Report from Dagstuhl Seminar 13331

Exponential Algorithms: Algorithms and Complexity Beyond Polynomial Time

Edited by

Thore Husfeldt[1], Ramamohan Paturi[2], Gregory B. Sorkin[3], and Ryan Williams[4]

1 **IT University of Copenhagen, DK,** thore.husfeldt@gmail.com
2 **UC San Diego, US,** paturi@cs.ucsd.edu
3 **London School of Economics, GB,** G.B.Sorkin@lse.ac.uk
4 **Stanford University, US,** rrwilliams@gmail.com

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 13331 "Exponential Algorithms: Algorithms and Complexity Beyond Polynomial Time". Problems are often solved in practice by algorithms with worst-case exponential time complexity. It is of interest to find the fastest algorithms for a given problem, be it polynomial, exponential, or something in between. The focus of the seminar is on finer-grained notions of complexity than NP-completeness and on understanding the exact complexities of problems. The report provides a rationale for the workshop and chronicles the presentations at the workshop. The report notes the progress on the open problems posed at the past workshops on the same topic. It also reports a collection of results that cite the presentations at the previous seminar. The docoument presents the collection of the abstracts of the results presented at the seminar. It also presents a compendium of open problems.

Seminar 11.–16. August, 2013 – www.dagstuhl.de/13331
1998 ACM Subject Classification F.1.3 Complexity Measures and Classes, F.2 Analysis of Algorithms and Problem Complexity
Keywords and phrases Algorithms, exponential time algorithms, exact algorithms, computational complexity, satisfiability
Digital Object Identifier 10.4230/DagRep.3.8.40
Edited in cooperation with Dominik Scheder

1 Executive Summary

Thore Husfeldt
Ramamohan Paturi
Gregory Sorkin
Ryan Williams

Background

Computational complexity has demonstrated that thousands of important computational problems, spanning the sciences, are intimately linked: either they all have polynomial time algorithms, or none does. Nearly all researchers believe that P $\neq$ NP, and that these problems do not all have low time complexity. However, they must be solved, one way or another, which means relaxing the requirements for "solving" a problem. One natural requirement to relax is the running time. Problems are often solved in practice by algorithms with worst-case exponential time complexity. It is of interest to find the *fastest* algorithm for a given problem, be it polynomial, exponential, or something in between.

This relaxation has revealed a finer-grained notion of problem complexity than NP-completeness. By definition all NP-complete problems are equivalent as far as the existence of polynomial time algorithms is concerned. However, the exact time complexities of these problems can be very different, just as their best approximation ratios can vary.

Algorithms for satisfiability represent well the progress in the field and the questions that arise. The theory of NP-completeness says that the Circuit Sat problem and 3-Sat are polynomial time equivalent. However, from the exact, exponential time perspective, the two problems look radically different.

For 3-Sat (and k-Sat in general), algorithms faster than the exhaustive search of 2^n assignments have been known for many years and are continually improved. The analysis of the randomized PPSZ algorithm for 3-Sat has recently been improved to $O(1.308^n)$, so currently the best known algorithm for this problem is also very simple. The best known deterministic algorithm runs in time $O(1.331^n)$, and is obtained by derandomizing earlier local search algorithms. A very natural DPLL-type algorithm for Formula Sat in fact has good performance on linear size formulas. All of these results represent major conceptual contributions.

No such progress has been made for general Circuit Sat. In fact, such results would have major implications in circuit complexity: even a $1.99^n \mathit{poly}(m)$ time algorithm for satisfiability of circuits with n inputs and m gates would imply exponential size *lower bounds* for solving problems with circuits. Between 3-Sat and Circuit Sat, there are also intermediate problems such as CNF-Sat that have resisted all efforts to produce an $O(1.99^n)$ time algorithm.

The basic algorithmic techniques to avoid exhaustive search are now consolidated in the field's first textbook, (Fomin and Kratsch, *Exact Exponential Algorithms*, Springer 2010) though they are still being extended and refined. For example, there is now a general framework for making various exponential time dynamic programming algorithms, such as standard algorithms for Knapsack and Subset Sum, run in polynomial space. The fast zeta transform, which plays a central role in the implementation of inclusion-exclusion algorithms, continues to be actively researched. And "measure-and-conquer" methods for analyzing branching/backtracking algorithms continue to be enhanced.

However, many other powerful techniques have been explored only recently. One idea is to find combinatorial structures (such as matchings) by looking for corresponding algebraic objects (such as polynomials). The idea dates to Edmonds if not Tutte, but was introduced by Koutis for exponential time path and packing problems, leading to an $2^k \mathit{poly}(n)$ algorithm to find a k-path in a graph and a breakthrough $O(1.67^n)$ time algorithm for finding a Hamiltonian path, improving the 50-year-old previously best algorithm.

Other open problems in the field have been attacked by intricate, dedicated analyses; for example, there is now an algorithm for scheduling partially ordered jobs in $O((2-\epsilon)^n)$ time.

Parameterized complexity is a closely related field that also investigates exponential time computation. Fundamentally, the field is interested in the dichotomy between algorithms that admit running times of the form $f(k)\mathit{poly}(n)$ (called fixed-parameter tractability) and those that do not, leading to qualitative hardness notions like $W[1]$-hardness. This field continues to make great progress, with the parameterized tractability of many fundamental problems just being discovered. Just recently the first fixed-parameter algorithms for finding topological minors and the multi-cut problem have been found.

However, many recent results in that area are interested in determining (typically exponential) growth rate of the function f, instead of just establishing its existence. For example, a recent, very successful focus of parameterized complexity is the existence of problem *kernels* of polynomial size, or their absence under assumptions from classical computational complexity. In another direction, very strong lower bounds for algorithms parameterized by treewidth can now be shown under hypotheses about the exponential time complexity of Sat.

A quantitative theory of computational complexity of hard problems would address questions like why it is that 3-Sat can be solved in 1.4^n but CNF-Sat apparently cannot be solve. Ideally, we could hope for a characterization of the exact complexity of NP-complete problems, perhaps under some plausible assumptions. There is a growing body of work on the exact complexity of NP-complete problems which draws heavily from parameterized complexity theory. The *Exponential Time Hypothesis* (ETH), which posits that 3-Sat cannot be solved in $2^{o(n)}$ time, has given a strong explanatory framework for why some classes of problems admit improved algorithms while others are resistant. The results surrounding ETH show that if 3-Sat could be solved in subexponential time, then many other NP problems would also have subexponential algorithms.

Another compelling conjecture is the *Strong Exponential Time Hypothesis* (SETH) that CNF Satisfiability cannot be solved in 1.999^n time on formulas with n variables and cn clauses (for sufficiently large c). SETH has implications for k-Sat, other graph problems, and parameterized computation. There is less consensus about the truth of SETH; nevertheless, studying its implications will help better understand what makes CNF so difficult. A counting version of the hypothesis, #ETH, has recently been introduced to study the exponential time complexity of counting problems, such as the permanent and the Tutte polynomial.

Connections to other fields are being discovered as well, such as the importance of exponential time algorithms to the study of lower bounds in circuit complexity, as mentioned above.

For another example, a celebrated recent result in the complexity of approximation algorithms exhibits an $\exp(O(n^{\varepsilon}))$ time approximation algorithm for Khot's Unique Games problem. This suggests that approximating unique games is a significantly easier task than solving NP-hard problems such as 3-Sat. The key to the algorithm is a new type of graph decomposition based on spectral methods. This decomposition method may well lead to more developments in exponential algorithms.

Furthermore, there are surprising connections between SETH and various other well-studied questions from other areas such as communication complexity and the 3-Sum hypothesis used in computational geometry and data structures. The instance compressibility notion introduced in the study of kernelisation turns out to be connected to the construction of hash functions.

These results show that increased attention to exponential time algorithms leads to progress of the highest caliber in well-established areas of the theory of computation.

2 Table of Contents

3 About the Meeting

The meeting was attended by 42 researchers, slightly smaller number than the number anticipated due to a few last-minute cancellations. The organizers are grateful to all who came, and regret that – due to a high acceptance rate – others who would have contributed could not be invited. The participants came all around the globe, predominantly from Europe and with a good showing from Germany and US: AU 1, AT 1, CH 1, DE 6, DK 3, FI 2, FR 2, GB 3, HU 1, IN 1, JP 3, NL 2, NO 3, PL 2, RU 1, SE 1, US 9.

Attendees include eight graduate students or postdoctoral fellows and eleven faculty who are at the beginning of their academic career. The meeting has a fair representation of researchers from each of the three fields, exact exponential algorithms, parameterized complexity, and computational complexity.

Almost all the talks were 45 minutes long. The after-lunch period was left free for informal discussions and small working groups, with talks again between 4 PM and dinner time. There were three open problem sessions, each taking up 30 minutes. These sessions brought out 19 open problems which were described in a latter section. Wednesday included traditional outing with the options of a long hike and a long run.

R. Williams started the session with an overview of the Exponential Time Hypothesis (ETH) and the Strong Exponential Time Hypothesis (SETH) discussing the motivation for the hypotheses and their many consequences. Bodlaender showed how algebraic methods can be used to obtain faster algorithms for several problems on tree decompositions with running time that is single exponential in the treewidth and linear in the number of vertices. Kulikov presented an improved exponential time algorithm for a special case of the classical problem, shortest superstring. Kowalik talked about fast self-reducibility with the number of oracle calls linear in the size of the solution. Wahlström presented a surprising result that gives a polynomial compression for the Steiner cycle problem.

