Report from Dagstuhl Seminar 16451

Structure and Hardness in P

Edited by

Moshe Lewenstein[1], Seth Pettie[2], and Virginia Vassilevska Williams[3]

1 **Bar-Ilan University – Ramat Gan, IL,** moshe@cs.biu.ac.il
2 **University of Michigan – Ann Arbor, US,** pettie@umich.edu
3 **Stanford University, US,** virgi@cs.stanford.edu

Abstract

This document contains description of the talks at the Dagstuhl seminar 16451 "Structure and Hardness in P". The main goal of the seminar was to bring together researchers from several disciplines and connect those who work on proving conditional lower bounds with those who or may benefit from it. This resulted in an extensive list of open problems which is also provided.

Seminar November 6–11, 2016 – http://www.dagstuhl.de/16451
1998 ACM Subject Classification F.1.3 Complexity Measures and Classes, F.2 Analysis of Algorithms and Problem Complexity
Keywords and phrases Algorithmic equivalences, Classifying P, Hardness assumptions, Lower bounds
Digital Object Identifier 10.4230/DagRep.6.11.1
Edited in cooperation with Ivan Mikhailin

1 Executive Summary

Moshe Lewenstein
Seth Pettie
Virginia Vassilevska Williams

The complexity class **P** (polynomial time) contains a vast variety of problems of practical interest and yet relatively little is known about the structure of P, or of the complexity of many individual problems in P. It is known that there exist contrived problems requiring Ω(n1.5) time or Ω(n2) time, and yet to date no unconditional nonlinear lower bounds have been proved for any problem of practical interest. However, the last few years have seen a new resurgence in conditional lower bounds, whose validity rests on the conjectured hardness of some archetypal computational problem. This work has imbued the class P with new structure and has valuable explanatory power.

To cite a small fraction of recent discoveries, it is now known that classic dynamic programming problems such as Edit Distance, LCS, and Fréchet distance require quadratic time (based on the conjectured hardness of k-CNF-SAT), that the best known triangle enumeration algorithms are optimal (based on the hardness of 3-SUM), that Valiant's context-free grammar parser is optimal (based on the hardness of k-CLIQUE), and that the best known approximate Nash equilibrium algorithm is optimal (based on the hardness of 3-SAT).

This Dagstuhl Seminar will bring together top researchers in diverse areas of theoretical computer science and include a mixture of both experts and non-experts in conditional lower bounds. Some specific goals of this seminar are listed below.

- Numerous important problems (such as Linear Programming) seem insoluble in linear time, and yet no conditional lower bounds are known to explain this fact. A goal is to discover conditional lower bounds for key problems for which little is currently known.
- Recent work has been based on both traditional hardness assumptions (such as the ETH, SETH, 3SUM, and APSP conjectures) and a variety of newly considered hardness assumptions (such as the OMv conjecture, the k-CLIQUE conjecture, and the Hitting Set conjecture). Almost nothing is known about the relative plausibility of these conjectures, or if multiple conjectures are, in fact, equivalent. A goal is to discover formal relationships between the traditional and newer hardness assumptions.
- A key goal of the seminar is to disseminate the techniques used to prove conditional lower bounds, particularly to researchers from areas of theoretical computer science that have yet to benefit from this theory. To this end the seminar will include a number of tutorials from top experts in the field.

2 Table of Contents

3 Overview of Talks

3.1 Hardness for Graph Problems

Amir Abboud (Stanford University, US)

This is a survey of the landscape of Hardness in P results that we have for graph problems.

3.2 Optimal Hashing for High-Dimensional Spaces

Alexandr Andoni, Thijs Laarhoven, Ilya Razenshteyn, and Erik Waingarten

Main reference A. Andoni, T. Laarhoven, I. Razenshteyn, E. Waingarten, "Optimal Hashing-based Time-Space Trade-offs for Approximate Near Neighbors", arXiv:1608.03580v1 [cs.DS], 2016.
URL https://arxiv.org/abs/1608.03580v1

We survey recent advances in the approximate nearest neighbor search problem in high-dimensional Euclidean/Hamming spaces, which go beyond the classic Locality Sensitive Hashing technique for the problem. The culmination of these advances is a new optimal hashing algorithm that achieves the full trade-off between space vs query time. For example, we obtain the first algorithm with near-linear space and sub-linear query time for any approximation factor greater than 1, which is perhaps the most important regime in practice.

Our algorithm also unifies, simplifies, and improves upon the previous data structures for the problem, combining elements of data-dependent hashing and Locality Sensitive Filtering.

Finally, we discuss matching lower bounds for hashing algorithms, as well as for 1- and 2-cell probe algorithms. In particular, the 2-cell probe lower bound exploits a connection to locally-decodable codes, and yields the first space lower bound that is not polynomially smaller than the 1-probe bound (for any static data structure).

3.3 Permanents as hardness for problems in P?

Andreas Björklund (Lund University, SE)

Joint work of Andreas Björklund, Virginia Vassilevska Williams, Ryan Williams

We know that solving the orthogonal vectors problem with two sets of n vectors of dimension d, for d superlogarithmic in n in less than quadratic time in n would violate SETH, even if the vectors are from $\{0,1\}^d$. This is used in many of the SETH hardness results for problems in P.

However, SETH is a hypothesis about one specific problem, that there are no significantly faster algorithms to solve CNF Sat on n variables than testing all assignments. Unfortunately there are very few fine-grained reductions between hard exponential time problems. Until we find them, there are several other hard problems we could make similar hypotheses about.

In this talk we take a look at the matrix permanent. We give a proof sketch that shows that you cannot count the pairs of orthogonal vectors in two sets of n vectors each from

$\{-1, 0, 1\}^d$, for d polylogarithmic in n, in truly subquadratic time in n, unless you can compute the permanent of a 0/1 matrix with bounded number of ones faster than the best algorithms we know of to date.

3.4 Hardness for Polytime String Problems

Karl Bringmann (MPI für Informatik – Saarbrücken, DE)

Joint work of Karl Bringmann, Marvin Künnemann, Allan Grønlund, Kasper Green Larsen

In this tutorial we surveyed recent conditional lower bounds for polynomial time problems on strings. We focused on hardness based on the Strong Exponential Time Hypothesis, specifically we discussed hardness of longest common subsequence [1, 2] as well as for pattern matching of regular expressions [3, 4].

References

1 Karl Bringmann and Marvin Künnemann. "Quadratic conditional lower bounds for string problems and dynamic time warping." FOCS 2015.
2 Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. "Tight hardness results for LCS and other sequence similarity measures." FOCS 2015.
3 Arturs Backurs and Piotr Indyk. "Which Regular Expression Patterns are Hard to Match?" FOCS 2016.
4 Karl Bringmann, Allan Grønlund, and Kasper Green Larsen. "A Dichotomy for Regular Expression Membership Testing." arXiv:1611.00918

3.5 Hardness of string problems with small alphabet size

Yi-Jun Chang (University of Michigan – Ann Arbor, US)

Main reference Y.-J. Chang, "Hardness of RNA folding problem with four symbols", in Proc. of the 27th Annual Symposium on Combinatorial Pattern Matching (CPM 2016), LIPIcs, Vol. 54, pp. 13:1–13:12, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2016.
URL http://dx.doi.org/10.4230/LIPIcs.CPM.2016.13

In FOCS 2015, Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams demonstrated conditional lower bounds for fundamental string problems such as RNA folding, Dyck edit distance, and $k - LCS$.

However, all these lower bound proofs require the alphabet size to be large enough to work. For RNA folding, the required alphabet size is 36, making the result biologically irrelevant. For k-LCS, the alphabet size needed is $O(k)$, and it is an open problem whether the same lower bound holds when the alphabet size is a constant independent of k.

In this talk, we show how we can lower the alphabet size requirement for the hardness proofs of RNA folding (from 36 to 4) and Dyck Edit distance (from 48 to 10). We will also discuss some open problems and future work directions.

3.6 Recent insights into counting small patterns

Radu Curticapean (Hungarian Academy of Sciences – Budapest, HU), Holger Dell (Universität des Saarlandes, DE), and Dániel Marx

We consider the problem of counting subgraphs. More specifically, we look at the following problems $\sharp Sub(C)$ for fixed graph classes C: Given as input a graph H from C (the pattern) and another graph G (the host), the task is to count the occurrences of H as a subgraph in G. Our goal is to understand which properties of the pattern class C make the problem $\sharp Sub(C)$ easy/hard. For instance, for the class of stars, we can solve this problem in linear time. For the class of paths however, it subsumes counting Hamiltonian paths and is hence $\sharp$ P-hard.

As it turns out, the notion of $\sharp$ P-hardness fails to give a sweeping dichotomy for the problems $\sharp Sub(C)$, since there exist classes C of intermediate complexity. However, adopting the framework of fixed-parameter tractability, and parameterizing by the size of the pattern, it was shown in 2014 how to classify the problems $\sharp Sub(C)$ as either polynomial-time solvable or $\sharp$ W[1]-hard: A class C lies on the polynomial-time side of this dichotomy iff the graphs appearing in C have vertex-covers of constant size.

In this talk, we introduce a new technique that allows us to view the subgraph counting problem from a new perspective. In particular, it allows for the following applications:

1. A greatly simplified proof of the 2014 dichotomy result, together with almost-tight lower bounds under ETH, which were not achievable before.
2. Faster algorithms for counting k-edge subgraphs, such as k-matchings, with running time n^{ck} for constants $c < 1$.

