Report from Dagstuhl Seminar 18421

Algorithmic Enumeration: Output-sensitive, Input-Sensitive, Parameterized, Approximative

Edited by

Henning Fernau[1], Petr A. Golovach[2], and Marie-France Sagot[3]

1 **Universität Trier, DE,** fernau@uni-trier.de
2 **University of Bergen, NO,** petr.golovach@ii.uib.no
3 **University Claude Bernard – Lyon, FR,** marie-france.sagot@inria.fr

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 18421 "Algorithmic Enumeration: Output-sensitive, Input-Sensitive, Parameterized, Approximative".

Enumeration problems require to list all wanted objects of the input as, e.g., particular subsets of the vertex or edge set of a given graph or particular satisfying assignments of logical expressions. Enumeration problems arise in a natural way in various fields of Computer Science, as, e.g., Artificial Intelligence and Data Mining, in Natural Sciences Engineering, Social Sciences, and Biology. The main challenge of the area of enumeration problems is that, contrary to decision, optimization and even counting problems, the output length of an enumeration problem is often exponential in the size of the input and cannot be neglected. This makes enumeration problems more challenging than many other types of algorithmic problems and demands development of specific techniques.

The principal goals of our Dagstuhl seminar were to increase the visibility of algorithmic enumeration within (Theoretical) Computer Science and to contribute to establishing it as an area of "Algorithms and Complexity". The seminar brought together researchers within the algorithms community, other fields of Computer Science and Computer Engineering, as well as researchers working on enumeration problems in other application areas, in particular Biology. The aim was to accelerate developments and discus new directions including algorithmic tools and hardness proofs.

Seminar October 14–19, 2018 – http://www.dagstuhl.de/18421
2012 ACM Subject Classification Mathematics of computing → Enumeration, Mathematics of computing → Combinatorial algorithms, Theory of computation → Design and analysis of algorithms, Theory of computation → Complexity classes
Keywords and phrases algorithms, input-sensitive enumeration, output-sensitive enumeration
Digital Object Identifier 10.4230/DagRep.8.10.63
Edited in cooperation with Benjamin Gras (University of Orleans, FR)

1 Executive Summary

Marie-France Sagot (University Claude Bernard – Lyon, FR)
Dieter Kratsch (Université de Lorraine, FR)
Henning Fernau (Universität Trier, DE)
Petr A. Golovach (University of Bergen, NO)

About fifty years ago, NP-completeness became the lens through which Computer Science views computationally hard (decision and optimization) problems. In the last decades various new approaches to solve NP-hard problems exactly have attracted a lot of attention, among them parameterized and exact exponential-time algorithms, typically dealing with decision and optimization problems.

While optimization is ubiquitious in computer science and many application areas, relatively little is known about enumeration within the "Algorithms and Complexity" community. Fortunately there has been important algorithmic research dedicated to enumeration problems in various fields of Computer Science, as, e.g., Artificial Intelligence and Data Mining, in Natural Sciences Engineering and Social Sciences.

Enumeration problems require to list all wanted objects of the input as, e.g., particular subsets of the vertex or edge set of a given graph or particular satisfying assignments of logical expressions. Contrary to decision, optimization and even counting problems, the output length of enumeration problems is often exponential in the size of the input and cannot be neglected. This motivates the classical approach in enumeration, now called output-sensitive, which measures running time in (input and) output length, and asks for output-polynomial algorithms and algorithms of polynomial delay. This approach has been studied since a long time and has produced its own important open questions, among them the question whether the minimal transversals of a hypergraph can be enumerated in output-polynomial time. This longstanding and challenging question has triggered a lot of research. It is open for more than fifty years and most likely the best known open problem in algorithmic enumeration.

Recently as a natural extension on research in exact exponential-time algorithms, a new approach, called input-sensitive, which measures the running time in the input length, has found growing interest. Due to the number of objects to enumerate (in the worst case), the corresponding algorithms have exponential running time. So far branching algorithms are a major tool. Input-sensitive enumeration is strongly related to lower and upper combinatorial bounds on the maximum number of objects to be enumerated for an input of given size. Such bounds can be achieved via input-sensitive enumeration algorithms but also by the use of combinatorial (non-algorithmic) means.

The area of algorithmic enumeration is in a nascent state, though it has a huge potential due to theoretical challenges and practical applications. While output-sensitive enumeration has a long history, input-sensitive enumeration has been initiated only recently. Natural and promising approaches like using parameterized or approximative approaches have not been explored yet in their full capacities.

The principal goals of our Dagstuhl seminar were to increase the visibility of algorithmic enumeration within (Theoretical) Computer Science and to contribute to establishing it as an area of "Algorithms and Complexity". The seminar brought together researchers within the algorithms community, other fields of Computer Science and Computer Engineering, as well as researchers working on enumeration problems in other application areas, in particular, in Biology. Besides the people already working with enumeration, researchers from other fields of Computer Science were invited. In particular, researchers who are interested in Parameterized Complexity and different aspects of counting problems were participating in the seminar. The aim was to accelerate developments and discuss new directions including algorithmic tools and hardness proofs.

The seminar collected 44 participants from 13 countries. The participants presented their recent results in 18 invited and contributed talks. Open problems were discussed in several open problem and discussion sessions.

2 Table of Contents

3 Overview of Talks

3.1 Modular counting of directed Hamiltonian cycles by enumerating solutions to quadratic equations

Andreas Björklund (Lund University, SE)

We show that the number of Hamiltonian cycles in an n-vertex directed graph can be counted modulo 2^k in expected time $O(1.619^n\binom{n}{k})$ for constant k. Our approach is based on an inclusion–exclusion formula over 2^n $n \times n$ matrix determinants. The speedup is obtained by observing that many of the determinants will be zero for the trivial reason of having k or more columns with every element even. We note that the determinants that are not trivially zero in the above sense can be seen as solutions to a special kind of quadratic equation system modulo 2. We show how to list the solutions in expected time $O(1.619^n\binom{n}{k})$.

3.2 Ideal-preferred Enumeration of Minimal Dominating Sets in Graphs

Oscar Defrain (Université Clermont Auvergne – Aubiere, FR)

Joint work of Oscar Defrain, Lhouari Nourine, Takeaki Uno

Due to its equivalence with the two problems of enumerating minimal transversals of a hypergraph, and minimal dominating sets of a graph, the dualization of a Boolean lattice is, in disguise, one of the most studied problem in algorithmic enumeration [1, 2, 3]. Its generalization to distributive lattices, however, is little-understood. In this work, we investigate ideal-preferred enumeration of minimal dominating sets in graphs, toward such a generalization, based on the recent framework of Staworko et al. on preferred consistent query answering in databases. We show that this problem is equivalent to the dualization of a distributive lattice, even when considering various combined restrictions on graphs classes and posets, including bipartite, split and co-bipartite graphs, and variants of neighborhood inclusion posets.