On Tuesday, several results were presented concerning improved algorithms for restricted circuit satisfiability problems.Tamaki talked about improved algorithm for formulas over full binary basis. Seto presented an improved algorithm for sparse instances of MaxSat. Schneider presented an algorithm for depth-2 threshold circuit satisfiability. Santhanam presented new algorithms for QBF satisfiability. Hertli presented a intriguing result that raises the hope for beating the current best algorithm for 3-SAT. In contrast, Nederlof presented a series of problems whose running time (in terms of the constant in the exponent) cannot be improved if SETH holds. The set of results presented on Tuesday enlarge our understanding of the possibility of improved algorithms for satisfiability as a function of the expressive power of the circuit class.

On Wednesday, Lokshtanov presented a nice account of the work on efficient computation of representative sets and its applications to exact algorithms and kernelisation. Lovett talked about the long-standing log rank conjecture in communication complexity and the recent progress he made. Drucker presented a new line of work which shows that the question of efficient algorithms for 3-SAT in probabilistic polynomial time can be tied to the relationship between NP and co-NP. Steurer talked about approximability in the region between polynomial and exponential time and how the results in this area an be unified using the sum-of-squares method.

On Thursday, several new algorithmic results were presented: Marx presented a subexponential time algorithm for the subset TSP problem for planar graphs. Saurabh presented an improved algorithm for the multicut problem. D. Kratsch presented an algorithm for the jumpnumber problem beating the 2^n upperbound. S. Kratsch showed how one can use matrix methods to obtain deterministic algorithms for problems such as Hamiltonian Cycle and Steiner Tree. Also Heggernes and Kaski presented results on enumeration of graph classes and counting thin subgraphs respectively. A highlight on Thursday was the result presented by V. Williams who showed that approximating the graph diameter beyond the 3/2 ratio is not possible unless SETH fails.

On Friday, Eberfeld talked about the development of a theory of parameterized space complexity. In addition, three algorithmic results were presented: Gaspers presented techniques to improve Measure and Conquer analyses and a faster algorithm for Max-2-CSP. Cygan presented a faster algorithm for the TSP problem when the average degree is bounded. Finally, Austrin presented improved time-space tradeoffs for the subset sum problem.

Several open problems from previous Dagstuhl Seminars (seminars 08431 and 10441) have been resolved. Ryan Williams asked about improved satisfiability algorithms for depth-2 threshold circuits in the seminar 10441. The question has been partially answered by Impagliazzo, Paturi and Schneider and presented in the seminar on Tuesday morning. Björklund, Kaski and Kowalik resolved an open problem posed by Koutis in the seminar 10441 regarding k-vector coloring. Binkele-Raible et al. resolved an open question posed by van Rooij in the seminar 08431 by providing an improved algorithm for computing irredundancy numbers. Heggernes, Kratsch, Lokshtanov, Raman and Saurabh showed that partitioning a permutation into monotone sequences is fixed parameter tractable when the parameter is the number of such sequences resolving an open problem stated by Kratsch in the seminar 08431.

We also found several research results which credit the seminar 10441. Junosza-Szaniawski, Kratocvhíl, Liedloff, Rossmanith, and Rzażewski cite the presentation by Rossmanith at seminar 10441 in their work on fast algorithms for $L(2,1)$-labeling of graphs. Björklund, Kaski, and Kowalik in their work on optimal graph motifs cite the presentation given by Koutis at the seminar 10441. Van Rooij generously credits the impact of the seminars 08431 and 10441 on his work in his PhD thesis. Dantsin and Hirsch's paper published in SAT 2011 notes that the work was done in part during the seminar 10441.

4 Organization

The seminar was organized by

Thore Husfeldt, IT University of Copenhagen, DK, thore@itu.dk
Ramamohan Paturi, University of California, San Diego, US, paturi@cs.ucsd.edu
Gregory Sorkin, London School of Economics, London, GB, G.B.Sorkin@lse.ac.uk
Ryan Williams, Stanford University, US, rrwilliams@gmail.com

Abstracts and open problems are being compiled and edited by
Dominik Scheder, Aarhus University, DK, and Simons Institute for the Theory of Computing, UC Berkeley, US, dominik.scheder@gmail.com,
who is also assisting in preparation of this final report.

We are grateful to the Dagstuhl personnel for their helpfulness and expertise, making the meeting smooth-running, pleasurable, productive, and easy to organize.

5 Overview of Talks

5.1 Space-Time Tradeoffs for Subset Sum

Per Austrin (KTH Stockholm, SE)

Joint work of Austrin, Per; Kaski, Petteri; Koivisto, Mikko; Määttä, Jussi
Main reference P. Austrin, P. Kaski, M. Koivisto, J. Määttä, "Space-Time Tradeoffs for Subset Sum: An Improved Worst Case Algorithm," arXiv:1303.0609v1 [cs.DS], 2013.
URL http://arxiv.org/abs/1303.0609v1

I discuss recent work on improved Space-Time Tradeoffs for the Subset Sum Problem.

5.2 Faster Algorithms on Tree Decompositions with Representative Sets: Expressibility and Experimentation

Hans L. Bodlaender (Utrecht University, NL)

Joint work of Bodlaender, Hans L.; Fafiani, Stefan; Kratsch, Stefan; Nederlof, Jesper

Recent work [1] (see also [2]) shows that algebraic methods can be used to obtain faster algorithms for several problems on tree decompositions, in particular for problems with connectivity requirements. The main underlying technique is to bring back a set of characteristics of partial solutions to a representative set. In this talk, we discuss a recent implementation of the method for the Steiner Tree problem [3], and a general characterization of a large set of problems that all allow such algorithms, running in time that is single exponential in the treewidth and linear in n.

References

1 Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, Jesper Nederlof: Deterministic Single Exponential Time Algorithms for Connectivity Problems Parameterized by Treewidth. ICALP (1) 2013: 196-207.
2 Marek Cygan, Stefan Kratsch, Jesper Nederlof: Fast Hamiltonicity checking via bases of perfect matchings. STOC 2013: 301-310.
3 Fafianie, Stefan; Bodlaender, Hans L.; Nederlof, Jesper. Speeding-up Dynamic Programming with Representative Sets – An Experimental Evaluation of Algorithms for Steiner Tree on Tree Decompositions. arXiv:1305.7448, 2013. To appear in proceedings IPEC 2013.

5.3 TSP and counting perfect matchings in bounded average degree graphs

Marek Cygan (University of Warsaw, PL)

Joint work of Cygan, Marek; Pilipczuk, Marcin
Main reference M. Cygan, M. Pilipczuk, "Faster Exponential-Time Algorithms in Graphs of Bounded Average Degree," in Proc. of the 40th Int'l Colloquium on Automata, Languages, and Programming (ICALP'13), LNCS, Vol. 7965, pp. 364–375, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-39206-1_31

We first show that the Traveling Salesman Problem in an n-vertex graph with average degree bounded by d can be solved in $O^*(2^{(1-\epsilon_d)n})$ time and exponential space for a constant ϵ_d depending only on d. Thus, we generalize the recent results of Björklund et al. [TALG 2012] on graphs of bounded degree. Then, we move to the problem of counting perfect matchings in a graph. We first present a simple algorithm for counting perfect matchings in an n-vertex graph in $O^*(2^{n/2})$ time and polynomial space; our algorithm matches the complexity bounds of the algorithm of Björklund [SODA 2012], but relies on inclusion-exclusion principle instead of algebraic transformations. Building upon this result, we show that the number of perfect matchings in an n-vertex graph with average degree bounded by d can be computed in $O^*(2^{(1-\epsilon_{2d})n/2})$ time and exponential space, where ϵ_{2d} is the constant obtained by us for the Traveling Salesman Problem in graphs of average degree at most $2d$. Moreover we obtain a simple algorithm that computes a permanent of an $n \times n$ matrix over an arbitrary commutative ring with at most dn non-zero entries using $O^*(2^{(1-1/(3.55d))n})$ time and ring operations, improving and simplifying the recent result of Izumi and Wadayama [FOCS 2012].

5.4 Nondeterministic Direct Product Reductions and the Success Probability of SAT Solvers

Andrew Drucker (MIT, US)

Main reference A. Drucker, "Nondeterministic Direct Product Reductions and the Success Probability of SAT Solvers," to appear in Proc. of IEEE FOCS 2013.

In this talk I will describe nondeterministic reductions which yield new direct product theorems (DPTs) for Boolean circuits. In our theorems one assumes that a function F is "mildly hard" against *nondeterministic* circuits of some size $s(n)$, and concludes that the t-fold direct product F^t is "extremely hard" against probabilistic circuits of only polynomially smaller size $s'(n)$. The main advantage of these results compared with previous DPTs is the strength of the size bound in our conclusion. As an application, we show that if NP is not in coNP/poly then, for every PPT algorithm attempting to produce satisfying assignments to Boolean formulas, there are infinitely many satisfiable instances on which the algorithm's success probability is nearly-exponentially small. This furthers a project of Paturi and Pudlák [STOC'10].

