Report from Dagstuhl Seminar 14071

Graph Modification Problems

Edited by

Hans L. Bodlaender[1], Pinar Heggernes[2], and Daniel Lokshtanov[3]

1 **Utrecht University, NL,** h.l.bodlaender@uu.nl
2 **University of Bergen, NO,** pinar.heggernes@ii.uib.no
3 **University of Bergen, NO,** daniello@ii.uib.no

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 14071 "Graph Modification Problems". The seminar was held from February 9 to February 14, 2014. This report contains abstracts for presentations about the recent developments on algorithms and structural results for graph modification problems, as well as related areas. Furthermore, the report contains a summary of open problems in this area of research.

Seminar February 9–14, 2014 – http://www.dagstuhl.de/14071

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity, G.2 Discrete Mathematics

Keywords and phrases graphs, algorithms, graph modification, fixed parameter tractable, graph classes

Digital Object Identifier 10.4230/DagRep.4.2.38

1 Executive Summary

Hans L. Bodlaender
Pinar Heggernes
Daniel Lokshtanov

A surprisingly high number of the interesting computational problems arising from theory and applications can be formulated as *graph modification* problems. Here we are given as input a graph G, and the goal is to apply certain operations on G (such as vertex deletions, edge deletions, additions or contractions) in order to obtain a graph H with some particular property. For an example the classical Vertex Cover problem can be formulated as trying to change G into an edgeless graph by performing the minimum possible number of vertex deletions. The Cluster Editing problem is to change G into a disjoint union of cliques with a minimum number of edge deletions or additions. Graph modification problems have been studied quite extensively, and both algorithms for these problems and structural aspects have been thoroughly explored.

Graph modification problems have received a significant amount of attention from the perspective of Parameterized Complexity. In parameterized complexity input comes with a parameter k and the goal is to design *fixed parameter tractable* algorithms, i.e. algorithms with running time $f(k)n^{O(1)}$ for some, hopefully not too fast growing function f. The parameter k can be the size of the solution sought for, or it could be a number describing how structured the input instance is. For an example k could be the *treewidth* of the input graph. Over the last few years, our understanding of the parameterized complexity of graph modification problems has greatly improved. Fixed parameter tractable algorithms have been found for a number of fundamental graph modification problems. For several problems, surprising new algorithms with *subexponential* ($2^{o(k)}$) dependence on k have been developed.

There is a strong connection between graph modification problems and *graph classes*. A graph class is simply a set of graphs satisfying some common properties. Thus many, if not all, graph modification problems can be phrased as modifying the input graph G by as few operations as possible to make it fit into a particular graph class. There is a large and active Graph Classes research community that primarily investigates how restricting the *input* graph to a particular graph class affects the computational complexity of computational problems. In the setting of graph modification problems we have no restrictions on the input graph, but the problem definitions dictate which graph class the *output* graph should belong to. The main objective of the seminar was to bring together experts within Parameterized Algorithms and experts within Graph Classes to join forces on graph modification problems. We also invited experts from related areas, such as Structural Graph Theory and Bioinformatics. Structural graph theory, in order to learn of the new powerful graph theoretic tools being developed, and hopefully to apply them on graph modification problems. Bioinformatics, in order to better understand the relationship between the idealized models we study and real-world applications of graph algorithms.

The scientific program of the seminar consisted of 21 talks. 4 of these talks were longer (45 or 90 minute) presentations covering some of the most exciting developments on graph modification problems and related areas. We had one long talk for each of the main topics covered by the seminar. On Monday, Marcin and Michał Pilipczuk gave a joint 90 minute talk ("Subexponential parameterized complexity of completion problems") on parameterized algorithms. On Tuesday, Paul Medvedev gave a 45 minute talk ("An introduction to genome assembly and its relation to problems on graphs") showcasing how graph algorithms can be used in Bioinformatics applications. On Wednsday, Kristina Vušković gave a 45 minute presentation ("Weighted Independent Set in bull-free graphs") about how deep structure theorems can be useful in algorithm design, and on Thursday, Andreas Brandstädt gave a presentation ("Clique separator decomposition for a subclass of hole-free graphs") on graph classes. We believe that the invited talks were a good starting point for cross-community collaboration. The remaining talks were 30 or 35 minute presentations on recent research of the participants. We made a point out of having fewer short talks, in order to leave more time for individual discussions and collaboration in groups, as well as for open problem sessions. The idea was to reserve almost all of the time between lunch and dinner for research. This was very well received by the participants. There were 3 fruitful open problem sessions, on Monday, Tuesday and Thursday. Notes on the presented problems can be found in this report.

2 Table of Contents

Open Problems

3 Overview of Talks

3.1 A few things about linear rankwidth

Isolde Adler (Goethe-Universität Frankfurt am Main, DE)

Joint work of Adler, Isolde; Farley, Art; Ganian, Robert; Kante, Mamadou; Kwon, O-joung; Proskurowski, Andrzej

Path-width can be seen as linearized variant of tree-width, and similarly, linear rank-width is the linearized version of rank-width. It is defined like rank-width by restricting the decomposition trees to being caterpillars. It is known that a graph class has bounded linear rank-width if and only if it has bounded linear clique-width. Many problems that are NP-hard in general become tractable on graphs on bounded (linear) rank-width. For instance, this is the case for all problems expressible in MSO_1 (monadic second order logic with quantification over vertex sets only).

While path-width is a well-studies notion, much less is yet known about linear rank-width.

If a graph class has bounded path-width, then it has bounded linear rank-width. The converse is not true: cliques and complete bipartite graphs have linear rank-width 1, but their path-width is unbounded.

Since computing linear rank-width is NP-hard in general, we are interested in finding graph classes that permit an efficient computation of linear rank-width.

In this talk we give a short overview of the state of the art and we present some results on trees and distance-hereditary graphs.

The talk includes results of joint work with Art Farley, Robert Ganian, Mamadou Kante and O-joung Kwon and Andrzej Proskurowski.

3.2 Parameterized complexity of three edge contraction problems with degree constraints

Rémy Belmonte (Kyoto University, JP)

Joint work of Belmonte, Rémy; Golovach, Petr A.; van 't Hof, Pim; Paulusma, Daniël
Main reference R. Belmonte, P. A. Golovach, P. van 't Hof, D. Paulusma, "Parameterized Complexity of Two Edge Contraction Problems with Degree Constraints," in Proc. of the 8th Int'l Symp. on Parameterized and Exact Computation (IPEC'13), LNCS, Vol. 8246, pp. 16–27, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-319-03898-8_3

For any graph class $\mathcal{H}$, the $\mathcal{H}$-Contraction problem takes as input a graph G and an integer k, and asks whether there exists a graph $H \in \mathcal{H}$ such that G can be modified into H using at most k edge contractions. We study the parameterized complexity of $\mathcal{H}$-Contraction for three different classes $\mathcal{H}$: the class $\mathcal{H}_{\leq d}$ of graphs with maximum degree at most d, the class $\mathcal{H}_{=d}$ of d-regular graphs, and the class of d-degenerate graphs. We completely classify the parameterized complexity of all three problems with respect to the parameters k, d, and $d+k$. Moreover, we show that $\mathcal{H}$-Contraction admits an $O(k)$ vertex kernel on connected graphs when $\mathcal{H} \in \{\mathcal{H}_{\leq 2}, \mathcal{H}_{=2}\}$, while the problem is W[2]-hard when $\mathcal{H}$ is the class of 2-degenerate graphs and hence is expected not to admit a kernel at all. In particular, our results imply that $\mathcal{H}$-Contraction admits a linear vertex kernel when $\mathcal{H}$ is the class of cycles.