3.7 Tight Bounds for Subgraph Isomorphism and Graph Homomorphism

Marek Cygan (University of Warsaw, PL)

Joint work of Marek Cygan, Fedor Fomin, Alexander Kulikov, Ivan Mihajlin, Alexander Golovnev, Jakub Pachocki, Arkadiusz Socała
Main reference M. Cygan, F. V. Fomin, A. Golovnev, A. S. Kulikov, I. Mihajlin, J. Pachocki, A. Socala, "Tight Bounds for Graph Homomorphism and Subgraph Isomorphism", in Proc. of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1643–1649, 2016.
URL http://dx.doi.org/10.1137/1.9781611974331.ch112

This tutorial will consists of two parts. First, we will show simple reductions that allow obtaining (slightly) sublinear reductions from CNF-SAT, leading to (slightly) superexponential lower bounds under the Exponential Time Hypothesis. The goal is to expose main ideas used in such reductions.

Second part will be about tight lower bounds for graph homomorphism and subgraph isomorphism (under the ETH), which is a joint work with Fedor Fomin, Alexander Kulikov, Ivan Mihajlin, Alexander Golovnev, Jakub Pachocki, Arkadiusz Socała.

3.8 Popular Conjectures as a Barrier for Dynamic Planar Graph Algorithms

Søren Dahlgaard (University of Copenhagen, DK)

Joint work of Amir Abboud
Main reference A. Abboud, S. Dahlgaard, "Popular Conjectures as a Barrier for Dynamic Planar Graph Algorithms", in Proc. of the 57th Annual Symposium on Foundations of Computer Science (FOCS'16), pp. 477–486, IEEE, 2016.
URL http://dx.doi.org/10.1109/FOCS.2016.58

We consider dynamic problems in planar graphs and present hardness results for dynamic shortest paths and related problems. This result is the first of its kind for planar graphs, and we believe that our techniques might be helpful in proving hardness for other problems in planar graphs. In particular we show that, based on the APSP-conjecture, no algorithm can perform dynamic shortest paths in planar graphs faster than $O(\sqrt{(}n))$ query and update time.

3.9 New upper bounds for some basic problems in P

Omer Gold (Tel Aviv University, IL)

Joint work of Omer Gold, Micha Sharir

I will provide an overview on our recent upper bounds for some basic (geometric) problems in P. Particularly, improved subquadratic bounds for 3-SUM, the first subquadratic-time algorithms for Dynamic Time Warping and Geometric Edit Distance, near-linear decision tree bounds for the discrete Fréchet distance under polyhedral metrics, and reduction relations between Dominance Products and high-dimensional Closest Pair problems. This overview is based on results that appear in [2, 1, 3, 4].

References

1 Omer Gold and Micha Sharir. *Improved Bounds for 3SUM, k-SUM, and Linear Degeneracy*. CoRR, abs/1512.05279, 2015
2 Omer Gold and Micha Sharir. *On the Complexity of the Discrete Fréchet Distance under L_1 and L_∞*. The 31st European Workshop on Computational Geometry (EuroCG), 2015
3 Omer Gold and Micha Sharir. *Dominance Products and Faster Algorithms for High-Dimensional Closest Pair under L_∞*. CoRR, abs/1605.08107, 2016
4 Omer Gold and Micha Sharir. *Dynamic Time Warping and Geometric Edit Distance: Breaking the Quadratic Barrier*. CoRR, abs/1607.05994, 2016

3.10 How Hard is it to Find (Honest) Witnesses?

Isaac Goldstein (Bar-Ilan University – Ramat Gan, IL)

Joint work of Isaac Goldstein, Tsvi Kopelowitz, Moshe Lewenstein, Ely Porat
Main reference I. Goldstein, T. Kopelowitz, M. Lewenstein, E. Porat: "How Hard is it to Find (Honest) Witnesses?", in Proc. of the 24th Annual European Symposium on Algorithms (ESA 2016), LIPIcs, Vol. 57, pp. 45:1–45:16, chloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2016.
URL http://dx.doi.org/10.4230/LIPIcs.ESA.2016.45

In recent years much effort was put into developing polynomial-time conditional lower bounds for algorithms and data structures in both static and dynamic settings. Along these lines we suggest a framework for proving conditional lower bounds based on the well-known 3-SUM conjecture. Our framework creates a compact representation of an instance of the 3-SUM problem using hashing and domain specific encoding. This compact representation admits false solutions to the original 3-SUM problem instance which we reveal and eliminate until we find a true solution. In other words, from all witnesses (candidate solutions) we figure out if an honest one (a true solution) exists. This enumeration of witnesses is used to prove conditional lower bound on reporting problems that generate all witnesses. In turn, these reporting problems are reduced to various decision problems. These help to enumerate the witnesses by constructing appropriate search data structures. Hence, 3SUM-hardness of the decision problems is deduced.

We utilize this framework to show conditional lower bounds for several variants of convolutions, matrix multiplication and string problems. Our framework uses a strong connection between all of these problems and the ability to find witnesses.

Specifically, we prove conditional lower bounds for computing partial outputs of convolutions and matrix multiplication for sparse inputs. These problems are inspired by the open question raised by Muthukrishnan 20 years ago. The lower bounds we show rule out the possibility (unless the 3-SUM conjecture is false) that almost linear time solutions to sparse input-output convolutions or matrix multiplications exist. This is in contrast to standard convolutions and matrix multiplications that have, or assumed to have, almost linear solutions.

Moreover, we improve upon the conditional lower bounds of Amir et al. for histogram indexing, a problem that has been of much interest recently. The conditional lower bounds we show apply for both reporting and decision variants. For the well-studied decision variant, we show a full tradeoff between preprocessing and query time for every alphabet size > 2. At an extreme, this implies that no solution to this problem exists with subquadratic preprocessing time and $O(1)$ query time for every alphabet size > 2, unless the 3-SUM conjecture is false. This is in contrast to a recent result by Chan and Lewenstein for a binary alphabet. While these specific applications are used to demonstrate the techniques of our framework, we believe that this novel framework is useful for many other problems as well.

3.11 Parameterised graph distance problems

Thore Husfeldt (IT University of Copenhagen, DK)

Main reference T. Husfeldt, "Computing Graph Distances Parameterized by Treewidth and Diameter", 11th Int'l Symposium on Parameterized and Exact Computation (IPEC 2016), LIPIcs, Vol. 63, pp. 16:1–16:11, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2016.
URL http://dx.doi.org/10.4230/LIPIcs.IPEC.2016.16

We study the complexity of computing the diameter and other distance measures in an unweighted, undirected graph. We sketch the ideas behind a tree decomposition-based repeated traversal that computes the diameter in time $n \exp(t \log d)$, where t is the treewidth and d is the diameter ([Husfeldt, IPEC 2016]), which matches a lower bound under the Strong Exponential Time Hypothesis of Abboud, Vassilevska Williams, and Wang [SODA 2016] for constant diameter. We observe that simple arguments establish tight bounds under the same hypothesis when the problem is paraemterised by vertex cover number and (with some help from the audience) domination number.

3.12 Finding Even Cycles

Mathias Bæk Tejs Knudsen (University of Copenhagen, DK)

Joint work of Mathias Bæk Tejs Knudsen, Søren Dahlgaard, Morten Stöckel.

We study the problem of finding a $2k$-cycle in a graph, for constant values of k. Previous results showed that it is possible to do this in time $O(n^2)$, and we have improved this to $O(m^(2k/(k+1)))$, where n and m is the number of nodes and edges in the graph. Since any graph with $m \gg n^(1+1/k)$ edges contains a $2k$-cycle, this bound is at least as good as the $O(n^2)$ bound.

I will tell a little bit about the result and then focus on why it seems difficult to show a conditional lower bound of $n^{2-o(1)}$, since it implies solving a problem related to the Erdos Girth Conjecture. However, this does not rule out the possibility, that it is easy (for whatever definition of "easy") to show a lower bound of $m^{2k/(k+1)-o(1)}$.

3.13 Birthday Repetition: Tool for proving quasi-poly hardness

Young Kun Ko (Princeton University, US)

In this talk we give a broad introduction to Birthday Repetition, a technique introduced to prove quasi-polynomial hardness of "Free" game assuming the Exponential Time Hypothesis. The main observation of the technique is Birthday Paradox, in particular that aggregating the variables in 2-CSPs to a tuple of size $\tilde{O}(\sqrt{n})$ then choosing two tuples at random will have a challenge from original 2-CSP with high probability. No prior material is assumed.

3.14 Tight Bounds for Gomory-Hu-like Cut Counting

Robert Krauthgamer (Weizmann Institute – Rehovot, IL)

Joint work of Rajesh Chitnis, Lior Kamma, and Robert Krauthgamer.
Main reference R. Chitnis, L. Kamma, R. Krauthgamer, "Tight Bounds for Gomory-Hu-like Cut Counting", in Proc. of the 42nd Int'l Workshop on Graph-Theoretic Concepts in Computer Science (WG'16), LNCS, Vol. 9941, pp. 133–144, Springer, 2016.
URL http://dx.doi.org/10.1007/978-3-662-53536-3_12

A classical result of Gomory and Hu from 1961 shows that in every edge-weighted undirected graph $G = (V, E, w)$, the minimum st-cut values, when ranging over all $s, t \in V$, take at most $|V| - 1$ distinct values. That is, these $\binom{|V|}{2}$ instances exhibit "redundancy" by factor $\Omega(|V|)$. They further showed how to construct a tree on V that stores all minimum st-cut values.

Motivated by this result, we obtain tight bounds for the redundancy factor of several generalizations of minimum st-cut, namely, Multiway-Cut, Multicut, and Group-Cut. A natural application of these bounds is to construct small data structures that store all the cut values for these problems, a la the Gomory-Hu tree. We initiate this direction by giving some upper and lower bounds.

