Open problems for PARUWa 2020*: Parameterized Algorithms Retreat of the University of Warsaw

Krynica, February 2020

1 Graver complexity of matrices with low primal treedepth

Michał Pilipczuk

The *treedepth* of a graph G is the least integer d for which there exists a rooted forest F on the vertex set of G such that the depth of F is at most d and every edge of G connects two vertices that are in the ancestor-descendant relation in F. For an integer-valued matrix A, we define its *primal graph* $G(A)$ as follows: the vertices of $G(A)$ are the rows of A, and two rows are adjacent in $G(A)$ if there is a column in which they both contain a non-zero entry. The *primal treedepth* of A is the treedepth of $G(A)$.

A non-zero integer vector $v = (a_1, \ldots, a_n)^\top$ belongs to the *Graver basis* of A if $v \in \ker A$ and there is no other non-zero integer vector $v' = (b_1, \ldots, b_n)^\top \in \ker A$, $v' \neq v$, that is *conformally smaller* than v in the following sense: for all $i \in \{1, \ldots, n\}$, a_i and b_i have the same sign and $|b_i| \leq |a_i|$.

It is known that there exists a function $f \colon \mathbb{N} \to \mathbb{N}$ such that

$$\|v\|_\infty \leq f(d)$$

for every vector v in the Graver basis of any matrix A with entries in $\{-1, 0, 1\}$ and of primal treedepth at most d [1]. Unfortunately, the known upper bound on $f(d)$ is non-elementary; it is essentially a tower of height d. The question is whether there exists an elementary upper bound on $f(d)$. For instance, for the *dual treedepth* (defined analogously for the transpose matrix) there is a doubly-exponential upper bound, which is tight [2]. A positive answer to the question would probably lead to a faster algorithm for Integer Linear Programming parameterized by the maximum absolute value in the input matrix A and its primal treedepth, while a counterexample asserting a negative answer could potentially be used as a gadget in a hardness proof.

References

[1] K. Klein. About the complexity of two-stage stochastic IPs. *CoRR*, abs/1901.01135, 2019.

[2] D. Knop, M. Pilipczuk, and M. Wrochna. Tight complexity lower bounds for Integer Linear Programming with few constraints. In *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019*, volume 126 of *LIPIcs*, pages 44:1–44:15. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2019.

2 ILP parameterized by the number of constraints

Michał Pilipczuk

In ILP Feasibility with box constraints we are given an integer linear program of the form:

$$\begin{aligned} Ax &= b \\ \ell_i \leq x_i &\leq r_i \quad \text{for all } i \in \{1, \ldots, n\}, \end{aligned}$$

*The retreat has been supported by the project that received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 714704 (PI: Marcin Pilipczuk).

where $x = (x_1, \dots, x_n)^\top$ is a vector of n variables, A is an integer-valued matrix with m rows and n columns, b is a vector of length m, and $\ell_i, r_i \in \mathbb{Z} \cup \{-\infty, \infty\}$ form a *box constraint* for the variable x_i. As usual, the question is whether there is an integer evaluation of the variables that satisfies all the constraints.

When A has entries in $\{-1, 0, 1\}$, this problem can be solved in time $2^{\mathcal{O}(m^2 \log m)} \cdot \mathrm{poly}(n, m)$ [1]. For the case when the box constraints are trivial — $\ell_i = 0$ and $r_i = \infty$, which effectively asserts non-negativeness of the variables — the problem can be solved in time $2^{\mathcal{O}(m \log m)} \cdot \mathrm{poly}(n, m)$ [1], which is tight under ETH [2]. However, for arbitrary box constraints, the gap between the $2^{\mathcal{O}(m^2 \log m)} \cdot \mathrm{poly}(n, m)$ upper bound and the $2^{o(m \log m)} \cdot \mathrm{poly}(n, m)$ lower bound remains. The problem is to close this gap.

References

[1] F. Eisenbrand and R. Weismantel. Proximity results and faster algorithms for Integer Programming using the Steinitz Lemma. In *29th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018*, pages 808–816. SIAM, 2018.

[2] D. Knop, M. Pilipczuk, and M. Wrochna. Tight complexity lower bounds for Integer Linear Programming with few constraints. In *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019*, volume 126 of *LIPIcs*, pages 44:1–44:15. Schloss Dagstuhl — Leibniz-Zentrum für Informatik, 2019.