3.3 Clique separator decomposition and modular decomposition for some subclasses of odd-hole-free graphs

Andreas Brandstädt (Universität Rostock, DE)

Joint work of Brandstädt, Andreas; Berry, Anne; Giakoumakis, Vassilis; Maffray, Frédéric; Mosca, Raffaele
Main reference A. Brandstädt, V. Giakoumakis, F. Maffray, "Clique separator decomposition of hole-free and diamond-free graphs and algorithmic consequences," Discrete Applied Math. 160 (2012):471–478.
URL http://dx.doi.org/10.1016/j.dam.2011.10.031

A *hole* is a chordless cycle of length at least 5. An *odd hole* is a hole with odd length. An *odd anti-hole* is the complement of an odd hole. A *diamond* is a 4-clique minus an edge. A *paraglider* is a graph having five vertices such that four of them induce a diamond, and the fifth is adjacent to exactly the vertices of degree 2 in the diamond. A *bull* is a graph having five vertices such that four of them induce a chordless path (a P_4) and the fifth is adjacent to exactly the vertices of degree 2 in the P_4. The famous Strong Perfect Graph Theorem by Chudnovsky et al. says that a graph is perfect if and only if it is odd-hole-free and odd-antihole-free. Graph decomposition is one of the fundamental tools for studying graph structure. Two of the most famous decomposition types are *modular decomposition* and *clique separator decomposition*.

Motivated by the study of graph classes related to perfect graphs and the fact that the complexity of the Maximum (Weight) Independent Set (MWIS) problem is an open question for hole-free graphs, we present the following results in the talk:

1. In a paper with Giakoumakis and Maffray, we characterize (hole, paraglider)-free graphs by the structure of their subgraphs having no clique separator. As a consequence, the MWIS problem is solvable in polynomial time on (hole, paraglider)-free graphs.
2. In a paper with Berry, Giakoumakis and Maffray, we describe the structure of (hole, diamond)-free graphs (which is a subclass of (hole, paraglider)-freegraphs) by the structure of their subgraphs having no clique separator and give an $\mathcal{O}(n^2)$ time recognition algorithm for this class.
3. In a paper with Raffaele Mosca, we give a polynomial time algorithm for the MWIS problem on (odd-hole, bull)-free graphs ((odd-hole, dart)-free graphs, respectively).

3.4 Linear recognition of almost (unit) interval graphs

Yixin Cao (Hungarian Academy of Sciences – Budapest, HU)

Main reference Y. Cao, "Linear Recognition of Almost (Unit) Interval Graphs," arXiv:1403.1515v1 [cs.DM], 2014.
URL http://arxiv.org/abs/1403.1515v1

Give a graph class $\mathcal{G}$ and a nonnegative integer k, we use $\mathcal{G}+kv$, $\mathcal{G}+ke$, and $\mathcal{G}-ke$ to denote the classes of graphs that can be obtained from some graph in $\mathcal{G}$ by adding k vertices, adding k edges, and deleting k edges, respectively. They are called *almost (unit) interval graphs* if $\mathcal{G}$ is the class of (unit) interval graphs. Almost (unit) interval graphs are well motivated from computational biology, where the data ought to be represented by a (unit) interval graph while we can only expect an almost (unit) interval graph for the best. For any fixed k, we give linear-time algorithms for recognizing all these classes, and in the case of membership, our algorithms provide also a specific (unit) interval graph as evidence.

When k is part of the input, all the recognition problems are NP-complete. Our results imply that all of them are fixed-parameter tractable parameterized by k, thereby resolving the long-standing open problem on the parameterized complexity of recognizing (unit-)interval $+ ke$, first asked by Bodlaender et al. [1]. Moreover, our algorithms for recognizing (unit-)interval $+ kv$ and (unit-)interval $- ke$ have single-exponential dependence on k and linear dependence on the graph size, which significantly improve all previous algorithms for recognizing the same classes. In particular, we show that: (n and m stand for the numbers of vertices and edges respectively in the input graph)

- interval $- ke$ can be recognized in time $O(6^k \cdot (n + m))$, improved from $O(k^{2k} \cdot n^3 m)$ [Heggernes et al., STOC 2007];
- unit-interval$-ke$ can be recognized in time $O(4^k \cdot (n+m))$, improved from $O(16^k \cdot (m+n))$ [Kaplan et al., FOCS 1994];
- interval $+ kv$ can be recognized in time $O(8^k \cdot (n + m))$, improved from $O(10^k \cdot n^9)$ [Cao and Marx, SODA 2014]; and
- unit-interval $+ kv$ can be recognized in time $O(6^k \cdot (n + m))$, improved from $O(6^k \cdot n^6)$ [Villanger, IPEC 2010].

These problems have natural optimization versions, which are known as graph modification problems. For those related to interval graphs, we show that under certain condition, there always exist optimum solutions that preserve all modules of the input graph. Another important ingredient of our algorithms is combinatorial and algorithmic characterizations of graphs free of small non-interval graphs. These studies might be of their own interest.

References

1 Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, Michael T. Hallett, and Harold T. Wareham. Parameterized complexity analysis in computational biology. *Computer applications in the biosciences*, 11(1):49–57, 1995.

3.5 Convexity in graphs

Tinaz Ekim (Bogaziçi University – Istanbul, TR)

Joint work of Ekim, Tinaz; Erey, Aysel

Let $G = (V, E)$ be a connected graph and $D \subseteq V(G)$. The geodetic closure of D, denoted by $I[D]$, consists of all vertices which lie on some shortest path between two vertices of D. We say that D is a geodetic set if $I[D] = V(G)$. The geodetic number, denoted by $g(G)$, is the cardinality of a minimum geodetic set in G, and a g-set is a geodetic set of minimum cardinality.

As it is NP-hard to compute the g-set already in chordal graphs, the complexity of the problem of finding a g-set is considered in special graph classes. Polynomial time algorithms are designed for split graphs and proper interval graphs among subclasses of chordal graphs, and for distance hereditary graphs, cographs and P_4-sparse graphs. We will briefly exhibit the block decomposition approach which yields a polynomial time algorithm to compute a g-set in monopolar chordal graphs and a superclass of block-cacti. Then we will discuss some other approaches to handle minimum geodetic set problem. In particular, we will consider the following questions: What are the graphs for which some greedy algorithm finds a g-set? What are the graphs for which the simplicial vertices form a g-set? Which graph modification would yield a graph having this property after k operations?