3.15 Advances in fully dynamic algorithms with worst-case update time

Sebastian Krinninger (MPI für Informatik – Saarbrücken, DE)

Joint work of Sebastian Krinninger, Ittai Abraham, Shiri Chechik

A major goal in dynamic graph algorithms is to strengthen amortized update time guarantees to to hard worst-case guarantees. As we have learned from recent conditional lower bounds, this unfortunately might not always be possible in many cases. In other cases, it is on open problem how much better the worst-case update time guarantees can get. I will give a short overview and then present my recent contributions in this area.

3.16 Deterministic Time-Space Tradeoffs for k-SUM

Andrea Lincoln (Stanford University, US), Joshua R. Wang, Ryan Williams, and Virginia Vassilevska Williams (Stanford University, US)

Main reference A. Lincoln, V. Vassilevska Williams, J. R. Wang, R. R. Williams, "Deterministic Time-Space Tradeoffs for k-SUM", arXiv:1605.07285v1 [cs.DS], 2016.
URL https://arxiv.org/abs/1605.07285v1

Given a set of numbers, the k-SUM problem asks for a subset of k numbers that sums to zero. When the numbers are integers, the time and space complexity of k-SUM is generally studied in the word-RAM model; when the numbers are reals, the complexity is studied in the real-RAM model, and space is measured by the number of reals held in memory at any point.

We present a time and space efficient deterministic self-reduction for the k-SUM problem which holds for both models, and has many interesting consequences. To illustrate:

- 3-SUM is in deterministic time $O(n^2 \lg\lg(n)/\lg(n))$ and space $O\left(\sqrt{\frac{n \lg(n)}{\lg\lg(n)}}\right)$. In general, any polylogarithmic-time improvement over quadratic time for 3-SUM can be converted into an algorithm with an identical time improvement but low space complexity as well.
- 3-SUM is in deterministic time $O(n^2)$ and space $O(\sqrt{n})$, derandomizing an algorithm of Wang.
- A popular conjecture states that 3-SUM requires $n^{2-o(1)}$ time on the word-RAM. We show that the 3-SUM Conjecture is in fact equivalent to the (seemingly weaker) conjecture that every $O(n^{.51})$-space algorithm for 3-SUM requires at least $n^{2-o(1)}$ time on the word-RAM.
- For $k \geq 4$, k-SUM is in deterministic $O(n^{k-2+2/k})$ time and $O(\sqrt{n})$ space.

3.17 Continuous Optimization Based Maximum Flow Algorithms Make Sense

Aleksander Madry (MIT – Cambridge, US)

I will explain how to compute the maximum flow in a graph by iteratively routing electrical flows in the residual graph. The resulting algorithm provides the state of the art running time bounds for the unit capacity maximum flow problem.

3.18 Shortest cycle approximation

Liam Roditty (Bar-Ilan University – Ramat Gan, IL)

Joint work of Liam Roditty, Virginia Vassilevska Williams

We study the problem of determining the girth of an unweighted undirected graph.

In this talk I will survey efficient approximation algorithms with additive and multiplicative approximations from the paper [1].

References

1 Liam Roditty, Virginia Vassilevska Williams. *Subquadratic time approximation algorithms for the girth.* Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, 2012.

3.19 Towards Conditional Lower Bounds for Tree Edit Distance

Oren Weimann (University of Haifa, IL)

Joint work of Karl Bringmann, Paweł Gawrychowski, Shay Mozes, Oren Weimann

The tree edit distance (TED) between two labeled trees T and T' is the minimum cost of transforming one tree into the other by a sequence of elementary operations consisting of deleting and relabeling existing nodes, as well as inserting new nodes. The current fastest algorithm for TED requires $O(n^3)$ time. In terms of conditional lower bounds, the problem has a string edit distance (hence SETH) flavor, yet a cubic (hence APSP) complexity. In my talk, I presented a joint work with Paweł Gawrychowski and Shay Mozes showing a possible first step towards APSP-hardness. Namely, a reduction from APSP to TED under the assumption that in TED we seek the edit distance between every subtree of T and every subtree of T'. All existing TED algorithms actually compute this all subtree-to-subtree information.

Right after the talk, Karl Bringmann who was in the audience came up with a way to remove the subtree-to-subtree assumption in the APSP to TED reduction, thus achieving APSP-hardness for TED. The reduction still required $\Omega(n)$ different labels. In the following few days at Dagstuhl, together with Karl we extended the reduction to a reduction from Max-weighted k-Clique to TED that requires only $O(1)$ different labels.

We are currently writing-up these results and plan to submit them soon to a conference. We are grateful for Dagstuhl and feel that such outcome as the above can only happen in meetings like Dagstuhl.

3.20 Fine-Grained Complexity and Conditional Hardness for Sparse Graphs

Vijaya Ramachandran (University of Texas – Austin, US)

Joint work of Udit Agarwal, Vijaya Ramachandran
Main reference U. Agarwal, V. Ramachandran, "Fine-Grained Complexity and Conditional Hardness for Sparse Graphs", arXiv:1611.07008v1 [cs.DS], 2016.
URL https://arxiv.org/abs/1611.07008v1

There is a large class of path and cycle problems on graphs that currently have $\tilde{O}(n^3)$[1] time algorithms. Graphs encountered in practice are typically sparse, with the number of edges m being close to linear in n, the number of vertices, or at least with $m << n^2$. When considering sparsity, the current time complexities of these problems split into two classes: the $\tilde{\Theta}(mn)$ class, which includes APSP, Betweenness Centrality, and Minimum-Weight-Cycle, among several other problems, and the $\Theta(m^{3/2})$ class, which includes all problems relating to enumerating and detecting triangles. Here n and m are the number of vertices and edges in the graph. We investigate the fine-grained complexity of these problems on sparse graphs, and our main results are the following:

[1] $\tilde{O}$ hides polylog factors. For APSP on dense graphs, we use it to also hide a larger, but sub-polynomial factor

1. **Reductions and Algorithms.** We define the notion of a sparse reduction that preserves graph sparsity, and we present several such reductions for graph problems in the $\tilde{O}(mn)$ class. This gives rise to a rich partial order on graph problems with $\tilde{O}(mn)$ time algorithms, with the Minimum-Weight-Cycle problem as a major source in this partial order, and APSP a major sink. Surprisingly, very few of the known subcubic results are sparse reductions (outside of a few reductions that place Centrality problems in the sub-cubic equivalence class). We develop new techniques in order to preserve sparsity in our reductions, many of which are nontrivial and intricate. Some of our reductions also lead to improved algorithms for various problems on finding simple cycles in undirected graphs.
2. **Conditional Hardness.** We establish a surprising conditional hardness result for sparse graphs: We show that if the Strong Exponential Time Hypothesis (SETH) holds, then several problems in the $\tilde{O}(mn)$ class, including certain problems that are also in the sub-cubic equivalence class such as Betweenness Centrality and Eccentricities, *cannot* have 'sub-mn' time algorithms, i.e., algorithms that run in $O(m^{\alpha} \cdot n^{2-\alpha-\epsilon})$ time, for constants $\alpha \geq 0$, $\epsilon > 0$. In particular, this result means that under SETH, the sub-cubic equivalence class is split into at least two classes when sparsity is taken into account, with triangle finding problems having faster algorithms than Eccentricities or Betweenness Centrality. This hardness result for the $\tilde{O}(mn)$ class is also surprising because a similar hardness result for the sub-cubic class is considered unlikely since this would falsify NSETH (Nondeterministic SETH).

3.21 Computing Min-Cut with truly subquadratic cut queries

Aviad Rubinstein (University of California – Berkeley, US)

Joint work of Aviad Rubinstein, Tselil Schramm, Matt Weinberg

I describe preliminary progress on the problem of computing an exact minimum cut of an unknown graph, when the graph is accessed via queries to a cut-value oracle.

3.22 On the oblivious adversary assumption in dynamic problems

Thatchaphol Saranurak (KTH Royal Institute of Technology – Stockholm, SE) and Danupon Nanongkai (KTH Royal Institute of Technology – Stockholm, SE)

Main reference D. Nanongkai, T. Saranurak, "Dynamic Spanning Forest with Worst-Case Update Time: Adaptive, Las Vegas, and $O(n^{1/2-\epsilon})$-Time", arXiv:1611.03745v1 [cs.DS], 2016.
URL https://arxiv.org/abs/1611.03745v1

Many dynamic randomized algorithms make the "oblivious adversary assumption" i.e. they assume that the adversary fixed all the updates before it sees any output of the dynamic algorithm. This is in contrast to an "adaptive adversary" that see all previous algorithm's outputs before it generates a new update. It is a fundamental question whether the true source of power of randomized dynamic algorithms is the randomness itself or in fact the oblivious adversary assumption.

For example, in dynamic spanning forest problem which plays a central role in the development of dynamic graph algorithms, there is a randomized algorithm with polylog worst-case update time against oblivious adversaries but the best-known algorithms against adaptive adversaries have $O(\sqrt{n})$ update time.

In this talk, I will try to motivate an approach for understanding the power of adaptive adversaries in dynamic spanning forest problem via a problem called "dynamic cut oracle". This problem is interesting for two reasons. First, if it is "hard", this would separate the two models of oblivious vs. adaptive adversaries. That is, it implies that there is no algorithm with polylog worst-case update time against adaptive adversaries for dynamic spanning forest. Second, the technique we used for studying dynamic cut oracle leads to a new exciting algorithm for dynamic spanning forest itself, which indicates that this problem might capture the hardness of dynamic spanning forest problem.