5.5 Space in Parameterized Complexity

Michael Elberfeld (ICSI – Berkeley, US)

Joint work of Elberfeld, Michael; Stockhusen, Christoph; Tantau, Till
Main reference M. Elberfeld, C. Stockhusen, T. Tantau, "On the Space Complexity of Parameterized Problems," in Proc. of the 7th Int'l Symp. on Parameterized and Exact Computation (IPEC'12), LNCS, Vol. 7535, pp. 206–217, Springer, 2012.
URL http://dx.doi.org/10.1007/978-3-642-33293-7_20

Besides running time, the (memory) space available severely limits the range of computational problems that can be solved efficiently. Classical computational complexity provides several ways to classify the space complexity of problems using space bounds or additional simultaneous time bounds. The talk reports on an ongoing effort to develop a theory of parameterized space complexity and algorithms that is based on (1) standard parameterizations of classical space complexity classes, (2) space-efficient notions of bounded fixed-parameter tractability, and (3) parameter-based simultaneous resource bounds. Here I will focus on the first aspect and talk about parameterizations of deterministic logarithmic space; and how they can be used to give new insights into the complexity of well-studied parameterized problems. The talk details results from a joint work with Christoph Stockhusen and Till Tantau published in the proceedings of IPEC 2012 as well as surveys results from the literature to give an overview of the theory developed so far.

5.6 Separate, Measure, and Conquer

Serge Gaspers (UNSW – Sydney, AU)

Joint work of Gaspers, Serge; Sorkin, Gregory B.

Graph separators have been used extensively to design subexponential time algorithms for graph classes and to guide heuristics in practical solvers. We present a method to take advantage of linear-sized separators to improve Measure and Conquer analyses. As case analyses, we give faster polynomial-space algorithms for Max 2-CSP and # Dominating Set.

5.7 Enumeration in Graph Classes

Pinar Heggernes (University of Bergen, NO)

Enumerating, counting, and determining the maximum number of various objects in graphs have long been established as important areas within graph theory and graph algorithms. As the number of enumerated objects is very often exponential in the size of the input graph, enumeration algorithms fall into two categories depending on their running time: those whose running time is measured in the size of the input, and those whose running time is measured in the size of the output. Based on this, we concentrate on the following two types of algorithms. 1. Exact exponential time algorithms. The design of these algorithms is mainly based on recursive branching. The running time is a function of the size of the input graph, and very often it also gives an upper bound on the number of enumerated objects any graph can have. 2. Output polynomial algorithms. The running time of these algorithms is polynomial in the number of the enumerated objects that the input graph actually contains. Some of these algorithms have even better running times in form of incremental polynomial or polynomial delay, depending on the time the algorithm spends between each consecutive object that is output. The methods for designing the two types of algorithms are usually quite different. Common to both approaches is that efforts have traditionally mainly been concentrated on arbitray graphs, whereas graphs with particular structure have largely been left unattended. In this talk we look at enumeration of objects in graphs with special structure. In particular, we focus on enumerating minimal dominating sets in various graph classes.

Algorithms of type 1: The number of minimal dominating sets that any graph on n vertices can have is known to be at most 1.7159^n. This upper bound might not be tight, since no examples of graphs with 1.5705^n or more minimal dominating sets are known. For several classes of graphs, like chordal, split, and proper interval graphs, we substantially improve the upper bound on the number of minimal dominating sets. At the same time, we give algorithms for enumerating all minimal dominating sets, where the running time of each algorithm is within a polynomial factor of the proved upper bound for the graph class in question. In some cases, we provide examples of graphs containing the maximum possible number of minimal dominating sets for graphs in that class, thereby showing the corresponding upper bounds to be tight.

Algorithms of type 2: Enumeration of minimal dominating sets in graphs has very recently been shown to be equivalent to enumeration of minimal transversals in hypergraphs. The question whether the minimal transversals of a hypergraph can be enumerated in output polynomial time is a fundamental and challenging question; it has been open for several decades and has triggered extensive research. We show that all minimal dominating sets of a line graph can be generated in incremental polynomial, and consequently output polynomial, time. We are able to improve the delay further on line graphs of bipartite graphs. Finally we show that our method is also efficient on graphs of large girth, resulting in an incremental polynomial time algorithm to enumerate the minimal dominating sets of graphs of girth at least 7. The presentation is based on joint works with Jean-François Couturier, Pim van 't Hof, and Dieter Kratsch [1], and with Petr Golovach, Dieter Kratsch, and Yngve Villanger [2].

References

1 J-F. Couturier, P. Heggernes, P. van 't Hof, and D. Kratsch. Minimal dominating sets in graph classes: Combinatorial bounds and enumeration. Theoretical Computer Science 487: 82–94 (2013).

2 P. Golovach, P. Heggernes, D. Kratsch, and Y. Villanger. An incremental polynomial time algorithm to enumerate all minimal edge dominating sets. ICALP 2013, Lecture Notes in Computer Science 7965: 485–496 (2013).

5.8 A Faster Algorithm For Unique 3-SAT

Timon Hertli (ETH Zürich, CH)

The PPSZ algorithm by Paturi, Pudlák, Saks, and Zane (FOCS 1998) is the fastest known algorithm for Unique k-SAT. We give an improved algorithm with exponentially faster bounds for Unique 3-SAT. We show that worst-case formulas for PPSZ are either sparse or allow us to extract nontrivial information. In the former case, we use an algorithm by Wahlström, in the latter case, we improve the very beginning of PPSZ.

5.9 Counting thin subgraphs via packings faster than meet-in-the-middle time

Petteri Kaski (Aalto University, FI)

Joint work of Björklund, Andreas; Kaski, Petteri; Kowalik, Łukasz
Main reference A. Björklund, P. Kaski, Ł. Kowalik, "Counting thin subgraphs via packings faster than meet-in-the-middle time," arXiv:1306.4111v1 [cs.DS], 2013; to appear in Proc. of the ACM-SIAM Symp. on Discrete Algorithms (SODA'14, 5–7 January 2014, Portland, OR).
URL http://arxiv.org/abs/1306.4111

Vassilevska and Williams (STOC 2009) showed how to count simple paths on k vertices and matchings on $k/2$ edges in an n-vertex graph in time $n^{k/2+O(1)}$. In the same year, two different algorithms with the same runtime were given by Koutis and Williams (ICALP 2009), and Björklund *et al.* (ESA 2009), via $n^{st/2+O(1)}$-time algorithms for counting t-tuples of pairwise disjoint sets drawn from a given family of s-sized subsets of an n-element universe. Shortly afterwards, Alon and Gutner (TALG 2010) showed that these problems have $\Omega(n^{\lfloor st/2 \rfloor})$ and $\Omega(n^{\lfloor k/2 \rfloor})$ lower bounds when counting by color coding. Here we show that one can do better, namely, we show that the "meet-in-the- middle" exponent $st/2$ can be beaten and give an algorithm that counts in time $n^{0.4547st+O(1)}$ for t a multiple of three. This implies algorithms for counting occurrences of a fixed subgraph on k vertices and pathwidth $p \ll k$ in an n-vertex graph in $n^{0.4547k+2p+O(1)}$ time, improving on the three mentioned algorithms for paths and matchings, and circumventing the color-coding lower bound.

5.10 Fast Extraction and Listing of Witnesses Using a Decision Oracle

Łukasz Kowalik (University of Warsaw, PL)

Joint work of Björklund, Andreas; Kaski, Petteri; Kowalik, Łukasz

The gist of many (NP-)hard combinatorial problems is to decide whether a universe of n elements contains a witness consisting of k elements that match some prescribed pattern. Example: find a k-path in an n-vertex graph. The state-of-art results provide a probabilistic, one-sided error oracles for testing whether such a k-path exists and then the pattern itself can be found by self-reducibility. We show that the self-reducibility can be done very efficiently, namely with $O(k)$ running time overhead (or in $O(k \log n)$ queries in the more abstract oracle setting). We also investigate the task of listing all witnesses: this can be done with $O(k(\log n + \log s))$ delay between two successive witnesses, where s is the total number of witnesses (unknown to the algorithm).

5.11 The jumpnumber problem: exact and parameterized

Dieter Kratsch (University of Metz, FR)

The Jump Number problem asks to find a linear extension of a given partially ordered set that minimizes the total number of jumps, i.e., the total number of consecutive pairs of elements that are incomparable originally. The problem is known to be NP-complete even on posets of height one and on interval orders. It has also been shown to be fixed-parameter tractable. Finally, the Jump Number problem can be solved in time $O^*(2^n)$ by dynamic programming. In this talk we present an exact algorithm to solve Jump Number in $O(1.8638^n)$ time. We also show that the Jump Number problem on interval orders can be solved by an $O(1.7593^n)$ time algorithm, and prove fixed-parameter tractability in terms of width w by an $O^*(2^w)$ time algorithm. Furthermore, we give an almost-linear kernel for Jump Number on interval orders when parameterized by solution size.