3.6 Tree deletion set has a polynomial kernel

Archontia C. Giannopoulou (Durham University, GB)

Joint work of Giannopoulou, Archontia C. ; Lokshtanov, Daniel; Saurabh, Saket; Suchy, Ondrej
Main reference A. C. Giannopoulou, D. Lokshtanov, S. Saurabh, O. Suchy, "Tree Deletion Set has a Polynomial Kernel (but no $\mathrm{OPT}^{\mathcal{O}(1)}$ approximation)," arXiv:1309.7891v1 [cs.DS], 2013.
URL http://arxiv.org/abs/1309.7891v1

In the Tree Deletion Set problem the input is a graph G together with an integer k. The objective is to determine whether there exists a set S of at most k vertices such that $G \setminus S$ is a tree. The problem is NP-complete and even NP-hard to approximate within any factor of OPT^c for any constant c. In this talk we give a $O(k^4)$ size kernel for the weighted Tree Deletion Set problem.

3.7 Editing to a connected graph of given degrees

Petr A. Golovach (University of Bergen, NO)

Main reference P. A. Golovach, "Editing to a Connected Graph of Given Degrees," arXiv:1308.1802v1 [cs.DS], 2013.
URL http://arxiv.org/abs/1308.1802v1

The aim of edge editing or modification problems is to change a given graph by adding and deleting of a small number of edges in order to satisfy a certain property. We consider the Edge Editing to a Connected Graph of Given Degrees problem that asks for a graph G, non-negative integers d, k and a function $\delta\colon V(G) \to \{1, \ldots, d\}$, whether it is possible to obtain a connected graph G' from G such that the degree of v is $\delta(v)$ for any vertex v by at most k edge editing operations. As the problem is NP-complete even if $\delta(v) = 2$, we are interested in the parameterized complexity and show that Edge Editing to a Connected Graph of Given Degrees admits a polynomial kernel when parameterized by $d + k$. For the special case $\delta(v) = d$, i.e., when the aim is to obtain a connected d-regular graph, the problem is shown to be fixed parameter tractable when parameterized by k only.

3.8 Characterizations of cographs as intersection graphs of paths on a grid

Martin Charles Golumbic (University of Haifa, IL)

Joint work of Cohen, Elad; Golumbic, Martin Charles; Ries, Bernard

A cograph is a graph which does not contain any induced path on four vertices. We characterize those cographs that are intersection graphs of paths on a grid in the following two cases: (i) the paths on the grid all have at most one bend and the intersections concern edges (the B_1-EPG graphs); (ii) the paths on the grid are not bended and the intersections concern vertices (the B_0-VPG graphs).

In both cases, we give a characterization by a family of forbidden induced subgraphs. We further present polynomial-time algorithms to recognize B_1-EPG cographs and B_0-VPG cographs using their cotree.

This work began during the previous Dagstuhl workshop in 2011.

3.9 A near-optimal planarization algorithm

Bart M. P. Jansen (University of Bergen, NO)

Joint work of Jansen, Bart Maarten Paul; Lokshtanov, Daniel; Saurabh, Saket
Main reference B. M. P. Jansen, D. Lokshtanov, S. Saurabh, "A near-optimal planarization algorithm," in Proc. of the 25th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA'14), pp. 1802–1811, SIAM, 2014.
URL http://dx.doi.org/10.1137/1.9781611973402.130

The problem of testing whether a graph is planar has been studied for over half a century, and is known to be solvable in $O(n)$ time using a myriad of different approaches and techniques. Robertson and Seymour established the existence of a cubic algorithm for the more general problem of deciding whether an n-vertex graph can be made planar by at most k vertex deletions, for every fixed k. Of the known algorithms for k-Vertex Planarization, the algorithm of Marx and Schlotter (WG 2007, Algorithmica 2012) running in time $2^{k^{O(k^3)}}n^2$ achieves the best running time dependence on k. The algorithm of Kawarabayashi (FOCS 2009), running in time $f(k)n$ for some $f(k) \geq 2^{k^{k^3}}$ that is not stated explicitly, achieves the best dependence on n.

We present an algorithm for k-Vertex Planarization with running time $2^{O(k \log k)}n$, significantly improving the running time dependence on k without compromising the linear dependence on n. Our main technical contribution is a novel scheme to reduce the treewidth of the input graph to $O(k)$ in time $2^{O(k \log k)}n$. It combines new insights into the structure of graphs that become planar after contracting a matching, with a Baker-type subroutine that reduces the number of disjoint paths through planar parts of the graph that are not affected by the sought solution. To solve the reduced instances we formulate a dynamic programming algorithm for Weighted Vertex Planarization on graphs of treewidth w with running time $2^{O(w \log w)}n$, thereby improving over previous double-exponential algorithms.

While Kawarabayashi's planarization algorithm relies heavily on deep results from the graph minors project, our techniques are elementary and practically self-contained. We expect them to be applicable to related edge-deletion and contraction variants of planarization problems.

3.10 Around the listing of minimal dominating sets

Mamadou Moustapha Kante (University Blaise Pascal – Aubiere, FR)

Joint work of Kante, Mamadou Moustapha; Limouzy, Vincent; Mary, Arnaud; Nourine, Lhouari; Uno, Takeaki

The Transversal Problem which consists in the enumeration of minimal hitting sets of a hypergraph in output-polynomial time, ie in time polynomial in the cumulated sizes of the input hypergraph and output set of minimal hitting sets is a long standing open problem (more than a 50 years old problem). Until now a few examples of tractable cases are known, the most general examples being the k-degenerate hypergraphs and the k-conformal ones. The best known algorithm is the quasi-polynomial algorithm by Fredman and Khachiyan [1]. A dominating set in a graph is a subset of vertices that hits every closed neighborhood. Hence, the enumeration of minimal dominating sets in a graph (DOM Problem) is a special case of the Transversal Problem. We first sketch the proof that the two problems are equivalent in the sense that there is a polynomial delay algorithm for the Transversal Problem iff there is one for the DOM Problem. In a second part we give examples of graphs where the DOM Problem is tractable by emphasing on used techniques:

1. Tractable Cases from Hypergraphs: k-degenerate graphs, undirected path-graphs, ...
2. The case of Bounded clique-width graphs: meta-theorem by Courcelle based on automata and logic. This case is interesting in its own since it transforms the DOM Problem into an enumeration of trees (simulating successful runs) in DAGS.
3. Transformations of instances into enumeration of paths in DAGS: Interval and permutation graphs. This allows to count in polynomial time, and can be extended to several other graph classes: circular-arc graphs, d-trapezoid, ...
4. Mix of hypergraph techniques and graph theoretic: a polynomial delay algorithm for the enumeration of minimal edge dominating sets.