3.23 Subquadratic Algorithms for Succinct Stable Matching

Stefan Schneider (University of California – San Diego, US)

Joint work of Marvin Künnemann, Daniel Moeller, Ramamohan Paturi, Stefan Schneider
Main reference M. Künnemann, D. Moeller, R. Paturi, S. Schneider, "Subquadratic Algorithms for Succinct Stable Matching", arXiv:1510.06452v5 [cs.DS], 2016.
URL https://arxiv.org/abs/1510.06452v5

We consider the stable matching problem when the preference lists are not given explicitly but are represented in a succinct way and ask whether the problem becomes computationally easier and investigate other implications. We give subquadratic algorithms for finding a stable matching in special cases of natural succinct representations of the problem, the d-attribute, d-list, geometric, and single-peaked models. We also present algorithms for verifying a stable matching in the same models. We further show that for $d = \omega(logn)$ both finding and verifying a stable matching in the d-attribute and d-dimensional geometric models requires quadratic time assuming the Strong Exponential Time Hypothesis. This suggests that these succinct models are not significantly simpler computationally than the general case for sufficiently large d.

3.24 RNA-Folding: From Hardness to Algorithms

Virginia Vassilevska Williams (Stanford University, US)

Joint work of Karl Bringmann, Fabrizio Grandoni, Barna Saha, Virginia Vassilevska Williams
Main reference V. Vassilevska Williams, "RNA-Folding - From Hardness to Algorithms", in Proc. of the 41st Int'l Symposium on Mathematical Foundations of Computer Science (MFCS 2016), LIPIcs, Vol. 58, pp. 5:1–5:1, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2016.
URL http://dx.doi.org/10.4230/LIPIcs.MFCS.2016.5

A fundamental problem in computational biology is predicting the base-pairing of an RNA secondary structure. Most algorithms for this rely on an algorithm for a simplified version of this problem, RNA-folding, defined as follows: given a sequence S of letters over the alphabet $\{A, U, C, G\}$ where A can only be paired with U and C can only be paired with G, determine the best "folding" of S, i.e. a maximum size *nested* pairing of the symbols of S. For instance, in the sequence ACUG the best pairing is either matching A with U, or matching C with G, but not both as that pairing wouldn't be nested.

A dynamic programming algorithm from 1980 by Nussinov and Jacobson solves the RNA-folding problem on an n letter sequence in $O(n^3)$ time. Despite many efforts, until recently, the best algorithms for RNA-folding only shaved small logarithmic factors over this cubic running time.

Recent work [1] explained why it has been so difficult to obtain faster algorithms: if one can solve RNA-folding on n length strings faster than one can currently multiply n by n matrices, then the Clique problem would have surprisingly fast algorithms.The current fastest algorithm to multiply n by n matrices runs in $O(n^{2.373})$ time and the fastest known Clique algorithms use this result. Obtaining an $O(n^{2.36})$ time algorithm for RNA-folding would thus be potentially difficult as it would imply a breakthrough for Clique algorithms and potentially also for matrix multiplication.

While this hardness result is appealing, it does not explain the seeming n^3 barrier. No better hardness seemed possible to us, and thus it became increasingly more plausible that RNA-folding should have a faster algorithm and in fact one using fast matrix multiplication. Indeed, this turned out to be true. In this talk I will strive to give some insight into the first truly subcubic time algorithm for the problem.

References

1 Abboud, Amir, Arturs Backurs, and Virginia Vassilevska Williams. *If the current clique algorithms are optimal, so is Valiant's parser.* Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on. IEEE, 2015.

3.25 Amortized Dynamic Cell-Probe Lower Bounds from Four-Party Communication

Huacheng Yu (Stanford University, US)

Joint work of Omri Weinstein, Huacheng Yu
Main reference O. Weinstein, H. Yu, "Amortized Dynamic Cell-Probe Lower Bounds from Four-Party Communication", in Proc. of the 57th Annual Symposium on Foundations of Computer Science (FOCS'16), pp. 305–314, IEEE, 2016.
URL http://dx.doi.org/10.1109/FOCS.2016.41

This paper develops a new technique for proving amortized, randomized cell-probe lower bounds on dynamic data structure problems. We introduce a new randomized nondeterministic four-party communication model that enables "accelerated", error-preserving simulations of dynamic data structures.

We use this technique to prove an $\Omega(n(\log n/\log\log n)^2)$ cell-probe lower bound for the dynamic 2D weighted orthogonal range counting problem (2D-ORC) with $n/poly\log n$ updates and n queries, that holds even for data structures with $\exp(-\tilde{\Omega}(n))$ success probability. This result not only proves the highest amortized lower bound to date, but is also tight in the strongest possible sense, as a matching upper bound can be obtained by a deterministic data structure with worst-case operational time. This is the first demonstration of a "sharp threshold" phenomenon for dynamic data structures.

Our broader motivation is that cell-probe lower bounds for exponentially small success facilitate reductions from dynamic to static data structures. As a proof-of-concept, we show that a slightly strengthened version of our lower bound would imply an $\Omega((\log n/\log\log n)^2)$

lower bound for the static 3D-ORC problem with $O(n \log^{O(1)} n)$ space. Such result would give a near quadratic improvement over the highest known static cell-probe lower bound, and break the long standing $\Omega(\log n)$ barrier for static data structures.

4 Open problems

The following open problems were contributed by the seminar attendees, and compiled and edited by the organizers.

4.1 Parameterizing problems in P by treewidth

Fedor V. Fomin (University of Bergen, NO)

Background. Let t be the treewidth of an input graph. Many **NP**-hard problems, particularly those expressible in MSOL, are solvable in $f(t)n$ time and there are lower bounds on the (exponential) function f conditioned on the Strong Exponential Time Hypothesis (SETH) [26]. For problems in **P** the picture is less clear. Consider your favorite problem Π in **P** solvable in $T_\Pi(n)$ time on a graph wtih n vertices. Some problems Π admit algorithms running in $\mathrm{poly}(t) \cdot o(T_\Pi(n))$ time whereas others do not. For example, Abboud et al. [5] proved that *Diameter* can be solved in $2^{O(t \log t)} n^{1+o(1)}$ time, yet a $2^{o(t)} n^{2-\epsilon}$ time algorithm would refute SETH. On the other hand, *maximum cardinality matching* can be solved in randomized $O(t^3 \cdot n \log n)$-time [31].

Question. Classify graph problems in **P** according to their dependence on treewidth. Which problems admit $f(t) \cdot n^{1+o(1)}$-time algorithms with polynomial f, and which require exponential f? A specific goal is the determine whether *maximum weight perfect matching* has an $\tilde{O}(\mathrm{poly}(t)n)$ algorithm, for integer weights from a polynomial range.

Main paper reference: Abboud et al. [5], Fomin et al. [31].

4.2 Approximate all-pairs shortest paths

Amir Abboud (Stanford University, US)

Background. In unweighted, undirected graphs, we can compute All Pairs Shortest Paths (APSP) in $O(n^3)$ time with a fast "combinatorial" algorithm, or in $O(n^\omega)$ time, where $\omega < 2.373$ is the matrix multiplication exponent. It is conjectured that a truly subcubic combinatorial algorithm does not exist, which is equivalent to the combinatorial Boolean matrix multiplication conjecture.

What about approximation algorithms? The best kind of approximation is an additive $+2$, so that for all pairs u, v we return a value that is between $d(u, v)$ and $d(u, v) + 2$. Dor, Halperin, and Zwick [30] presented a *combinatorial* algorithm with runtime $\tilde{O}(n^{7/3})$. Note that this runtime is currently even better that $O(n^\omega)$, and has the advantage of being practical.

Questions. Is there a conditional lower bound for +2-APSP? Can we show that a combinatorial algorithm must spend $n^{7/3-o(1)}$ time? Would a faster non-combinatorial algorithm require improvements to ω? Alternatively, is there an $\tilde{O}(n^2)$ time algorithm for +2-APSP?

Main paper reference: Dor, Halperin, and Zwick [30].

4.3 Approximate diameter

Amir Abboud (Stanford University, US)

Background. Computing the diameter of a sparse graph in truly subquadratic time refutes SETH: Roditty and Vassilevska Williams [62] showed that a $(3/2-\varepsilon)$-approximation to the diameter requires $n^{2-o(1)}$ time, even on a sparse unweighted undirected graph under SETH. On the other hand, there are algorithms [62, 22] that give a (roughly) 3/2 approximation in $\tilde{O}(m\sqrt{n})$ time on unweighted graphs, or $\tilde{O}(\min\{m^{3/2}, mn^{2/3}\})$ time on weighted graphs. Extending these algorithms further, Cairo et al. [19] showed that for all integers $k \geq 1$, there is an $\tilde{O}(mn^{1/(k+1)})$ time algorithm that approximates the diameter of an undirected unweighted graph within a factor of (roughly) $2 - 1/2^k$.

Question. If we insist on near-linear runtime, what is the best approximation factor we can get? It is easy to see that a 2-approximation can be achieved in linear time, but what about an α-approximation, where $3/2 \leq \alpha < 2$?

Main paper reference: Roditty and Vassilevska W. [62].

4.4 Finding cycles and approximating the girth

Mathias Bæk Tejs Knudsen (University of Copenhagen, DK) and Liam Roditty (Bar-Ilan University – Ramat Gan, IL)

Background. Consider an unweighted undirected graph $G = (V, E)$. The *girth* of G is the length of the shortest cycle. The problem of detecting 3-cycles (and odd cycles of any length) is reducible to matrix multiplication and there are reductions in the reverse direction; see [66]. Yuster and Zwick [67] showed that detecting $2k$-cycles can be computed in $O(f(k)n^2)$ time, where f is exponential.

