamic Graph Algorithms

Organizers: A. Bernstein, S. Chechik, S. Forster, T. Kopelowitz

November 13-18 2022

Motivation

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 property of 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 will bring together leading researchers in dynamic algorithms and related areas of graph algorithms. Some specific goals for the seminar are outlined below.

- Because the field has grown so much in recent years, one of the main motivations for this seminar is to establish the main challenges that remain – both a list of specific open problems that continue to resist progress, as well as a discussion of more general limitations to our current upper and lower bound techniques. The seminar will also provide a venue for the community to actively shape the direction of the field going forward.

- Much of the recent progress in the field has stemmed from a new set of techniques, many of which originated in other areas of graph or approximation algorithms. One of the goals of this seminar is thus to discuss how recent developments in graph algorithms broadly construed can be applied to the dynamic setting in particular, and to disseminate some of the recent techniques that have already had tremendous success in this regard.

- The experimental evaluation of dynamic graph algorithms is still in its infancy. There are no established data sets or even evaluation methodologies. This seminar aims to include the leading researchers in the area of algorithms engineering of dynamic graph algorithms to address and, hopefully, resolve these fundamental questions.

Preliminary Schedule: Dagstuhl Seminar 22461 “Dynamic Graph Algorithms”

	Monday (Nov 14)	Tuesday (Nov 15)	Wednesday (Nov 16)		Thursday (Nov 17)	Friday (Nov 18)
07:30 AM	Breakfast	Breakfast	Breakfast		Breakfast	Breakfast
until 08:45 AM						
09:15 AM	Welcome and Ice-Breaker Session	Talk: Nicole Megow	Talk: Adam Karczmarz		Talk: Oren Weimann	Talk: Gramoz Goranci
09:45 AM		Talk: William Kuszmaul	Talk: Yasamin Nazari		Talk: Nick Fischer	Talk: Sebastian Forster
10:15 AM	Coffee Break	Coffee Break	Coffee Break		Coffee Break	Coffee Break
11:00 AM	Talk: Thatchaphol Saranurak	Open Problems Session	Talk: Nikos Parotsidis		Open Problems Session	Collaboration Time
11:30 AM	Talk: David Wajc		Talk: David Tench			
12:00 PM	Lunch	Lunch	Lunch		Lunch	Lunch
12:30 PM						
01:00 PM	Collaboration Time	Collaboration Time	**Track A**	**Track B**	Collaboration Time	*End of Seminar*
01:30 PM			Collaboration Time	Hike		
02:00 PM						
02:30 PM						
03:00 PM	Coffee Break	Coffee Break	Coffee Break		Coffee Break	
03:30 PM						
04:00 PM	Talk: Keren Censor-Hillel	Talk: Christian Schulz	Collaboration Time		Talk: Uri Zwick	
04:30 AM	Talk: Kuba Łącki	Talk: Kathrin Hanauer			Talk: Peter Kiss	
05:00 PM	*Break*	*Break*			Talk: Cliff Stein	
06:00 PM	Dinner	Dinner	Dinner		Dinner	
06:30 PM						

Abstracts

Censor-Hillel, Keren: Dynamic Distributed Subgraph Finding

I will discuss exciting recent progress in distributed subgraph finding (static and dynamic) and describe some of the many intriguing open questions.

Fischer, Nick: Stronger 3-SUM Lower Bounds for Approximate Distance Oracles via Additive Combinatorics

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.

Forster, Sebastian: Deterministic Incremental APSP with Polylogarithmic Update Time and Stretch

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.

Goranci, Gramoz:

Hanauer, Kathrin: Fully Dynamic Graph Algorithms in Practice: (Some) Lessons Learned

Karczmarz, Adam: Strongly polynomial dynamic algorithms for minimum-weight cycle and related problems

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.

Kiss, Peter: Dynamic Algorithms for Packing-Covering LPs via Multiplicative Weight Updates

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 paper, 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 ϵ-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.

Kuszmaul, William: Balanced Allocations: The Heavy Case With Deletions

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.

Łącki, Kuba: Optimal Decremental Connectivity in Non-Sparse Graphs

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.

This is joint work with Anders Aamand, Adam Karczmarz, Nikos Parotsidis, Peter Rasmussen and Mikkel Thorup

Megow, Nicole: Online Routing and Network Design with Predictions

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.

Nazari, Yasamin: Deterministic Fully Dynamic Distance Approximation

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.

Based on joint work with Jan van den Brand and Sebastian Forster.

Parotsidis, Nikos: Scalable dynamic graph processing with low latency: insights and challenges

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.

Schulz, Christian: Recent Results in Engineering Dynamic Graph Algorithms

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.

Tench, David: Dynamic Graph Sketching: To Infinity And Beyond

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. In this talk I will explore 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, I will give an overview of the many open problems in this area, focusing on improving query performance of graph sketching algorithms.

Wajc, David: Dynamic Matching with Better-than-2 Approximation in Polylogarithmic Update Time

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.

Weimann, Oren: Dynamic Distance Oracles in Planar Graphs

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 my talk I 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.

Zwick, Uri: Optimal resizable arrays

Open Problems

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

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 $\widetilde{O}(1)$ amortized update time and $\widetilde{O}(\sqrt{n})$ query time [4]. However, to the best my knowledge, no tradeoff with $\widetilde{O}(1)$ 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 1.1 *Suppose a plane digraph G is given. Initially, all edges are switched off, and G undergoes edge switch-ons. Design a data structure supporting $\widetilde{O}(1)$-time arbitrary-pair reachability queries (in the switched on subgraph) and edge switch-ons within $\widetilde{O}(n^{1.99})$ total update time.*

[Contributed by Adam Karczmarz]

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.

2 Breaking CountMin Sketches Inside a Greedy Outer Loop

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

1. Pick a random permutation π 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 π), $Sketch_\pi(S_1) < Sketch_\pi(S_2)$. 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 π, 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. π.
   (b) Replace S_i, S_j in the collection of sets by their union, $S_i \cup S_j$.

Exhibit an initial state such that with probability at least 0.1 over the choices of π, 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 π).

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`.

[Contributed by Richard Peng]