5.12 Applications of matrix rank for faster dynamic programming on tree and path decompositions

Stefan Kratsch (TU Berlin, DE)

Joint work of Bodlaender, Hans; Cygan, Marek; Kratsch, Stefan; Nederlof, Jesper
Main reference H. Bodlaender, M. Cygan, S. Kratsch, J. Nederlof, "Solving weighted and counting variants of connectivity problems parameterized by treewidth deterministically in single exponential time," arXiv:1211.1505v1 [cs.DS], 2013.
URL http://arxiv.org/abs/1211.1505v1
Main reference H. Bodlaender, M. Cygan, S. Kratsch, J. Nederlof, "Deterministic Single Exponential Time Algorithms for Connectivity Problems Parameterized by Treewidth," in Proc. of the 40th Int'l Colloquium on Automata, Languages, and Programming (ICALP'13), LNCS, Vol. 7965, pp. 196–207, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-39206-1_17

In the talk we discuss properties of two matrices that relate to dynamic programming on tree and path decompositions. The partition matrix has rows and columns indexed by the partitions of t elements; all entries are 1 or 0 depending on whether the meet of the two indexing partitions gives the unit partition. The matching matrix is a submatrix of the partition matrix, obtained by restricting the latter to those partitions that are perfect matchings, i.e., partitions into sets of size two each. We give factorizations for both matrices (and matching lower bounds), proving that the ranks are exactly 2^{t-1} and $2^{t/2-1}$, respectively. This leads to several interesting algorithmic results, among them: 1) The first deterministic algorithms for solving connectivity problems like Hamiltonian Cycle and Steiner Tree in time $O^*(c^{tw})$ when provided a tree decomposition of width tw. 2) A $O((2+\sqrt{(}2))^{pw})$ time randomized algorithm for Hamiltonian Cycle, given a path decomposition of width pw. The latter runtime is tight under Strong ETH.

5.13 Shortest superstring for short strings

Alexander S. Kulikov (Steklov Institute – St. Petersburg, RU)

Joint work of Golovnev, Alexander; Kulikov, Alexander S.; Mihajlin, Ivan
Main reference A. Golovnev, A. S. Kulikov, I. Mihajlin, "Solving 3-Superstring in $3^{n/3}$ Time," in Proc. of the 38th Int'l Symp. on Mathematical Foundations of Computer Science (MFCS'13), LNCS, Vol. 8087, pp. 480–491, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-40313-2_43

In the shortest common superstring problem (SCS) one is given a set $s_1, \ldots, s_n$ of n strings and the goal is to find a shortest string containing each s_i as a substring. While many approximation algorithms for this problem have been developed, it is still not known whether it can be solved exactly in fewer than 2^n steps. In this paper we present an algorithm that solves the special case when all of the input strings have length 3 in time $3^{n/3}$ and polynomial space. The algorithm generates a combination of a de Bruijn graph and an overlap graph, such that a SCS is then a shortest directed rural postman path (DRPP) on this graph. We show that there exists at least one optimal DRPP satisfying some natural properties. The algorithm works basically by exhaustive search, but on the reduced search space of such paths of size $3^{n/3}$.

5.14 Efficient Computation of Representative Sets and Applications

Daniel Lokshtanov (University of Bergen, NO)

We give computationally efficient variants of "Bolobás' lemma" and its generalization to half-spaces, and consider applications in exact and parameterized algorithms as well as in kernelization. The talk is a mash-up of arxiv:1304.4626, arXiv:1111.2195 as well as a few other results.

5.15 Communication is bounded by root of rank

Shachar Lovett (University of California – San Diego, US)

Main reference S. Lovett, "Communication is bounded by root of rank," arXiv:1306.1877v2 [cs.CC], 2013.
URL http://arxiv.org/abs/1306.1877v2

The log rank conjecture is one of the fundamental open problems in communication complexity. It speculates that if for a boolean $f(x, y)$, its associated matrix $M_{x,y} = f(x, y)$ has rank r (over the reals), then there exists a deterministic protocol computing f which uses only polylog(r) bits of communication. There is a trivial protocol which uses r bits of communication, and further progress on this problem reduced it to $O(r)$ bits [Kotlov-Lovász'96, Kotlov '97] and more recently to $O(r/\log(r))$ bits assuming a number theoretic conjecture [BenSasson-L-RonZewi'12]. In this work, we prove an (unconditional) upper bound of $O(r^{1/2}\log(r))$ bits.

5.16 A subexponential parameterized algorithm for Subset TSP on planar graphs

Dániel Marx (Hungarian Academy of Sciences, HU)

Joint work of Klein, Philip N.; Marx, Dániel

Given a graph G and a subset S of vertices, the Subset TSP problem asks for a shortest closed walk in G visiting all vertices of S. The problem can be solved in time $2^k \cdot n^{O(1)}$ using the classical dynamic programming algorithms of Bellman and of Held and Karp, where $k = |S|$ and $n = |V(G)|$. Our main result is showing that the problem can be solved in time $(2^{O(\sqrt{k}\log k)} + W) \cdot n^{O(1)}$ if G is a planar graph with weights that are integers no greater than W. While similar speedups have been observed for various paramterized problems on planar graphs, our result cannot be simply obtained as a consequence of bounding the treewidth of G or invoking bidimensionality theory. Our algorithm consists of two steps: (1) find a locally optimal solution, and (2) use it to guide a dynamic program. The proof of correctness of the algorithm depends on a treewidth bound on a graph obtained by combining an optimal solution with a locally optimal solution.

5.17 On Problems as Hard as CNF-SAT

Jesper Nederlof (Utrecht University, NL)

Joint work of Cygan, Marek ; Dell, Holger; Lokshtanov, Daniel; Marx, Dániel; Nederlof, Jesper; Okamoto, Yoshio; Paturi, Ramamohan; Saurabh, Saket; Wahlstrom, Magnus
Main reference M. Cygan, H. Dell, D. Lokshtanov, D. Marx, J. Nederlof, Y. Okamoto, R. Paturi, S. Saurabh, M. Wahlström, "On Problems as Hard as CNF-SAT," in Proc. of the 2012 IEEE 27th Annual Conf. on Computational Complexity (CCC'12), pp. 74–84, IEEE, 2012.
URL http://dx.doi.org/10.1109/CCC.2012.36

While exhaustive search remains asymptotically the fastest known algorithm for some basic problems, difficult and non-trivial exponential time algorithms have been found for a myriad of problems, including Graph Coloring, Hamiltonian Path, Dominating Set and 3-CNF-Sat. In some instances, improving these algorithms further seems to be out of reach. The CNF-Sat problem is the canonical example of a problem for which the trivial exhaustive search algorithm runs in time $O(2^n)$, where n is the number of variables in the input formula. While there exist non-trivial algorithms for CNF-Sat that run in time $o(2^n)$, no algorithm was able to improve the growth rate 2 to a smaller constant, and hence it is natural to conjecture that 2 is the optimal growth rate. The strong exponential time hypothesis (SETH) by Impagliazzo and Paturi [JCSS 2001] goes a little bit further and asserts that, for every $\epsilon < 1$, there is a (large) integer k such that k-CNF-SAT cannot be computed in time $2^{\epsilon n}$. In this talk, we show that, for every $\epsilon < 1$, the problems Hitting Set, Set Splitting, and NAE-Sat cannot be computed in time $O(2^{\epsilon n})$ unless SETH fails. Here n is the number of elements or variables in the input. For these problems, we actually get an equivalence to SETH in a certain sense. We conjecture that SETH implies a similar statement for Set Cover, and prove that, under this assumption, the fastest known algorithms for Steinter Tree, Connected Vertex Cover, Set Partitioning, and the pseudo-polynomial time algorithm for Subset Sum cannot be significantly improved. Finally, we justify our assumption about the hardness of Set Cover by showing that the parity of the number of set covers cannot be computed in time $O(2^{\epsilon n})$ for any $\epsilon < 1$ unless SETH fails.

5.18 New Algorithms for QBF Satisfiability, and Implications for Circuit Complexity

Rahul Santhanam (University of Edinburgh, GB)

Joint work of Santhanam, Rahul; Williams, Ryan

We study the complexity of the satisfiability problem for quantified Boolean formulas. We show that satisfiability of quantified CNFs of size poly(n) on n variables with at most q quantifier alternations can be solved in time $2^{n-n^{1/q}}$ by zero-error randomized algorithms. This is the first improvement over brute-force search even for $q = 1$. We also show how to achieve non-trivial savings when $q = \omega(\log(n))$. We then draw a connection between algorithms for QBF satisfiability and lower bounds against non-uniform NC^1. We show that NEXP not in NC^1/poly if (i) There are zero-error randomized algorithms solving satisfiability of quantified CNFs of size poly(n) on n variables with at most q quantifier alternations in time $2^{n-n^{\omega_q(1/q)}}$, or (ii) There are zero-error randomized algorithms solving satisfiability of quantified CNFs of size poly(n) on n variables in time $2^{n-\omega(\log(n))}$. Thus even minor improvements of our algorithmic results will yield new circuit lower bounds.

5.19 Better Than Trivial Algorithm for Multicut

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

Joint work of Saurabh, Saket; Lokshtanov, Daniel; Suchy, Ondrej

There have been several exact algorithms finding a vertex subset of a graph on n vertices satisfying certain properties like acyclicity, bipartiteness in time c^n, where $c < 2$ is a constant. However, there has been very few exact exponential algorithms for cut problems like Multiway Cut or Multicut. In Multicut, we are given an undirected graph $G = (V, E)$ and a family $T = \{(s_i, t_i) \mid s_i, t_i \in V\}$ and objective is to find a minimum size set S such that in $G - S$ there is no path between any pair of vertices in T. In this talk we present an exact algorithm for Multicut running in time 1.987^n. The algorithm is based on branch and bound and utilizes an interesting measure.

5.20 A Satisfiability Algorithm for Sparse Depth Two Threshold Circuits

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

Joint work of Schneider, Stefan; Impagliazzo, Russell; Paturi, Ramamohan

We give a nontrivial algorithm for the satisfiability problem for threshold circuits of depth two with a linear number of wires which improves over exhaustive search by an exponential factor. We also get an algorithm for 0-1 Integer Linear Programming with a linear number of constraints. The key idea is to reduce the satisfiability problem to the Vector Domination problem, the problem of checking whether there are two vectors in a given collection of vectors such that one dominates the other component-wise. This is joint work with Russell Impagliazzo and Ramamohan Paturi.