3 Computing the highway dimension

Andreas Emil Feldmann

The *highway dimension* is a graph parameter that models transportation networks, and was introduced by Abraham et al. [2]. In an edge-weighted graph $G = (V, E)$, given a vertex $v \in V$ and a scale $r \in \mathbb{R}^+$, we define a set system that consists of shortest paths at distance at most $2r$ from v: for every shortest path P in G (between any vertex pair according to the edge weights) of length in the interval $(r, 2r]$, the set system contains the set of vertices of P if P contains at least one vertex at distance at most $2r$ from v. The *highway dimension* of G is the smallest integer h such that there exists a hitting set $H \subseteq V$ of size at most h for every such set system (i.e., for every vertex and scale).

It is known that computing the highway dimension is NP-hard by a reduction from Vertex Cover [5]. On the other hand, assuming (w.l.o.g.) that all shortest paths are unique, the set systems of shortest paths as defined above have VC-dimension 2 [1]. This implies the existence of a polynomial time $O(\log h)$-approximation algorithm [1]. The main question is whether computing the highway dimension is FPT. While the above-mentioned reduction from Vertex Cover does not rule out an FPT algorithm, it is known that computing hitting sets for set systems of VC-dimension 2 is W[1]-hard [4]. The set systems constructed for the latter reduction however are not easily converted into set systems of shortest paths, and thus it is not clear whether this can be used to show hardness to compute the highway dimension.

Even if it turns out that computing the highway dimension is W[1]-hard, it would still be very useful to have a fixed-parameter (or polynomial-time) approximation algorithm beating the known $O(\log h)$ factor. In fact, it is not ruled out that a parameterized approximation scheme exists to compute the highway dimension, and a related result [3] on the Max Coverage problem indicates that this might be possible. An improved (parameterized) approximation algorithm for highway dimension would on one hand improve the runtime of many known algorithms for graphs of low highway dimension, and would on the other hand have practical applications to verify that transportation networks of various kinds (for instance derived from road networks, public transportation, and air traffic) have small highway dimension.

References

[1] I. Abraham, D. Delling, A. Fiat, A. Goldberg, and R. Werneck. VC-dimension and shortest path algorithms. In *ICALP*, pages 690–699, 2011.

[2] I. Abraham, A. Fiat, A. V. Goldberg, and R. F. Werneck. Highway dimension, shortest paths, and provably efficient algorithms. In *SODA*, pages 782–793, 2010.

[3] A. Badanidiyuru, R. Kleinberg, and H. Lee. Approximating low-dimensional coverage problems. In *Proceedings of the twenty-eighth annual symposium on Computational geometry*, pages 161–170. ACM, 2012.

[4] K. Bringmann, L. Kozma, S. Moran, and N. S. Narayanaswamy. Hitting set for hypergraphs of low vc-dimension. In *24th Annual European Symposium on Algorithms, ESA 2016, August 22-24, 2016, Aarhus, Denmark*, pages 23:1–23:18, 2016.

[5] A. E. Feldmann, W. S. Fung, J. Könemann, and I. Post. A $(1+\varepsilon)$-embedding of low highway dimension graphs into bounded treewidth graphs. *SIAM Journal on Computing*, 47(4):1275–1734, 2018.

4 Steiner trees with few Steiner vertices

Andreas Emil Feldmann

The Steiner Tree problem is a classic well-studied problem. The input consists of an edge-weighted graph $G = (V, E)$, and terminal set $R \subseteq V$. The goal is to compute a minimum weight tree $T \subseteq G$ containing all of R, and possibly some non-terminals of $V \setminus R$ (which are called *Steiner vertices*). It is well-known that Steiner Tree is FPT parameterized by the number of terminals $|R|$ and can be solved in $3^{|R|}n^{O(1)}$ time [3] (for unweighted graphs even in $2^{|R|}n^{O(1)}$ time [1]). An alternative (and in some sense dual) parameter is the number $k = |V(T) \setminus R|$ of Steiner vertices in the optimum solution T, and a folklore result says that the problem is W[2]-hard for this parameter k. At the same time the Steiner Tree problem is APX-hard [2], and thus has no PTAS, unless P=NP. On the other hand, a fixed-parameter approximation scheme with runtime $2^{O(k^2/\varepsilon^4)} \cdot n^{O(1)}$ exists [4]. In the unweighted case this can be improved to $2^{O(k/\varepsilon)} \cdot n^{O(1)}$.