References

1 Michael L. Fredman, Leonid Khachiyan: On the Complexity of Dualization of Monotone Disjunctive Normal Forms. J. Algorithms 21(3): 618–628 (1996)

3.11 On the variants of tree-width

O-joung Kwon (KAIST – Daejeon, KR)

Joint work of Kwon, O-joung; Ok, Seongmin

In this talk, we consider the notion of spaghetti treewidth, directed spaghetti treewidth, and strongly chordal treewidth, which are variants of tree-width. For each of these graph parameters, we show that the class of graphs with this parameter at most two is closed under taking of minors, and give the obstruction set for this class. We also characterize the class, in terms of a tree of cycles with additional conditions. We also show that for an integer k larger than 2, the classes of graphs with spaghetti treewidth, directed spaghetti treewidth, or strongly chordal treewidth, respectively at most k, are not closed under taking minors.

3.12 Introduction to genome assembly and its relation to problems on graphs

Paul Medvedev (Pennsylvania State University, US)

In this talk, we give a short tutorial on genome assembly, focusing on the algorithmic aspects. We first describe the biological problem and then formulate the shortest superstring model and show its limitations. We then describe the de Bruijn graph model, showing its limitations as well as strengths.

In the second part of the talk, we describe a recent algorithm to collapse all the chains in a de Bruijn graph using a small amount of memory. The algorithm works by partitioning the node in the graph using a hash function so that only the nodes with the same hash value need to be loaded into memory at the same time. The hash function is based on the idea of frequency-based minimizers, which allow the nodes to be evenly distributed and the hash function to exhibit structural locality. This second part is joint work with Rayan Chikhi and Antoine Limasset that will appear at RECOMB 2014.

3.13 On the recognition of four-directional orthogonal ray graphs

George B. Mertzios (Durham University, GB)

Orthogonal ray graphs are the intersection graphs of horizontal and vertical rays (i.e. half-lines) in the plane. If the rays can have any possible orientation (left/right/up/down) then the graph is a 4-directional orthogonal ray graph (4-DORG). Otherwise, if all rays are only pointing into the positive x and y directions, the intersection graph is a 2-DORG. Similarly, for 3-DORGs, the horizontal rays can have any direction but the vertical ones can only have the positive direction. The recognition problem of 2-DORGs, which are a nice subclass of bipartite comparability graphs, is known to be polynomial, while the recognition problems for 3-DORGs and 4-DORGs are open. Recently it has been shown that the recognition of unit grid intersection graphs, a superclass of 4-DORGs, is NP-complete. In this paper we prove that the recognition problem of 4-DORGs is polynomial, given a partition $\{L, R, U, D\}$ of the vertices of G (which corresponds to the four possible ray directions). For the proof, given the graph G, we first construct two cliques G_1, G_2 with both directed and undirected edges. Then we successively augment these two graphs, constructing eventually a graph $\widetilde{G}$ with both directed and undirected edges, such that G has a 4-DORG representation if and only if $\widetilde{G}$ has a transitive orientation respecting its directed edges. As a crucial tool for our analysis we introduce the notion of an S-orientation of a graph, which extends the notion of a transitive orientation. We expect that our proof ideas will be useful also in other situations. Using an independent approach we show that, given a permutation π of the vertices of U (π is the order of y-coordinates of ray endpoints for U), while the partition $\{L, R\}$ of $V \setminus U$ is not given, we can still efficiently check whether G has a 3-DORG representation.

3.14 Vector connectivity in graphs

Martin Milanic (University of Primorska, SI)

Joint work of Boros, Endre; Cicalese, Ferdinando; Heggernes, Pinar; van 't Hof, Pim; Milanič, Martin; Rizzi, Romeo;
Main reference P. Heggernes, P. van 't Hof, Pim, M. Milanič, "Vector connectivity in graphs," Networks, online version, February 2014.
URL http://dx.doi.org/10.1002/net.21545

Motivated by challenges related to domination, connectivity, and information propagation in social and other networks, we introduce and study the Vector Connectivity problem. This problem takes as input a graph G and an integer $r(v)$ for every vertex v of G, and the objective is to find a vertex subset S of minimum cardinality such that every vertex v either belongs to S, or is connected to at least $r(v)$ vertices of S by disjoint paths. If we require each path to be of length exactly 1, we get the well-known vector domination problem, which is a generalization of the dominating set and vertex cover problems. Consequently, the vector connectivity problem becomes NP-hard if an upper bound on the length of the disjoint paths is also supplied as input. Due to the hardness of these domination variants even on restricted graph classes, like split graphs, Vector Connectivity seems to be a natural problem to study for drawing the boundaries of tractability for this type of problems. We show that Vector Connectivity can actually be solved in polynomial time on split graphs, in addition to cographs and trees.

We also show that the problem is NP-hard for planar line graphs and for planar bipartite graphs, APX-hard on general graphs, and can be approximated in polynomial time within a factor of $\log n + 2$ on all n-vertex graphs.

Vertex covers and dominating sets in a graph G can be easily characterized as hitting sets of derived hypergraphs (of G itself, and of the closed neighborhood hypergraph of G, respectively). Using Menger's Theorem, we give a similar characterization of vector connectivity sets.

References

1. Endre Boros, Pinar Heggernes, Pim van 't Hof, and Martin Milanič. Vector connectivity in graphs. *Networks*, http://dx.doi.org/10.1002/net.21545.
2. Ferdinando Cicalese, Martin Milanič, and Romeo Rizzi. On the complexity of the vector connectivity problem. Submitted.

3.15 Parameterized algorithms for Max Colorable Induced Subgraph problem on perfect graphs

Neeldhara Misra (Indian Institute of Science, IN)

Joint work of Misra, Neeldhara; Panolan, Fahad; Rai, Ashutosh; Raman, Venkatesh; Saurabh, Saket
Main reference N. Misra, F. Panolan, A. Rai, Ve. Raman, S. Saurabh, "Parameterized Algorithms for Max Colorable Induced Subgraph Problem on Perfect Graphs," in Proc. of the 39th Int'l Workshop on Graph-Theoretic Concepts in Computer Science (WG'13), LNCS, Vol. 8165, pp. 370–381, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-45043-3_32

We explore the parameterized complexity of Max Colorable Induced Subgraph on perfect graphs. The problem asks for a maximum sized q-colorable induced subgraph of an input graph G. Yannakakis and Gavril (IPL 1987) showed that this problem is NP-complete even on split graphs (which is a proper subset of perfect graphs, chordal graphs and co-chordal graphs). However, they showed that for fixed q, the problem is solvable in time $n^{O(q)}$ on chordal graphs. A natural question is whether the problem is fixed parameter tractable (FPT) when parameterized by the number of colors q, that is, whether the problem admits an algorithm with running time $f(q)n^{O(1)}$. A simple reduction shows that the problem is W[2]-hard parameterized by q, even on split graphs. Thus, we study this problem with another natural parameter – the solution size – l.