5.21 Solving Sparse Instances of Max SAT via Width Reduction and Greedy Restriction

Kazuhisa Seto (The University of Electro-Communications – Tokyo, JP)

Joint work of Seto, Kazuhisa; Tamaki, Suguru; Sakai Takayuki

We present a moderately exponential time polynomial space algorithm for sparse instances of Max SAT. For instances with n variables and cn clauses, our algorithm runs in time $O(2^{(1-\mu_c)n})$, where $\mu_c = \Omega(\frac{1}{c^2 \log^2 c})$. Previously, an exponential space algorithm with $\mu_c = \Omega(\frac{1}{c \log c})$ was shown by Dantsin and Wolpert [SAT 2006] and a polynomial space algorithm with $\mu_c = \Omega(\frac{1}{2^{O(c)}})$ was shown by Kulikov and Kutzkov [CSR 2007]. Our algorithm is based on the combination of two techniques, width reduction of Schuler and greedy restriction of Santhanam. The approach is flexible in the sense that we also obtain algorithms for Max Conj SAT, Max Not-All-Equal SAT and Max Exact-One SAT by slightly modifying the algorithm for Max SAT.

5.22 Time vs Approximation Trade-offs and Strong Relaxations

David Steurer (Cornell University, US)

I will survey recent results about approximability in the regime of intermediate complexity – strictly between polynomial and exponential time. A unifying theme are strong relaxations, especially the semidefinite programs obtained from sum-of-squares methods.

5.23 A Satisfiability Algorithm and Average-Case Hardness for Formulas over the Full Binary Basis

Suguru Tamaki (Kyoto University, JP)

Joint work of Seto, Kazuhisa; Suguru Tamaki
Main reference K. Seto, S. Tamaki, "A Satisfiability Algorithm and Average-Case Hardness for Formulas over the Full Binary Basis," Computational Complexity, 22(2):245–274, 2013.
URL http://dx.doi.org/10.1007/s00037-013-0067-7

We present a moderately exponential time algorithm for the satisfiability of Boolean formulas over the full binary basis. For formulas of size at most cn, our algorithm runs in time $2^{(1-\mu_c)n}$ for some constant $\mu_c > 0$. As a byproduct of the running time analysis of our algorithm, we obtain strong average-case hardness of affine extractors for linear- sized formulas over the full binary basis.

5.24 SETH and some natural graph problems in poly-time

Virginia Vassilevska Williams (Stanford University, US)

Joint work of Roditty, Liam; Vassilevska Williams, Virginia
Main reference L. Roditty, V. Vassilevska Williams, "Fast approximation algorithms for the diameter and radius of sparse graphs," in Proc. of the 45th Annual ACM Symp. on Theory of Computing (STOC'13), pp. 515–524, ACM, 2013.
URL http://dx.doi.org/10.1145/2488608.2488673

We discuss several reductions from CNF-SAT to some natural graph problems that relate the complexity of these problems to the Strong Exponential Time Hypothesis. For instance, we show that unless SAT is in $(2-\epsilon)^n$ time for some $\epsilon > 0$, any $(3/2-\delta)$-approximation algorithm for the graph diameter must take $\Omega(n^{2-\epsilon})$ time for all $\epsilon > 0$. There is an $\tilde{O}(m\sqrt{n})$-time 3/2-approximation for the problem, and our reduction shows that this algorithm is in a sense optimal, assuming Strong ETH. We also show that some dynamic graph problems admit lower bounds based on Strong ETH. For instance, maintaining the number of strongly connected components of a graph under edge insertions or deletions requires amortized update time $\Omega(m^{1-\epsilon})$ for all $\epsilon > 0$, thus showing that the trivial update algorithm may be optimal.

5.25 Abusing the Tutte matrix: A polynomial compression for the Steiner Cycle problem

Magnus Wahlstroem (Royal Holloway University of London, GB)

Main reference M. Wahlström, "Abusing the Tutte Matrix: An Algebraic Instance Compression for the K-set-cycle Problem," in Proc. of the 30th Int'l Symp. on Theoretical Aspects of Computer Science (STACS'13), LIPIcs, Vol. 20, pp. 341–352, Schlos Dagstuhl – Leibniz-Zentrum für Informatik, 2013.
URL http://dx.doi.org/10.4230/LIPIcs.STACS.2013.341

We give an algebraic, determinant-based algorithm for the Steiner Cycle problem, i.e., the problem of finding a cycle through a set K of specified elements. Our approach gives a simple FPT algorithm for the problem, matching the $O^*(2^{|K|})$ running time of the algorithm of Björklund et al. (SODA, 2012). Furthermore, our approach is open for treatment by classical algebraic tools (e.g., Gaussian elimination), and we show that it leads to a polynomial compression of the problem, i.e., a polynomial-time reduction of the Steiner Cycle problem into an algebraic problem with coding size $O(|K|^3)$. This is surprising, as several related problems (e.g., k-Cycle and the Disjoint Paths problem) are known not to admit such a reduction unless the polynomial hierarchy collapses. Furthermore, despite the result, we are not aware of any witness for the Steiner Cycle problem of size polynomial in $|K| + \log n$, which seems (for now) to separate the notions of polynomial compression and polynomial kernelization (as a polynomial kernelization for a problem in NP necessarily implies a small witness).

5.26 The Organizers Try to Explain ETH Using Somebody Else's Slides

Ryan Williams (Stanford University, US)

To begin the seminar with a common focal point, I presented an overview of the Exponential Time Hypothesis (ETH) and the Strong Exponential Time Hypothesis (SETH). Namely, I discussed how these hypotheses are formulated, their consistency with current knowledge, their many consequences for parameterized and exact exponential time algorithms, and (briefly) their plausibility. For this purpose, I worked with a modification of excellent slides by Daniel Lokshtanov (hence the talk title), based on the survey [1].

References

1 Daniel Lokshtanov, Dániel Marx, Saket Saurabh. Lower bounds based on the Exponential Time Hypothesis. *Bulletin of the EATCS* 105, 41–72, 2011.

6 Open Problems

6.1 Fast algorithm for DRPP w.r.t. the number of weakly connected components

Alexander S. Kulikov

In the *directed rural postman problem* (DRPP) one is given a directed weighted multigraph $G = (V, A)$ together with a subset $R \subseteq A$ of its arcs and the goal is to find a shortest closed walk in this graph going through all the arcs from R. Although DRPP is NP-hard in general case, it can be solved in polynomial time if the arcs from R form a single weakly connected component [1] (weakly connected components of a directed graph are connected components in this graph with all directed arcs replaced by undirected edges).

Question

Is DRPP fixed-parameter tractable w.r.t. the number of weakly connected components k? Can it be solved in time $O^*(2^k)$?

Remark. Shortly after the seminar Gutin, Wahlström, and Yeo [2] answered the first question in the affirmative. One month later they answered the second question as well.

References

1 N. Christofides, V. Campos, A. Corberan, and E. Mota. An algorithm for the Rural Postman problem on a directed graph. In *Netflow at Pisa*, volume 26 of *Mathematical Programming Studies*, pages 155–166. Springer Berlin Heidelberg, 1986.
2 G. Gutin, M. Wahlström, and A. Yeo.Parameterized rural postman and conjoining bipartite matching problems. *arXiv preprint arXiv:1308.2599*, 2013.

6.2 Shortest common superstring problem

Alexander S. Kulikov

In the *shortest common superstring problem* (SCS) one is given n strings $s_1, \ldots, s_n$ and the goal is to find a shortest string containing each s_i as a substring. It can be solved in $O^*(2^n)$ time and space by dynamic programming. Using inclusion-exclusion one can reduce the space to polynomial.

Question

Can SCS be solved in time $O^*(1.99^n)$? Or, at least, can it be solved in time $O^*((2-c)^n)$ where $c = c(r) > 0$ for a special case when all input strings have length at most r? Can one reduce TSP with n vertices to SCS with n strings (this would show that SCS is as hard as TSP)?

Remark. Note also that the well-known greedy conjecture (saying that one gets a 2-approximation for SCS by overlapping $n-1$ times two strings with maximal overlap) is still open.

6.3 The exponential complexity of some SUBSET SUM variants

Andrew Drucker

Question

Here are some variants of the NP-complete SUBSET SUM problem.

In the THRESHOLD PROBABILITY problem, one is given integers $a_1, \ldots, a_n$ and a threshold value T, and asked to compute the probability over a random $S \subseteq [n]$ that $\sum_{i \in S} a_i \geq T$. This problem lies in $\#P$.

In the RANKED SUBSET SUM problem, one is given integers $a_1, \ldots, a_n$ and a number $k \in \{1, 2, \ldots, 2^n\}$, and is asked to compute the k^{th} largest value $\sum_{i \in S} a_i \geq T$, ranging over all possible $S \subseteq [n]$. It is not clear to me whether this problem is even in $\#P$. It does lie in $P^{\#P}$, however. (Proof sketch: we use binary search. With a $\#P$ oracle we can determine the rank order of a given subset S's sum. Using the power of the Polynomial Hierarchy we can choose a subset nearly-uniformly from those ranked above or below S. And computations in the Polynomial Hierarchy can be done in $P^{\#P}$ by Toda's theorem.)