References

1. Fredman, M. L., Khachiyan, L. (1996). On the complexity of dualization of monotone disjunctive normal forms. Journal of Algorithms, 21(3).
2. Staworko, S., Chomicki, J., Marcinkowski, J. (2012). Prioritized repairing and consistent query answering in relational databases. Annals of Mathematics and Artificial Intelligence, 64(2-3), 209–246.
3. Babin, M. A., Kuznetsov, S. O. (2017). Dualization in lattices given by ordered sets of irreducibles. Theoretical Computer Science, 658, 316–326.

3.3 Enumerating Vertices of Covering Polyhedra with Totally Unimodular Constraint Matrices

Khaled M. Elbassioni (Masdar Institute – Abu Dhabi, AE) and Kazuhisa Makino (University of Tokyo, JP)

Main reference Khaled M. Elbassioni, Kazuhisa Makino: "Enumerating Vertices of 0/1-Polyhedra associated with 0/1-Totally Unimodular Matrices", in Proc. of the 16th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2018, June 18-20, 2018, Malmö, Sweden, LIPIcs, Vol. 101, pp. 18:1–18:14, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2018.
URL http://dx.doi.org/10.4230/LIPIcs.SWAT.2018.18

We give an incremental polynomial time algorithm for enumerating the vertices of any polyhedron $P = P(A, 1) = \{x \in \mathbb{R}^n \mid Ax \geq 1,\ x \geq 0\}$, when A is a totally unimodular matrix. Our algorithm is based on decomposing the hypergraph transversal problem for unimodular hypergraphs using Seymour's decomposition of totally unimodular matrices, and may be of independent interest.

3.4 The Minimal Extension of a Partial Solution

Henning Fernau (Universität Trier, DE)

Joint work of Katrin Casel, Henning Fernau, Mehdi Khosravian Ghadikolaei, Jérôme Monnot, Florian Sikora
Main reference Katrin Casel, Henning Fernau, Mehdi Khosravian Ghadikolaei, Jérôme Monnot, Florian Sikora: "On the Complexity of Solution Extension of Optimization Problems", CoRR, Vol. abs/1810.04553, 2018.
URL http://arxiv.org/abs/1810.04553

The very general problem of determining the quality of a given partial solution occurs basically in every algorithmic approach which computes solutions in some sense gradually. Pruning search-trees, proving approximation guarantees or the efficiency of enumeration strategies usually requires a suitable way to decide if a partial solution is a reasonable candidate to pursue. Consider for example the classical concept of minimal dominating sets for graphs. The task of finding a maximum cardinality minimal dominating set (or an approximation of it) as well as enumerating all minimal dominating sets naturally leads to solving the following extension problem: Given a graph $G = (V, E)$ and a vertex set $P \subseteq V$, does there exists a *minimal* dominating set S with $P \subseteq S$.

In an attempt to study the nature of such extension tasks, we propose a general, partial-order based framework to express a broad class of what we refer to as *extension problems*. In essence, we consider optimisation problems in NPO with an additionally specified set of presolutions (including the solutions) and a partial order on those. This partial order $\preccurlyeq$ reflects not only the notion of *extension* but also of *minimality* as follows. For a presolution P and a solution S, S *extends* P if $P \preccurlyeq S$. A solution S is *minimal*, if there exists no solution $S' \neq S$ with $S' \preccurlyeq S$. The resulting extension problem is then formally the task to decide for a given presolution P, if there exits a minimal solution S which extends P.

We consider a number of specific problems which can be expressed in this framework. Possibly contradicting intuition, these problems tend to be NP-hard, even for problems where the underlying optimisation problem can be solved in polynomial time. This raises the question of how fixing a presolution causes this increase in difficulty. In this regard, we study the parameterised complexity of extension problems with respect to parameters related to the presolution. We further discuss relaxation of the extension constraint asking only for a solution S which extends some presolution $P' \preccurlyeq P$. Here we do not want just any such presolution P' but we want P' to be as close to P as possible, in the sense that there exits no presolution $P'' \neq P'$ with $P' \preccurlyeq P'' \preccurlyeq P$ which can also be extended. These considerations yield some insight into the difficult aspects of extension problems.

3.5 Enumeration of Preferred Extensions in Almost Oriented Digraphs

Serge Gaspers (UNSW – Sydney, AU)

Joint work of Serge Gaspers, Rayi Li

In this talk, we present enumeration algorithms to list all preferred extensions of an argumentation framework. This task is equivalent to enumerating all semikernels of a directed graph. For directed graphs on n vertices, all preferred extensions can be enumerated in $O^*(3^{n/3})$ time and there are directed graphs with $\Omega(3^{n/3})$ preferred extensions. We give faster enumeration algorithms for directed graphs with at most $0.8004n$ vertices occurring in 2-cycles. In particular, for oriented graphs one of our algorithms runs in time $O(1.2321^n)$, and we show that there are oriented graphs with $\Omega(3^{n/6}) > \Omega(1.2009^n)$ preferred extensions.

A combination of three algorithms leads to the fastest enumeration times for various proportions of the number of vertices in 2-cycles. The most innovative one is a new 2-stage sampling algorithm, combined with a new parameterized enumeration algorithm, analyzed with a combination of the recent monotone local search technique (STOC 2016) and an extension thereof (ICALP 2017).

3.6 Listing All Maximal k-Plexes in Temporal Graphs

Anne-Sophie Himmel (TU Berlin, DE)

Joint work of Matthias Bentert, Anne-Sophie Himmel, Hendrik Molter, Marco Morik, Rolf Niedermeier, René Saitenmacher
Main reference Matthias Bentert, Anne-Sophie Himmel, Hendrik Molter, Marco Morik, Rolf Niedermeier, René Saitenmacher: "Listing All Maximal k-Plexes in Temporal Graphs" in Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, (ASONAM '18), p.41–46. IEEE, 2018.
URL https://doi.org/10.1109/ASONAM.2018.8508847

Many real-world networks evolve over time, that is, new contacts appear and old contacts may disappear. They can be modeled as temporal graphs where interactions between vertices (in case of social networks these would represent people) are represented by time-stamped edges. One of the most fundamental problems in (social) network analysis is community detection, and one of the most basic primitives to model a community is a clique. Addressing the problem of finding communities in temporal networks, Viard et al. [TCS 2016] introduced Δ-cliques as a natural temporal version of cliques. Himmel et al. [SNAM 2017] showed how to adapt the well-known Bron-Kerbosch algorithm to enumerate Δ-cliques. We continue this work and improve and extend this algorithm to enumerate temporal k-plexes (notably, cliques are the special case $k = 1$).