The first problem is to improve the runtime of the approximation scheme. For instance, it might be possible to match the runtime for the unweighted case even in the weighted case. The second problem is to prove any runtime lower bound for fixed-parameter approximation schemes, e.g., of the form $2^{o(k/\varepsilon)} \cdot n^{O(1)}$, using some reasonable complexity assumption (e.g., ETH or Gap-ETH).

References

[1] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto. Fourier meets Möbius: fast subset convolution. In *STOC 2007*, pages 67–74, 2007.

[2] M. Chlebík and J. Chlebíková. Approximation hardness of the Steiner tree problem on graphs. In *SWAT 2002*, pages 170–179.

[3] S. E. Dreyfus and R. A. Wagner. The Steiner problem in graphs. *Networks*, 1(3):195–207, 1971.

[4] P. Dvorák, A. E. Feldmann, D. Knop, T. Masarík, T. Toufar, and P. Veselý. Parameterized approximation schemes for steiner trees with small number of steiner vertices. In *35th Symposium on Theoretical Aspects of Computer Science, STACS*, pages 26:1–26:15, 2018.

5 Dominating Set parameterized by treedepth

Michał Pilipczuk

The *treedepth* of a graph G is the least integer d for which there exists a rooted forest F on the vertex set of G such that the depth of F is at most d and every edge of G connects two vertices that are in the ancestor-descendant relation in F. Such a forest F is called the *treedepth decomposition* of G. It is easy to see that the treedepth of a graph is not larger than its pathwidth plus one, which in turn is not larger than the treewidth plus one.

Standard dynamic programming solves the DOMINATING SET problem in time $3^t \cdot n^{\mathcal{O}(1)}$, where t is the width of a given tree decomposition. This is known to be tight even for pathwidth: unless SETH fails, there is no algorithm for DOMINATING SET with running time $(3-\varepsilon)^p \cdot n^{\mathcal{O}(1)}$, where p is the width of a given path decomposition [1]. However, it is unclear what happens for treedepth: is there an algorithm for DOMINATING SET with running time $2.99^d \cdot n^{\mathcal{O}(1)}$, where d is the depth of a given treedepth decomposition? Many tight lower bounds given in [1] for other problems can be extended to the treedepth paramerization, but this is not known in the particular case of DOMINATING SET.

REFERENCES

[1] D. Lokshtanov, D. Marx, and S. Saurabh. Known algorithms on graphs of bounded treewidth are probably optimal. *ACM Trans. Algorithms*, 14(2):13:1–13:30, 2018.

6 Kernelization for H-COLORING parameterized by vertex cover

Bart M.P. Jansen, Astrid Pieterse, Paweł Rzążewski

Jansen and Pieterse [1] considered the q-COLORING problem parameterized by the size k of a vertex cover. They obtained a kernel of size $\widetilde{\mathcal{O}}(k^{q-1})$, for every $q \geq 3$, which appears to be essentially optimal under standard assumptions.

Furthermore, the authors generalized their argument to the H-COLORING problem, in which we ask whether a given a graph G has a homomorphism to a fixed graph H. For this problem the size of the kernel is $\widetilde{\mathcal{O}}(k^{\Delta(H)})$.

Since K_q-COLORING is equivalent to q-COLORING and $\Delta(K_q) = q-1$, the bound $\widetilde{\mathcal{O}}(k^{\Delta(H)})$ is essentially tight for complete graphs. The only other family of graphs H, for which we know a matching lower bound, are odd cycles. Is $\widetilde{\mathcal{O}}(k^{\Delta(H)})$ optimal for other graphs H?

REFERENCES

[1] B. M. P. Jansen and A. Pieterse. Optimal data reduction for graph coloring using low-degree polynomials. *Algorithmica*, 81(10):3865–3889, 2019.

7 Many visits TSP

Lukasz Kowalik

The many-visits traveling salesperson problem (MV-TSP) asks for an optimal tour of n cities in a complete weighted digraph that visits each city v a prescribed number k_v of times. It is allowed to use the same edge multiple times and visiting a city twice in a row may incur a non-zero cost.

A standard DP works in time $O(n^2 \prod_v (k_v + 1))$. This is terrible for large k_v's. In 1984 Cosmadakis and Papadimitriou proposed an algorithm in time $n^{O(n)} + n^{O(1)} \log k$, where $k = \sum_v k_v$. Recently, Berger et al. [1, 2] improved it to $16^{n+o(n)} + n^{O(1)} \log k$ using a simpler algorithm. Can we improve the constant 16? Why should it be harder than the classic TSP?