It may well be known that each of these problems is $\#P$-hard. However, my question is whether a stronger statement is true: namely, whether either or both of these problems requires $2^{\Omega(n)}$ time (or something close) under the *counting Exponential Time Hypothesis (#ETH)* of Dell, Husfelt, Marx, and Taslaman [1]. These would give more basic examples of problems to which this hardness assumption applies.

References

1 Holger Dell and Thore Husfeldt and Dániel Marx and Nina Taslaman and Martin Wahlen, *Exponential Time Complexity of the Permanent and the Tutte Polynomial*, CoRR, abs/1206.1775, 2012, http://arxiv.org/abs/1206.1775.

6.4 Maximum Acyclic Subgraph

Eunjung Kim

Given a directed graph $D = (V, A)$, the problem Maximum Acyclic Subgraph, or MAS, asks to find a subdigraph $D' = (V, A')$ of D such that $|A'|$ is maximized. The problem MAS can be solved in time $O(2^n)$, where $n = |V|$, using dynamic programming [1].

Question

Is there an algorithm for MAS running in time $O(c^n)$ for a constant $c < 2$?

Remark. The problem MAS can be viewed as one of the family of arity k ordering CSP. An instance of ordering CSP consists of a set of variables V and a set of constraints $\mathcal{C}$, in which constraints are tuples of elements of V . The goal is to find a total ordering of the variables, $\pi : V \to \{1, \ldots, |V|\}$, which satisfies as many constraints as possible. A constraint $(v_1, \ldots, v_k)$ is satisfied by an ordering π when $\pi(v_1) < \pi(v_2) < \ldots < \pi(v_k)$. An instance has arity k if all the constraints involve at most k elements. The problem MAS is equivalent to arity 2 ordering CSP.

While the dynamic programming approach for ordering problem such as in [1] yields an $O(2^n)$-time algorithm for arity 3 ordering CSP, it is unlikely to solve arity k ordering CSP in time $O(2^{o(n \log n)})$ for $k \geq 4$ under ETH, see [2]. Such dichotomy leads to a further question: can arity 3 ordering CSP be solved in time $O(c^n)$ for $c < 2$?

References

1 Hans L. Bodlaender, Fedor V. Fomin, Arie M. C. A. Koster, Dieter Kratsch, and Dimitrios M. Thilikos. A note on exact algorithms for vertex ordering problems on graphs. *Theory Comput. Syst.*, 50(3):420–432, 2012.

2 Eun Jung Kim and Daniel Gonçalves. On exact algorithms for the permutation CSP. *Theoretical Computer Science*, Vol. 514, pp. 109–116, 2012. ISSN 0304-3975. http://dx.doi.org/10.1016/j.tcs.2012.10.035.

6.5 Enumerating all story arcsets (SAS) of a digraph with polynomial delay

Ewald Speckenmeyer

The problem was motivated by a biological question w.r.t. metabolic networks.
$G = (V, E)$ be a digraph. A subset $F \subseteq E$ of arcs is a **feedback arc set** (**FAS**) if $G - F$ is acydic.

Result

Schwikowski, Sp.: On enumerating all minimal solutions of feedback problems; DAM 2002

All minimal FAS of a digraph can be enumerated with polynomial delay.
$\rightsquigarrow$ All maximal arc-induced acydic digraphs of a digraph can be enumerated with polynomial delay.

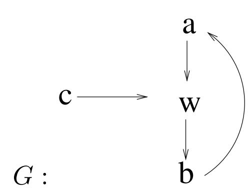
Nodeset $V = B \cup W$ consists of **black** nodes from B and **white** nodes from W. Definitions used here are from:

- Acuña, et al.: Telling stories; TCS 2012
- Borassi, et al: Telling stories fast; SEA 2013.

A **pitch** of $G = (B \cup W, E)$ is an acydic digraph G' of G, containing all nodes from B, a subset $W' \subseteq W$, and a subset $E' \subseteq E$, s.t. no node from W' is source or target node of G'. A **story** S of G is a maximal pitch of G. The corresponding arc set $E - E'$ is called **SAS** of G. **Telling stories** means enumerating all stories S of G.

Example

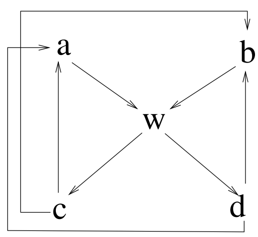
$B = \{a, b, c\}, W = \{w\}$



3 minimal FAS: $\{a \to w\}$, $\{\mathbf{w} \to \mathbf{b}\}$ $\{b \to a\}$. $E(G) - \{\mathbf{w} \to \mathbf{b}\}$ is no SAS. An SAS induces a FAS, but not vice versa!

Example

(from Acuña, et al.)



$S = \{a \to w,\ b \to w,\ w \to c,\ w \to d\}$ is a SAS, but $E - S$ is **not** a minimal FAS of G.

Open Problem

Enumerate all stories of a digraph with polynimial delay. A first step towards solving the problem by Borassi, et.al.: Enumerate pitches with linear delay.

6.6 Exact Algorithm For Subgraph Isomorphism

Fedor V. Fomin

For any graph G with n vertices and m edges, it is possible to decide in time $2^{m+o(n)}$ if G contains a given graph F as a subgraph: try all possible edge subsets of G and for each subset check if the obtained graph is isomorphic to F. Another approach is to try all the permutations of the vertices of G and F, and for each of these permutations, to compare vertex neighborhoods. This will give us running time $O(n!n^2) = 2^{O(n \log n)}$.

Question

Can subgraph isomorphism can be solved in time $2^{o(n \log n)}$?

6.7 Exact algorithms for Special Treewidth and Spaghetti Treewidth

Hans Bodlaender

Courcelle [2] introduced the notion of *Special Treewidth.* Some resuls (e.g., fixed parameter tractability) on the notion were recently obtained [1]. A *special tree decomposition* is a rooted tree decomposition, such that for each vertex, the bags containing the vertex form a rooted path in the tree from the tree decomposition. The special treewidth is the minimum width of a special tree decomposition. (A rooted path is a path from a vertex to one of its ancestors.)

In [1], we also introduced the notion of spaghetti treewidth. A *spaghetti tree decomposition* is a tree decompositions, such that for each vertex, the bags containing the vertex from a pah in the tree. (I.e., it is not necessarily rooted.)

Computing the special treewidth and spaghetti treewidth are NP-hard.

Question

Find efficient exact algorithms to compute the special treewidth or spaghetti treewidth of a given graph.

Remark. The existence of $O*(2^n)$ algorithms for these problems is not known.

References

1 Hans L. Bodlaender, Vincent Kreuzen and Stefan Kratsch. *Fixed-Parameter Tractability and Characterizations of Small Special Treewidth* , To appear in the proceedings of WG 2013.

2 Bruno Courcelle. Special tree-width and the verification of monadic second-order graph properties. Proceedings FSTTCS 2010, LIPIcs, Vol. 8, pp. 13–29, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, DOI: 10.4230/LIPIcs.FSTTCS.2010.13, 2010.

6.8 Treewidth of Planar Graphs

Hans Bodlaender

The famous Ratcatcher algorithm by Seymour and Thomas gives a polynomial time algorithm to compute the branchwidth of a planar graph. This directly implies that the treewidth of a planar graph can be approximated with a factor 1.5. The treewidth of a planar graph with n vertices is bounded by $O(\sqrt{n})$.

Question

How hard is the following problem: Given is a planar graph G, and an integer k. Is the treewidth of G at most k? Is it NP-complete? Is it polynomial time solvable?

Comment. This is a long standing open problem.

References

1 aul D. Seymour, Robin Thomas: Call Routing and the Ratcatcher. Combinatorica 14(2): 217-241 (1994)

6.9 Improving pseudo polynomial time algorithms for Subset Sum

Jesper Nederlof

The Subset Sum problem is defined as follows: Given integers $w_1, \ldots, w_n$ and budget w, decide whether there exists $X \subseteq \{ 1, \ldots, n\}$ such that $\sum_{i \in X} w_i = w$. A classical example of an exact algorithm for a NP-hard problem is the algorithm of Bellman from the 50's that solves the Subset Sum problem in $O(nw)$ time and $O(w)$ space (note that this resource bound assumes that computation and storage involving arbitrary integers requires constant time / storage).

Question 1

Does there exist $\epsilon > 0$ such that Subset Sum be solved in $w^{1-\epsilon} n^{O(1)}$?

In [1], the authors relate the above question to improvements of naive algorithms for Set Cover. However, there are no known consequences of the Strong ETH in this context.

In 2010 [4] (see also [2, 3]), alternatives for Bellman's algorithm were presented that are significantly more space efficient. In particular, Subset Sum can be solved in $\tilde{O}(n^3 t \log t)$ time and $\tilde{O}(n^2)$ space (note that these resource bounds are stated in terms of bit operations / storage). It would be interesting to see whether the time usage of this algorithm can be improved in order to practically compete/outperform Bellman's algorithm:

Question 2

Can Subset Sum be solved in $\tilde{O}(nt \log t)$ time and $\tilde{O}(n)$ space?

After posing this question, Ryan Williams mentioned that, together with Huacheng Yu, he made partial progress in this direction.