Question. For any fixed constant k, give a conditional lower bound, showing that there does not exist an algorithm deciding whether G contains a $2k$-cycle in time $O(f(k)n^{2-\varepsilon})$ for any $\varepsilon > 0$, or one running in $O(f(k)m^{2k/(k+1)-\varepsilon})$ time, where m is the number of edges.

Main paper reference: Yuster and Zwick [67].

Question. Prove or disprove the following conjecture: There exists a truly subquadratic algorithm for finding a 4-cycle in a graph if and only if there exists a truly subquadratic algorithm for finding a multiplicative $(2-\varepsilon)$-approximation of the girth.

Question. Prove or disprove the following conjecture from [63]: the problem of detecting a 3-cycle in a graph G without 4- and 5-cycles requires $n^{2-o(1)}$ time. Note that if there exists a subquadratic $(2-\epsilon)$-approximation for the girth, it must be able to detect 3-cycles in graphs without 4- and 5-cycles. See [63] for more details.

Main paper reference: Roditty and Vassilevska W. [63].

4.5 Minimum cycle problem in directed graphs

Virginia Vassilevska Williams (Stanford University, US)

Background. Given an unweighted *directed* graph $G = (V, E)$ on n vertices, the problem is to find a shortest cycle in G. The potentially simpler *Girth* problem asks to compute just the length of the shortest cycle.

The girth and the minimum cycle can be computed in $O(n^\omega)$ time exactly, as shown by Itai and Rodeh [49], where $\omega < 2.373$. It is easy to see that the minimum cycle problem is at least as hard as finding a triangle in a graph. In fact, even obtaining a $(2-\delta)$-approximation for the girth for any constant $\delta > 0$ is at least as hard as triangle detection. The fastest algorithm for the Triangle problem in n node graphs runs in $O(n^\omega)$ time.

Question. Is there any $O(1)$-approximation algorithm for the girth that runs faster than $O(n^\omega)$ time? In recent work, Pachocki, Roditty, Sidford, Tov, and Vassilevska Williams [59] showed that for any integer k, there is an $\tilde{O}(mn^{1/k})$ time $O(k \log n)$ approximation algorithm for the Minimum Cycle problem. Thus, in nearly linear time, one can obtain an $O(\log^2 n)$-approximation. Can one improve the approximation factor further? Can one even obtain a constant factor approximation in linear time?

Main paper reference: Pachocki et al. [59].

4.6 Linear Programming

Aleksander Madry (MIT – Cambridge, US)

Background. Consider a linear program of the following form: minimize $c^T x$ subject to $Ax \geq b$, where A is an d-by-n constraint matrix. Suppose that we could solve any such LP in time

$$\widetilde{O}\left(\left(\mathrm{nnz}(A) + d^2\right) d^\delta \log L\right),$$

where $\mathrm{nnz}(A)$ is the number of non-zero entries of A, L is the bound on the bit complexity of the input entries, and δ is a positive constant.

Question. Is there some value of δ for which the above (hypothetical) running time bound would disprove any of the popular hardness conjectures?

In [57], it is shown that one can achieve the above running time bound for $\delta = \frac{1}{2}$.

Main paper reference: Lee and Sidford [57].

4.7 Fully dynamic APSP

Sebastian Krinninger (MPI für Informatik – Saarbrücken, DE)

Background. In the fully dynamic all-pairs shortest paths (APSP) problem we are interested in maintaining the distance matrix of a graph under insertions and deletions of nodes. Demetrescu and Italiano [28] showed that the distance matrix can be updated in *amortized* time $\tilde{O}(n^2)$ after each node update. The current fastest *worst case* algorithms have update times of $\tilde{O}(n^{2+2/3})$ (randomized Monte Carlo [7]) and $\tilde{O}(n^{2+3/4})$ (deterministic [65]).

Questions. Can the worst case update time $\tilde{O}(n^2)$ be achieved? A barrier for current algorithmic approaches is $n^{2.5}$. Is there a conditional lower bound showing this to be a true barrier?

Main paper reference: Abraham et al. [7].

4.8 Dynamic reachability in planar graphs

Søren Dahlgaard (University of Copenhagen, DK)

Background. *Dynamic reachability* in a planar graph G is the problem of maintaining a data structure supporting the following operations: (i) Insert a directed edge (u, v) into G, (ii) delete an edge from G, and (iii) query whether v is reachable from u in G.

An algorithm with update and query time $\tilde{O}(\sqrt{n})$ is known (Diks and Sankowski [29]) for dynamic *plane graphs*—that is, the graph is dynamic but the plane embedding is fixed.

Question. Does an $n^{1/2-\Omega(1)}$ algorithm exist or is there a conditional $n^{1/2-o(1)}$ hardness result? Any polynomial hardness result would be interesting. A good place to start for the latter part would be the recent paper by Abboud and Dahlgaard [3] about hardness for dynamic problems in planar graphs.

Main paper reference: Abboud and Dahlgaard [3].

4.9 Static hardness for planar graphs

Søren Dahlgaard (University of Copenhagen, DK)

Background. An important direction is to show conditional hardness for important problems, even on restricted (easier) classes of graphs, e.g., planar graphs. Abboud and Dahlgaard [3] recently showed hardness for several *dynamic* problems in planar graphs, but nothing is known for static problems.

Question. On planar graphs, many problems (such as shortest paths, multi-source multi-sink max-flow, etc.) run in near-linear time. Can we show that some problem does not? No hardness results are known for any static problem in **P** on planar graphs. Two candidate problems to consider are *diameter* and *sum of distances*. Both require subquadratic time (Cabello [18]), but it may still be possible to show a hardness result, e.g., $n^{3/2-o(1)}$ hardness.

Main paper reference: Cabello [18].

4.10 Sparse reductions for graph problems

Vijaya Ramachandran (University of Texas – Austin, US)

Background. Many graph problems are known to be as hard as APSP on dense graphs [66, 4, 64], in the sense that a subcubic algorithm for any of them implies a subcubic algorithm for all of them. When the graph sparsity is taken into account, these problems currently are no longer in a single class: many have $\tilde{O}(mn)$-time algorithms whereas finding minimum weight triangle and related problems have $\tilde{O}(m^{3/2})$-time algorithms. Most known fine-grained reductions between graph problems do not preserve the graph sparsity. Until recently, the only examples of sparseness-preserving truly subcubic reductions appeared in [4]. Agarwal and Ramachandran [8] presented several more such reductions, strengthening the connections between problems with $\tilde{O}(mn)$-time algorithms. A reduction from CNF-SAT to Diameter was presented in [62] to give SETH-hardness results for Diameter and Eccentricities. The notion of a *sub-mn time bound* was formalized later, in [8], where it was observed that the reduction in [62] gives SETH-hardness for any sub-mn time bound for these problems.

Questions. Is there a sparseness-preserving, $\tilde{O}(n^2)$ time reduction from undirected weighted All Nodes Shortest Cycles (ANSC) to APSP? Is there a sparseness-preserving, $\tilde{O}(m+n)$ time reduction from undirected Min-Wt-Cycle to either Radius or Eccentricities? Is it SETH-hard to find a sub-mn bound for Min-Wt-Cycle or an $O(n^2+$ sub-$mn)$ bound on APSP?

Main paper reference: Agarwal and Ramachandran [8].

4.11 Hardness for partially dynamic graph problems

Søren Dahlgaard (University of Copenhagen, DK)

Background. Many results show hardness for *fully*-dynamic problems in graphs, but the techniques do not seem to extend well to amortized lower bounds in the incremental and decremental cases. (See Abboud and Vassilevska Williams [2], Henzinger, Krinninger, Nanongkai, and Saranurak [46], Kopelowitz, Pettie and Porat [55], and Dahlgaard [27] for some initial results on incremental/decremental problems.)

Question. Develop general techniques for showing amortized hardness of partially dynamic problems in graphs. One candidate problem is decremental single-source reachability. A result of Chechik, Hansen, Italiano, Lacki, and Parotsidis [23] shows that $\tilde{O}(m\sqrt{n})$ total time is sufficient. Is it necessary?

4.12 Hardness of vertex connectivity

Veronika Loitzenbauer (Universität Wien, AT)

Background. A connected undirected graph is k-vertex (resp. edge) connected if it remains connected after any set of at most $k-1$ vertices (edges) is removed from the graph. A strongly connected directed graph is k-vertex (edge) connected if it remains strongly connected after any set of at most $k-1$ vertices (edges) is removed from the graph. The vertex (edge) connectivity of a graph is the maximum value of k such that the graph is k-vertex (edge) connected.

The edge-connectivity λ of an undirected graph can be determined in time $O(m \log^2 n \log^2 \log n)$ [47, 54], and for directed graphs in time $O(\lambda m \log(n^2/m))$ [37]. In contrast, the vertex-connectivity κ can only be computed in time $O((n + \min\{\kappa^{5/2}, \kappa n^{3/4}\})m)$ [38], where for undirected graphs m can be replaced by kn.

Question. To *check* k-vertex connectivity means to either confirm that $\kappa \geq k$ or to find a set of $k-1$ vertices that disconnects the graph. Even when k is constant, no $o(n^2)$ time (or $o(mn)$ time for directed graphs) algorithms are known for checking k-connectivity. Is there a conditional superlinear lower bound?

Main paper reference: Gabow [38].

