Open problem list for the Fine-grained Complexity Design program

September 23, 2015

Problems from older open problem lists of FPT community

1 Even Set aka Minimum Codeword

Long standing; appeared, e.g., in [1].

In the Even Set problem, the input consists of a family $\mathcal{F}$ of subsets of a universe U and an integer k; the question is to find a nonempty set $A \subseteq U$ of size at most k such that $|A \cap F|$ is even for every $F \in \mathcal{F}$. Alternatively, the question can be stated as finding a nonzero codeword of Hamming weight at most k in a linear code over $\mathbb{F}_2$. The question of parameterized complexity of this problem, parameterized by k, remains open.

Note that if we require the set A to be of size exactly k, or we require the intersections to be odd, the problem becomes W[1]-hard.

References

[1] M. R. Fellows, J. Guo, D. Marx, and S. Saurabh. Data Reduction and Problem Kernels (Dagstuhl Seminar 12241). *Dagstuhl Reports*, 2(6):26–50, 2012.

2 Planar Independent Set above guarantee

Appeared in [3, 1].

A direct consequence of the Four Colour Theorem is that every n-vertex planar graph admits an independent set of size at least $n/4$. Consider the following question: given an n-vertex planar graph G and an integer k, we ask if G admits an independent set of size at least $(n+k)/4$. Is this problem FPT when parameterized by k?

A related question of being "above guarantee" for the triangle-free planar graphs has been resolved positively in [2].

References

[1] G. Borradaile, P. Klein, D. Marx, and C. Mathieu. Algorithms for Optimization Problems in Planar Graphs (Dagstuhl Seminar 13421). *Dagstuhl Reports*, 3(10):36–57, 2014.

[2] Z. Dvorak and M. Mnich. Large independent sets in triangle-free planar graphs. *CoRR*, abs/1311.2749, 2013.

[3] M. R. Fellows, J. Guo, D. Marx, and S. Saurabh. Data Reduction and Problem Kernels (Dagstuhl Seminar 12241). *Dagstuhl Reports*, 2(6):26–50, 2012.

3 Subgraph isomorphism in planar graphs parameterized by the difference

Appeared in [1].

Consider a Subgraph Isomorphism problem, parameterized by the difference $|E(G)| - |E(H)|$. Is it fixed-parameter tractable on planar graphs? Recall that the Graph Isomorphism problem on planar graphs is polynomial, due to (a) uniqueness of the embedding of 3-connected planar graphs, and (b) uniqueness of the Tutte's decomposition into 3-connected components.
In the setting of *counting* subgraphs, a related problem is #W[1]-hard: Given a planar graph G and a number k, it is #W[1]-hard to count those matchings of G that leave exactly k vertices unmatched. [3, 2]

References

[1] G. Borradaile, P. Klein, D. Marx, and C. Mathieu. Algorithms for Optimization Problems in Planar Graphs (Dagstuhl Seminar 13421). *Dagstuhl Reports*, 3(10):36–57, 2014.

[2] R. Curticapean. *The simple, little and slow things count: on parameterized counting complexity*. PhD thesis, Saarland University, August 2015.

[3] R. Curticapean and M. Xia. Parameterizing the permanent: Genus, apices, minors, evaluation mod 2^k. In *56th IEEE Annual Symposium on Foundations of Computer Science, FOCS. To appear.*

4 Eulerian SCC Deletion

Appeared originally in [1], asked in [3, 2].

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. Somehow related deletion problems were studied in [3]. 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 [3].

References

[1] K. Cechlárová and I. Schlotter. Computing the deficiency of housing markets with duplicate houses. In V. Raman and S. Saurabh, editors, *IPEC*, volume 6478 of *Lecture Notes in Computer Science*, pages 72–83. Springer, 2010.

[2] M. Cygan, L. Kowalik, and M. Pilipczuk. Open problems from workshop on kernels, 2013. http://worker2013.mimuw.edu.pl/slides/worker-opl.pdf.

[3] M. Cygan, D. Marx, M. Pilipczuk, M. Pilipczuk, and I. Schlotter. Parameterized complexity of Eulerian deletion problems. *Algorithmica*, 68(1):41–61, 2014.

5 Chain SAT

Asked in [1].