We define a Δ-k-plex as a set of vertices with a lifetime, where during the lifetime each vertex has an edge to all but at most $k-1$ vertices at least once within any consecutive $\Delta+1$ time steps. We develop a recursive algorithm for enumerating all maximal Δ-k-plexes and perform experiments on real-world social networks that demonstrate the feasibility of our approach. In particular, for the special case of Δ-1-plexes (that is, Δ-cliques), we observe that our algorithm is significantly faster than the previous algorithm by Himmel et al. [SNAM 2017] for enumerating Δ-cliques.

3.7 Constructive and nonconstructive enumeration of designs

Petteri Kaski (Aalto University, FI)

This talk looks at enumeration problems in the study of combinatorial designs where the enumeration is to be carried out up to isomorphism given by a group action. We look at two examples, namely one-factorizations of the complete graph and Latin squares, enumerated up to isomorphism by the sequences A000474 and A003090, respectively, in the Online Encyclopedia of Integer Sequences. A key technique used in both cases is McKay's method of canonical extensions [1].

References

1 Brendan D. McKay, Isomorph-free exhaustive generation, J. Algorithms 26 (1998), no. 2, 306–324.

3.8 Node Similarity with q-Grams for Real-World Labeled Networks

Andrea Marino (University of Pisa, IT)

Joint work of Conte, Alessio; Ferraro, Gaspare; Grossi, Roberto; Sadakane, Kunihiko; Uno, Takeaki
Main reference Alessio Conte, Gaspare Ferraro, Roberto Grossi, Andrea Marino, Kunihiko Sadakane, Takeaki Uno: "Node Similarity with q -Grams for Real-World Labeled Networks", in Proc. of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018, pp. 1282–1291, ACM, 2018.
URL http://dx.doi.org/10.1145/3219819.3220085

We study node similarity in labeled networks, using the label sequences found in paths of bounded length q leading to the nodes. (This recalls the q-grams employed in document resemblance, based on the Jaccard distance.) When applied to networks, the challenge is two-fold: the number of q-grams generated from labeled paths grows exponentially with q, and their frequency should be taken into account: this leads to a variation of the Jaccard index known as Bray-Curtis index for multisets. We describe NSIMGRAM, a suite of fast algorithms for node similarity with q-grams, based on a novel blend of color coding, probabilistic counting, sketches, and string algorithms, where the universe of elements to sample is exponential. In particular, after a preprocessing node coloring phase, our method estimates the similarity between two nodes x and y, sampling colorful q-paths, i.e. q-paths containing different colors ending in x and y, and using the Bray-Curtis index for the corresponding multisets of q-grams. We provide experimental evidence that our measure is effective and our running times scale to deal with large real-world networks. One lesson learned is that, when dealing with network analytic, in order to avoid to do statistics on an exponential number of solutions, sometimes we can successfully focus just on special solutions, e.g. colorful solutions, or consider just part of them by applying sampling.

3.9 On Solving Enumeration Problems with SAT Oracles

João Marques-Silva (University of Lisbon, PT)

Joint work of Mark H. Liffiton, Alessandro Previti, Ammar Malik, João Marques-Silva
Main reference Mark H. Liffiton, Alessandro Previti, Ammar Malik, João Marques-Silva: "Fast, flexible MUS enumeration", Constraints, Vol. 21(2), pp. 223–250, 2016.
URL http://dx.doi.org/10.1007/s10601-015-9183-0

Enumeration is often necessary in the context of reasoning about computationally hard problems, including NP-complete, PSPACE-complete, or harder. One example is enumeration of solutions. However, there are many other enumeration problems, that find important practical applications. One example is the enumeration of diagnoses or explanations of overconstrained systems. This talk overviews some of these enumeration problems, and reports on the progress achieved in recent years.

3.10 Minimal Connected Dominating Sets below 2^n

Michal Pilipczuk (University of Warsaw, PL)

Joint work of Daniel Lokshtanov, Michal Pilipczuk, Saket Saurabh
Main reference Daniel Lokshtanov, Michal Pilipczuk, Saket Saurabh: "Below all subsets for Minimal Connected Dominating Set", CoRR, Vol. abs/1611.00840, 2016.
URL http://arxiv.org/abs/1611.00840

We prove that the number of minimal connected dominating sets in an n-vertex graph is always bounded by $(2-\varepsilon)^n$, for some $\varepsilon > 0$, and moreover they can be enumerated in time $O((2-\varepsilon)^n)$. The proof relies on a fine understanding of combinatorial properties of minimal connected dominating sets and a probabilistic argument, which shows that if a graph is "robustly dense", then the probability that a random subset of its vertices is a minimal connected dominating set is exponentially small.

3.11 Enumerating minimal dominating sets in triangle-free graphs

Jean-Florent Raymond (TU Berlin, DE)

Joint work of Marthe Bonamy, Oscar Defrain, Marc Heinrich, Jean-Florent Raymond
Main reference Marthe Bonamy, Oscar Defrain, Marc Heinrich, Jean-Florent Raymond: "Enumerating minimal dominating sets in triangle-free graphs", CoRR, Vol. abs/1810.00789, 2018.
URL http://arxiv.org/abs/1810.00789

It is a long-standing open problem whether the minimal dominating sets of a graph can be enumerated in output-polynomial time. In this talk I will present the following results obtained jointly with Oscar Defrain, Marthe Bonamy, and Marc Heinrich:

- the enumeration of minimal dominating sets in triangle-free graphs can be performed in output-polynomial time;
- deciding if a set of vertices of a bipartite graph can be completed into a minimal dominating set is a NP-complete problem.

3.12 The Maximum Number of Minimal Dominating Sets in a Tree

Günter Rote (FU Berlin, DE)

Main reference Günter Rote: "The Maximum Number of Minimal Dominating Sets in a Tree", in Proc. of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019, pp. 1201–1214, SIAM, 2019.
URL https://doi.org/10.1137/1.9781611975482.73

A tree with n vertices has at most $95^{n/13}$ minimal dominating sets. The corresponding growth constant $95^{1/13} \approx 1.4194908$ is best possible.

I show how these results are obtained in a semi-automatic way with computer help, starting from the dynamic-programming recursion for computing the number of minimal dominating sets of a given tree. This recursion defines a bilinear operation on sixtuples, and the growth constant arises as a kind of "dominant eigenvalue" of this operation.

