

08431 Abstracts Collection
Moderately Exponential Time Algorithms
— **Dagstuhl Seminar** —

Fedor V. Fomin¹, Kazuo Iwama² and Dieter Kratsch³

¹ University of Bergen, Norway

`fedor.fomin@ii.uib.no`

² Kyoto University, Japan

`iwama@kuis.kyoto-u.ac.jp`

³ Université de Metz; Paul Verlaine - Metz, France

`kratsch@univ-metz.fr`

Abstract. From 19/10/2008 to 24/10/2008, the Dagstuhl Seminar 08431 “Moderately Exponential Time Algorithms” was held in Schloss Dagstuhl – Leibniz Center for Informatics. During the seminar, several participants presented their current research, and ongoing work and open problems were discussed. Abstracts of the presentations given during the seminar as well as abstracts of seminar results and ideas are put together in this paper. The first section describes the seminar topics and goals in general. Links to extended abstracts or full papers are provided, if available.

Keywords. Algorithms, Exponential time algorithms, Graphs, SAT

08431 Executive Summary – Moderately Exponential Time Algorithms

The Dagstuhl seminar on **Moderately Exponential Time Algorithms** took place from 19.10.08 to 24.10.08. This was the first meeting of researchers working on exact and “fast exponential time” algorithms for hard problems. In total 54 participants came from 18 countries.

Moderately exponential time algorithms for NP-hard problems are a natural type of algorithms and research on them dates back to Held and Karp’s paper on the travelling salesman problem in the sixties. However until the year 2000, papers were published only sporadically (with the exception of work on satisfiability problems maybe). Some important and fundamental techniques have not been recognized at full value or even been forgotten, as e.g. the Inclusion-Exclusion method from Karp’s ORL paper in 1982.

Recently the situation has changed — there is a rapidly increasing interest in exponential time algorithms on hard problems and papers have been accepted for high-level conferences in the last few years. There are many (young) researchers that are attracted by moderately exponential time algorithms, and this interest

is easy to explain, the field is still an unexplored continent with many open problems and new techniques are still to appear to solve such problems. To mention a few example:

- There is a trivial algorithm that for a given SAT formula Φ with m clauses and n variables determines in time roughly $O(2^n + m)$ whether there is a satisfying assignment for Φ . Despite of many attempts, no algorithm of running time $O(c^n + m)$, for some $c < 2$ is known. So what happens here? Is it just because we still do not have appropriate algorithmic techniques or are there deeper reasons for our failure to obtain faster algorithms for some problems? It would be very exciting to prove that (up to some reasonable conjecture in complexity theory) there exists a constant $c > 1$ such that SAT cannot be solved in time c^n .
- One of the most frequently used methods for solving NP-hard problems is Branch & Reduce. The techniques to analyze such algorithms, that we know so far, are based on linear recurrences and are far from being precise. The question here is: How to analyze Branch & Reduce algorithms to establish their worst case running time?
- The algorithm deciding whether a given graph on n vertices has a Hamiltonian cycle has running time $2^n \cdot n^{O(1)}$ and it is known since the 1960s. Amazingly, all progress in algorithms for the last 40 years did not have any impact on the solution of this problem. Are there new techniques which can be applied to crack this problem?

Despite of the growing interest and the new researchers joining the potential community there has not been a workshop on moderately exponential time algorithms longer than one day since the year 2000. The major goal of the proposed Dagstuhl seminar was to unite for one week many of the researchers being interested in the design and analysis of moderately exponential algorithms for NP-hard problems. The Dagstuhl seminar was a unique opportunity to bring together those people, to share insights and methods, present and discuss open problems and future directions of research in the young domain.

There were 27 talks and 2 open problem sessions. Talks were complemented by intensive informal discussions, and many new research directions and open problems will result from these discussions. The warm and encouraging Dagstuhl atmosphere stimulated new research projects. We expect many new research results and collaborations growing from the seeds of this meeting.

Joint work of: Fomin, Fedor V.; Iwama, Kazuo; Kratsch, Dieter

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2008/1797>

Efficient approximation of some NP-hard problems

Bruno Escoffier (Université Paris-Dauphine)

In this talk we propose a way to bring together two domains that are the polynomial approximation and the exact computation for NP-hard problems: We illustrate how one can match ideas from both areas in order to design approximation algorithms achieving ratios unachievable in polynomial time (unless a very unlikely complexity conjecture is confirmed) with worst-case complexity much lower (though super-polynomial) than that of an exact computation. We propose several techniques to get interesting tradeoffs between running time and approximation ratios for paradigmatic optimization problems, mainly independent set, vertex cover and set cover.

Keywords: Exponential algorithms, approximation algorithms

Joint work of: Escoffier, Bruno; Paschos, Vangelis

Harvesting Reference Search Trees

Henning Fernau (Universität Trier)

The POWER DOMINATING SET problem is an extension of the well-known domination problem on graphs in a way that we enrich it by a second propagation rule: Given a graph $G(V, E)$ a set $P \subseteq V$ is a power dominating set if every vertex is observed after we have applied the next two rules exhaustively. First, a vertex is observed if $v \in P$ or it has a neighbor in P . Secondly, if an observed vertex has exactly one unobserved neighbor u , then also u will be observed as well. We show that POWER DOMINATING SET remains NP-hard on cubic graphs. We designed an algorithm solving this problem in time $O^*(1.76^n)$ on general graphs, using polynomial space. To achieve this we have coined a new notion of search trees called reference search trees. This setting also allows to trade exponential space for time, yielding an $O^*(1.65^n)$ running time (using $O^*(1.53^n)$ space).

Keywords: Exact algorithms for hard problems; power dominating set

Joint work of: Fernau, Henning; Raible, Daniel

See also: An extended abstract of this paper appeared in the proceedings of ISAAC 2008.

See also: Henning Fernau and Daniel Raible, Harvesting Reference Search Trees, ISAAC 2008, Springer LNCS 5369, pp. 136-147, 2008.

Quantum Search for Problems with Moderately Exponential Time Complexity

Martin Fürer (Pennsylvania State University)

In his seminal paper, Grover points out the prospect of faster solutions for an NP-complete problem like SAT. If there are n variables, then an obvious classical deterministic algorithm checks out all 2^n truth assignments in about 2^n steps, while his quantum search algorithm can find a satisfying truth assignment in $O(2^{n/2})$ steps. The method uses the simple structure of the deterministic search algorithm and does not directly extend to arbitrary exponential time searches.