4.13 Parity and mean-payoff games

Veronika Loitzenbauer (Universität Wien, AT)

Background. *Parity games*, and their generalization *mean-payoff games*, are among the rare "natural" problems in NP ∩ co-NP (and in UP ∩ co-UP [52]) for which no polynomial-time algorithm is known. Both parity games and mean-payoff games are 2-player games played by taking an infinite walk on a directed graph; one of the vertices is designated the start vertex. In parity games each vertex is labeled by an integer in $[0, c]$; in mean payoff games each edge is labeled by an integer in $[-W, W]$. (See [53] for a description of the game.) The algorithmic question is to decide, for each start vertex, which of the two players wins the game and to construct a corresponding winning strategy. Parity games can be reduced to mean-payoff games with $W = n^c$. Very recently, quasi-polynomial $O(n^{\log c})$ time algorithms for parity games were discovered [20, 51]. The best known algorithms for mean-payoff games run in pseudo-polynomial time $O(mnW)$ [16] and randomized sub-exponential time $O(2^{\sqrt{n \log n}} \log W)$ [15].

Questions. Is there a polynomial-time algorithm for parity or mean-payoff games? Are there conditional superlinear lower bounds on these problems?

4.14 Unknotting

Seth Pettie (University of Michigan – Ann Arbor, US)

Background. A *knot* is a closed, non-self-intersecting polygonal chain in $\mathbb{R}^3$. Two knots are *equivalent* if one can be continuously deformed into the other without self-intersection. The *unknot* problem is to decide if a knot is equivalent to one that is embeddable in the plane.

Knots can be represented combinatorially, by projecting the polygonal chain onto $\mathbb{R}^2$, placing a vertex wherever two edges intersect. The result is a 4-regular planar graph (possibly with loops and parallel edges) where each vertex carries a bit indicating which pair of edges is "over" and which pair is "under." *Reidemeister moves* (a small set of transformations on the knot diagram) suffice to transform any knot diagram to one of its equivalent representations.

The complexity of *unknot* and related problems (e.g., are two knots equivalent?, can two knots simultaneously embedded in $\mathbb{R}^3$ be untangled?) are known to be in **NP** [45] and solvable in $2^{O(n)}$ time [45, 50].

Questions. Given a plane knot diagram with n intersections, can *unknot* or *knot-equivalence* be solved in time near-linear in n? If not, are there conditional lower bounds that show even some polynomial hardness?

4.15 3-Collinearity (general position testing)

Omer Gold (Tel Aviv University, IL)

Background. A set S of n points in $\mathbb{R}^2$ is said to be in *general position* if there do not exist three points in S that lie on a line. The *3-Collinearity* problem is to test whether S is in general position. The 3-Collinearity problem is known to be as hard as 3SUM, and an algorithm that runs in $O(n^2)$ time is known.

Questions. The question is whether the $O(n^2)$ algorithm is optimal or whether it can be solved in $o(n^2)$ time. Recent subquadratic algorithms for 3SUM [12, 44, 34, 41] indicate that polylogarithmic improvements should be possible. A related question is whether there is an $O(n^{2-\epsilon})$-depth decision tree for 3-Collinearity; see [44, 13].

Main paper reference: Gajentaan and Overmars [39].

4.16 Element uniqueness in $X + Y$

Omer Gold (Tel Aviv University, IL)

Background. Given two sets X and Y, each of n real numbers, determine whether all the elements of $X + Y = \{x + y \mid x \in X, y \in Y\}$ are distinct. A somewhat stronger variant of this problem is to sort $X + Y$.

The decision tree complexity of sorting $X + Y$ and Element Uniqueness in $X + Y$ was shown to be $O(n^2)$ by Fredman [33].

Question. Can these problems can be solved in $o(n^2 \log n)$ time, even for the special case $X = Y$?

4.17 Histogram indexing

Isaac Goldstein (Bar-Ilan University – Ramat Gan, IL)

Background. The *histogram* $\psi(T)$ of a string $T \in \Sigma^*$ is a $|\Sigma|$-length vector containing the number of occurrences of each letter in T. The *histogram indexing problem* (aka *jumbled indexing*) is to preprocess a string T to support the following query: given a histogram vector ψ, decide whether there is a substring T' of T such that $\psi(T') = \psi$.

The state-of-the-art algorithm for histogram indexing [21] preprocesses a binary text T in $O(n^{1.859})$ time and answers queries in $O(1)$ time. Over a d-letter alphabet the preprocessing and query times are $\tilde{O}(n^{2-\delta})$ and $\tilde{O}(n^{2/3+\delta(d+13)/6})$, for any $\delta \geq 0$. On the lower bound side [10, 42], the 3SUM conjecture implies that it is impossible to simultaneously improve $n^{2-\delta}$ preprocessing and $n^{\delta(d/2-1)}$ query time by polynomial factors, where $\delta \leq 2/(d-1)$ and $d \geq 3$.

Question. Are there any non-trivial lower bounds on histogram indexing when $d = 2$? Is it possible to close the gap between the lower and upper bounds in general, or to base the hardness off of a different conjecture than 3SUM?

Main paper reference: Chan and Lewenstein [21].

4.18 Integer programming

Fedor V. Fomin (University of Bergen, NO)

Background. The objective of *Integer Programming* (IP) is to decide, for a given $m \times n$ matrix A and an m-vector $b = (b_1, \dots, b_m)$, whether there is a non-negative integer n-vector x such that $Ax = b$. In 1981, Papadimitriou [61] showed that (IP) is solvable in pseudo-polynomial time on instances for which the number of constraints m is constant. The rough estimation of the running time of Papadimitriou's algorithm is $n^{O(m)} \cdot d^{O(m^2)}$, where d bounds the magnitude of any entry in A and b. The best known lower bound is $n^{o(\frac{m}{\log m})} d^{o(m)}$ [32], assuming the Exponential Time Hypothesis (ETH).

Question. Is it possible to narrow the gap between algorithms for IP and the ETH-hardness of IP?

Main paper reference: Fomin et al. [32].

4.19 All-pairs min-cut and generalizations

Robert Krauthgamer (Weizmann Institute – Rehovot, IL)

Background. The *all-pairs min-cut* problem is, given an edge-capacitated undirected graph $G = (V, E, c)$, to compute the minimum s-t cut over all pairs $s, t \in V$. Gomory and Hu [43] showed the problem is reducible to $n-1$ s-t min-cut instances, and moreover, all $\binom{n}{2}$ min-cuts can be represented by a capacitated tree T on the vertex set V. On unweighted graphs, the construction of T takes time $\tilde{O}(mn)$ [14, 60].

Generalizations of this problem include finding the min-cut separating every triple $(r, s, t) \in V^3$, which is **NP**-hard, and finding the min-cuts separating all pairs of k-sets $\{s_1, \dots, s_k\}$ from $\{t_1, \dots, t_k\}$. See [24].

Questions. Are there superlinear conditional lower bounds for all-pairs min-cut/Gomory-Hu tree construction? (Refer to [6] for conditional lower bounds for variants of the problem on directed graphs.) Are there non-trivial conditional lower bounds for all-triplets approximate min-cut, or all-k-sets min-cut?

4.20 Parameterizing string algorithms by compressibility

Oren Weimann (University of Haifa, IL)

Background. The broad idea can be illustrated with a lower bound for string edit distance: Given two strings of length N whose compressed length (say, using Lempel-Ziv compression) is n, it is known that their edit distance can be computed in $O(nN)$ time. Is it possible to prove an $\Omega(nN)$ conditional lower bound? The known conditional lower bound [11, 1, 17] reduces CNF-SAT (with n variables) to string edit distance by creating two strings each consisting of $O(2^{n/2})$ blocks. To make such a reduction suitable for proving $\Omega(nN)$ lower bound, one needs to generate instead two strings whose length is much more than $2^{n/2}$ but that compress to much less than $2^{n/2}$.

4.21 Reductions from low complexity to high complexity

Amir Abboud (Stanford University, US)

Background. We know that improving the runtime of our 10-Clique algorithms improves the runtime of our 100-Clique algorithms. E.g., if 10-Clique can be solved in $O(n^5)$, then 100-Clique can be solved in $O(n^{50})$. In general, we have many examples of reductions showing that a faster algorithm for a problem with best known runtime $O(n^a)$, implies a faster algorithm for a problem with runtime $O(n^b)$, where $a \leq b$.

However, we have no interesting reductions in the other way, showing that improvements over n^b imply improvements over n^a, where $a < b$. In particular, we do not know how to use an algorithm that solved 100-Clique in $O(n^{50})$ or even $O(n^{11})$ time, to speed up the known algorithms for 10-Clique.

Could it be that such reductions, from low complexity to high complexity, do not exist? It is not hard to construct artificial problems where this can be done, but what about the natural problems we typically study: Clique, Orthogonal Vectors, k-SUM, APSP, LCS, etc. Can we show that a fine-grained reduction from 10-Clique to 100-Clique is unlikely due to some surprising consequences? Another candidate is 3SUM (for which the complexity is n^2) vs. APSP (for which the complexity is $N^{1.5}$, where N is the input size). We repeatedly ask if faster 3SUM implies faster APSP, but maybe proving such a result (via fine-grained reductions) has unexpected consequences?

On the other hand, it would be of great interest to find examples of such reductions between interesting and natural problems.

4.22 Stable matching in the two-list model

Stefan Schneider (University of California – San Diego, US)

Background. Gale and Shapley's stable matching [40] algorithm runs in $O(n^2)$ time (linear in the input size) and it is known that $\Omega(n^2)$ is optimal if the preference lists are arbitrary. Künnemann, Moeller, Paturi, and Schneider [56] studied the complexity of stable matching when the preference lists are constrained, and encoded in some succinct manner. Many succinct input models nonetheless require $n^{2-o(1)}$ time, conditioned on SETH.

Question. A problem left open by [56] is *two-list stable matching.* A matching market in the two-list model consists of two sets M and W, both of size n, and permutations π_1, π_2 on M and σ_1, σ_2 on W. The preference list of each agent $m \in M$ is either σ_1 or σ_2 and the preference list of each agent $w \in W$ is either π_1 or π_2. The input size is $O(n)$. The goal is to find a stable matching in the resulting matching market. Can this problem be solved in linear time, or is there a superlinear conditional lower bound?