In the ℓ-Chain SAT problem we are given a set of n Boolean variables, a set of constraints of the form $x_1 \Rightarrow x_2 \Rightarrow \ldots \Rightarrow x_r$ where $r \leq \ell$, and an integer k. The question is to delete at most k constraints to obtain a satisfiable instance. For a fixed integer ℓ, does this problem admit an FPT algorithm, parameterized by k?

References

[1] R. H. Chitnis, L. Egri, and D. Marx. List H-coloring a graph by removing few vertices. In H. L. Bodlaender and G. F. Italiano, editors, *ESA*, volume 8125 of *Lecture Notes in Computer Science*, pages 313–324. Springer, 2013.

6 Solving Integer Linear Programming in single-exponential time in the number of variables

Appeared in [2].

Given a matrix $A \in \mathbb{Z}^{m\times p}$ and a vector $b \in \mathbb{Z}^m$, we look for a vector $x \in \mathbb{Z}^p$ such that $Ax \leq b$. Is there a $c^p \cdot m^{\mathcal{O}(1)}$-time algorithm? The best known has $p^{\mathcal{O}(p)}$ dependency on p in the running time bound [1].

References

[1] R. Kannan. Minkowskis convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415440, 1987.

[2] S. Kratsch, D. Lokshtanov, D. Marx, and P. Rossmanith. Optimality and tight results in parameterized complexity (Dagstuhl Seminar 14451). *Dagstuhl Reports*, 4(11):1–21, 2015.

7 Faster Subset Sum

Long-standing, see [1].

In the Subset Sum problem we are given n integers $x_1, x_2, \ldots, x_n$ and an integer M and we ask if there exists a set $S \subseteq \{1, 2, \ldots, n\}$ such that $\sum_{i\in S} x_i = M$. A brute-force algorithm solves this problem in $\mathcal{O}^*(2^n)$ time and polynomial space, whereas a simple meet-in-the-middle approach gives $\mathcal{O}^*(2^{n/2})$ time and space. Can any of these bounds be exponentially improved? That is, we ask for an $\mathcal{O}(c^n)$-time algorithm with polynomial space for some $c < 2$ or an $\mathcal{O}(c^{n/2})$-time algorithm for some $c < 2$ that may use exponential space.
Recent advances on the second question include [3, 2].

References

[1] P. Austrin, P. Kaski, M. Koivisto, and J. Määttä. Space-time tradeoffs for subset sum: An improved worst case algorithm. In F. V. Fomin, R. Freivalds, M. Z. Kwiatkowska, and D. Peleg, editors, *ICALP (1)*, volume 7965 of *Lecture Notes in Computer Science*, pages 45–56. Springer, 2013.

[2] P. Austrin, P. Kaski, M. Koivisto, and J. Nederlof. Dense subset sum may be the hardest. *CoRR*, abs/1508.06019, 2015.

[3] P. Austrin, P. Kaski, M. Koivisto, and J. Nederlof. Subset Sum in the Absence of Concentration. In E. W. Mayr and N. Ollinger, editors, *32nd International Symposium on Theoretical Aspects of Computer Science (STACS 2015)*, volume 30 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 48–61, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

8 From Parity Set Cover to Parity SAT

Appeared in [3].

Given a universe U of size n, a set family $\mathcal{F} \subseteq 2^U$, and an integer t, $\oplus$-Set Cover asks if the number of subsets $\mathcal{X} \subseteq \mathcal{F}$ of size at most t such that $\bigcup \mathcal{X} = U$ is odd. Similarly, $\oplus$-SAT asks if the number of satisfying assignments of a SAT formula is odd.

A serf-reduction is a reduction that preserves sub-exponential running times. Meaning that the resulting instance should be linear in the original instance and not require more than sub-exponential time to compute [2].

Cygan et al. [1] gave a serf-reduction from $\oplus$-SAT to $\oplus$-SET COVER. Is there a serf-reduction in the other direction?

REFERENCES

[1] M. Cygan, H. Dell, D. Lokshtanov, D. Marx, J. Nederlof, Y. Okamoto, R. Paturi, S. Saurabh, and M. Wahlström. On problems as hard as CNF-SAT. In *Proceedings of the 27th Conference on Computational Complexity, CCC 2012, Porto, Portugal, June 26-29, 2012*, pages 74–84. IEEE, 2012.