We design two parameterized algorithms for the problem. The first one runs in time $5.44^l(n + \#\alpha(G))^{O(1)}$ where $\#\alpha(G)$ is the number of maximal independent sets of the input graph and the second algorithm runs in time $q^{(l+o(l))}n^{O(1)}$. Observe that since $q < l$ for all non-trivial situations, we have that the second algorithm is FPT in l alone. The first algorithm is efficient when the input graph contains only polynomially many maximal independent sets; for example split graphs and co-chordal graphs. Finally, we show that (under standard complexity-theoretic assumptions) the problem does not admit a polynomial kernel even on split graphs and on perfect graphs the problem does not admit a polynomial kernel even for fixed values of $q > 1$.

3.16 Certifying FPT-algorithms

Haiko Müller (University of Leeds, GB)

Joint work of Mueller, Haiko; Wilson, Samuel
Main reference H. Müller, S. Wilson, "An FPT Certifying Algorithm for the Vertex-Deletion Problem," in Proc. of the 24th Int'l Workshop on Combinatorial Algorithms (IWOCA'13), LNCS, Vol. 8288, pp. 468–472, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-45278-9_45

We propose a scheme of certifying FPT-algorithms for vertex-deletion problems on graphs. For a class C of graphs that is closed under a partial order $<$ these algorithms decide, for a fixed integer k, whether a given graph G has a set U of at most k vertices such that $G - U$ belongs to C. That is, these algorithms recognize the class $C + kv$ of graphs in time $f(k)n^c$ for some constant c. In the affirmative case the algorithm should also provide the user with such a set U of vertices, and otherwise it should point out an obstruction of $C + kv$ in G. For instance, if $<$ is the ordering defined by induced subgraphs then the obstruction will be a minimal forbidden subgraph.

We give conditions on the partial order $<$ that are necessary or sufficient for such certifying FPT-algorithms to exist for all classes C that are closed under $<$ and have a finite obstruction set with respect to $<$. Moreover we illustrate these conditions by examples, namely the partial orders defined by vertex deletion, edge deletion, vertex dissolution and edge contraction, or combinations thereof.

3.17 On the complexity of degree anonymization

Andre Nichterlein (TU Berlin, DE)

Joint work of Bazgan, Cristina; Froese, Vincent; Hartung, Sepp; Nichterlein, Andre; Niedermeier, Rolf; Suchy, Ondrej
Main reference S. Hartung, A. Nichterlein, R. Niedermeier, and O. Suchy, "A refined complexity analysis of degree anonymization on graphs," in Proc. of the 40th Int'l Colloquium on Automata, Languages, and Programming (ICALP'13), LNCS, Vol. 7966, pp. 594–606, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-39212-2_52

Motivated by a growing interest in graph anonymization (in particular with respect to social networks), we study the NP-hard Degree Anonymity problem asking whether a graph can be made k-anonymous by adding at most a given number of edges. Herein, a graph is k-anonymous if for every vertex in the graph there are at least $k-1$ other vertices of the same degree. We show that the problem is intractable when considering the standard parameter solution size, even when searching for parameterized approximation algorithms. Contrasting these negative results, we prove fixed-parameter tractability for the parameter maximum vertex degree and experimentally evaluate the corresponding algorithm.

3.18 Subexponential parameterized complexity of completion problems

Marcin Pilipczuk and Michał Pilipczuk (University of Bergen, NO)

Let Π be a fixed hereditary graph class. In the Π Completion problem, given a graph G and an integer k, we ask whether it is possible to add at most k edges to G to obtain a member of Π. In the recent years completion problems received significant attention from the perspective of parameterized complexity, with the standard parameterization by k.

In our tutorial we first survey the history of the study of parameterized complexity of completion problems, including the breakthrough paper of Villanger et al [6] that settles fixed-parameter tractability of Interval Completion, as well as recent advancements on polynomial kernelization. Then, we move to the main topic of the tutorial, namely subexponential parameterized algorithms.

First fixed-parameter algorithms for completion problems focused mostly on the 'forbidden induced subgraphs' definition of the graph class Π in question. In 2012 Fomin and Villanger [4] came up with a novel idea to instead focus on some structural definition of the class Π, trying to build the modified output graph by dynamic programming. Slightly simplifying, we may say that the main technical contribution of [4] is a bound of at most $k^{\mathcal{O}(\sqrt{k})}$ reasonable 'partial chordal graphs' for an input instance (G, k) of Chordal Completion. Consequently, Chordal Completion can be solved in $k^{\mathcal{O}(\sqrt{k})} + n^{\mathcal{O}(1)}$ time. Following the approach of Fomin and Villanger, in the past two years subexponential parameterized algorithms were shown for the class of chain [4], split [5], threshold [3], trivially perfect [3], pseudosplit [3] and, very recently, proper interval [2] and interval [1] graphs. Moreover, a few lower bounds for related graph classes were found [3].

In our tutorial we present the approach of Fomin and Villanger on the example of Trivially Perfect Completion, and then survey the main ideas needed in the remaining algorithms.

References

1 Ivan Bliznets, Fedor V. Fomin, Marcin Pilipczuk, Michał Pilipczuk. *A subexponential parameterized algorithm for Interval Completion*. ArXiv report `arxiv.org/abs/1402.3473` (2014)

2 Ivan Bliznets, Fedor V. Fomin, Marcin Pilipczuk, Michał Pilipczuk. *A subexponential parameterized algorithm for Proper Interval Completion*. ArXiv report `arxiv.org/abs/1402.3472` (2014)

3 Pål Grønås Drange, Fedor V. Fomin, Michał Pilipczuk, Yngve Villanger. *Exploring Subexponential Parameterized Complexity of Completion Problems*. STACS 2014, to appear. Available at `arxiv.org/abs/1309.4022`

4 Fedor V. Fomin, Yngve Villanger. *Subexponential parameterized algorithm for minimum fill-in*. SODA 2012:1737–1746

5 Esha Ghosh, Sudeshna Kolay, Mrinal Kumar, Pranabendu Misra, Fahad Panolan, Ashutosh Rai, M. S. Ramanujan. *Faster Parameterized Algorithms for Deletion to Split Graphs*. SWAT 2012:107–118

6 Yngve Villanger, Pinar Heggernes, Christophe Paul, Jan Arne Telle. *Interval Completion Is Fixed Parameter Tractable*. SIAM J. Comput. 38(5):2007–2020 (2009)

3.19 Optimal Erdős Pósa for pumpkins revisited

Dimitrios M. Thilikos (ALGCo project team, CNRS, LIRMM, Montpellier, FR, and Department of Mathematics, National and Kapodistrian University of Athens, GR)

Joint work of Chatzidimitriou, Dimitris; Sau, Ignasi; Raymond, Jean-Florent; Thilikos, Dimitrios M.