We also derive an output-sensitive algorithm for listing all minimal dominating sets with linear set-up time and linear delay between reporting successive solutions. It is open whether the delay can be reduced to a constant delay, for an appropriate modification of the problem statement.

3.13 Efficiently Enumerating Hitting Sets of Hypergraphs Arising in Data Profiling

Martin Schirneck (Hasso-Plattner-Institut – Potsdam, DE)

Joint work of Thomas Bläsius, Tobias Friedrich, Julius Lischeid, Kitty Meeks, Martin Schirneck

A reoccurring task in the design and profiling of relational data is the discovery of hidden dependencies between attributes, e.g., unique column combinations. Enumerating them is equivalent to the classical transversal hypergraph problem. We present a backtracking algorithm for the enumeration of inclusion-wise minimal hitting sets that achieves polynomial delay on hypergraphs for which the size of the largest minimal transversal is bounded.

This algorithm solves the extension problem for minimal hitting sets as a subroutine. The extension problem is known to be NP-complete. We show that it also remains hard in the parameterised sense. In fact, it is one of only a few natural W[3]-complete problems when parameterised by the size of the set to be extended. Despite the hardness results, we show that a careful implementation of the extension oracle can help avoiding the worst case on hypergraphs arising in the profiling of real-world databases, leading to significant speed-ups in practice.

This work is to appear at the 2019 Meeting on Algorithm Engineering and Experiments (ALENEX).

3.14 A panorama of enumeration complexity

Yann Strozecki (University of Versailles, FR)

Joint work of Yann Strozecki, Arnaud Mary, Florent Capelli
Main reference Florent Capelli, Yann Strozecki: "Incremental delay enumeration: Space and time", Discrete Applied Mathematics, 2018.
URL https://doi.org/10.1016/j.dam.2018.06.038

We review different complexity measures for enumeration algorithms: total time, incremental time and delay. To each of these measures, we associate one ore several complexity classes. We show how those classes relate to classical decision or search problems and how they relate together. In particular, we prove strict inclusions modulo complexity hypotheses such as ETH or TFNP≠NP. We then present a framework of saturation problems designed to investigate the limit between incremental polynomial time and polynomial delay. It allows to prove that a large number of natural saturation problems are solvable in polynomial delay and capture interesting problems such as the enumerations of maximal cliques in a graph or of the circuits in a matroid representable over a finite field. We also present low complexity classes with polynomial time precomputation and related algorithms: constant delay (Gray codes, amortization or queries over simple structures) and polynomial delay in the size of a single solution.

3.15 How to use the Solutions of Enumeration in Practice

Takeaki Uno (National Institute of Informatics – Tokyo, JP)

There are many applications of optimizations and enumerations in practice. However, users are often, always, not satisfied with the solution given by those algorithms. For the optimization, there is no alternatives even though they have several implicit unwritten rules, and for enumeration that is the given huge number of solutions. Actually, the users usually want to choose something well, thus want to understand the problems and data easily, what is important, and what solutions are possible. In this sense, users want to be given good explanations of problems and solutions, in an understandable and abstracted ways. For this sake, we propose three ways; one is abstract the solutions by clustering the solutions, one is decomposing the problem/solutions into parts composing the solutions, and one is to extract important vertices and edges of the solutions according to the distribution of the solutions to the enumeration.

3.16 A Complexity Theory for Hard Enumeration Problems

Heribert Vollmer (Leibniz Universität Hannover, DE)

Joint work of Nadia Creignou, Markus Kröll, Reinhard Pichler, Sebastian Skritek, Heribert Vollmer
Main reference Nadia Creignou, Markus Kröll, Reinhard Pichler, Sebastian Skritek, Heribert Vollmer: "On the Complexity of Hard Enumeration Problems", CoRR, Vol. abs/1610.05493, 2016.
URL http://arxiv.org/abs/1610.05493

We introduce a hierarchy complexity classes and reductions for enumeration problems. The hierarchy resembles the polynomial-time hierarchy for decision problems. We provide tools to prove membership in the hierarchy, e.g., based on self-reducibility. Furthermore, we introduce a new reduction among enumeration problems, based on oracle computations, i.e., a Turing-type reduction. We prove a basic "Completeness Theorem", providing an easy to obtain hardness for classes in our hierarchy. We demonstrate the applicability of our approach by providing a number of completeness results for well-studied enumeration problems from the areas of propositional logic, artificial intelligence, and database theory.

4 Open problems

4.1 The DIVERSE X Paradigm

Michael R. Fellows (University of Bergen, NO) and Frances Rosamond (University of Bergen, NO)

4.1.1 Introduction

Diverse X refers to a parameterized problem such as the following (being somewhat informal, to begin with):

DIVERSE VERTEX COVERS

Input: A graph G on n vertices, and positive integers k, r and T.

Parameter: (k, r)

Output: A set S of r different vertex covers V_i of G, each of size at most k, such that a diversity measurement for the set of solutions S is at least T.

To summarize the role of these different numbers: k measures the quality of the solutions (the vertex covers in S). A vertex cover of a graph is good when it is small; r measures the size of the sample S of these high quality solutions; T measures the diversity of the sample. For example, we might take the diversity measurement to be the sum of the Hamming distances between the pairs of n-length 0/1 indicator vectors, where the sum is taken over all possible pairs V_i and V_j of the solutions in S.

This is clearly a very general paradigm for problem parameterization. The problem defined above is a variation on the classic NP-hard decision problem X = VERTEX COVER.

Theorem. DIVERSE VERTEX COVERS is FPT.

4.1.2 Motivation for the Paradigm

A motivation to look at this parameterization has been developing for a long time, since some conversations in the early 1990s with biologists, in the collaborative setting of computational biology.[1]

Biologists typically do not want just *one* high quality solution to a combinatorial optimization problem relevant to their investigations. For example, they don't want just one high quality multiple sequence alignment. Because they have all of this other information. Call this *side information* which is not included in the simple combinatorial model. Given two sequence alignments of equally good score, they might prefer one of them, on the basis of side information, for further investigation.[2]

Biologists typically also do not want *all* of the solutions of quality k. There are usually too many to deal with.

A moment's reflection makes clear that this is the natural situation now in all areas of science and information systems. There is more and more data coming from all different directions. Relevant side information is *the rule* in practical computing, not the exception.

This direction of investigation is truly fundamental and profound!

The traditional approach in computational complexity, focusing on the existence (or construction) of a single high-quality solution, lending to optimality issues (in the worst-case asymptotic framework), is therefore almost always nonsense!

This generality and significance deserves a brave and colorful name: *The Second Main Heresy.*

In applied settings, they generally don't care about optimality. They want a diverse sample of high quality solutions, to which they can apply side information in picking one to work with.