We present a simple method to obtain the full $T(n)$ to $O^*(\sqrt{T(n)})$ speed-up for most of the many moderately exponential time algorithms for NP-hard problems. The method works whenever the widely used technique of recursive decomposition is employed.

Keywords: Quantum Computing, Recursive Decomposition, Moderately Exponential Time

See also: Martin Fürer, Solving NP-Complete Problems with Quantum Search, LATIN 2008, Springer LNCS 4957, pp. 784-792, 2008.

A universally fastest algorithm for Max 2-Sat, Max 2-CSP, and everything in between

Serge Gaspers (University of Bergen)

We introduce “hybrid” Max 2-CSP formulas consisting of “simple clauses”, namely conjunctions and disjunctions of pairs of variables, and general integer-valued 2-variable clauses, which can be any functions of pairs of boolean variables. This allows an algorithm to use both efficient reductions specific to AND and OR clauses, and other powerful reductions that require the general CSP setting. Parametrizing an instance by the fraction p of non-simple clauses, we give an exact, polynomial-space algorithm that is the fastest known for Max 2-Sat (and other formulas with $p = 0$), tied for fastest for general Max 2-CSP ($p = 1$), and the only efficient algorithm for mixtures of simple and general clauses ($0 < p < 1$). The algorithm uses new reductions introduced here, and known reductions adapted to our hybrid setting. Each reduction imposes constraints on various parameters, and the running-time bound is an “objective function” of these parameters and p . The optimal running-time bound is obtained by solving a convex nonlinear program, which can be done efficiently and with a certificate of optimality.

Keywords: Max-2-CSP, Max-2-SAT, Reduction Rules, Convex Non-linear program

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

On random ordering constraints

Andreas Goerdt (TU Chemnitz)

Ordering constraints are analogous to instances of the satisfiability problem in conjunctive normalform, but instead of a boolean assignment we consider a linear ordering of the variables in question. A clause becomes true given a linear ordering iff the relative ordering of its variables obeys the constraint considered.

The naturally arising satisfiability problems are **NP**-complete for many types of constraints. The present paper seems to be one of the first looking at random ordering constraints.

Experimental evidence suggests threshold phenomena as in the case of random k -SAT instances and thus natural problems to be proved. We state some basic observations and prove two results:

First, random instances of the cyclic ordering and betweenness constraint have a sharp threshold for unsatisfiability. The proof is an application of the threshold criterion due to Friedgut.

Second, random instances of the cyclic ordering constraint are satisfiable with high probability if the number of randomly picked clauses is $< 1 \cdot n$, where n is the number of variables considered.

Keywords: Constraints, random structures, logic

Faster Steiner Tree Computation in Polynomial Space

Fabrizio Grandoni (Università di Roma II)

Given an n -node graph and a subset of k terminal nodes, the **NP**-hard Steiner tree problem is to compute a minimum-size tree which spans the terminals. All the known algorithms for this problem which improve on trivial $O(1.62^n)$ -time enumeration are based on dynamic programming, and require exponential space.

Motivated by the fact that exponential-space algorithms are typically impractical, in this paper we address the problem of designing faster polynomial-space algorithms. Our first contribution is a simple $O(6^k n^{O(\log k)})$ -time polynomial-space algorithm, based on a variant of the classical tree-separator theorem. This improves on trivial $O(n^{k+O(1)})$ enumeration for, roughly, $k \leq n/4$.

Combining the algorithm above (for small k), with an improved branching strategy (for large k), we obtain an $O(1.60^n)$ -time polynomial-space algorithm. The refined branching is based on a charging mechanism which shows that, for large values of k , convenient local configurations of terminals and non-terminals must exist. The analysis of the algorithm relies on the Measure & Conquer approach: the non-standard measure used here is a linear combination of the number of nodes and number of non-terminals.

As a byproduct of our work, we also improve the (exponential-space) time complexity of the problem from $O(1.42^n)$ to $O(1.36^n)$.

Keywords: Steiner tree, separators, measure and conquer

Joint work of: Fomin, Fedor V.; Grandoni, Fabrizio; Kratsch, Dieter

On the Induced Matching Problem

Iyad A. Kanj (DePaul University, Chicago)

We study extremal questions on induced matchings in certain natural graph classes.

We argue that these questions should be asked for twinless graphs, that is graphs not containing two vertices with the same neighborhood. We show that planar twinless graphs always contain an induced matching of size at least $n/40$ while there are planar twinless graphs that do not contain an induced matching of size $(n + 10)/27$. We derive similar results for outerplanar graphs and graphs of bounded genus.

These extremal results can be applied to the area of parameterized computation. For example, we show that the induced matching problem on planar graphs has a kernel of size at most $40k$ that is computable in linear time; this significantly improves the results of Moser and Sikdar (2007). We also show that we can decide in time $O(91^k + n)$ whether a planar graph contains an induced matching of size at least k .

Keywords: Induced matching, planar graphs, outerplanar graphs, kernel, parameterized algorithms

Joint work of: Kanj, Iyad A.; Pelsmajer, Michael J.; Xia, Ge; Schaefer, Marcus

The fast intersection transform with applications to counting paths

Petteri Kaski (University of Helsinki)

We contribute by studying an “intersection transform” of functions defined on subsets of a ground set.

Let U be an n -element set (the ground set), let R be a ring, and denote by 2^U the set of all subsets of U . The intersection transform maps a function $f : 2^U \rightarrow R$ to the function $f\iota : \{0, 1, \dots, n\} \times 2^U \rightarrow R$, defined for all $j = 0, 1, \dots, n$ and $Y \subseteq U$ by $f\iota_j(Y) = \sum_{\substack{X \subseteq U \\ |X \cap Y| = j}} f(X)$. Our interest here is in particular to restrict

(or “trim”) the domains of the input f and the output $f\iota$ from 2^U to given subsets of 2^U . For a subset $\mathcal{F} \subseteq 2^U$, denote by $\downarrow\mathcal{F}$ the *down-closure* of \mathcal{F} , that is, the family of sets consisting of all the sets in \mathcal{F} and their subsets. The notation $O^*(\cdot)$ suppresses a factor polynomial in n .