REFERENCES

[1] A. Berger, L. Kozma, M. Mnich, and R. Vincze. Time- and space-optimal algorithms for the many-visits TSP. *CoRR*, abs/1804.06361, 2018.

[2] A. Berger, L. Kozma, M. Mnich, and R. Vincze. A time- and space-optimal algorithm for the many-visits TSP. In *SODA*, pages 1770–1782. SIAM, 2019.

8 Max Independent Set in H-free graphs

Édouard Bonnet

For a fixed graph H, the class of *H-free graphs* is the set of all graphs which do not contain H as an induced subgraph. Establishing a complete dichotomy, separating the graphs H for which Max Independent Set restricted to H-free graphs is in P and those for which it remains NP-complete, has been raised as an open question not long after the theory of intractability emerged. On the complexity side, if a connected graph H is not a path nor a subdivided claw (the claw is the biclique $K_{1,3}$) then Max Independent Set is NP-complete on H-free graphs [1]. Let us denote by P_ℓ the path on ℓ vertices, and for three positive integers i, j, k, let $S_{i,j,k}$ be the tree with exactly one vertex of degree three, from which start three paths with i, j, and k edges, respectively. The set $\{S_{i,j,k}\}_{i,j,k}$ thus consists of all the subdivisions of the claw. On the positive side, there is a polynomial-time algorithm on P_6-free graphs, by Grzesik et al. [6], and on fork-free graphs (the fork is $S_{1,1,2}$) [2, 7]. The mainstream belief seems to be that all the remaining open cases are tractable, eventhough the P_7-free case is wide open. Remarkably, a QPTAS was recently obtained on P_t-free graphs (and even $S_{i,j,k}$-free graphs) [5].

In [3, 4], we started to draw the frontier between the parameterized tractability (FPT) and intractability (W[1]-hard): what are the graphs H for which Max Independent Set on H-free graphs is FPT (and what are the graphs H for which it remains W[1]-complete)? Echoing the classical yet-to-obtain dichotomy, the problematic cases for H are roughly paths and subdivided claws where all the vertices can be substituted by cliques. The "first open questions" are when H is a P_4 *of cliques* (substitution by cliques of the four vertices of P_4), P_7, or even P_5 with the additional ambition to find a simpler argument than the polytime algorithm.

References

[1] V. E. Alekseev. The effect of local constraints on the complexity of determination of the graph independence number. *Combinatorial-Algebraic Methods in Applied Mathematics*, pages 3–13, 1982. in Russian.

[2] V. E. Alekseev. Polynomial algorithm for finding the largest independent sets in graphs without forks. *Discrete Applied Mathematics*, 135(1-3):3–16, 2004.

[3] É. Bonnet, N. Bousquet, P. Charbit, S. Thomassé, and R. Watrigant. Parameterized complexity of independent set in h-free graphs. In *13th International Symposium on Parameterized and Exact Computation, IPEC 2018, August 20-24, 2018, Helsinki, Finland*, pages 17:1–17:13, 2018.

[4] É. Bonnet, N. Bousquet, S. Thomassé, and R. Watrigant. When maximum stable set can be solved in FPT time. In *30th International Symposium on Algorithms and Computation, ISAAC 2019, December 8-11, 2019, Shanghai University of Finance and Economics, Shanghai, China*, pages 49:1–49:22, 2019.

[5] M. Chudnovsky, M. Pilipczuk, M. Pilipczuk, and S. Thomassé. Quasi-polynomial time approximation schemes for the maximum weight independent set problem in h-free graphs. *CoRR*, abs/1907.04585, 2019.

[6] A. Grzesik, T. Klimosova, M. Pilipczuk, and M. Pilipczuk. Polynomial-time algorithm for maximum weight independent set on P_6-free graphs. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1257–1271, 2019.

[7] V. V. Lozin and M. Milanič. A polynomial algorithm to find an independent set of maximum weight in a fork-free graph. *J. Discrete Algorithms*, 6(4):595–604, 2008.

9 Tight bounds for constant congestion brambles

Marcin Pilipczuk

Let G be a (simple, undirected) graph. A *bramble* is a collection $\mathcal{B}$ of connected subgraphs of G such that for every two $B_1, B_2 \in \mathcal{B}$, either $V(B_1) \cap V(B_2) \neq \emptyset$ or there is an edge of G with one endpoint in

B_1 and one endpoint in B_2. A *hitting set* of a bramble is a set $X \subseteq V(G)$ that intersects every element of a bramble and the *order* of a bramble is the minimum size of a hitting set. The *size* of the bramble is $|\mathcal{B}|$ Finally, the *congestion* (sometimes called *depth*) of a bramble is the maximum, over all vertices $v \in V(G)$, of the number of elements of $\mathcal{B}$ that contain v. Note that the size of the bramble is bounded by the product of the order and the congestion.