In order to get an r-sized sample of quality k, and high diversity, you might have to move away from optimality. So this is sort of like approximation, but different. There will likely be interesting tradeoffs motivated by practical considerations and the properties of real datasets, and the playground for metatheorems should be rich.

4.1.3 What is a Diverse Collection of High Quality Solutions Good For?

Beyond the basic vanilla issue raised in the above section, one can point to several established areas, and speculative possibilities.

Concorde-style memetic heuristics. The Concorde heuristic for TSP patented by Paul Seymour and Bill Cook, when Paul was at Bell Labs, maintains a (smallish) population of high quality solutions, and alternates between two phases. In one phase, each solution in the population is improved (hopefully) by a local search heuristic, applied separately to each solution in the population. In the second phase (termed *recombination*) a small number of

---

1 The conversations were with Ben Koop and/or Chris Upton. Why did it take so long for the DIVERSE X research direction to gel? The conversations were in the early days of parameterized complexity, and my recollection is that we were already taking so much heat for the very idea of a relevant secondary measurement apart from the input size — that while they had an undeniably good point, our plate was already full of trouble!

2 It can even happen that the crucial side information is *secret*, as in an amusing anecdote about a scheduling problem that I heard about from Karsten Weihe at a Dagstuhl Workshop around 1998.

tours (like ten or so) from the population (randomly chosen), are combined by taking the union of the edges involved in these ten tours.

It turns out, empirically, for many applications of the TSP problem, the union of ten high quality tours tends to be a graph of treewidth less than 15, and on such a graph, TSP can be solved exactly and optimally (essentially by an FPT dynamic programming algorithm where the treewidth parameter is 15). The solution found will be at least as good as any of the tours in the union. This is then added to the population (which is maintained at a constant size by some pruning of less efficient solutions). These two phases are repeated some number of times.

Prominence in data – training sets for machine learning, and uses in data visualization. The work of Gunnar Carlsson and Harlan Sexton.

Maximizing relative prominence in the setting of abstract simplicial complexes is the same as sum of pairwise Hamming distances.

Speculative Uses: Heuristic Kernelization. If you have a highly diverse r-sized sample S of good solutions (k-small) for VERTEX COVER, then the vertices v that:

(i) Occur in many of the solutions in S, should perhaps be inducted into the solution: $G' = G - v$ and $k' = k - 1$. (ii) Occur in none of solutions in S, should perhaps be banned from a solution. $G' = G - -N[v]$ and $k' = k - deg(v)$.

Either of these simplifies the instance and is a heuristic kernelization.

Speculative Uses: Evidentiary Interpretation of Universal Quantification. Suppose we are interested in the graph property involving alternating quantification:

For every independent set J in G of size k, there exists a set J' of size k such tha $J \cup J'$ is a dominating set in G.

Considered classically, this is likely hard for Π_2^P, but if we could find a *very diverse* collection of k-independent sets, for each J_i of which, there is a J_i' such that $J_i \cup J_i'$ is a dominating set, then we might regard this as evidence that the graph has the property. Thus using diversity to interpret Π_2^P into NP in a heuristic, evidentiary manner.

This could potentially be quite interesting, as most of mathematical thought involves at most two, sometimes three, very rarely four alternating quantifications. It is probably the same with cognition in general.

This could also be interesting in the context of the Szeider-deHaan program.

4.1.4 Some Concrete Open Problems

Classical P-time Self-reducibility. DIVERSE X, generally speaking, is a search problem. In the setting of classical complexity (P *vs* NP and all that), search problems are sort of swept under the rug (see the discussion in the Garey and Johnson book). It is generally easy to work out a search algorithm given an oracle for the decision problem.[3]

Given an oracle for the decision problem ("Does G have a dominating set of size k?") is there (or, when is there) a P-time oracle algorithm that constructively produces a set S of r different solutions, of diversity measure at least T?

[3] It should be more widely understood that this is a *skeleton in the closet* of classical complexity. The problem is in defining what one means by *P-time self-reducibility*, the usual name for this passage from decision to construction. Formally, decision problems are defined by a formal language, but the notion of a solution requires some sort of relational framework. There is no generally agreed-upon definition of P-time self-reducibility. An attempt to frame things in relational terms was made in [1]. This might be relevant to the DIVERSE X program.

What about FPT self-reducibility for DIVERSE X? For various X and various diversity measures.

Dependence of Positive Results on Properties of the Diversity Measure. Can we give meta-theorems where if the diversity measure has certain properties, then positive results hold?

References

1 K. Abrahamson, M. Fellows, M. Langston and B. Moret. "Constructive Complexity." *Discrete Appl. Math.* 34 (1991), 3–16.

4.2 Various enumeration perspectives

Henning Fernau (University of Trier, Germany)

4.2.1 Planar graphs

- Planarity often gives opportunities for improved algorithms in decision or optimization problems: square root phenomenon, PTAS, ...
- Many worst-case examples in enumeration are planar. No hope for improvements here?
- Enumeration problems involving connectivity are notoriously hard.
  What about considering such problems on planar graphs?
  Intuition: There are not too many ways to interconnect planar graphs ...

4.2.2 Enumerating representative solutions

- One of the motivations for enumeration:
  User do not only want to see one optimal solution, because there might be ill-formalized aspects of the problem that require to also look at seemingly non-optimal solutions.
- Yet, users do not want to look at zillions of solutions.
- What about presenting a selection of k solutions that are far apart from each other, kind of being a representative selection of the whole space of solutions?

4.2.3 Enumeration beyond graphs

- Typically, the enumeration community focuses on problems concerning graphs or (sometimes) logic. What about other areas?
- To give some concrete example and also to indicate the different character of such problems, we suggest looking at *synchronizing words*, a notion related to the possibly most famous open combinatorial question in Formal Languages, the so-called Černý Conjecture.
- A word $x \in \Sigma^*$ is called *synchronizing* for a DFA A, $A = (S, \Sigma, \delta, s_0, F)$, if there is a state s_f, such that for all states s, $\delta^*(s, x) = s_f$.
- Deciding, given (A, k), if DFA A has synchronizing word of length at most k is NP-complete [1].
- For the purpose of enumeration, in order to define a good notion of minimality, several natural options exist, depending on the chosen partial ordering on the set of words. The most common are: the prefix, suffix, subword (infix), subsequence (scattered subword), lexicographic, or length-lexicographic orderings.