Theorem 1. *There exists an algorithm that, given $\mathcal{F} \subseteq 2^U$ and $\mathcal{G} \subseteq 2^U$ as input, in time $O^*(|\downarrow\mathcal{F}| + |\downarrow\mathcal{G}|)$ constructs an R -arithmetic circuit with input gates for $f : \mathcal{F} \rightarrow R$ and output gates that evaluate to $f \iota : \{0, 1, \dots, n\} \times \mathcal{G} \rightarrow R$.*

We apply Theorem 1 in the context of counting paths in graphs. Denote by H the entropy function $H(p) = -p \log p - (1-p) \log(1-p)$, $0 \leq p \leq 1$.

Theorem 2. *There exists an algorithm that, given as input (i) a directed graph D with n vertices and bounded integer weights at the edges, (ii) two vertices, s and t , and (iii) a length $\ell = 0, 1, \dots, n-1$, counts, by total weight, the number of paths of length ℓ from s to t in D in time $O^*(\exp(H(\ell/(2n)) \cdot n))$.*

For example, Theorem 2 implies that we can count in $O(1.7548^n)$ time with length $\ell = 0.5n$ and in $O(1.999999999^n)$ time with length $\ell = 0.9999n$. For length $\ell = n-1$ the bound reduces to the classical bound $O^*(2^n)$. [See arXiv:0809.2489 for a preprint.]

Joint work of: Björklund, Andreas; Husfeldt, Thore; Kaski, Petteri; Koivisto, Mikko

Full Paper:

<http://arxiv.org/abs/0809.2489>

Computer Aided Analysis of Independent Set Algorithms

Joachim Kneis (RWTH Aachen)

There are several exact branching algorithms for computing maximum independent sets in graphs. Some of them are very simple while others employ dozens of case distinctions and data reductions. We present a computer aided approach to analyze the running time of such algorithms, especially of very simple algorithms. We generate all possible local neighborhoods of a node with small degree and simulate the algorithms on these neighborhoods. This leads to millions of different cases but also gives better upper bounds on the running time than a similar analysis by hand could yield.

Keywords: Independent Sets, Computer Aided Analysis

Joint work of: Kneis, Joachim; Langer, Alexander; Rossmanith, Peter

Exponential-Time Approximation of Hard Problems

Lukasz Kowalik (University of Warsaw)

We study optimization problems that are neither approximable in polynomial time (at least with a constant factor) nor fixed parameter tractable, under widely believed complexity assumptions. Specifically, we focus on Maximum Independent Set, Vertex Coloring, Set Cover, and Bandwidth.

In recent years, many researchers design exact exponential-time algorithms for these and other hard problems. The goal is getting the time complexity still of order $O(c^n)$, but with the constant c as small as possible. In this work we extend this line of research and we investigate whether the constant c can be made even smaller when one allows constant factor approximation. In fact, we describe a kind of approximation schemes — trade-offs between approximation factor and the time complexity.

We study two natural approaches. The first approach consists of designing a backtracking algorithm with a small search tree. We present one result of that kind: a $(4r - 1)$ -approximation of Bandwidth in time $O^*(2^{n/r})$, for any positive integer r .

The second approach uses general transformations from exponential-time exact algorithms to approximations that are faster but still exponential-time. For example, we show that for any reduction rate r , one can transform any $O^*(c^n)$ -time⁴ algorithm for SET COVER into a $(1 + \ln r)$ -approximation algorithm running in time $O^*(c^{n/r})$. We believe that results of that kind extend the applicability of exact algorithms for NP-hard problems.

Keywords: Exponential-time algorithm, approximation, bandwidth, set cover

Joint work of: Cygan, Marek; Kowalik, Lukasz; Pilipczuk, Marcin; Wykurz, Mateusz

New Bounds for MAX-SAT by Clause Learning

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

To solve a problem on a given CNF formula F a splitting algorithm recursively calls for $F[v]$ and $F[\neg v]$ for a variable v . Obviously, after the first call an algorithm obtains some information on the structure of the formula that can be used in the second call. We use this idea to design new surprisingly simple algorithms for the MAX-SAT problem. Namely, we show that MAX-SAT for formulas with constant clause density can be solved in time c^n , where $c < 2$ is a constant and n is the number of variables, and within polynomial space. We also prove that MAX-2-SAT can be solved in time $2^{m/6}$, where m is the number of clauses. To illustrate the idea we will show in the talk how to solve SAT with constant clause density in time c^n , where $c < 2$ is a constant.

Keywords: MAX-SAT, MAX-2-SAT, splitting algorithm, clause density

The general theory of branching heuristics, and (real) SAT solving

Oliver Kullmann (Swansea University)

⁴ $O^*(f(n))$ notation suppresses polynomial factors

The general methods for upper bounds are rather old, but there might be surprises still awaiting us. I hope to show you some interesting directions.

In the chapter “Fundamentals of Branching Heuristics” of the forthcoming “Handbook of Satisfiability”, the general theory of branching heuristics, based on the notion of “distances” in branching trees, and combining the approach by Knuth (34 years ago) for measuring backtracking trees with the “tau-method” invented in my Diplom thesis (17 years ago), is outlined, both from a theoretical as well as from a practical point of view.

I want to give an overview on this theory and the main open problems and challenges (including practical applications to SAT solving).

Especially the implications for practical SAT solving became clear to me only recently, with the occasion of writing this handbook chapter, and I hope you might find some interest in possibly applying your techniques to “real world” SAT solving.

Keywords: SAT solving, branching heuristics, distances, branching tuples, backtracking trees

Full Paper:

<http://www.swan.ac.uk/compsci/research/reports/2008/index.html>

Exact exponential-time algorithms for $L(2, 1)$ -labeling of graphs

Mathieu Liedloff (Université d'Orleans)

The Frequency assignment problem asks for assigning frequencies to transmitters in a broadcasting network with the aim of avoiding undesired interference.

One of the well elaborated graph theoretical models is the notion of distance constrained labeling of graphs.

Given a graph $G = (V, E)$, an $L(2, 1)$ -labeling of span k is a mapping from V to $\{0, \dots, k\}$ such that :

- any two adjacent vertices are mapped onto integers that are at least 2 apart;
- and
- every two vertices with a common neighbor are mapped onto distinct integers.