[2] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

[3] S. Kratsch, D. Lokshtanov, D. Marx, and P. Rossmanith. Optimality and tight results in parameterized complexity (Dagstuhl Seminar 14451). *Dagstuhl Reports*, 4(11):1–21, 2015.

9 Faster algorithms for TSP and related problems

Appeared in [3].

The problem of finding a minimum/maximum cost Hamiltonian cycle can be solved in $\mathcal{O}^*(2^n)$ time by a standard dynamic-programming algorithm. Can it be solved in $\mathcal{O}(c^n)$ time for some $c < 2$? The existence of a Hamiltonian cycle in undirected graphs can be detected in $\mathcal{O}(1.66^n)$ time [1], but the directed case is open (but the parity of the number of such cycles can be found quicker [2]). Also, even for a possibly simpler case of SHORTEST SUPERSTRING (given n strings, what is the shortest string that contains every input string as a subword) we do not know a faster algorithm than $\mathcal{O}^*(2^n)$.

REFERENCES

[1] A. Björklund. Determinant sums for undirected hamiltonicity. *SIAM J. Comput.*, 43(1):280–299, 2014.

[2] A. Björklund and T. Husfeldt. The parity of directed hamiltonian cycles. In *FOCS*, pages 727–735. IEEE Computer Society, 2013.

[3] T. Husfeldt, R. Paturi, G. B. Sorkin, and R. Williams. Exponential Algorithms: Algorithms and Complexity Beyond Polynomial Time (Dagstuhl Seminar 13331). *Dagstuhl Reports*, 3(8):40–72, 2013.

10 Faster algorithm for balanced QBF with constant number of alternations

Appeared in [1], motivated by [2].

Is there an $\varepsilon > 0$ and an integer $l \geq 1$ such that for every integer k and every integer n, for every k-CNF formula ϕ over n variables the question whether

$$\forall_{x_1,x_2,\ldots,x_{n/l}} \exists_{x_{n/l+1},\ldots,x_{2n/l}} \forall_{x_{2n/l+1},\ldots,x_{3n/l}} \ldots \exists_{x_{n-n/l+1},\ldots,x_n} \phi(x_1,\ldots,x_n)$$

is true can be decided in $\mathcal{O}(2^{(1-\varepsilon)n})$ time?

There are algorithms working in $2^{n-n^{1/l}} n^{\mathcal{O}(1)}$ time and in $2^{n-\Omega(l)} n^{\mathcal{O}(1)}$ time [2], in a more general setting where we do not require the formula to be balanced (i.e., the quantifier blocks may have varying lenghts). Significantly improved algorithms for the general case and $l = \mathcal{O}(\log n)$ alternations would have important implications in circuit complexity; see [2] for details. However, this particular question asks for potentially easier case of constant number of alternations with the balancedness assumption.

References

[1] S. Kratsch, D. Lokshtanov, D. Marx, and P. Rossmanith. Optimality and tight results in parameterized complexity (Dagstuhl Seminar 14451). *Dagstuhl Reports*, 4(11):1–21, 2015.

[2] R. Santhanam and R. Williams. Beating exhaustive search for quantified boolean formulas and connections to circuit complexity. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 231–241, 2015.

11 Faster algorithm for Maximum Acyclic Subgraph

Appeared in [1].

Given a directed graph G, the Maximum Acyclic Subgraph problem asks for an acyclic subgraph with maximum possible number of edges. A simple dynamic-programming algorithm solves this problem in $\mathcal{O}^*(2^n)$ time for n-vertex graphs. Is it possible to obtain a $\mathcal{O}(c^n)$-time algorithm for some $c < 2$?
Note that for the "induced subgraph" problem, where we maximize the number of vertices in the subgraph, the problem is equivalent to Directed Feedback Vertex Set, and the answer is positive [2].

References

[1] T. Husfeldt, R. Paturi, G. B. Sorkin, and R. Williams. Exponential Algorithms: Algorithms and Complexity Beyond Polynomial Time (Dagstuhl Seminar 13331). *Dagstuhl Reports*, 3(8):40–72, 2013.

[2] I. Razgon. Computing minimum directed feedback vertex set in $O(1.9977^n)$. In G. F. Italiano, E. Moggi, and L. Laura, editors, *ICTCS*, pages 70–81. World Scientific, 2007.