- In the meantime, we (together with S. Hoffmann) could at least determine the status of most of the extension problems that can be associated to these enumeration problem. More precisely, given A and a word u, the question if there is some synchronizing word w that is larger than u (in the considered ordering) and minimal among the synchronizing words can be solved in polynomial time for the prefix, suffix and lexicographic orderings but is NP-hard for the subsequence or length-lexicographic orderings. The status with respect to the subword ordering is unkonwn.

References

1 D. Eppstein. Reset sequences for monotonic automata. *SIAM Journal on Computing*, 19(3):500–510, 1990.

4.3 Enumeration of tree width-t-modulators

Fedor V. Fomin (University of Bergen, NO) and Saket Saurabh (Institute of Mathematical Sciences – Chennai, IN)

Let $t \geq 1$ be a fixed integer. Given a graph G, a set $S \subseteq V(G)$ is called *treewidth-t-modulator* if $G - S$ has treewidth at most t. Furthermore if $G[S]$ is connected it is called *connected treewidth-t-modulator* Observe that for $t = 1$, treewidth-1-modulator and connected treewidth-1-modulator are known as *feedback vertex set* and *connected feedback vertex set*, respectively. It is known that the number of minimal feedback vertex sets on a graph on n vertices is at most $\mathcal{O}(1.8527^n)$ [1].

1. Show that there exist a constant c such that any graph on n-vertices has at most

$$\left(2 - \frac{1}{c}\right)^n n^{\mathcal{O}(1)}$$

minimal connected feedback vertex sets.
Potential approach seems to be via methods given in [2] and [3].

2. Show that for evert $t \geq 2$, there exist a constant c_t such that any graph on n-vertices has at most

$$\left(2 - \frac{1}{c_t}\right)^n n^{\mathcal{O}(1)}$$

minimal treewidth-2-modulator sets. Even for $t = 2$, this is open. One could also study this problem when the objective is to upper bound the number of minimal vertex sets, whose deletion results in an outer-planar graph.
Potential approach seems to be via methods given in [1].

References

1 Serge Gaspers, Edward J. Lee: Exact Algorithms via Multivariate Subroutines. ICALP 2017: 69:1–69:13
2 Daniel Lokshtanov, Michal Pilipczuk, Saket Saurabh: Below all subsets for Minimal Connected Dominating Set. CoRR abs/1611.00840 (2016)
3 Fedor V. Fomin, Serge Gaspers, Artem V. Pyatkin, Igor Razgon: On the Minimum Feedback Vertex Set Problem: Exact and Enumeration Algorithms. Algorithmica 52(2): 293–307 (2008)

4.4 Minimal separators in graphs

Serge Gaspers (UNSW – Sydney, AU)

It is known that every graph on n vertices has $O(1.6181^n)$ minimal separators [2]. However, it is unknown whether this bound is tight. In 2008, a family of graphs with $\Omega(3^{n/3}) = \Omega(1.4422^n)$ minimal separators was shown [1], and this lower bound has recently been improved to $\Omega(1.4457^n)$ [3]]. The question is to improve either the upper bound or the lower bound.

References

1 Fedor V. Fomin, Dieter Kratsch, Ioan Todinca, Yngve Villanger: Exact Algorithms for Treewidth and Minimum Fill-In. SIAM J. Comput. 38(3): 1058–1079, 2008.
2 F. V. Fomin and Y. Villanger, Treewidth computation and extremal combinatorics, Combinatorica 32(3), 289–308, 2012.
3 Serge Gaspers and Simon Mackenzie. On the Number of Minimal Separators in Graphs. Journal of Graph Theory 87(4): 653–659, 2018.

4.5 Enumeration of vertex subsets with non-local properties

Petr A. Golovach (University of Bergen, NO)

The most standard approach for the input-sensitive enumeration of vertex subsets of a graph satisfying a given property Π is using the recursive branching algorithms (see the book [1]). Still, this approach has limitations. In particular, the technique works well for properties that are local in some sense. For example, a set of vertices X is an inclusion-maximal independent set of a graph G if and only if $|X \cap N[v]| = 1$ for every $v \in V(G)$ and this property is crucial for for the recursive branching algorithm that enumerates maximal independent sets of an n-vertex graph in time $\mathcal{O}^*(3^{n/3})$. It looks that for non-local properties, we have to develop new techniques. The following specific problems could be interesting as starting points.

1. A set $D \subseteq V(G)$ is a *connected dominating set* of G if D is a dominating set and $G[D]$ is connected. Lokshtanov, Pilipczuk and Saurabh [2] proved that all inclusion-minimal connected dominating sets of an n-vertex graph can be enumerated in time $\mathcal{O}(2^{(1-\varepsilon)n})$ where $\epsilon < 10^{-50}$ slightly improving upon the trivial $\mathcal{O}^*(2^n)$ algorithm. Is it possible to improve this results?
2. A set $D \subseteq V(G)$ is *irredundant* if for every $u \in D$, there is $v \in N[u]$ (called a *private vertex*) such that $v \notin N[w]$ for every $w \in D \setminus \{u\}$. Is it possible to enumerate all inclusion-maximal irredundant sets of an n-vertex graph faster that $\mathcal{O}^*(2^n)$?

References

1 Fedor V. Fomin, Dieter Kratsch: Exact Exponential Algorithms. Springer 2010
2 Daniel Lokshtanov, Michal Pilipczuk, Saket Saurabh: Below all subsets for Minimal Connected Dominating Set. CoRR abs/1611.00840 (2016)

4.6 Enumeration of minimal dominating sets for graph classes

Dieter Kratsch (Université de Lorraine, FR)

The number of minimal dominating sets of an n-vertex graphs has generated plenty of research in the last 15 years. In fact even in 1985 all known was that it could not be more than order of $2^n/\log n$.

Due to groundbreaking work in input-sensitive algorithms by Fomin, Grandoni, Pyatkin and Stepanov (2008) we know that there is an upper bound of 1.7159^n. This one is achieved by an intricate measure and conquer analysis of the branching algorithm enumerating all minimal dominating sets of the input graph. The best known lower bound for general n-vertex graphs is $15^{n/6}$, approximately 1.5704^n, is folklore (see e.g. the master thesis of Serge Gaspers in 2005, see also the above mentioned paper by Fomin et al). Hence there is a huge gap between upper and lower bound.

- Any even small improvement upon the lower or upper bound for graphs in general would be highly appreciated and may lead to improvements on various similar questions (minimal feedback vertex sets, minimal separators, minimal connected dominating sets, minimal subset feedback vertex sets, etc.).

A natural approach to achieve matching upper and lower bounds is to restrict lower and upper bounds to some particular class of graphs. Starting with a paper by Couturier Heggernes, Kratsch, van't Hoff (2013) matching upper and lower bounds for the number of minimal dominating sets in n-vertex graphs of a certain graph class have been obtained. It is worth mentioning that in almost all of these cases the upper bound has been obtained by the use of structural properties of the graph class (and without complicated measure-and-conquer analysis).