An $L(2, 1)$ -labeling of span k is a locally injective homomorphism into the complement of the path of length k . Moreover it is known that for every fixed integer $k \geq 4$, deciding whether a such $L(2, 1)$ -labeling of span k exists is NP-complete.

We first give an exact algorithm for locally injective homomorphisms in time $O^*((\Delta(H) - 1)^n)$.

We derive an $*O((k-2)^n)$ time algorithm for computing, if one exists, an $L(2,1)$ -labeling of span k from this result. This branching algorithm only needs a polynomial space.

Then we give a branch-and-reduce algorithm for deciding the existence of an $L(2,1)$ -labeling of span 4 in time $O(1.3161^n)$. By using a refined running-time analysis based on the so-called Measure-and-Conquer technique, we show that $O(1.3006^n)$ is a worst-case upper bound on its running-time.

Finally we discuss a dynamic programming approach to compute the minimum span of an $L(2,1)$ -labeling in time faster than $O^*(4^n)$. Whereas this algorithm needs an exponential space, we note that its running-time does not depend on k .

Keywords: Moderately exponential-time algorithms, graph labeling problem, $L(2,1)$ -labeling

Joint work of: Havet, Frédéric; Klazar, Martin; Kratochvil, Jan; Kratsch, Dieter; Liedloff, Mathieu

See also: A preliminary version of the paper was presented at MFCS 2007: Jan Kratochvil, Dieter Kratsch, Mathieu Liedloff, Exact algorithms for $L(2,1)$ -labeling of graphs, Proceedings of MFCS 2007, LNCS 4708, pp. 513-524, Springer-Verlag

Low-distortion Embeddings - Graph metrics into the line

Daniel Lokshтанov (University of Bergen)

We revisit the issue of low-distortion embedding of metric spaces into the line from an algorithmic perspective. Let $M = M(G)$ be the shortest path metric of an unweighted graph $G = (V, E)$ on n vertices. We describe two algorithms for the problem of finding a low distortion non-contracting embedding of M into the line.

We give an $O(nd^4(2d+1)^{2d})$ time algorithm that for an unweighted graph metric M and integer d either constructs an embedding of M into the line with distortion at most d , or concludes that no such embedding exists. We find the result surprising, because the considered problem bears a strong resemblance to the notoriously hard Bandwidth Minimization problem which does not admit any FPT algorithm unless an unlikely collapse of parameterized complexity classes occurs.

We give a $O(5^n)$ algorithm for the same problem. This algorithm outperforms our first one in the case that the distortion d is big, that is at least of order $\frac{n}{\log n}$.

Keywords: Distortion, Embeddings, Line, FPT, Exact Algorithms

Joint work of: Fellows, Michael R.; Fomin, Fedor V.; Lokshтанov, Daniel; Losievskaja, Elena; Rosamond, Frances A.; Saurabh, Saket

Dominating Set and other distance problems in non-minor closed graph families

Matthias Mnich (TU Eindhoven)

The Minimum Dominating Set (MDS) problem is W[2]-hard in general graphs, and fixed-parameter tractable in planar graphs and families of graphs excluding a fixed minor. How general can graphs be without losing fixed-parameter tractability of the MDS problem? We consider the MDS problem in two graph families that have unbounded tree-width when parameterized by the size of a minimum dominating set.

The first such family are map graphs, which generalize planar graphs and allow for arbitrarily large cliques. We show that map graphs are not closed under contractions, and that their clique-width is bounded in terms of the size of a minimum dominating set. Then we give a linear kernel for the MDS problem on map graphs, by a region-decomposition approach of the embedded graph that is inspired by the linear kernel for planar graphs.

The second such family are line graphs of graphs with parameter-treewidth property, which include e.g. planar graphs. For this class of graphs we obtain a fixed-parameter algorithm for the MDS problem.

These two results are the beginning of a deeper investigation of how graph operators like “line-graph” can serve as a tool in design of kernels for parameterized problems.

Keywords: Dominating Set, fixed-parameter tractability, tree-width, clique-width

Computational Models with Nontrivial Upperbounds for Satisfiability

Ramamohan Paturi (UC at San Diego, La Jolla)

In this paper, we explore which computational models have nontrivial upperbounds for satisfiability. We show that certain versions of depth-3 unbounded fan-in circuits have nontrivial upperbounds for satisfiability. We also exhibit a connection between $\Pi\Sigma k$ -CNF and k -CNF formulas with regards to the complexity of solving the satisfiability problem.

Joint work of: Paturi, Ramamohan; Impagliazzo, Russell; Calabro, Chris

Computing Minimum Directed Feedback Vertex Set in $O(1.9977^n)$

Igor Razgon (Univ. College Cork)

In this paper we propose an algorithm which, given a directed graph G , finds the minimum directed feedback vertex set (FVS) of G in $O^*(1.9977^n)$ time and polynomial space. To the best of our knowledge, this is the first algorithm computing the minimum directed FVS faster than in $O(2^n)$. The algorithm is based on the branch-and-prune principle. The minimum directed FVS is obtained through computing of the complement, i.e. the maximum induced directed acyclic graph. To evaluate the time complexity, we use the measure-and-conquer strategy according to which the vertices are assigned with weights and the size of the problem is measured in the sum of weights of vertices of the given graph rather than in the number of the vertices.

Keywords: Directed Feedback Vertex Set, Exact Algorithms, Measure and Conquer

Spanning Trees of Bounded Degree Graphs

John Mike Robson (LaBRI, Bordeaux)

We consider lower bounds on the number of spanning trees of connected graphs with degree bounded by d .

The question is of interest because such bounds may improve the analysis of the improvement produced by memorisation in the runtime of exponential algorithms.

The value of interest is the constant β_d such that all connected graphs with degree bounded by d have at least β_d^μ spanning trees where μ is the cyclomatic number or excess of the graph, namely $m - n + 1$.

We conjecture that β_d is achieved by the complete graph K_{d+1} but we have not proved this for any d greater than 2. Instead we give weaker lower bounds on β_d for $d \leq 11$.

First we establish lower bounds on the factor by which the number of spanning trees is multiplied when one new vertex is added to an existing graph so that the new vertex has degree c and the maximum degree of the resulting graph is at most d . In all the cases analysed, this lower bound $f_{c,d}$ is attained when the graph before the addition was a complete graph of order d but we have not proved this in general.