Main paper reference: Kuennemann et al. [56]

4.23 Boolean vs. real maximum inner product

Stefan Schneider (University of California – San Diego, US)

Background. In the *maximum inner product* problem we are given two sets of d-dimensional vectors U and V of size n as well as a threshold l. The problem is to decide if there is a pair $u \in U, v \in V$ such that their inner product $u \cdot v$ is at least l. If the vectors are Boolean, then a randomized algorithm by Alman and Williams [9] solves the problem in time $n^{2-1/\Theta(c \log^2 c)}$ where $d = c \log n$. In contrast, if the vectors are real or integer, then using ray-shooting techniques [58] we can solve the problem in time $n^{2-1/\Theta(d)}$. This leaves a large gap between the two problems. In particular, the Boolean case is strongly subquadratic if $d = O(\log n)$, while the real case is only strongly subquadratic for constant d. The conditional lower bounds of [9] show that any $n^{2-\epsilon}$ algorithm when $d = \omega(\log n)$ refutes SETH.

Questions. Can the gap between the boolean and integer/real case be closed, with a better maximum inner product algorithm? If the gap is natural, can it be explained with a stronger conditional lower bound on (real or integer) maximum inner product?

4.24 Hardness of Approximating NP-hard Problems

Seth Pettie (University of Michigan – Ann Arbor, US)

Background. Many approximation algorithms for **NP**-hard problems run in polynomial time, but not linear time. This is often due to the use of general LP or SDP solvers, but not always. To take two examples, the *chromatic index* (edge coloring) and *minimum degree spanning tree* problems are **NP**-hard, but can both be approximated to within 1 of optimal in $\tilde{O}(m\sqrt{n})$ time [36] and $\tilde{O}(mn)$ time [35], respectively.

Question. Prove superlinear conditional lower bounds on the time complexity of *any* approximation problem, whose exact version is **NP**-hard.

4.25 Chromatic index/edge coloring

Marek Cygan (University of Warsaw, PL)

Background. The *chromatic index* of a graph is the least number of colors needed for a proper edge-coloring. Vizing's theorem implies that the chromatic index is either Δ or $\Delta + 1$ (where Δ is the maximum degree), but determining which one is NP-hard. The NP-hardness reduction of Holyer [48] reduces 3SAT to a 3-regular graph on $O(n)$ vertices, so the ETH implies a $2^{\Omega(n)}$ lower bound. There is an $O^*(2^m)$ algorithm for chromatic index, by reduction to vertex coloring, so the hardness is well understood when $m = O(n)$.

Questions. Does the ETH rule out a $2^{o(m)}$ algorithm for chromatic index on dense graphs? Is there, for example, an $n^{O(n)}$ or $2^{n^{2-\epsilon}}$-time algorithm?

4.26 Communication Complexity of Approximate Hamming Distance

Raphaël Clifford (University of Bristol, GB)

Background. Consider strings P of length n and T of length $2n$. Alice has the whole of P and the first half of T. That is she has P and $T[0, \ldots, n-1]$. Bob has the second half of T, that is $T[n, \ldots, 2n-1]$. Alice sends one message to Bob and Bob has to output a $(1+\epsilon)$ multiplicative approximation of $\mathrm{HD}(P, T[i, \ldots, i+n])$ for all $i \in [n]$ where HD is the Hamming Distance.

In [25] a $O(\sqrt{n} \log n/\epsilon^2)$ bit communication protocol was given.

Question. Is there a matching lower bound for the randomized one-way communication complexity of this problem?

Main paper reference. Clifford and Starikovskaya [25].

References

1 A. Abboud, A. Backurs, and V. Vassilevska Williams. Tight hardness results for LCS and other sequence similarity measures. In *Proceedings 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 59–78, 2015.

2 A. Abboud and V. Vassilevska Williams. Popular conjectures imply strong lower bounds for dynamic problems. In *Proceedings 55th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 434–443, 2014.

3 Amir Abboud and Søren Dahlgaard. Popular conjectures as a barrier for dynamic planar graph algorithms. In *Proceedings 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 477–486, 2016. doi:10.1109/FOCS.2016.58.

4 Amir Abboud, Fabrizio Grandoni, and Virginia Vassilevska Williams. Subcubic equivalences between graph centrality problems, APSP and diameter. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1681–1697, 2015. doi:10.1137/1.9781611973730.112.

5 Amir Abboud, Virginia Vassilevska Williams, and Joshua R. Wang. Approximation and fixed parameter subquadratic algorithms for radius and diameter in sparse graphs. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 377–391, 2016. doi:10.1137/1.9781611974331.ch28.

6 Amir Abboud, Virginia Vassilevska Williams, and Huacheng Yu. Matching triangles and basing hardness on an extremely popular conjecture. In *Proceedings 47th Annual ACM Symposium on Theory of Computing*, pages 41–50, 2015. doi:10.1145/2746539.2746594.

7 Ittai Abraham, Shiri Chechik, and Sebastian Krinninger. Fully dynamic all-pairs shortest paths with worst-case update-time revisited. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 440–452, 2017. doi:10.1137/1.9781611974782.28.

8 Udit Agarwal and Vijaya Ramachandran. Fine-grained complexity and conditional hardness for sparse graphs. *CoRR*, abs/1611.07008, 2016.

9 Josh Alman and Ryan Williams. Probabilistic polynomials and hamming nearest neighbors. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*. IEEE, 2015.

10 Amihood Amir, Timothy M. Chan, Moshe Lewenstein, and Noa Lewenstein. On hardness of jumbled indexing. In *International Colloquium on Automata, Languages and Programming (ICALP)*, pages 114–125, 2014.

11 A. Backurs and P. Indyk. Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). In *Proceedings 47th Annual ACM Symposium on Theory of Computing (STOC)*, pages 51–58, 2015.

12 I. Baran, E. D. Demaine, and M. Pătraşcu. Subquadratic algorithms for 3SUM. *Algorithmica*, 50(4):584–596, 2008.

13 L. Barba, J. Cardinal, J. Iacono, S. Langerman, A. Ooms, and N. Solomon. Subquadratic algorithms for algebraic generalizations of 3SUM. *CoRR*, abs/1612.02384, 2016.

14 Anand Bhalgat, Ramesh Hariharan, Telikepalli Kavitha, and Debmalya Panigrahi. An $o(mn)$ Gomory-Hu tree construction algorithm for unweighted graphs. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC)*, pages 605–614, 2007. doi:10.1145/1250790.1250879.

15 Henrik Björklund and Sergei G. Vorobyov. A combinatorial strongly subexponential strategy improvement algorithm for mean payoff games. *Discrete Applied Mathematics*, 155(2):210–229, 2007. doi:10.1016/j.dam.2006.04.029.

16 Lubos Brim, Jakub Chaloupka, Laurent Doyen, Raffaella Gentilini, and Jean-François Raskin. Faster algorithms for mean-payoff games. *Formal Methods in System Design*, 38(2):97–118, 2011. doi:10.1007/s10703-010-0105-x.

17 K. Bringmann and M. Künnemann. Quadratic conditional lower bounds for string problems and dynamic time warping. In *Proceedings 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 79–97, 2015.

18 Sergio Cabello. Subquadratic algorithms for the diameter and the sum of pairwise distances in planar graphs. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2143–2152, 2017. doi:10.1137/1.9781611974782.139.

19 Massimo Cairo, Roberto Grossi, and Romeo Rizzi. New bounds for approximating extremal distances in undirected graphs. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 363–376, 2016.

20 C. S. Calude, S. Jain, B. Khoussainov, W. Li, and F. Stephan. Deciding parity games in quasipolynomial time. In *Proceedings 49th Annual ACM Symposium on Theory of Computing (STOC)*, 2017.

21 Timothy Chan and Moshe Lewenstein. Clustered integer 3SUM via additive combinatorics. In *Proceedings 47th ACM Symposium on Theory of Computing (STOC)*, pages 31–40, 2015.

22 S. Chechik, D. Larkin, L. Roditty, G. Schoenebeck, R. E. Tarjan, and V. Vassilevska Williams. Better approximation algorithms for the graph diameter. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1041–1052, 2014.

23 Shiri Chechik, Thomas Dueholm Hansen, Giuseppe F. Italiano, Jakub Lacki, and Nikos Parotsidis. Decremental single-source reachability and strongly connected components in $\tilde{O}(m\sqrt{n})$ total update time. In *Proceedings 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 315–324, 2016. doi:10.1109/FOCS.2016.42.

24 Rajesh Chitnis, Lior Kamma, and Robert Krauthgamer. Tight bounds for gomory-hu-like cut counting. In *Proceedings 42nd International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, pages 133–144, 2016. doi:10.1007/978-3-662-53536-3_12.

25 Raphaël Clifford and Tatiana A. Starikovskaya. Approximate hamming distance in a stream. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 20:1–20:14, 2016.

26 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

27 Søren Dahlgaard. On the hardness of partially dynamic graph problems and connections to diameter. In *Proceedings 43rd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 48:1–48:14, 2016. doi:10.4230/LIPIcs.ICALP.2016.48.

28 C. Demetrescu and G. F. Italiano. A new approach to dynamic all pairs shortest paths. *Journal of the ACM*, 51(6):968–992, 2004.

29 Krzysztof Diks and Piotr Sankowski. Dynamic plane transitive closure. In *Proceedings of 15th Annual European Symposium on Algorithms (ESA)*, pages 594–604, 2007. doi:10.1007/978-3-540-75520-3_53.