Given two graphs H and G, we denote by $\mathbf{pack}^{\mathbf{v}}_{H}(G)$ the maximum number of vertex-disjoint minor models of H in G. We denote by $\mathbf{pack}^{\mathbf{e}}_{H}(G)$ the maximum number of edge-disjoint minor models of H in G. We also denote by $\mathbf{cover}^{\mathbf{v}}_{H}(G)$ the minimum number of vertices that intersect all minor models of H in G. Similarly, by $\mathbf{cover}^{\mathbf{e}}_{H}(G)$ we denote the minimum number of edges that intersect all minor models of H in G. Finally, we denote by θ_r the multi-graph containing two vertices and r parallel edges between them (also known as *the r-pumpkin*).

We prove the following results.

▶ **Theorem 1.** *There exists a function $f : \mathbb{N} \to \mathbb{N}$ such that for every two positive integers r, q, and every graph G excluding K_q as a minor, it holds that $\mathbf{cover}^{\mathbf{v}}_{\theta_r}(G) \leq f(r) \cdot \mathbf{pack}^{\mathbf{v}}_{\theta_r}(G) \cdot \log q$.*

▶ **Theorem 2.** *There exists a function $f : \mathbb{N} \to \mathbb{N}$ such that for every two positive integers r, q, and every graph G excluding K_q as a minor, it holds that $\mathbf{cover}^{\mathbf{e}}_{\theta_r}(G) \leq f(r) \cdot \mathbf{pack}^{\mathbf{e}}_{\theta_r}(G) \cdot \log q$.*

The above results also imply that, for every r, the problems of computing the values of $\mathbf{pack}^{\mathbf{v}}_{\theta_r}$, $\mathbf{cover}^{\mathbf{v}}_{\theta_r}$, $\mathbf{pack}^{\mathbf{e}}_{\theta_r}$, and $\mathbf{cover}^{\mathbf{e}}_{\theta_r}$ admit $\log(OPT)$-approximation (deterministic and polynomial) algorithms.

3.20 Parameterized complexity dichotomy for Steiner multicut

Erik Jan van Leeuwen (MPI für Informatik – Saarbrücken, DE)

Joint work of Bringmann, Karl; Hermelin, Danny; Mnich, Matthias; van Leeuwen, Erik Jan

We consider the Steiner Multicut problem, which asks, given an undirected graph G, a collection $T_1, \ldots, T_t \subseteq V(G)$, of terminal sets of size at most p, and an integer k, whether there is a set S of at most k edges or nodes such that of each set T_i at least one pair of terminals is in different connected components of $G \setminus S$. This problem generalizes several well-studied graph cut problems, in particular the Multicut problem, which corresponds to the case $p = 2$. We provide a dichotomy of the parameterized complexity of Steiner Multicut on general graphs. That is, for any combination of k, t, p, and the treewidth $tw(G)$ as a constant, parameter, or unbounded, and for all versions of the problem (edge deletion, and node deletion with and without deletable terminals), we prove either that the problem is fixed-parameter tractable or that the problem is hard (W[1]-hard or even (para-) NP-complete). Among the many results in the paper, we highlight that:

- The edge deletion version of Steiner Multicut is fixed-parameter tractable for the parameter $k + t$ on general graphs (but has no polynomial kernel, even on trees).
- In contrast, both node deletion versions of Steiner Multicut are W[1]-hard for the parameter $k + t$ on general graphs.

- All versions of Steiner Multicut are W[1]-hard for the parameter k, even when $p = 3$ and the graph is a tree plus one node. This means that the known parameterized algorithms of Marx and Razgon, and Bousquet et al. (STOC 2011) for Multicut do not generalize to even the most basic instances of Steiner Multicut.

Since we allow k, t, p, and $tw(G)$ to be any constant, our characterization includes a dichotomy for Steiner Multicut on trees (i.e., for $tw(G) = 1$) as well as a polynomial-time versus NP-hardness dichotomy (by restricting $k, t, p, tw(G)$ to a constant or unbounded).

3.21 Parametrized algorithm for weighted independent set problem in bull-free graphs

Kristina Vušković (University of Leeds, GB)

Joint work of Thomassé, Stéphan; Trotignon, Nicolas; Vušković, Kristina
Main reference S. Thomassé, N. Trotignon, K. Vušković, "Parameterized algorithm for weighted independent set problem in bull-free graphs," arXiv:1310.6205v1 [cs.DM], 2013.
URL http://arxiv.org/abs/1310.6205v1

The bull is the graph obtained from a triangle by adding two pendant nonadjacent edges. A graph is bull-free if it does not contain a bull as an induced subgraph. We show the existence of an FPT algorithm for weighted independent set problem for bull-free graphs (parametrized by solution size). While a polynomial kernel is unlikely to exist for this problem, we show however that the problem has a polynomial size Turing-kernel. As a byproduct, if we forbid odd holes in addition to the bull, we show the existence of a polynomial time algorithm for the independent set problem. We also prove that the chromatic number of a bull-free graph is bounded by a function of its clique number and the maximum chromatic number of its triangle-free induced subgraphs. All our results rely on a decomposition theorem of bull-free graphs due to Chudnovsky which is modified here, allowing us to provide extreme decompositions, adapted to our computational

4 Open Problems

4.1 Treecost as a parameterized problem

Hans L. Bodlaender (Utrecht University, NL)

Let f be a function on the natural numbers. Consider the following problem. Given a graph G, and an integer L, is there a chordal supergraph H of G such that the sum over all maximal cliques C in H of $f(|C|)$ is at most L.

What is the complexity of the problem when parameterized by L?

4.2 Two simple edge editing problems

Henning Fernau (Universität Trier, DE)

We propose two graph modification problems mentioned by Damaschke and Mogren in [1].

- Edit Into Clique & Isolates
  Given a graph G of order n and an integer k, is it possible to turn G into one clique K_ℓ and a collection of $n-\ell$ isolates by adding or removing at most k edges from G?
  This problem is termed $K_1[0]$ Bag Editing in [1].
- Edit Into Biclique & Isolates
  Given a graph G of order n and an integer k, is it possible to turn G into one biclique $K_{j,\ell}$ and a collection of $n-j-\ell$ isolates by adding or removing at most k edges from G?
  This corresponds to P_3 Bag Editing from [1] by graph complementation.

In both cases, it was open whether the problem is NP-hard or whether it can be solved in polynomial time.

Edit Into Clique & Isolates was shown NP-hard in the course of the seminar by André Nichterlein.

References

1 P. Damaschke and O. Mogren. Editing the simplest graphs. In S. Prasant Pal and K. Sadakane, editors, *Algorithms and Computation – 8th International Workshop, WALCOM*, volume 8344 of *LNCS*, pages 249–260. Springer, 2014.