In the Knapsack problem we are additionally given values $v_1, \ldots, v_n$, goal v and we are asked to find $X \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in X} v_i \geq v$ and $\sum_{i \in X} w_i \leq w$. Bellman's approach solves the Knapsack problem in $O(n \min\{v, w\})$ and $O(\min\{v, w\})$ space. However, considering space efficient algorithms, Knapsack can currently only be solved in $\tilde{O}(n^4 vw \log(vw))$ time and $\tilde{O}(n^2 \log(vw))$ space. It would be interesting to improve this:

Question 3

Can Knapsack be solved in $\min\{v, w\} n^{O(1)}$ time and $(n \log(vw))^{O(1)}$ space?

Other non-trivial improvements or time/space tradeoffs in this context would also be interesting, but it seems hard to mix the greedy aspect of Bellman's approach with the coefficient interpolation strategy of the space efficient approach.

References

1. Marek Cygan, Holger Dell, Daniel Lokshtanov, Dániel Marx, Jesper Nederlof, Yoshio Okamoto, Ramamohan Paturi, Saket Saurabh, and Magnus Wahlström. On Problems as Hard as CNF-sat. In *IEEE Conference on Computational Complexity*, pages 74–84. IEEE, 2012. ISBN 978-1-4673-1663-7.
2. Michael Elberfeld, Andreas Jakoby, and Till Tantau. Logspace Versions of the Theorems of Bodlaender and Courcelle. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, FOCS '10, pages 143–152, Washington, DC, USA, 2010. IEEE Computer Society. ISBN 978-0-7695-4244-7. 10.1109/FOCS.2010.21. URL http://dx.doi.org/10.1109/FOCS.2010.21.
3. Daniel M. Kane. Unary Subset-Sum is in Logspace. *CoRR*, abs/1012.1336, 2010.
4. Daniel Lokshtanov and Jesper Nederlof. Saving Space by Algebraization. In *Proceedings of the 42nd ACM symposium on Theory of computing*, STOC '10, pages 321–330, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0050-6. 10.1145/1806689.1806735. URL http://doi.acm.org/10.1145/1806689.1806735.

6.10 Channel assignment

Łukasz Kowalik

In the channel assignment problem, we are given a symmetric weight function $w : V^2 \to \mathbb{N}$ (we assume that $0 \in \mathbb{N}$). The elements of V are called vertices (as w induces a graph on the vertex set V with edges corresponding to positive values of w). We say that w is ℓ-bounded when for every $x, y \in V$ we have $w(x, y) \leq \ell$. An assignment $c : V \to \{1, \ldots, s\}$ is called *proper* when for each pair of vertices x, y we have $|c(x) - c(y)| \geq w(x, y)$. The number s is called the *span* of c. The goal is to find a proper assignment of minimum span. Note that the special case when w is 1-bounded corresponds to the classical graph coloring problem.

Currently the fastest algorithm for the channel assignment problem, due to Cygan and Kowalik [1] runs in $O^*((\ell + 1)^n)$ time, where n denotes the number of vertices.

Traxler [2] has shown that for any constant c, the Constraint Satisfaction Problem (CSP) has no $O(c^n)$-time algorithm, assuming the Exponential Time Hypothesis (ETH). More precisely, he shows that ETH implies that CSP requires $d^{\Omega(n)}$ time, where d is the domain size. On the other hand, graph coloring, which is a variant of CSP with unbounded domain, admits a $O^*(2^n)$-time algorithm. The channel assignment problem is a generalization of graph coloring and a special case of CSP. In that context, the central open problem in the complexity of the channel assignment problem is the following:

Open Problem

Find an $O^*(c^n)$-time algorithm for a constant c independent of ℓ or to show that such the algorithm does not exist, assuming ETH (or other well-established complexity conjecture).

References

1 M. Cygan and Ł. Kowalik. Channel Assignment via Fast Zeta Transform. *Information Processing Letters*, 111(15):727-730, 2011.
2 P. Traxler. The Time Complexity of Constraint Satisfaction. In *Proc. IWPEC 2008*, LNCS 5018, pages 190–201, 2008.

6.11 Improve exponential time algorithm for Bandwidth

Marek Cygan

In the Bandwidth problem one is given an undirected graph $G = (V, E)$ together with an integer b. The question is whether there exists a bijection $\pi : V \to \{1, \ldots, n\}$, such that for each edge $uv \in E$ we have $|\pi(u) - \pi(v)| \leq b$. Bandwidth is an NP-complete problem, and currently the fastest known exact exponential time algorithms work in:

- $O(4.383^n)$ time and exponential space [1],
- $O(9.363^n)$ time and polynomial space [2].

Question

Can the above algorithms be improved, in particular is there an $O^*(4^n)$ time algorithm for Bandwidth?

References

1 Marek Cygan and Marcin Pilipczuk, *Exact and approximate bandwidth*, Theoretical Computer Science 411(40-42): 3701-3713 (2010).
2 Marek Cygan and Marcin Pilipczuk, *Bandwidth and distortion revisited*, Discrete Applied Mathematics, 160(4-5): 494-504 (2012).

6.12 Relation between permanent computation and SETH

Marek Cygan

One can compute permanent of a given matrix in $O^*(2^n)$ time either by dynamic programming or by the inclusion-exclusion principle. A slightly faster $2^{n-\Omega(\sqrt{n/\log n})}$ was recently obtained by Björklund [1]. Can we relate the hardness of permanent computation to SETH, by showing that one of those problems is at least as hard as the other?

Remark. Preferably the reduction would involve binary matrices only, but the general case would also be interesting.

References

1 Andreas Björklund, *Below All Subsets for Permutational Counting Problems*, CoRR, abs/1211.0391, 2012, http://arxiv.org/abs/1211.0391.

6.13 Exact Counting of Linear Extensions

Mikko Koivisto

Counting the linear extensions of a given partially ordered set (poset) is #P-complete [1]. While the problem admits a FPRAS [4], the practical value of the existing schemes [2] is questionable, as the running time grows roughly as $n^6\epsilon^{-2}$ for n-element posets and relative error ϵ. Fast exact algorithms could be more practical when n is around 100 and good accuracy is needed. Currently the best known worst-case bound is $2^n n^{O(1)}$, based on simple dynamic programming across the downsets of the poset [3, 5].

Question

Is there a $O(c^n)$-time algorithm for counting linear extensions, for some $c < 2$? Is the problem fixed-parameter tractable with respect to the treewidth of the Hasse diagram (i.e., the transitive reduction of the poset)?

Remark. The problem can be solved in $n^{t+O(1)}$ time for treewidth t (unpublished work).

References

1 G. Brightwell and P. Wrinkler, *Counting linear extensions*, Order 8 (1991) 225–242.
2 R. Bubley and M. Dyer, *Faster random generation of linear extensions*, Discrete Mathematics 201 (1999) 81–88.
3 K. De Loof, H. De Meyer, and B. De Baets, *Exploiting the lattice of ideals representation of a poset*, Fundamenta Informaticae 71 (2006) 309–321.
4 M. Dyer, A. Frieze, and R. Kannan, *A random polynomial-time algorithm for approximating the volume of convex bodies*, Journal of the ACM 38 (1991) 1–17.
5 T. Niinimäki and M. Koivisto, *Annealed importance sampling for structure learning in Bayesian networks*, 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013), pp. 1579–1585, AAAI, 2013.

6.14 Vector Positive Explanation

Rolf Niedermeier

The problem Vector Postitive Explanation is motivated by applications in context of radiation therapy.

Input: A vector $A \in IN_0^n$ and an integer $k > 0$.
Question: Can A be explained by at most k positive segments?

Herein, a *segment* is a vector in $\{0, a\}^n$ for some $a \in \mathbb{Z} \setminus \{0\}$ where all a-entries occur consecutively, and a segment is positive if a is positive. An *explanation* is a set of segments that component-wisely sum up to the input vector.

The parameter we are interested in is the maximum difference between two consecutive vector entries $\delta := \max_{0 \leq i \leq n} |A[i] - A[i+1]|$, assuming that $A[0] = A[n+1] = 0$.

Example. The vector $(4, 3, 3, 4)$ can be explained by $(3, 3, 3, 3)$, $(1, 0, 0, 0)$, and $(0, 0, 0, 1)$. Herein, $\delta = 4$.

Question

Is Vector Positive Explanation parameterized by δ fixed-parameter tractable?

Remark. The variant where also negative segments are allowed is known to be fixed-parameter tractable when parameterized by δ (using an integer linear program formulation and applying Lenstra's famous result).

References

1 Robert Bredereck and Jiehua Chen and Sepp Hartung and Christian Komusiewicz and Rolf Niedermeier and Ondrej Suchý, *On Explaining Integer Vectors by Few Homogenous Segments*, Proc. WADS 2013, LNCS 8037, pages 207-218, 2013. Journal version under review and available upon request.

6.15 The Complexity of #k-SAT

Rahul Santhanam

Question

Is there a deterministic algorithm for #k-SAT running in time $2^{n-\Omega(n/k)}$?

Remarks. There is a probabilistic algorithm due to Impagliazzo, Mathews and Paturi with the required time complexity, but it is not known how to derandomize the algorithm. To the best of my knowledge, it is also unknown whether there is an algorithm for k-SAT running in time $2^{n-n/k}$.

6.16 Strong Backdoor detection

Serge Gaspers

Let $\mathcal{C}$ be a class of CNF formulas. For example, the class $\mathcal{W}_t$ contains all CNF formulas whose incidence graphs have treewidth at most t. The *incidence graph* of a CNF formula is a bipartite graph whose vertex set is bipartitioned into the set of variables and the set of clauses of F; a variable is adjacent to a clause if the variable is contained in the clause.