12 Algorithms for Cutwidth: exponential and on tree decompositions

The question on exponential-time algorithm appeared in [2] and in [1]. The question on treewidth algorithm appeared in [3].

The cutwidth of a graph G is the smallest integer k such that the vertices of G can be arranged in a linear layout $v_1, v_2, \ldots, v_n$ such that for every $1 \leq i < n$ there are at most k edges between $\{v_1, v_2, \ldots, v_i\}$ and $\{v_{i+1}, v_{i+2}, \ldots, v_n\}$.

1. A simple dynamic-programming algorithm finds cutwidth of an n-vertex graph in time $\mathcal{O}^*(2^n)$. Does there exist an algorithm running in time $\mathcal{O}(c^n)$ for some $c < 2$? Note that the problem can be solved in time $2^k n^{\mathcal{O}(1)}$ for graphs with vertex cover of size at most k [1].

2. Is there an XP-algorithm parameterized by treewidth or by the feedback vertex set number?

References

[1] M. Cygan, D. Lokshtanov, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. On cutwidth parameterized by vertex cover. *Algorithmica*, 68(4):940–953, 2014.

[2] T. Husfeldt, R. Paturi, G. B. Sorkin, and R. Williams. Exponential Algorithms: Algorithms and Complexity Beyond Polynomial Time (Dagstuhl Seminar 13331). *Dagstuhl Reports*, 3(8):40–72, 2013.

[3] S. Kratsch, D. Lokshtanov, D. Marx, and P. Rossmanith. Optimality and tight results in parameterized complexity (Dagstuhl Seminar 14451). *Dagstuhl Reports*, 4(11):1–21, 2015.

13 Lower bounds for determining treewidth

Appeared in [1].

Is there an interesting lower bound on time needed to compute the treewidth of an input graph? For example, a refutation of a $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$-time algorithm under ETH?

References

[1] S. Kratsch, D. Lokshtanov, D. Marx, and P. Rossmanith. Optimality and tight results in parameterized complexity (Dagstuhl Seminar 14451). *Dagstuhl Reports*, 4(11):1–21, 2015.

14 Faster algorithm for computing chromatic number

Based on [2], asket also in Mohan's third talk at the boot camp and in [3, Section 5]

With the use of Fast Subset Convolution or the inclusion–exclusion principle we can compute the chromatic number of an n-vertex graph in time and space $\mathcal{O}^*(2^n)$ or in time $\mathcal{O}^*(2.246^n)$ and polynomial space [2]. Can any of the running time bounds be improved? In particular, can we compute chromatic number in $\mathcal{O}^*(2^n)$ time and polynomial space? Or refute $\mathcal{O}^*((2-\varepsilon)^n)$-time algorithm under some plausible hypothesis?
Note: We can decide in time $O^*(1.3289^n)$ whether $\chi(G) \leq 3$ holds [1], and in time $O^*(1.7272^n)$ whether $\chi(G) \leq 4$ holds [4].

References

[1] R. Beigel and D. Eppstein. 3-coloring in time O(1.3289^n). *J. Algorithms*, 54(2):168–204, 2005.

[2] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto. Fourier meets Möbius: fast subset convolution. In D. S. Johnson and U. Feige, editors, *STOC*, pages 67–74. ACM, 2007.

[3] M. Cygan, H. Dell, D. Lokshtanov, D. Marx, J. Nederlof, Y. Okamoto, R. Paturi, S. Saurabh, and M. Wahlström. On problems as hard as CNFSAT. *CoRR*, abs/1112.2275, 2011.

[4] F. V. Fomin, S. Gaspers, and S. Saurabh. Improved exact algorithms for counting 3- and 4-colorings. In *Computing and Combinatorics, 13th Annual International Conference, COCOON 2007, Banff, Canada, July 16-19, 2007, Proceedings*, pages 65–74, 2007.

15 Chromatic index in c^n time

Folklore.

The chromatic index of a graph is the minimum number of colors in which we can color the edges of the graph such that no two edges with a common endpoint have the same color. The classic theorem of Vizing asserts that the chromatic index of a graph is either Δ or $\Delta+1$, where Δ is the maximum degree. However, detecting which is the case is NP-hard. Morever, it is open whether it can be done in $\mathcal{O}(c^n)$ time on n-vertex graphs, for some constant c.