- An interesting class that despite various efforts still has non matching upper and lower bounds are the chordal graphs.
- Nothing has been published on bipartite graphs.
- Possibly the most challenging case are permutation graphs. For at least a year we tried to prove that $15^{n/6}$, approximately 1.5705^n, is a matching upper bound. However we never succeeded to fix all the holes in our proofs. Thus we give up on this one some years ago. Now with some new ideas we conjecture that 1.5705^n is an upper bound for permutation graphs and it might even be an upper bound for cocomparability graphs.

4.7 Enumeration of k-colorable induced subgraphs

Daniel Lokshtanov (University of California – Santa Barbara, US)

Question 1: Is the following statement true? For every integer $k > 0$ there exists reals $c_k < 2$ and $a_k > 0$ such that for every graph G on n vertices the number of inclusion maximal k-colorable induced subgraphs of G is at most $a_k \cdot c_k^n$.

Question 2: For a graph G and integer k, define the family $F_k(G)$ to contain all vertex sets S that are the union of k (not necessarily distinct) maximal independent sets of G. Is the following statement true? For every integer $k > 0$ there exists reals $c_k < 2$ and $a_k > 0$ such that for every graph G on n vertices $|F_k(G)| \leq a_k \cdot c_k^n$?

Remarks: A positive answer to question 2 implies a positive answer to question 1, but not necessarily vice versa. Question 2 was posed by Boris Bukh at ICGT 2018.

4.8 Primal and Dual Representations

Kazuhisa Makino (University of Tokyo, JP)

Enumeration is one of the fundamental topics of discrete mathematics. Recently, enumerar-ation (or generation) algorithms have recently been acheived much attention in computer science. In this note, we describe three famous problems, which are *not* known to be solved in output-polynomial (or polynomial total) time. They can be viewed as the decision problems on *primal and dual representations* for polytopes, monotone Boolean functions, and Horn Boolean functions.

Problem 1
Input: A set of points $S \in \mathbb{R}^n$ and a set of inequalities H in $\mathbb{R}^n$.
Question: Is the convex hull of S represented by H ?

Problem 2
Input: A monotone DNF φ and a monotone CNF ψ.
Question: Is φ logically equivalent to ψ ?

Clearly they belong to co-NP. As for Problem 1, one of the natural representations of polytope P is the set of extreme points in P, and the other one is the set of facets of P. It is not known whether Problem 1 can be solved in polynomial time, or it is co-NP-compelte. It is known that Problem 1 can be solved in polynomial time then the corresponding generation problems such as (1-1) given a set of point S, generating all facets of the polytope defined by S, or (1-2) given a set of inequalites H (that defines a polytope), generate all the extreme points of the polytope defined by H, can be solved in polynomial total time, On the other hand, if it is co-NP-complete, the correponding generation problems have no polynomial total time algorithm, unless P=NP.

As for Problem 2, monotone Boolean functions have two natural representations: monotone DNFs φ and monotone CNFs ψ. It is known that Problem 2 can be solved in quasi-polynomial time. However, it is still open whether Problem 2 can be solved in polynomial time, or it is co-NP-compelte, where most experts in complexity theory believe that Problem 2 is not co-NP-complete. Similarly to Problem 1, Problem 2 can be solved in polynomial time if and only if the corresponding generation problems can be solved in polynomial total time.

Let us consider primal and dual representations of Horn Boolean function $f : \{0,1\}^n \to \{0,1\}$. A CNF is called Horn if each clause contains at most one positive literal, and a Boolean function is called *Horn* if it can be represented by a Horn CNF. It is well-known that a Boolean function f is Horn if and only if its false set is closed under intersection, i.e., for any pair of vectors v and w in $\{0,1\}^n$ such that $f(v) = f(w) = 1$, we have $f(v \wedge w) = 1$. Here for vectors v and w in $\{0,1\}^n$, $v \wedge w$ denotes the vector in $\{0,1\}^n$ defined by $(v \wedge w)_i = v_i \wedge w_i$

for all $i = 1, \ldots, n$. Thus Horn functions have alternative representation. For a set S of vectors in $\{0,1\}^n$, the intersection closure $Cl_\wedge(S)$ is the set defined by

$$Cl_\wedge(S) = \{ \bigwedge_{v \in W} v \mid W \subseteq S\}.$$

It is not difficult to see that any set T of vectors in $\{0,1\}^n$ that is closed under intersection has a unique (inclusion-)minimal set S such that $T = Cl_\wedge(S)$. Such a set is called *characteristic set* of T.

Problem 3
Input: A set of vectors $S \in \{0,1\}^n$ and a Horn CNF ψ
Question: Is $Cl_\wedge(S)$ a set of true vectors of ψ ?

Problem 3 is polynomially equivalent to the problem that decides whether given characteristifc set and Horn CNF represent the same Horn Boolean function. Again the complexity status of Problem 3 is still open, although Problem 3 is polynomially equivalent to Problem 2 if ψ in Problem 3 contains all prime implicates, which implies that Problem 3 is at least as difficult as Problem 2. Moreover, Problem 3 can be solved in polynomial time if and only if the corresponding generation problems can be solved in polynomial total time.

References

1 Thomas Eiter, Kazuhisa Makino, Georg Gottlob: Computational aspects of monotone dualization: A brief survey. Discrete Applied Mathematics 156: 2035–2049 (2008)

2 L. Lovasz. Combinatorial optimization: some problems and trends. DIMACS Technical Report 92-53, Rutgers University, 1992.

3 Roni Khardon: Translating between Horn Representations and their Characteristic Models. J. Artif. Intell. Res. 3: 349–372 (1995)

4.9 Minimal dominating sets enumeration and hypergraph colorings

Michal Pilipczuk (University of Warsaw, PL)

It is known that the problem of finding the next solution for enumeration of minimal dominating sets in a graph reduces to the following problem. We are given two hypergraphs H_1 and H_2 over the same universe U with the property that every hyperedge of H_1 intersects every hyperedge of H_2. The question is whether one can color every element of U red or blue so that red vertices form a transversal of H_1 while blue vertices form a transversal of H_2. It is known that this problem can be solved in quasi-polynomial time (in the size of input), which immediately leads to incremental quasipolynomial-time enumeration algorithm.