Brambles are tight duals to treewidth: the maximum possible order of a bramble in G is exactly treewidth$(G) + 1$. However, as shown by Grohe and Marx [1], sometimes brambles of high order need to have exponential size: for infinitely many k, there exists a graph of treewidth k such that for all $1/2 > \delta > 0$ every bramble of order $k^{1/2+\delta}$ has size $2^{\Omega(k^{\delta})}$. Furthermore, if one asks for brambles of polynomial size, they show a matching bound: there is always a bramble of order $\Omega(k^{1/2}/\log^2 k)$ and congestion $\widetilde{\mathcal{O}}(k)$ (so size $\widetilde{\mathcal{O}}(k^{3/2})$).

For many applications, complex objects (such as brambles) of *constant* congestion are of particular importance. The lower bound of Grohe and Marx excludes the existence of constant congestion brambles of order $k^{1/2+\delta}$ for any $\delta > 0$. However, there is no matching upper bound: the best construction is due to Reed and Wood [2] that gives a bramble of congestion 2 and order $\Omega(k^{1/4}/\sqrt{\log k})$.

We conjecture that there exists a constant c and a polynomial p such that for every graph G if $k = \text{treewidth}(G)+1$, then G contains a bramble of congestion at most c and order at least $k^{1/2}/p(\log k)$.

References

[1] M. Grohe and D. Marx. On tree width, bramble size, and expansion. *J. Comb. Theory, Ser. B*, 99(1):218–228, 2009.

[2] B. A. Reed and D. R. Wood. Polynomial treewidth forces a large grid-like-minor. *Eur. J. Comb.*, 33(3):374–379, 2012.

10 Polynomial kernel for Genuinely Planar Edge Multicut

Bart M.P. Jansen, Stefan Kratsch, Marcin Pilipczuk, Manuel Sorge, and Erik Jan van Leeuwen

The input to Edge Multicut consists of an undirected graph G, a set $\mathcal{T} \subseteq \binom{V(G)}{2}$ of terminal pairs (called *requests*) and an integer (parameter) k. The goal is to select a set $X \subseteq E(G)$ of at most k edges such that for every $st \in \mathcal{T}$, s and t lie in different connected components of $G - X$.

The instance is *planar* if G is planar and *genuinely planar* if $G + \mathcal{T} := (V(G), E(G) \cup \mathcal{T})$ is planar (i.e., you can draw G and all the requests of $\mathcal{T}$ as extra edges in one planar drawing). Does Genuinely Planar Edge Multicut admit a polynomial kernel when parameterized by k?

The Multiway Cut version, where $\mathcal{T} = \binom{T}{2}$ for some $T \subseteq V(G)$, is known to have a polynomial kernel for fixed $|T|$ in arbitrary graphs [4] and polynomial kernel in planar graphs [5], also for the vertex-deletion variant [3]. Edge Multicut has no polynomial kernel in general graphs [1] and we found a way to modify the above lower bound to exclude a polynomial kernel for Planar Edge Multicut, even in graphs of constant treewidth.

We also proved that Genuinely Planar Edge Multicut is polynomial-time solvable if all the requests are in one face of G. The algorithm here is an involved dynamic programming algorithm inspired by the algorithm for Planar Steiner Tree with all terminals on one face [2].

The vertex-deletion version of Genuinely Planar Edge Multicut is trivially NP-hard by a reduction from Planar Vertex Cover. However, the NP-hardness of edge deletion is much trickier, but true.

We conjecture that Genuinely Planar Edge Multicut admits a polynomial kernel and we expect that the kernelization algorithm uses the toolbox for Planar Edge Multiway Cut [5].

References

[1] M. Cygan, S. Kratsch, M. Pilipczuk, M. Pilipczuk, and M. Wahlström. Clique cover and graph separation: New incompressibility results. *TOCT*, 6(2):6:1–6:19, 2014.

[2] R. E. Erickson, C. L. Monma, and A. F. J. Veinott. Send-and-split method for minimum-concave-cost network flows. *Mathematics of Operations Research*, 12(4):634–664, 1987.