Next we show that, for any cut of size c cutting a graph G of degree bounded by d into two connected components G_1 and G_2 , the number of spanning trees of G is at least the product of this number for G_1 and G_2 multiplied by the same factor $f_{c,d}$.

Finally we examine the process of repeatedly cutting a graph until no edges remain.

The number of spanning trees is at least the product of the multipliers associated with all the cuts. Some obvious constraints on the number of cuts of each size give linear constraints on the normalised numbers of cuts of each size which are then used to lower bound β_d by the solution of a linear program.

The lower bound obtained is significantly improved by imposing a rule that, at each stage, a cut of the minimum available size is chosen and adding some new constraints implied by this rule.

Keywords: Spanning trees, memorisation, cyclomatic number, bounded degree graphs, cut, linear program

A New Algorithm for Finding Trees With Many Leaves

Peter Rossmanith (RWTH Aachen)

We present an algorithm that finds trees with at least k leaves in undirected and directed graphs.

These problems are known as Maximum Leaf Spanning Tree for undirected graphs, and, respectively, Directed Maximum Leaf Out-Tree and Directed Maximum Leaf Spanning Out-Tree in the case of directed graphs.

The run time of our algorithm is $O(\text{poly}(|V|) + 4^k k^2)$ on undirected graphs, and $O(4^k |V| \cdot |E|)$ on directed graphs.

This improves over the previously fastest algorithms for these problems with run times of $O(\text{poly}(|V|) + 6.75^k \text{poly}(k))$ and $2^{O(k \log k)} \text{poly}(|V|)$, respectively.

Keywords: Maximum leaf spanning tree, exact algorithms, efficient algorithms

Joint work of: Kneis, Joachim; Langer, Alexander; Rossmanith, Peter

Exact Algorithms for Counting Subgraphs via Homomorphisms

Saket Saurabh (University of Bergen)

Counting homomorphisms between graphs has applications in variety of areas, including extremal graph theory, properties of graph products, partition functions in statistical physics and property testing of large graphs. In this work we show a new application of counting graph homomorphisms to the area of exact algorithms.

We introduce a generic approach for counting subgraphs in a graph.

The main idea is to relate counting subgraphs to counting graph homomorphisms. This approach provides new algorithms and unifies several well known results in the area of exact algorithms including the recent algorithm of Björklund,

Husfeldt and Koivisto for computing the chromatic polynomial of a given graph, the classical algorithm of Karp for counting hamiltonian cycles, and Ryser’s formula for counting perfect matchings in a bipartite graph. By combining our method with ideas from succinct representation of various data structures and partition functions, we obtain several new results.

In this talk we will present a few new and old results.

Keywords: Subgraph Isomorphism, Graph Homomorphisms, Inclusion-Exclusion, Chromatic Number

Joint work of: Amini, Omid; Fedor, Fomin V.; Saurabh, Saket

Sparse Algebraic Equations over Finite Fields

Igor A. Semaev (University of Bergen)

A system of algebraic equations over a finite field is called sparse if each equation depends on a low number of variables. Finding efficiently solutions to the system is an underlying hard problem in the cryptanalysis of modern ciphers. In this talk we survey the family of Agreeing-Gluing algorithms for solving such equations. In contrast with other known approaches as Gröbner basis methods and SAT-solving heuristic algorithms(e.g. MiniSat), the asymptotic average complexity of which is not rigorous or even not known, rigorous estimates for the average time complexity for some of the Agreeing-Gluing algorithms can be provided. They are much better than conjectural complexity of Gröbner basis methods. In characteristic 2 an exciting difference with the worst case complexity provided by SAT solving methods is observed.

Multiple Right Hand Sides linear equations are based on a more general notion of sparseness related to Linear Algebra. This is a convenient tool for representing cipher equations as modern block ciphers are combinations of sparse nonlinear S-boxes and affine transforms. So a more general Agreeing-Gluing approach is developed to solving them. Experimental results overcome significantly what was previously achieved with Gröbner basis methods.

Keywords: Sparse algebraic equations, finite fields, Agreeing-Gluing algorithms, SAT solving methods, Gröbner basis methods

Polynomial Constraint Satisfaction Problems (PCSP)

Gregory Sorkin (IBM TJ Watson Research Center)

Max 2-Sat, Maximum Independent Set, and Maximum Cut are examples of Max 2-CSP: maximization problems with arbitrary real-valued “score” functions on pairs of variables (binary or otherwise). Generalizing the score domain from reals to formal polynomials (or other rings), and replacing the Max-Sum with a Sum-of-Products, gives a class “Polynomial 2-CSP” (PCSP) that includes 2-CSP, its

counting extension, and many problems not in 2-CSP, such as the minimum bisection of a graph, the partition function of an Ising model, sparsest cut, Max Clique, and Max Ones 2-Sat. Remarkably, PCSP can be solved as efficiently as 2-CSP by all the best algorithms we know, notably: (1) algorithms based on constraint-graph reduction, (2) the polynomial-expected-time specialization of that technique for semi-random CSPs up to the giant-component threshold, (3) dynamic-programming algorithms based on tree decomposition, and (4) the split-and-list matrix-multiplication algorithm of Williams. This gives the first efficient polynomial-space algorithms we know of for graph bisection and the Ising partition function.

Joint work of: Sorkin, Gregory; Scott, Alexander D.

The Time Complexity of Constraint Satisfaction

Patrick Traxler (ETH Zürich)

We present two results about the time complexity of (d, k) -CSP, the problem of deciding satisfiability of a constraint system C with n variables, domain size d , and at most k variables per constraint. Assuming the Exponential Time Hypothesis, two special cases, namely $(d, 2)$ -CSP with bounded variable frequency and d -UNIQUE-CSP, already require exponential time $\Omega(d^{cn})$ for some $c \geq 0$ independent of d and n . UNIQUE-CSP is the special case for which it is guaranteed that every input constraint system has at most 1 satisfying assignment.