4.3 More graph editing problems

Henning Fernau (Universität Trier, DE)

The typical graph editing problems seen during this seminar are to delete at most i vertices, or to delete at most j edges, or to insert at most k edges from a given graph G of order n and size m to obtain a target graph that satisfies certain properties P. Often, such properties can be specified by (induced) subgraphs or similar substructures. There are several related problems that might be worth studying, as well. From the perspective of parameterized complexity, the "dual problems" could be interesting to study. This could mean:

- Delete vertices from G such that the target graph satisfying P contains at least $i_d = n-i$ vertices. In other words, does there exist an induced subgraph of G on at least i_d vertices that satisfies P?
- Delete edges from G such that the target graph satisfying P contains at least $j_d = m-j$ edges. In other words, does there exist a subgraph of G on at least j_d vertices that satisfies P?
- Add edges to G such that the target graph satisfying P contains at least $k_d = (n^2 - n)/2 - m + k$ edges. The upper bound $(n^2-n)/2-m$ is derived from the fact that adding edges corresponds to deleting edges in the complement graph.

Also, there are natural lower bounds for these problems in terms of packings, assuming that P is given by a set of forbidden structures (e.g., forbidden induced subgraphs) S_P. We would arrive at problems like:

- Can we find a vertex-disjoint packing of G with i_p objects from S_P?
- Can we find an edge-disjoint packing of G with j_p objects from S_P?

Possibly, stranger problems would show up when defining packing problems for edge-addition problems. This seems to necessitate a forbidden substructure characterization of the complement graphs. There could be also other related packing problems, for instance:

- Can we add some vertices and edges to the graph G so that the resulting graph H admits a "perfect packing" with at most ℓ objects from S_P?

Here, "perfect packing" could either mean that all vertices or that all edges of H are covered by the at most ℓ objects from S_P. An example for such a problem can be found in [1]. This might also answer a question raised by one of the participants of the seminar about the meaningful existence of vertex addition problems.

To our knowledge, far less recent work on the graph (modification) problems sketched above has been done. In particular, general question on when such problems are hard or easy in the parameterized sense could be posed. Also, the existence of sub-exponential algorithms for such types of problems should be interesting to look into.

Clearly, $i_p \leq i$ and $j_p \leq j$, so that also the question of "parameterizing above guaranteed value" shows up, which has not been in the focus of talks from the seminar, either.

References

1 D. S. Hochbaum and A. Levin. Covering the edges of bipartite graphs using graphs. *Theoretical Computer Science*, 411(1):1–9, 2010.

4.4 Modification into graph classes

George B. Mertzios (Durham University, GB)

Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two graph classes. Then, given a graph $G = (V, E) \in \mathcal{C}_1$, what is the complexity of each of the problems:

- compute a minimum set F of edges such that the graph $G' = (V, E \cup F)$ belongs to class $\mathcal{C}_2$ (*completion* problem);
- compute a minimum set $F \subseteq E$ of edges such that the graph $G' = (V, E \setminus F)$ belongs to class $\mathcal{C}_2$ (*edge deletion* problem);
- compute a minimum set $U \subseteq V$ of vertices such that the graph $G' = G[U]$ belongs to class $\mathcal{C}_2$ (*vertex deletion* problem).

For which classes $\mathcal{C}_1, \mathcal{C}_2$ are the above problems solvable in polynomial or FPT time? For instance, what is the complexity of these problems in the case where $\mathcal{C}_1$ is the class of interval graphs and $\mathcal{C}_2$ is the class of proper interval graphs?

4.5 H-minor sequences

Daniel Paulusma (Durham University, GB)

Joint work of Golovach, Petr; Paulusma, Daniel; Stewart, Iain
Main reference P. A. Golovach, D. Paulusma, I. A. Stewart, "Graph editing to a fixed target," in Proc. of the 24th Int'l Workshop on Combinatorial Algorithms (IWOCA'13), LNCS, Vol. 8288, pp. 192–205, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-45278-9_17

Call a sequence of operations, each of type "edge contraction", "edge deletion" or "vertex deletion", that modifies a graph G into a graph H an *H-minor sequence.* The *length* of an H-minor sequence is the number of its operations.

For a fixed graph H, let H-Minor Sequence be the problem that asks whether a given graph G has an H-minor sequence of length at most ℓ for some given integer ℓ. There are many graphs H for which this problem is known to be polynomial-time solvable, and many graphs H for which this problem is known to be NP-complete.

Let C_k be the cycle on k vertices. It is known that C_k-Minor Sequence is polynomial-time solvable for every $k \leq 4$. We pose the following problem:

*Determine the computational complexity of C_k-*Minor Sequence *for any fixed $k \geq 5$.*

4.6 Open problems from the tutorial on subexponential parameterized complexity of completion problems

Marcin Pilipczuk (University of Bergen, NO)

Let Π be a fixed hereditary graph class. In the Π Completion problem, given a graph G and an integer k, we ask whether it is possible to add at most k edges to G to obtain a member of Π. During the tutorial on subexponential parameterized complexity of completion problems and the discussion on open problem session following the tutorial the following interesting open problems were identified.

1. For most of known subexponential parameterized algorithm for completion problems the dependency on the parameter in the running time is $k^{\mathcal{O}(\sqrt{k})}$ or better, with the exception of Proper Interval Completion where the dependency is $k^{\mathcal{O}(k^{2/3})}$ [2]. Can it be improved to $k^{\mathcal{O}(\sqrt{k})}$?
2. The running time of the algorithm for Split Completion [4] has dependency $2^{\mathcal{O}(\sqrt{k})}$ on the parameter. Can we obtain such a dependency for other problems?
3. We believe that for the discussed graph classes Π, no FPT algorithm with dependency $2^{o(\sqrt{k})}$ on the parameter should exist, as it would be also a $2^{o(n)}$-time algorithm. Can we prove this conjecture for some discussed graph classes Π, under the assumption of the Exponential Time Hypothesis? We remark that to achieve this goal most likely one would need to reengineer the known NP-hardness reductions for these completion problems, as the only currently known reductions use Optimal Linear Arrangement as a pivot problem, causing at least cubic blowup in the parameter.
4. In scope of the techniques used in the recent subexponential parameterized algorithm for Interval Completion [1], a question of a polynomial kernel for this problem is appealing.

5. We conjecture that for an instance (G, k) of Chordal Completion, one can enumerate a family $\mathcal{F}$ of $n^{\mathcal{O}(\sqrt{k})}$ subsets of $V(G)$ such that for any chordal supergraph H of G with $|E(H) \setminus E(G)| \leq k$, all maximal cliques of H belong to $\mathcal{F}$. This statement does not follow from the work of Fomin and Villanger [3], as in some cases the algorithm of [3] identifies and executes a subexponential branching.
6. For the search of further subexponential parameterized algorithms for completion problems, the following interesting graph classes have been identified: weakly chordal graphs, strongly chordal graphs, permutation graphs, perfect graphs, 3-leaf powers and path graphs. Of particular importance is the case of Perfect Completion, where no fixed-parameter algorithm is known.