[3] B. M. P. Jansen, M. Pilipczuk, and E. J. van Leeuwen. A deterministic polynomial kernel for odd cycle transversal and vertex multiway cut in planar graphs. In R. Niedermeier and C. Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany*, volume 126 of *LIPIcs*, pages 39:1–39:18. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2019.

[4] S. Kratsch and M. Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 450–459. IEEE Computer Society, 2012.

[5] M. Pilipczuk, M. Pilipczuk, P. Sankowski, and E. J. van Leeuwen. Network sparsification for steiner problems on planar and bounded-genus graphs. *ACM Trans. Algorithms*, 14(4):53:1–53:73, 2018.

11 Hardness of Euclidean TSP with Hyperplane Neighborhoods

Krzysztof Fleszar

Problem statement. Given n hyperplanes in $\mathbb{R}^d$, find a minimum-length tour visiting at least one point from each hyperplane.

Known results. Efficiently solvable for dimension $d = 2$ (lines in the plane) [2], PTAS for constant d (e.g., planes in $\mathbb{R}^3$) [1].

Open problem. Is the problem NP-hard for $d \geq 3$?

References

[1] A. Antoniadis, K. Fleszar, R. Hoeksma, and K. Schewior. A PTAS for Euclidean TSP with hyperplane neighborhoods. In *Proc. of ACM-SIAM Symposium on Discrete Algorithms (SODA 2019)*, pages 1089–1105. Society for Industrial and Applied Mathematics, 2019.

[2] H. Jonsson. The traveling salesman problem for lines in the plane. *Information Processing Letters*, 82(3):137–142, 2002.

12 Polynomial kernel for Bicolored P_3-free Edge Deletion

Astrid Pieterse, Eva-Maria Hols, Michał Pilipczuk, Marcin Pilipczuk, Stefan Kratsch, Manuel Sorge

Grüttemeier et al. [1] studied the Bicolored P_3-free Edge Deletion problem wherein we are given a graph whose edges are colored with two colors and an integer k. We want to decide whether we can delete at most k edges such that in the resulting graph there is no induced P_3 with two different colors. Grüttemeier et al. obtained a polynomial-size problem kernel with respect to k and the maximum degree. Is there a polynomial-size problem kernel with respect to only k?

During the last Karpacz workshop we worked on this problem, with not much progress. However, there is a problem of similar nature that could be helpful to study. An edge-coloring of a complete graph in three colors is called a *Gallai coloring* if there is not triangle with all three edges with different colors. The problem is as follows: given some edge-coloring of a complete graph with three colors, is it possible to change colors on at most k edges to obtain a Gallai coloring? Is there a polynomial-size kernel for this problem with respect to parameter k? The structure of such colorings was studied by Gallai in the 60s, see modern discussion in [2].

References

[1] N. Grüttemeier, C. Komusiewicz, J. Schestag, and F. Sommer. Destroying bicolored p_3s by deleting few edges. In *Computing with Foresight and Industry - 15th Conference on Computability in Europe, CiE 2019, Durham, UK, July 15-19, 2019, Proceedings*, volume 11558 of *Lecture Notes in Computer Science*, pages 193–204. Springer, 2019.

[2] A. Gyárfás and G. Simony. Edge colorings of complete graphs without tricolored triangles. *Journal of Graph Theory*, 46(3):211–216, 2004.

13 Matching-mimicking networks

Magnus Wahlström

This problem is essentially from David Eppstein's blog (`https://11011110.github.io/blog/2019/05/25/more-matching-mimicking.html`) but was also previously investigated in a paper of Eppstein and Vazirani [1]. Given a graph $G = (V, E)$ and a set of k terminals $T \subseteq V$, the *matching pattern* of G and T is the collection

$$\mathcal{T} = \{T' \subseteq T \mid G - (T \setminus T') \text{ has a perfect matching}\}.$$

A *matching-mimicking network* for G and T is a graph $G' = (V', E')$ with $T \subseteq V'$ such that (G, T) and (G', T) have the same matching pattern. The question, of course, if we can compute such a graph G' in polynomial time and with $|V(G')|$ polynomial in k. No reasonable bound is known, even for existence (that is, I don't think the problem has been studied too extensively). But there is a polynomial-sized (randomized) algebraic representation: There is a $|T| \times |T|$ skew-symmetric matrix A such that the matching pattern of (G, T) corresponds to sets $T' \subseteq T$ such that $A[T', T']$ is non-singular. (I'm cutting a small technical corner here, but not a significant one.) I'm not aware of any formal writeup of this fact, but it follows directly from applying the so-called *pivoting* operation to the Tutte matrix of G (see Geelen at al. [2] for a clear and succinct presentation of the technical terms here, even though the rest of [2] is on a much more complex topic than we're considering here).