Keywords: Constraint Satisfaction, Exponential Time Complexity

Full Paper:

<http://www.springerlink.com/content/agl0m2vq05457w03/>

Treewidth Computation and Extremal Combinatorics

Yngve Villanger (University of Bergen)

For a given graph G and integers $b, f \geq 0$, let S be a subset of vertices of G of size $b + 1$ such that the subgraph of G induced by S is connected and S can be separated from other vertices of G by removing f vertices. We prove that every graph on n vertices contains at most $n \binom{b+f}{b}$ such vertex subsets. This result from extremal combinatorics appears to be very useful in the design of several enumeration and exact algorithms. In particular, we use it to provide algorithms that for a given n -vertex graph G

- compute the treewidth of G in time $O(1.7549^n)$ by making use of exponential space and in time $O(2.6151^n)$ and polynomial space;
- decide in time $O(kn^5 \cdot (\frac{2n+k+1}{3})^{k+1})$ if the treewidth of G is at most k ;

- list all minimal separators of G in time $O(1.6181^n)$ and all potential maximal cliques of G in time $O(1.7549^n)$.

This significantly improves previous algorithms for these problems.

Joint work of: Fomin, Fedor V.; Villanger, Yngve

Full Paper:

<http://www.springerlink.com/content/lr624r224737414g/>

A faster non-parameterized 3-Hitting Set algorithm

Magnus Wahlstrom (MPI für Informatik, Saarbrücken)

We present an improved algorithm for the non-parameterized case of the 3-Hitting Set problem (3HS), with a running time of $O(c^n)$ for $c < 1.6$ (breaking the bound $O(1.6181^n)$ of the $T(n) = T(n-1) + T(n-2)$ recurrence for the first time). Algorithmically, the speedups come from an application of an improved FPT algorithm for sparse cases, with a stronger bound on hitting set size, and from applications of search tuples (i.e. minimality constraints for the search), as used by Chen et al for Vertex Cover (MFCS-2006). Analysis-wise, we count the 2-edges in a new way, combining the 2-clause-counting approach of our previous 3HS algorithm with the matching number-based counting of Zhang (TCS, 1996).

Keywords: 3-Hitting Set, FPT Algorithms, Matching

Joint work of: Wahlstrom, Magnus; Kutzkov, Konstantin

Counting the number of Dominating Sets

Johan van Rooij (Utrecht University)

Inclusion/exclusion and measure and conquer are two techniques that are very popular for the design of exponential time algorithms today. Set cover is a generic problem for both techniques: inclusion/exclusion gives a $O(2^m)$ algorithm for this problem, and measure and conquer was introduced on this problem giving $O(1.2353^{n+m})$.

In this paper we show that a combined approach is possible. We propose a branching algorithm analysed by measure and conquer which has a standard branching rule and a second one inspired by inclusion/exclusion. This will be combined with pathwidth approaches on sparse instances, as has been done in the previous algorithm by Fomin et al. As a result we obtain an algorithm counting the number of set covers of cardinality κ for each $0 \leq \kappa \leq n$ separately in $O(1.2276^{n+m})$. When we apply this to the standard set cover formulation

of dominating set this allows us to count the number of dominating sets of cardinality κ in $O(1.5069^n)$. Compare this to the current fastest algorithm that computes the minimum dominating set and runs in $O(1.5063^n)$.

Our approach with two branching rules has another application. Following a recent result of Gaspers et al. for dominating set on special graph classes, we show that on these special graph classes we can obtain a faster algorithm for the general dominating set problem by exploiting our counting techniques.

Keywords: Dominating Set, Counting, Inclusion-Exclusion, Branching

Joint work of: Bodlaender, Hans L.; Rooij, Johan van; Nederlof, Jesper

08431 Open Problems – Moderately Exponential Time Algorithms

Fedor Fomin, *Subgraph Isomorphism*.

In SUBGRAPH ISOMORPHISM problem we are given two graphs G and F , and the question is to decide if G contains F as a subgraph. There are many important special cases of this problem like HAMILTONIAN CYCLE or BANDWIDTH, that can be solved in time $2^{O(n)}$, where n is the number of vertices in G . However, no such algorithm with such a running time is known for SUBGRAPH ISOMORPHISM. Even the existence of such an algorithm for the special case when the maximum vertex degree of F is at most 3 is open.

Johan van Rooij, *Pathwidth of sparse graphs*.

Many graph problems can be solved in moderately exponential time on graphs of bounded degree. One approach is to create a path decomposition of these graphs and then solve the problem by dynamic programming. For cubic n -vertex graphs Fomin et al. proved that for large enough graphs the pathwidth can be bounded by $\frac{n}{6}$ and for maximum degree four graphs by $\frac{n}{3}$. Recent results by Rossmanith show that a number of problems can be solved in the same exponential time on tree decompositions as on path decompositions.

This leads to the natural question: does there exist similar but stronger bounds on the treewidth of bounded degree graphs for which a tree decomposition can be found in polynomial time? Also, can we derive stronger bounds on the treewidth or pathwidth of bounded degree bipartite graphs?

Johan van Rooij, *Capacitated domination*. There are many NP-hard graph problems that can trivially be solved in $\mathcal{O}(2^n n^{\mathcal{O}(1)})$ by enumerating all vertex subsets, checking for each subset whether it satisfies certain properties in polynomial time, and returning the smallest or largest such subset. Many such problems such as INDEPENDENT SET or DOMINATING SET can actually be solved much faster, while other problems such as CAPACITATED DOMINATING SET seem to be stuck to this 2^n barrier.

In the capacitated domination problem each vertex v is supplied with a number c_v ; this vertex can dominate only at most c_v vertices in its neighbourhood.

It is not surprising that we cannot do better than 2^n for this problem yet (this was given as an open problem at IWPEC 2008) since the polynomial time algorithm verifying that a given vertex subset is a capacitated dominating set involves a flow algorithm or bipartite matching which is more complicated than simple neighbourhood observations as is the case for an independent set or a dominating set.

Johan van Rooij, *Irredundant Set*. Consider the IRREDUNDANT SET problem. An irredundant set can be described in the following way. Consider a number of kings we want to place on the vertices our graph (the irredundant set vertices). A king claims his own vertex and all its neighbours as its own, but a king only has right of existence if he can rule some undisputed vertex of his own. For example, a king has no right of existence if all its neighbouring vertices contain a king, or if has one neighbouring king (which puts his own vertex in dispute) and all other neighbouring vertices also have some neighbour with a king. For positive examples, take any independent set or any inclusion minimal dominating set.