References

1 Ivan Bliznets, Fedor V. Fomin, Marcin Pilipczuk, Michał Pilipczuk. *A subexponential parameterized algorithm for Interval Completion*. ArXiv report `arxiv.org/abs/1402.3473` (2014)

2 Ivan Bliznets, Fedor V. Fomin, Marcin Pilipczuk, Michał Pilipczuk. *A subexponential parameterized algorithm for Proper Interval Completion*. ArXiv report `arxiv.org/abs/1402.3472` (2014)

3 Fedor V. Fomin, Yngve Villanger. *Subexponential parameterized algorithm for minimum fill-in*. SODA 2012:1737–1746

4 Esha Ghosh, Sudeshna Kolay, Mrinal Kumar, Pranabendu Misra, Fahad Panolan, Ashutosh Rai, M. S. Ramanujan. *Faster Parameterized Algorithms for Deletion to Split Graphs*. SWAT 2012:107–118

4.7 Open problem: Eulerian SCC Deletion

Marcin Pilipczuk (University of Bergen, NO)

In the Eulerian SCC Deletion problem, given a directed graph G and an integer k, we ask whether it is possible to delete at most k arcs from G to obtain a graph where each strongly connected component contains an Euler tour. Is Eulerian SCC Deletion fixed-parameter tractable, when parameterized by k?

A few remarks are in place. The question of fixed-parameter tractability of Eulerian SCC Deletion was originally posted by Cechlárová and Schlotter in [1], where it appeared naturally in modelling of housing markets. A somehow related deletion problems were studied in [2]. However, it is not hard to reduce Directed Feedback Vertex Set to Eulerian SCC Deletion, and, hence, we expect that a hypothetical fixed-parameter algorithm for Eulerian SCC Deletion would need to use substantially different techniques than the ones developed in [2].

References

1 Katarína Cechlárová, Ildikó Schlotter. *Computing the Deficiency of Housing Markets with Duplicate Houses*. IPEC 2010:72–83

2 Marek Cygan, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, Ildikó Schlotter. *Parameterized Complexity of Eulerian Deletion Problems*. Algorithmica 68(1):41–61 (2014)

4.8 Existence of Polynomial Kernel for Edge-Disjoint Paths

Erik Jan van Leeuwen (MPI für Informatik – Saarbrücken, DE)

The Vertex-Disjoint Paths problem takes as input a graph G and a set of k pairs of terminals in G, and one should decide whether there exist k pairwise vertex-disjoint paths in G such that the vertices in each terminal pair are connected to each other by one of the paths. In the Edge-Disjoint Paths problem, the paths should be edge-disjoint instead of vertex-disjoint. It is known that both problems are NP-hard on general graphs [4, 2], but fixed-parameter tractable when parameterized by k [5]. Recently, in joint work with Pinar Heggernes, Pim van 't Hof, and Reza Saei, I showed that both problems remain NP-hard on the class of split graphs, which are graphs whose vertex set can be partitioned into an independent set and a clique. Moreover, we showed that both problems admit a polynomial kernel on split graphs when parameterized by k [3]. This is the first polynomial kernel for both problems on graph classes. On general graphs, it is known that Vertex-Disjoint Paths does not admit a polynomial kernel when parameterized by k, unless NP $\subseteq$ coNP/poly [1]. However, for Edge-Disjoint Paths, no such result seems to be known. Therefore, we ask whether or not there exists a polynomial kernel for Edge-Disjoint Paths on general graphs when parameterized by k?

References

1. Bodlaender, H.L., Thomasse, S., Yeo, A.: Kernel bounds for disjoint cycles and disjoint paths. Theor. Comp. Sci. 412(35), 4570–4578 (2011)
2. Even, S., Itai, A., Shamir, A.: On the complexity of timetable and multicommodity flow problems. SIAM J. Comp. 5, 691–703 (1976)
3. Heggernes, P., Hof, P. van 't, Saei, R., van Leeuwen, E.J.: Finding Disjoint Paths in Split Graphs. In: Geffert, V., Preneel, B., Rovan, B., Stuller, J., Tjoa, A.M. (eds.), Proc. SOFSEM 2014, LNCS 8327, pp. 315–326. Springer-Verlag, Berlin.
4. Lynch, J.F.: The equivalence of theorem proving and the interconnection problem ACM SIGDA Newsletter 5(3), 31–36 (1975)
5. Robertson, N., Seymour, P.D.: Graph minors XIII. The disjoint paths problem. J. Comb. Theory B 63(1), 65–110 (1995)

Participants

- Isolde Adler
Goethe-Universität Frankfurt am Main, DE
- Rémy Belmonte
Kyoto University, JP
- Hans L. Bodlaender
Utrecht University, NL
- Andreas Brandstädt
Universität Rostock, DE
- Yixin Cao
Hungarian Academy of Sciences – Budapest, HU
- Derek G. Corneil
University of Toronto, CA
- Marek Cygan
University of Warsaw, PL
- Tinaz Ekim
Bogaziçi Univ. – Istanbul, TR
- Henning Fernau
Universität Trier, DE
- Fedor V. Fomin
University of Bergen, NO
- Archontia C. Giannopoulou
Durham University, GB
- Petr A. Golovach
University of Bergen, NO
- Martin Charles Golumbic
University of Haifa, IL
- Alexander Grigoriev
Maastricht University, NL
- Pinar Heggernes
University of Bergen, NO
- Pim van 't Hof
University of Bergen, NO
- Bart M. P. Jansen
University of Bergen, NO
- Mamadou Moustapha Kanté
Blaise Pascal Université – Aubière, FR
- Eun Jung Kim
University Paris-Dauphine, FR
- Dieter Kratsch
University of Metz, FR
- Stefan Kratsch
TU Berlin, DE
- O-joung Kwon
KAIST – Daejeon, KR
- Van Bang Le
Universität Rostock, DE
- Daniel Lokshtanov
University of Bergen, NO
- Dániel Marx
Hungarian Acad. of Sciences, HU
- Paul Medvedev
Pennsylvania State Univ., US
- Daniel Meister
Universität Trier, DE
- George B. Mertzios
Durham University, GB
- Martin Milanic
University of Primorska, SI
- Neeldhara Misra
Indian Institute of Science – Bangalore, IN
- Haiko Müller
University of Leeds, GB
- André Nichterlein
TU Berlin, DE
- Rolf Niedermeier
TU Berlin, DE
- Charis Papadopoulos
University of Ioannina, GR
- Christophe Paul
University of Montpellier 2, FR
- Daniel Paulusma
Durham University, GB
- Marcin Pilipczuk
University of Bergen, NO
- Michał Pilipczuk
University of Bergen, NO
- Andrzej Proskurowski
University of Oregon, US
- Dieter B. Rautenbach
Universität Ulm, DE
- Saket Saurabh
The Institute of Mathematical Sciences – Chennai, IN
- Dimitrios M. Thilikos
National and Kapodistrian University of Athens, GR
- Erik Jan van Leeuwen
MPI für Informatik – Saarbrücken, DE
- Kristina Vušković
University of Leeds, GB