The Strong $\mathcal{C}$-Backdoor Detection problem [3] takes as input a CNF formula F on n variables and an integer k, and the question is whether there exists a subset of variables $S \subseteq \mathsf{var}(F)$ with $|S| \leq k$ such that for each assignment $\alpha : S \to \{0, 1\}$ we have that $F[\alpha] \in \mathcal{C}$. Here, the formula $F[\alpha]$ is obtained from F by removing all clauses containing a literal set to 1 by α, and by removing the literals set to 0 from all remaining clauses.

Question 1

Let $t > 0$. Is there an algorithm for Strong $\mathcal{W}_t$-Backdoor Detection with running time $O((3-\epsilon)^n)$ for some $\epsilon > 0$?

Note that a $O(3^n)$ time algorithm is trivially obtained by going through all $S \subseteq \mathsf{var}(F)$ and all assignments to S. Since backdoors are mainly useful when they are small and since the main reason for computing backdoors is to solve SAT and #SAT, we arrive at the second question.

Question 2

Let $t > 0$. Is there an algorithm for Strong $\mathcal{W}_t$-Backdoor Detection restricted to instances where $k \le n/4$ with running time $O((2-\epsilon)^n)$ for some $\epsilon > 0$?

Note that the trivial algorithm restricted to subsets of size at most $n/4$ executes at least $\binom{n}{n/4}2^{n/4} \approx 2.0868\ldots^n$ steps. Both questions remain open and interesting for $t = 1$. For further background on these problems, see [2, 1].

References

1. Serge Gaspers and Stefan Szeider, *Strong Backdoors to Bounded Treewidth SAT*, Proceedings of FOCS 2013, 2013.
2. Serge Gaspers and Stefan Szeider, *Backdoors to Satisfaction*, The Multivariate Algorithmic Revolution and Beyond, pages 287-317, 2012.
3. Ryan Williams, Carla P. Gomes, and Bart Selman, *Backdoors To Typical Case Complexity*, Proceedings of IJCAI 2003, pages 1173-1178, 2003.

6.17 Monochromatic Rectangles in Low-Rank Matrices

Shachar Lovett

Conjecture: Let A be a real $n \times n$ matrix of rank r, such that (say) $n^2/2$ of its entries are zero. Then there exists a rectangle $R \subset [n] \times [n]$ of size $|R| \ge n^2 \exp(-O(\sqrt{r}))$ such that A restricted to R is zero.

If true, the bound is best possible. Take $A = BB^t$ where B is an $n \times r$ matrix whose rows are all the $\{0,1\}^r$ vectors of hamming weight $\sqrt{r}/10$. We have $n = r^{\Theta(\sqrt{r})}$. Then 99% of the elements of A are zero, but the largest zero rectangle corresponds to choosing rows corresponding to vectors supported on the first half of the coordinates, and columns corresponding to vectors supported on the last half of the coordinates.

6.18 Cutwidth

Saket Saurabh

Question

The cutwidth of a graph G is the smallest integer k such that the vertices of G can be arranged in a linear layout $[v_1, \ldots, v_n]$ in such a way that, for every $i = 1, \ldots, n-1$, there are at most k edges with one endpoint in $\{v_1, \ldots, v_i\}$ and the other in $\{v_{i+1}, \ldots, v_n\}$. Does there exist an algorithm to find cutwidth of a graph on n vertices that runs in time c^n, where $c < 2$ is a constant.

Remark. There does exist an algorithm on bipartite graphs that run in time 1.415^n [1].

References

1 M. Cygan, D. Lokshtanov, M. Pilipczuck, M. Pilipczuck and S. Saurabh, *On Cutwidth Parameterized by Vertex Cover*, IPEC, LNCS 7112, 246-258, 2011.

6.19 Converting CNF to DNF

Stefan Schneider

Question

Given a k-CNF, how large is the smallest equivalent DNF, measured in the number of terms?

This question is discussed by Miltersen, Radhakrishnan and Wegener [1], who show that $2^{n(1-\frac{1}{100k})}$ terms are sufficient. The bound coincides up to the constant in the exponent with block parity, that requires $2^{n(1-\frac{1}{k})}$ terms (if k divides n). The question asks if we can close the gap between the lower bound and the upper bound.

The proof of Miltersen, Radhakrishnan and Wegener is based on Håstad's Switching Lemma [2]. Since the Switching Lemma is not tight up to constants, the resulting bound on the number of terms has this constant in the exponent.

To close this gap, one might try to prove a bound based on the Satisfiability Coding Lemma [3], which shows (among others) that no k-CNF can have more than $2^{n(1-\frac{1}{k})}$ isolated solutions, where an isolated solution is a satisfying assignment such that no other assignment of Hamming distance 1 is also satisfying.

References

1 P. Miltersen, J. Radhakrishnan and I. Wegener , *On converting CNF to DNF*, Theor. Comput. Sci, 2005.
2 J. Håstad, *Almost optimal lower bounds for small depth circuits*, STOC, 1986.
3 R. Paturi, P. Pudlák and F. Zane, *Satisfiability Coding Lemma*, FOCS, 1997.

6.20 Directed Hamiltonicity

Thore Husfeldt

Despite recent progress in exponential time algorithms for Hamiltonicity [2, 3, 5, 4], Open Problem 3.1 in Woeginger's influential survey [7] remains open in its general form: we do not have an algorithm for Hamilonicity in directed graphs (nor for TSP with unbounded weights) whose exponent is better than the dynamic programming solution of Bellman and Held–Karp from 1962 [1, 6].

Question

Construct an algorithm for the Hamiltonian cycle problem in directed graphs with time complexity $O(1.999^n)$.

References

1 R. Bellman. Dynamic programming treatment of the travelling salesman problem, J. Assoc. Comput. Mach. 9, pp. 61–63, 1962.

2 A. Björklund. Determinant sums for undirected Hamiltonicity. In Proceedings of the 51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23–26, 2010, Las Vegas, Nevada, USA. IEEE Computer Society 2010, 51st FOCS, pages 173–182.

3 A. Björklund. Below all subsets for permutational counting problems. arXiv:1211.0391, 2012.

4 A. Björklund, T. Husfeldt: The Parity of Directed Hamiltonian Cycles. 54th Annual Symposium on Foundations of Computer Science, FOCS 2013, October 27–29, 2013, Berkeley, California, arXiv:1301.7250, 2013.

5 M. Cygan, S. Kratsch, and J. Nederlof. Fast Hamiltonicity checking via bases of perfect matchings. 45th ACM Symposium on the Theory of Computing, STOC 2013, June 1–4, 2013, Palo Alto, USA, arXiv:1211.1506, 2012.

6 M. Held and R. M. Karp. A dynamic programming approach to sequencing problems, J. Soc. Indust. Appl. Math. 10, pp. 196–210, 1962.

7 G. J. Woeginger. Exact algorithms for NP-hard problems: A survey. In Combinatorial Optimization – Eureka! You shrink!, M. Juenger, G. Reinelt and G. Rinaldi (eds.). Lecture Notes in Computer Science 2570, Springer 2003, pages 185–207.

Participants

- Per Austrin
KTH Stockholm, SE
- Hans L. Bodlaender
Utrecht University, NL
- Marek Cygan
University of Warsaw, PL
- Holger Dell
University of Wisconsin – Madison, US
- Andrew Drucker
MIT, US
- Michael Elberfeld
ICSI – Berkeley, US
- Fedor V. Fomin
University of Bergen, NO
- Serge Gaspers
The University of New South Wales, AU
- Pinar Heggernes
University of Bergen, NO
- Timon Hertli
ETH Zürich, CH
- Thore Husfeldt
IT Univ. of Copenhagen, DK
- Kazuo Iwama
Kyoto University, JP
- Petteri Kaski
Aalto University, FI
- Eun Jung Kim
University Paris-Dauphine, FR
- Joachim Kneis
RWTH Aachen, DE
- Mikko Koivisto
University of Helsinki, FI
- Łukasz Kowalik
University of Warsaw, PL
- Dieter Kratsch
University of Metz, FR
- Stefan Kratsch
TU Berlin, DE
- Alexander S. Kulikov
Steklov Institute – St. Petersburg, RU
- Daniel Lokshtanov
University of Bergen, NO
- Shachar Lovett
University of California – San Diego, US
- Dániel Marx
Hungarian Acad. of Sciences, HU
- Matthias Mnich
Universität des Saarlandes, DE
- Jesper Nederlof
Utrecht University, NL
- Rolf Niedermeier
TU Berlin, DE
- Ramamohan Paturi
University of California – San Diego, US
- Peter Rossmanith
RWTH Aachen, DE
- Rahul Santhanam
University of Edinburgh, GB
- Saket Saurabh
The Institute of Mathematical Sciences – Chennai, IN
- Dominik Scheder
Aarhus University, DK
- Stefan Schneider
University of California – San Diego, US
- Kazuhisa Seto
The Univ. of Electro-Communications – Tokyo, JP
- Gregory B. Sorkin
London School of Economics, GB
- Ewald Speckenmeyer
Universität Köln, DE
- David Steurer
Cornell University, US
- Stefan Szeider
TU Wien, AT
- Navid Talebanfard
Aarhus University, DK
- Suguru Tamaki
Kyoto University, JP
- Virginia Vassilevska Williams
Stanford University, US
- Magnus Wahlström
Royal Holloway University of London, GB
- Ryan Williams
Stanford University, US