30 D. Dor, S. Halperin, and U. Zwick. All-pairs almost shortest paths. *SIAM Journal on Computing*, 29(5):1740–1759, 2000.

31 Fedor V. Fomin, Daniel Lokshtanov, Michal Pilipczuk, Saket Saurabh, and Marcin Wrochna. Fully polynomial-time parameterized computations for graphs and matrices of low treewidth. In *Proceedings 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1419–1432, 2017. doi:10.1137/1.9781611974782.92.

32 Fedor V. Fomin, Fahad Panolan, M. S. Ramanujan, and Saket Saurabh. On integer programming and the path-width of the constraint matrix. *CoRR*, abs/1607.05342v2, 2016.

33 M. L. Fredman. How good is the information theory bound in sorting? *Theoretical Computer Science*, 1(4):355–361, 1976.

34 A. Freund. Improved subquadratic 3SUM. *Algorithmica*, pages 1–19, 2015. doi:10.1007/s00453-015-0079-6.

35 M. Fürer and B. Raghavachari. Approximating the minimum-degree Steiner tree to within one of optimal. *J. Algorithms*, 17(3):409–423, 1994. doi:10.1006/jagm.1994.1042.

36 H. Gabow, T. Nishizeki, O. Kariv, D. Leven, and O. Terada. Algorithms for edge-coloring graphs. Technical Report 41/85, Tohoku University, 1985.

37 Harold N. Gabow. A matroid approach to finding edge connectivity and packing arborescences. *Journal of Computer and System Sciences*, 50(2):259–273, 1995. doi:10.1006/jcss.1995.1022.

38 Harold N. Gabow. Using expander graphs to find vertex connectivity. *J. ACM*, 53(5):800–844, 2006. doi:10.1145/1183907.1183912.

39 Anka Gajentaan and Mark H. Overmars. On a class of $o(n^2)$ problems in computational geometry. *Comput. Geom.*, 45(4):140–152, 2012.

40 David Gale and Lloyd S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

41 O. Gold and M. Sharir. Improved bounds for 3SUM, K-SUM, and linear degeneracy. *CoRR*, abs/1512.05279, 2015.

42 Isaac Goldstein, Tsvi Kopelowitz, Moshe Lewenstein, and Ely Porat. How hard is it to find (honest) witnesses? In *Proceedings 24th Annual European Symposium on Algorithms (ESA)*, pages 45:1–45:16, 2016.

43 R. E. Gomory and T. C. Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9, 1961.

44 A. Grønlund and S. Pettie. Threesomes, degenerates, and love triangles. In *Proceedings 55th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 621–630, 2014.

45 Joel Hass, J. C. Lagarias, and Nicholas Pippenger. The computational complexity of knot and link problems. *J. ACM*, 46(2):185–211, 1999. `doi:10.1145/301970.301971`.

46 M. Henzinger, S. Krinninger, D. Nanongkai, and T. Saranurak. Unifying and strengthening hardness for dynamic problems via the online matrix-vector multiplication conjecture. In *Proceedings 47th Annual ACM Symposium on Theory of Computing (STOC)*, pages 21–30, 2015.

47 Monika Henzinger, Satish Rao, and Di Wang. Local flow partitioning for faster edge connectivity. In *Proceedings 28th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1919–1938, 2017.

48 Ian Holyer. The NP-completeness of edge-coloring. *SIAM J. Comput.*, 10(4):718–720, 1981. `doi:10.1137/0210055`.

49 A. Itai and M. Rodeh. Finding a minimum circuit in a graph. *SIAM J. Comput.*, 7(4):413–423, 1978.

50 W. Jaco and J. L. Tollefson. Algorithms for the complete decomposition of a closed 3-manifold. *Ill. J. Math.*, 39:358–406, 1995.

51 M. Jurdziński and R. Lazić. Succinct progress measures for solving parity games. *CoRR*, abs/1702.05051, 2017.

52 Marcin Jurdziński. Deciding the winner in parity games is in UP $\cap$ co-UP. *Information Processing Letters*, 68(3):119–124, 1998. `doi:10.1016/S0020-0190(98)00150-1`.

53 Marcin Jurdziński, Mike Paterson, and Uri Zwick. A Deterministic Subexponential Algorithm for Solving Parity Games. *SIAM Journal on Computing*, 38(4):1519–1532, 2008. `doi:10.1137/070686652`.

54 Ken-ichi Kawarabayashi and Mikkel Thorup. Deterministic global minimum cut of a simple graph in near-linear time. In *Proceedings 47th ACM Symposium on Theory of Computing (STOC)*, pages 665–674, 2015. `doi:10.1145/2746539.2746588`.

55 T. Kopelowitz, S. Pettie, and E. Porat. Higher lower bounds from the 3SUM conjecture. In *Proceedings 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1272–1287, 2016. `doi:10.1137/1.9781611974331.ch89`.

56 Marvin Künnemann, Daniel Moeller, Ramamohan Paturi, and Stefan Schneider. Subquadratic algorithms for succinct stable matching. *CoRR*, abs/1510.06452, 2015.

57 Yin Tat Lee and Aaron Sidford. Efficient inverse maintenance and faster algorithms for linear programming. In *Proceedings 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 230–249, 2015. `doi:10.1109/FOCS.2015.23`.

58 Jiří Matoušek. Efficient partition trees. *Discrete & Computational Geometry*, 8(1):315–334, 1992.

59 Jakub Pachocki, Liam Roditty, Aaron Sidford, Roei Tov, and Virginia Vassilevska Williams. Approximating cycles in directed graphs: Fast algorithms for girth and roundtrip spanners. *CoRR*, abs/1611.00721, 2016. URL: http://arxiv.org/abs/1611.00721.

60 Debmalya Panigrahi. Gomory-Hu trees. In Ming-Yang Kao, editor, *Encyclopedia of Algorithms*, pages 858–861. Springer New York, 2016. doi:10.1007/978-1-4939-2864-4_168.

61 Christos H. Papadimitriou. On the complexity of integer programming. *J. ACM*, 28(4):765–768, 1981. doi:10.1145/322276.322287.

62 L. Roditty and V. Vassilevska Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In *Proceedings 45th ACM Symposium on Theory of Computing (STOC)*, pages 515–524, 2013.

63 Liam Roditty and Virginia Vassilevska Williams. Subquadratic time approximation algorithms for the girth. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 833–845, 2012.

64 Barna Saha. Language edit distance and maximum likelihood parsing of stochastic grammars: Faster algorithms and connection to fundamental graph problems. In *Proceedings 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 118–135, 2015. doi:10.1109/FOCS.2015.17.

65 M. Thorup. Worst-case update times for fully-dynamic all-pairs shortest paths. In *Proceedings 37th ACM Symposium on Theory of Computing (STOC)*, pages 112–119, 2005.

66 V. Vassilevska Williams and R. Williams. Subcubic equivalences between path, matrix and triangle problems. In *Proceedings 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 645–654, 2010.

67 R. Yuster and U. Zwick. Finding even cycles even faster. *SIAM J. Discrete Mathematics*, 10(2):209–222, 1997.

Participants

- Amir Abboud
Stanford University, US
- Alexandr Andoni
Columbia University – New York, US
- Arturs Backurs
MIT – Cambridge, US
- Andreas Björklund
Lund University, SE
- Karl Bringmann
MPI für Informatik – Saarbrücken, DE
- Maike Buchin
Ruhr-Universität Bochum, DE
- Yi-Jun Chang
University of Michigan – Ann Arbor, US
- Raphaël Clifford
University of Bristol, GB
- Radu Curticapean
Hungarian Academy of Sciences – Budapest, HU
- Marek Cygan
University of Warsaw, PL
- Søren Dahlgaard
University of Copenhagen, DK
- Holger Dell
Universität des Saarlandes, DE
- Fedor V. Fomin
University of Bergen, NO
- Omer Gold
Tel Aviv University, IL
- Isaac Goldstein
Bar-Ilan University – Ramat Gan, IL
- Thore Husfeldt
IT University of Copenhagen, DK
- Mathias Bæk Tejs Knudsen
University of Copenhagen, DK
- Young Kun Ko
Princeton University, US
- Tsvi Kopelowitz
University of Michigan – Ann Arbor, US
- Robert Krauthgamer
Weizmann Institute – Rehovot, IL
- Sebastian Krinninger
MPI für Informatik – Saarbrücken, DE
- Marvin Künnemann
University of California – San Diego, US
- Moshe Lewenstein
Bar-Ilan University – Ramat Gan, IL
- Andrea Lincoln
Stanford University, US
- Veronika Loitzenbauer
Universität Wien, AT
- Aleksander Madry
MIT – Cambridge, US
- Kurt Mehlhorn
MPI für Informatik – Saarbrücken, DE
- Ivan Mikhailin
University of California – San Diego, US
- Danupon Nanongkai
KTH Royal Institute of Technology – Stockholm, SE
- Ramamohan Paturi
University of California – San Diego, US
- Seth Pettie
University of Michigan – Ann Arbor, US
- Ely Porat
Bar-Ilan University – Ramat Gan, IL
- Vijaya Ramachandran
University of Texas – Austin, US
- Liam Roditty
Bar-Ilan University – Ramat Gan, IL
- Aviad Rubinstein
University of California – Berkeley, US
- Thatchaphol Saranurak
KTH Royal Institute of Technology – Stockholm, SE
- Stefan Schneider
University of California – San Diego, US
- Arkadiusz Socala
University of Warsaw, PL
- Virginia Vassilevska Williams
Stanford University, US
- Joshua R. Wang
Stanford University, US
- Oren Weimann
University of Haifa, IL
- Philip Wellnitz
MPI für Informatik – Saarbrücken, DE
- Huacheng Yu
Stanford University, US