When looking at the 2^n vertex subset problems, the IRREDUNDANT SET problem lies in between both worlds: it can be verified that vertex subset is an irredundant set by only considering its distance two neighbourhood, while we were unable to solve this problem faster than $\mathcal{O}(2^n n^{\mathcal{O}(1)})$. Therefore, we post it as an open problem to compute the upper or lower irredundance numbers of a graph faster than $\mathcal{O}(2^n n^{\mathcal{O}(1)})$: the largest irredundant set or the smallest inclusion maximal irredundant set.

We note that the irredundance numbers are not just any numbers to compute: they have been studied extensively in graph theory before. For example, consider the (by some well known) chain:

$$ir(G) \leq \gamma(G) \leq i(G) \leq \alpha(G) \leq \Gamma(G) \leq IR(G)$$

Where $\alpha(G)$ is the cardinality of a maximum independent set of G , $i(G)$ is the cardinality of a minimum inclusion maximal independent set of G , $\gamma(G)$ is the cardinality of the minimum dominating set of G , $\Gamma(G)$ is the cardinality of a maximum inclusion minimal dominating set of G , and $ir(G)$ and $IR(G)$ correspond to the lower and upper irredundance numbers of G , respectively. Finally, irredundance is the property that makes a dominating set inclusion minimal.

Petteri Kaski, *Counting edge-colorings of the complete graph*. A complete graph K_{2n} always admits a coloring of its edges with colors $\{1, 2, \dots, 2n - 1\}$ so that edges sharing an endvertex have distinct colors.

Question 1. Can one count the number of distinct edge-colorings of K_{2n} in time $2^{o(n^2)}$?

Remark. An algorithm with $O^*(2^{n(n-1)/2})$ running time follows by counting the vertex-colorings of the line graph of K_{2n} with $2n - 1$ colors. See

- A. Björklund, T. Husfeldt, M. Koivisto, Set partitioning via inclusion-exclusion, *SIAM J. Comput.*, to appear.

Question 2. What is the number of edge-colorings for $2n = 16$?

Remark. For $2n = 14$ the number is

$$\begin{aligned} &13! \cdot 98758655816833727741338583040 \\ &= 614972203951464612786852376432607232000. \end{aligned}$$

See

- P. Kaski, P. R. J. Östergård, There are 1,132,835,421,602,062,347 non-isomorphic one-factorizations of K_{14} , *J. Combin. Designs*, to appear. doi: 10.1002/jcd.20188

Petteri Kaski Disjoint triples of subsets. Let U be an n -element set. Denote by $\binom{U}{k}$ the set of all k -subsets of U . Given $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3 \subseteq \binom{U}{k}$ as input, the task is to determine whether there exists a triple $(X_1, X_2, X_3) \in \mathcal{F}_1 \times \mathcal{F}_2 \times \mathcal{F}_3$ with $X_1 \cap X_2 = X_1 \cap X_3 = X_2 \cap X_3 = \emptyset$.

Question. For which values of $1/4 \leq \alpha \leq 1/3$ and $k = \alpha n$ does there exist an algorithm with running time $O((2 - \epsilon_\alpha)^n)$, with $\epsilon_\alpha > 0$ independent of n ?

Remarks. A positive answer for $\alpha = 1/3$ implies an $O((2 - \epsilon)^n)$ algorithm for the Hamilton Cycle/Path problem. For $\alpha < 1/4$ a positive answer is obtained by combining a trimmed fast subset convolution of f_1, f_2 with the fast intersection transform of f_3 , where f_1, f_2, f_3 are indicator functions of $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$. See

- A. Björklund, T. Husfeldt, P. Kaski, M. Koivisto, Fourier meets Möbius: fast subset convolution, Proceedings of the 39th Annual ACM Symposium on Theory of Computing (San Diego, CA, June 11-13, 2007), Association for Computing Machinery, New York, 2007, pp. 67-74;
- A. Björklund, T. Husfeldt, P. Kaski, M. Koivisto, Trimmed Moebius inversion and graphs of bounded degree, Proceedings of the 25th Annual Symposium on Theoretical Aspects of Computer Science (Bordeaux, February 21-23, 2008) (S. Albers and P. Weil, Eds.), IBFI Schloss Dagstuhl, Wadern, Germany, 2008, pp. 85-96;
- A. Björklund, T. Husfeldt, P. Kaski, M. Koivisto, The fast intersection transform with applications to counting paths, arXiv:0809.2489.

Dieter Kratsch, *Number of minimal dominating sets.*

Let $ds(n)$ be the maximum number of minimal dominating sets in a graph on n vertices. It is known that $ds(n) \geq 15^{n/6} \geq 1.5704^n$. Fomin, Grandoni, Pyatkin and Stepanov showed that $ds(n) \leq 1.7159^n$ by means of a moderately exponential-time algorithm enumerating all minimal covers of a set cover instance.

- Determine $ds(n)$. For which value of α is $ds(n) \approx \alpha^n$?

Dieter Kratsch, *Partition into increasing or decreasing subsequences*.

The problem to partition a permutation into the smallest possible number of increasing or decreasing subsequences is known to be NP-hard. When combining two old results on the problem one obtains a subexponential time algorithm (of running time $O(n^{\sqrt{2n}})$) to solve the problem.

- Can you find a faster subexponential time algorithm for the problem?
- Is the problem fixed-parameter tractable when the parameter is the number of increasing or decreasing subsequences in the partition?

Mikko Koivisto, *Reducibility among Problems in 2^n* .

For some extensively studied problems – such as TSP, Graph Coloring, #Hamiltonian Cycles, Permanent—the fastest algorithms currently known require time $2^n \text{poly}(n)$. Show that if one of these problems can be solved in time c^n for some $c < 2$, then also the other problems in "the class" can be solved in time d^n for some $d < 2$.

Daniel Paulusma, *Disconnected Cut*. Let $G = (V, E)$ be a finite, undirected, connected graph without multiple edges and without loops. Let $U \subset V$. Then $G[U]$ denotes the subgraph of G induced by U . We say that U is a *disconnected cut* if both $G[U]$ and $G[V \setminus U]$ are disconnected.

What is the computational complexity of the following problem?

DISCONNECTED CUT

Instance: A graph $G = (V, E)$ (of diameter 2)

Question: Does G have a disconnected cut?