The question of matching-mimicking networks appears in some sense to be a more difficult version of cut-covering sets, which have been used in the matroid-based kernels for cut problems [3]: Given a digraph D and a set of terminals $T \subseteq V(D)$ with $|T| = k$, we can find a set $Z \subseteq V(D)$ with $|Z| = O(k^3)$ such that for any two sets $A, B \subseteq T$, there is an (A, B)-min cut X in D with $X \subseteq Z$. It may be no surprise that such cut-questions correspond to the bipartite case of matching-mimicking networks. There is even a variant of a representative set statement for delta-matroids (unpublished, but I'm happy to share my notes), but it doesn't seem to solve the problem without some further combinatorial ingredients.

References

[1] D. Eppstein and V. V. Vazirani. NC algorithms for computing a perfect matching, the number of perfect matchings, and a maximum flow in one-crossing-minor-free graphs. In *SPAA*, pages 23–30. ACM, 2019.

[2] J. F. Geelen, S. Iwata, and K. Murota. The linear delta-matroid parity problem. *J. Comb. Theory, Ser. B*, 88(2):377–398, 2003.

[3] S. Kratsch and M. Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 450–459. IEEE Computer Society, 2012.

14 Constant-factor approximation for loaded treewidth

Manuel Sorge

A *loaded* graph is a graph with a vertex bipartition into *heavy* and *light* vertices. The *load* of a tree decomposition of a graph is the least integer c such that each bag contains at most c heavy vertices. In recent unpublished work Ganian, Schidler, Szeider, and Sorge introduced and studied loaded treewidth. It is not too difficult to observe that it is fixed-parameter tractable with respect to k to determine for a given c and graph whether there is a tree decomposition of width at most k and load at most c. Treewidth has nice constant-factor approximation algorithms that run in $2^{O(k)} \cdot$ poly-time. Does the same hold for loaded treewidth? That is, is there a constant d and a $2^{O(k)} \cdot$ poly-time algorithm that, given a graph G and integers c, k, computes a tree decomposition for G of width at most dk and load at most c or correctly reports that no tree decomposition of width at most k and load c exists?

The constant-factor algorithms for treewidth recursively identify balanced separators for some subset W of vertices. The separator and each part of W then form a bag in the resulting decomposition. However, there is an example that shows that each balanced separator of the required size needs to increase the number of heavy vertices in the resulting bag.

15 Scheduling on m parallel machines exactly

Céline M. F. Swennenhuis, Karol Węgrzycki

You are given m identical parallel machines and n tasks, each with some preprocessing time p. Let us define $C_i(s)$ to be the completion time of the job i in a given schedule s. The aim of scheduling is to allocate the jobs to the machines in order to minimize the *completion time* $C_{\max} = \max_{1 \leq i \leq n}(C_i)$.

For $m = 2$ parallel machines, the problem is related to Partition Problem, and admits a naive $\mathcal{O}^\star(2^n)$ time, polynomial space algorithm and $\mathcal{O}^\star(2^{n/2})$ time, exponential space algorithm.

Lenté et al. [2], considered a problem for $m = 3$ and proposed $\mathcal{O}^\star(3^n)$ time $\mathcal{O}^\star(1)$ space algorithm and $\mathcal{O}^\star(3^{n/2})$ time $\mathcal{O}^\star(3^{n/2})$ space algorithm. For general m they show $\mathcal{O}^\star(m^{n/2})$ time algorithm.

1. Can we improve their polynomial space algorithm, for some m? (perhaps with technique of [1])
2. Partition admits $\mathcal{O}^\star(2^{n/2})$ time, $\mathcal{O}^\star(2^{n/4})$ space alg. Can we say something similar about P3$||C_{\max}$?
3. Give $\mathcal{O}^\star((3-\varepsilon)^{n/2})$ algorithm for P3$||C_{\max}$ or propose a conditional lower bound.

References

[1] N. Bansal, S. Garg, J. Nederlof, and N. Vyas. Faster space-efficient algorithms for subset sum and k-sum. In H. Hatami, P. McKenzie, and V. King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 198–209. ACM, 2017.