There are two question about the problem stated above:

1. Can one exclude a polynomial-time algorithm for the problem under ETH? To do so, one would need to devise a reduction from 3SAT that takes an instance φ of total size N, and in time $2^{\mathcal{O}(N^{1-\varepsilon})}$ outputs an equivalent instance of the problem in question of size $2^{\mathcal{O}(N^{1-\varepsilon})}$, for some $\varepsilon > 0$.
2. Is the problem in coNP? If this was true, then this would exclude the existence of a reduction as above, unless 3SAT is solvable in co-nondeterministic subexponential time.

4.10 Generation of representative solutions

Yann Strozecki (University of Versailles, FR)

The aim of many enumeration algorithms is to build explicitly a set of elements so that an end user (a chemist, a biologist, a network engineer ...) can inspect them and select the best for him. When the user can express its preferences as a value to optimize, it is a classical optimization problem. If there are several critieria, we have multi-criteria problem and the set of pareto optimal solutions can be enumerated (or approximated). However, most of the time, the user is unable to give its preferences explicitly and formally. Hence we must generate the whole set of solutions, which can be slow and is often too large to be used by the user or requires complex data analysis methods.

It would be much better to generate only a small subsets of the solutions but diverse enough to represent well the whole set of solutions. There are several way of formalizing this problem which can or have been proposed during the workshop: generating k solutions which are different enough, generating a sufficiently large subset of solutions, representing succinctly the set of solutions using cartesian product, unions, wildcards ... Here we assume that we have a distance over the set of solutions. We say that a subset of solutions C is a d-cover of the whole set of solutions S if all solutions in S are at distance at most d of a solution in C. We say that an enumeration problem is efficiently approximable if for every d, there is an algorithm which produces a d-cover in a time polynomial in the size of the smallest d-cover. We can relax this definition by allowing non solutions to be in the cover and also by allowing randomized algorithms.

When the set of solutions S is given explicitly, the task of finding a small d-cover is a clustering problem, which is often hard. Hence, this notion is relevant only if the set of solutions is very simple and structured. As open problems and important practical examples we ask whether the two following problems are efficiently approximable for the Hamming distance between solutions:

- the set of (minimal) spanning trees of a graph
- the set of (minimal) $s - t$ paths

4.11 Strong polynomial delay conjectures

Yann Strozecki (University of Versailles, FR)

Joint work of Yann Strozecki, Florent Capelli
Main reference Florent Capelli, Yann Strozecki: "Enumerating models of DNF faster: breaking the dependency on the formula size", CoRR, Vol. abs/1810.04006, 2018.
URL http://arxiv.org/abs/1810.04006

A good enumeration algorithm is in polynomial delay, that is the delay between the production of two solutions is polynomial in the "input". However, for many problems, the input can be much larger than the size of the generated solutions. Hence, we should rather design strong polynomial delay algorithms, where after a precomputation phase polynomial in the input, the solutions are generated with a delay polynomial in the size of the previous solution.

The problem of enumerating the models of a DNF formula is denoted by EnumDNF. This problem seems exceedingly simple: we just need to compute the union of the models of the terms and the models of one term can be obtained in constant delay by Gray code enumeration. There are linear delay algorithm solving EnumDNF, for instance using a classical branching method with the relevant data structure. Let n be the number of variables and m the number of terms of a DNF. Note that a model is of size n while m can be exponentially larger than n. Hence the known algorithms are not in strong polynomial delay. While this problem seems simple, we conjecture that dealing with non disjoint union is hard and that there are no better algorithms than the classical one.

Weak conjecture: EnumDNF cannot be solved in strong polynomial delay.

Strong conjecture: EnumDNF cannot be solved in delay $O(m^{1-\epsilon}n^k)$ for any k and $\epsilon > 0$.

Remark: the weak conjecture is false for monotone DNF formulas. The strong conjecture is false if one considers amortized delay. We feel that the weak conjecture could be proved (assuming complexity hypothesis) by looking at the total time. Indeed, it is of the form $m^a + sn^b$ where s is the number of solutions, a and b are constants. When s is large this number is almost independent from m, which seems to strong.

Participants

- Faisal N. Abu-Khzam
Lebanese American University – Beirut, LB
- Ricardo Andrade
University Claude Bernard – Lyon, FR
- Cristina Bazgan
University Paris-Dauphine, FR
- Daniel Binkele-Raible
BEONTRA – Karlsruhe, DE
- Andreas Björklund
Lund University, SE
- Oscar Defrain
Université Clermont Auvergne – Aubiere, FR
- Arnaud Durand
Paris Diderot University, FR
- Martin Dyer
University of Leeds, GB
- Khaled M. Elbassioni
Masdar Institute – Abu Dhabi, AE
- Michael R. Fellows
University of Bergen, NO
- Henning Fernau
Universität Trier, DE
- Fedor V. Fomin
University of Bergen, NO
- Serge Gaspers
UNSW – Sydney, AU
- Petr A. Golovach
University of Bergen, NO
- Benjamin Gras
University of Orleans, FR
- Anne-Sophie Himmel
TU Berlin, DE
- Leandro Ishi Soares de Lima
University Claude Bernard – Lyon, FR
- Giuseppe F. Italiano
LUISS University, IT
- Mamadou Moustapha Kanté
Université Clermont Auvergne – Aubiere, FR
- Petteri Kaski
Aalto University, FI
- Mathieu Liedloff
University of Orleans, FR
- Daniel Lokshtanov
University of California – Santa Barbara, US
- Kazuhisa Makino
University of Tokyo, JP
- Alberto Marchetti-Spaccamela
Sapienza University of Rome, IT
- Andrea Marino
University of Pisa, IT
- João Marques-Silva
University of Lisbon, PT
- Arnaud Mary
University Claude Bernard – Lyon, FR
- Lhouari Nourine
University Clermont Auvergne, FR
- Michal Pilipczuk
University of Warsaw, PL
- Jean-Florent Raymond
TU Berlin, DE
- Frances A. Rosamond
University of Bergen, NO
- Günter Rote
FU Berlin, DE
- Marie-France Sagot
University Claude Bernard – Lyon, FR
- Saket Saurabh
Institute of Mathematical Sciences – Chennai, IN
- Mohamed Yosri Sayadi
Université de Lorraine – Metz, FR
- Martin Schirneck
Hasso-Plattner-Institut – Potsdam, DE
- Blerina Sinaimeri
Université Lyon I, FR
- Leen Stougie
CWI – Amsterdam, NL
- Yann Strozecki
University of Versailles, FR
- Takeaki Uno
National Institute of Informatics – Tokyo, JP
- Heribert Vollmer
Leibniz Universität Hannover, DE
- Kunihiro Wasa
National Institute of Informatics – Tokyo, JP
- Gerhard J. Woeginger
RWTH Aachen, DE
- Petra Wolf
Universität Tübingen, DE