Saying that a graph $G = (V, E)$ has a disconnected cut is equivalent to saying that

- V can be partitioned into four nonempty sets V_1, V_2, V_3, V_4 such that there is no edge $uv \in E$ with $uv \in (V_1 \times V_3) \cup (V_2 \times V_4)$;
- G allows a vertex-surjective homomorphisms to the reflexive four-cycle (a cycle on four vertices with a self-loop in every vertex);
- $\overline{G} = (V, \{uv \mid uv \notin E\})$ allows a spanning subgraph that consists of two *bicliques*, i.e., two nontrivial vertex-disjoint complete bipartite graphs.

Ryan Williams, *Solving k -path in $O^*(2^k)$ time deterministically*.

Can the k -path problem be solved in $O^*(2^k)$ time, deterministically? The approach will probably have to be quite different from the known randomized algorithm, since that uses polynomial identity testing as a key subroutine.

Ryan Williams, *Hybrid algorithm for vertex cover*. A *hybrid algorithm* (cf. Vassilevska-Williams-Woo, SODA'06) is a collection of three algorithms A_1, A_2, A_3 , with the following curious property. A_1 is a polytime algorithm that always returns "approximate" or "exact". A_2 is a polytime approximation algorithm that only works on some inputs. A_3 is an exact (exponential) algorithm that only works on some inputs.

On each instance x of a problem,

- if $A_1(x)$ = “approximate” then $A_2(x)$ approximately solves instance x .
- if $A_1(x)$ = “exact” then $A_3(x)$ exactly solves instance x .

The overall research goal in hybrid algorithms is to find those that beat the worst case inapproximability with A_2 , and get subexponential time with A_3 . For example, there is a hybrid algorithm for Maximum Independent Set for all $\varepsilon > 0$ with the property that if A_2 runs then it outputs an $n^{1-\varepsilon}$ -approximation in polytime, and if A_3 runs then it outputs a maximum independent set in $2^{\varepsilon n}$ time, where ε' decreases as ε decreases. Neither of these two cases are expected to be achievable on all inputs, unless some very surprising things happen. In other words, the set of graphs for which it is hard to approximate Independent Set is a subset of those graphs for which a maximum independent set can be found rather quickly!

In general, hybrid algorithms help us get a better understanding of the relationships between hardness of approximation and hardness of exact solution. The major open problem here is to obtain a hybrid algorithm for Vertex Cover: is there a hybrid algorithm for Vertex Cover which either approximately solves within a $(2 - \varepsilon)$ factor in polynomial time, or exactly solves in $2^{\varepsilon n}$ time, for ε' which decreases as ε decreases? Or, is there some plausible evidence that no such hybrid algorithm exists? (Does ETH fail if the algorithm exists?)

Lukasz Kowalik, *Edge coloring*

In the edge coloring problem, the input is an undirected graph G of n vertices and m edges and the goal is to assign colors to edges so that incident edges get distinct colors. The number of distinct colors used should be as small as possible.

Clearly, one can reduce this problem to a vertex-coloring problem, by making a new graph G' (called *line graph*) with vertices corresponding to edges of G and such that two vertices in G' are adjacent if the relevant edges in G' are incident. Vertex-coloring G' using k colors is equivalent to edge-coloring G using k colors. It follows that we can solve the edge coloring problem in $O(2^m)$ -time and space by the algorithm of Björklund, Husfeldt and Koivisto [FOCS 2006].

On the other hand, there was some work on edge-coloring cubic graphs: Eppstein and Beigel [J. Algorithms 2005] gave an $O(1.415^n)$ -time algorithm and later Kowalik [WG 2006] gave an $O(1.344^n)$ -time algorithm. Both these algorithm use the special properties of the edge coloring problem (in other words, they use the structure of the line graph).

The first open problem is giving an algorithm for a general case that is substantially faster than a current best vertex-coloring algorithm applied to the line graph, in other words an algorithm for general graphs which uses the structure of the line graph.

The second open problem here is the question whether one can solve the (general) edge-coloring problem in $O(c^n)$ time, for some constant c . We believe that such an algorithm does not exist, and the goal is to prove it under some complexity hypothesis (like ETH).

Yoshio Okamoto, *Bicriteria Minimum-Cost Spanning Tree Problem*.

Input: A connected undirected graph $G = (V, E)$, two non-negative edge costs $c_1, c_2: E \rightarrow \mathbb{R}$, and two non-negative real numbers $b_1, b_2 \in \mathbb{R}$.

Output: YES if there exists a spanning tree T of G such that $\sum_{e \in T} c_1(e) \leq b_1$ and $\sum_{e \in T} c_2(e) \leq b_2$; NO otherwise.

Question: Devise an algorithm for the problem above running in $O^*(c^{|E|})$ with $c < 2$.

Remark: The problem itself is known to be NP-complete (via the reduction of the partition problem) [P. Camerini, G. Galbiati, and F. Maffioli. in *Theory of Algorithms*, North-Holland, Amsterdam.] There are a pseudo-polynomial-time algorithm using the idea from Barahona and Pulleyblank [Disc. Appl. Math. 1987], and a polynomial-time approximation scheme by Goemans and Ravi [SWAT 1996] (for the definition of a polynomial-time approximation scheme for bicriteria problems, see their paper). As far as I know, the problem has not been studied in the context of moderately exponential-time algorithms. We only know the trivial algorithm that enumerates all spanning trees of a given graph.

Yoshio Okamoto, *Forest Counting in Graph Classes*

Input: A undirected graph $G = (V, E)$ from a fixed graph class \mathcal{G} .

Output: The number of forests in G . Here, a forest means an edge-subset $F \subseteq E$ that does not embrace any cycle.

Question: Is the problem #P-complete or polynomial-time solvable when \mathcal{G} is the class of cographs? What if \mathcal{G} is the class of unit interval graphs?

Remark: The case of cographs was studied by Giménez, Hliněný, and Noy [SIAM J. Disc. Math. 2006], and they gave an exact algorithm running in $O^*(\exp(|V|^{1/3}))$ time. The case of unit interval graphs was studied by Gebauer and Okamoto [Intern. J. of Foundation of Comp. Sci., to appear], and they gave an exact algorithm running in $O^*(1.9706^{|E|})$ time. They also prove that the problem is #P-complete when \mathcal{G} is the class of chordal graphs.

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2008/1798>