[2] C. Lenté, M. Liedloff, A. Soukhal, and V. T'Kindt. On an extension of the sort & search method with application to scheduling theory. *Theor. Comput. Sci.*, 511:13–22, 2013.

16 Graph modification problems for permutation graphs

Michał Pilipczuk

For a hereditary class of graphs $\mathcal{C}$, the parameterized $\mathcal{C}$ Vertex Deletion problem is as follows: given a graph G and parameter k, can one delete at most k vertices from G to obtain a graph from $\mathcal{C}$? When $\mathcal{C}$ is characterized by a finite number of forbidden subgraphs, then this problem is FPT by straightforward branching. However, for more complex classes $\mathcal{C}$, the line between being FPT and W-hard becomes quite thin and establishing fixed-parameter tractability for specific classes $\mathcal{C}$ often requires a good insight into the structure of graphs from $\mathcal{C}$. For instance, Interval Vertex Deletion is FPT [1], but Perfect Vertex Deletion is already W[2]-hard [3].

One class that so far received little attention is the class of *permutation graphs*. Here, a graph G is a permutation graph if it admits an intersection model that looks as follows: there are two parallel lines ℓ_1 and ℓ_2 in the plane, every vertex of G is modelled as a segment with one endpoint on ℓ_1 and second on ℓ_2, and two vertices are adjacent if and only if their segments intersect. To the best of our knowledge, it is still open whether Permutation Graphs Vertex Deletion is FPT. We expect that the answer might be positive, as permutation graphs admit a lot of modular-like structure that might be useful, as was the case for interval graphs [1]. Fixed-paramter tractability of the edge modification variants also seems open. Finally, there is non-trivial combinatorial recognition algorithm for permutation graphs (see e.g. [2]) that uncovers connections to comparability and co-comparability graphs. To the best of our knowledge, the parameterized complexity of the modification problems for those classes is open as well.

References

[1] Y. Cao and D. Marx. Interval deletion is fixed-parameter tractable. *ACM Trans. Algorithms*, 11(3):21:1–21:35, 2015.

[2] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, New York, 1980.

[3] P. Heggernes, P. van't Hof, B. M. P. Jansen, S. Kratsch, and Y. Villanger. Parameterized complexity of vertex deletion into perfect graph classes. In *Fundamentals of Computation Theory - 18th International Symposium, FCT 2011*, volume 6914 of *Lecture Notes in Computer Science*, pages 240–251. Springer, 2011.

17 Packing arc-disjoint triangles in tournaments

Michał Pilipczuk

We consider the following parameterized problem: given a tournament T and integer k, is it possible to find k arc-disjoint directed triangles in T? It is known that this problem can be solved in time $2^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ and there is a lower bound excluding time complexity $2^{o(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$ [2]. Since the dual Feedback Arc Set in Tournaments can be solved in time $2^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)}$ [1, 3, 4], it is natural to ask whether a similar running time can be achieved also for packing arc-disjoint triangles.

References

[1] N. Alon, D. Lokshtanov, and S. Saurabh. Fast FAST. In *ICALP 2009*, volume 5555 of *Lecture Notes in Computer Science*, pages 49–58. Springer, 2009.

[2] S. Bessy, M. Bougeret, R. Krithika, A. Sahu, S. Saurabh, J. Thiebaut, and M. Zehavi. Packing arc-disjoint cycles in tournaments. In *MFCS 2019*, volume 138 of *LIPIcs*, pages 27:1–27:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[3] U. Feige. Faster FAST (Feedback Arc Set in Tournaments). *CoRR*, abs/0911.5094, 2009.

[4] F. V. Fomin and M. Pilipczuk. On width measures and topological problems on semi-complete digraphs. *J. Comb. Theory, Ser. B*, 138:78–165, 2019.

Change log

21 Feb 2020	tournaments from Michał
03 Feb 2020	loaded treewidth (Manuel), scheduling (Céline and Karol), modification (Michał)
30 Jan 2020	a problem from Magnus
29 Jan 2020	bicolored P_3s by Manuel
28 Jan 2020	a problem from Krzysztof
21 Jan 2020	constant congestion brambles problem by Marcin
20 Jan 2020	a problem from Édouard
17 Jan 2020	a problem from Łukasz
08 Jan 2020	a problem from Bart, Astrid, and Paweł
07 Jan 2020	two problems from Andreas and one problem from Michał
11 Dec 2019	first version: two problems from Michał