16 Subexponential algorithms for planar problems

Appeared in [1].

Despite the robustness of the bidimensionality framework, for a few planar problems we still do not know whether they admit a subexponential algorithm. This includes:

- Longest Path in directed planar graphs;

- Weighted k-Path in undirected graphs (maximum weight path on k vertices);
- Exact k-Cycle (does there exist a cycle on exactly k vertices);
- Steiner Tree, parameterized by the number of terminals;
- Subgraph Isomorphism, parameterized by the size of the pattern graph.

References

[1] G. Borradaile, P. Klein, D. Marx, and C. Mathieu. Algorithms for Optimization Problems in Planar Graphs (Dagstuhl Seminar 13421). *Dagstuhl Reports*, 3(10):36–57, 2014.

17 Faster FPT algorithm for Feedback Vertex Set

Based on [2] and [4].

The fastest known FPT algorithms for Feedback Vertex Set run in $\mathcal{O}^*(3^k)$ randomized time [2] and $\mathcal{O}^*(3.62^k)$ deterministic time [4]. Can they be improved? In particular, we expect that it should be possible to obtain an $\mathcal{O}^*(3^k)$-time deterministic algorithm for the problem. Note that a similar result has been obtained for Connected Vertex Cover [1].
We remark here that the authors of [4] in their technical report [3] observed that it is relatively easy (but very tedious) to improve slightly the base of the exponent, at the cost of very extensive case analysis. This is not an improvement we are looking for.

References

[1] M. Cygan. Deterministic parameterized connected vertex cover. In F. V. Fomin and P. Kaski, editors, *SWAT*, volume 7357 of *Lecture Notes in Computer Science*, pages 95–106. Springer, 2012.

[2] M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J. M. M. van Rooij, and J. O. Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In R. Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 150–159. IEEE Computer Society, 2011.

[3] T. Kociumaka and M. Pilipczuk. Faster deterministic feedback vertex set. *CoRR*, abs/1306.3566, 2013.

[4] T. Kociumaka and M. Pilipczuk. Faster deterministic feedback vertex set. *Inf. Process. Lett.*, 114(10):556–560, 2014.

18 Faster algorithms for Odd Cycle Transversal and related problems

Appeared in [1].

The currently fastest FPT algorithm for Odd Cycle Transversal and Vertex Cover above LP runs in $\mathcal{O}^*(2.318^k)$ time [3]. The base of the exponent comes from branching vectors with complicated case analysis, so we do not expect it to be optimal. Can it be significantly improved? For example, to $\mathcal{O}^*(2^k)$?
A related question is to improve the $\mathcal{O}^*(2^k)$ algorithm for Edge Bipartization [2] has been very recently resolved in [4].

References

[1] M. Cygan, L. Kowalik, and M. Pilipczuk. Open problems from update meeting on graph separation problems, 2013. http://worker2013.mimuw.edu.pl/slides/update-opl.pdf.

[2] J. Guo, J. Gramm, F. Hüffner, R. Niedermeier, and S. Wernicke. Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization. *J. Comput. Syst. Sci.*, 72(8):1386–1396, 2006.

[3] D. Lokshtanov, N. S. Narayanaswamy, V. Raman, M. S. Ramanujan, and S. Saurabh. Faster parameterized algorithms using linear programming. *ACM Transactions on Algorithms*, 11(2):15:1–15:31, 2014.

[4] M. Pilipczuk, M. Pilipczuk, and M. Wrochna. Edge bipartization faster than 2^k. *CoRR*, abs/1507.02168, 2015.

19 A single-exponential algorithm for Directed FVS

Appeared in [2].

Since 2008 we know that Directed Feedback Vertex Set is fixed-parameter tractable, but the only known algorithm runs in $\mathcal{O}^*(k!4^k)$ time [1]. The $k!$ factor comes out from considering all orderings of the modulator set in the iterative compression step; the rest of the algorithm runs in $\mathcal{O}^*(2^{\mathcal{O}(k)})$ time. Can this step be avoided, so that DFVS would be solved in $\mathcal{O}^*(2^{\mathcal{O}(k)})$ time? Or maybe it is impossible, assuming ETH?

References

[1] J. Chen, Y. Liu, S. Lu, B. O'Sullivan, and I. Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *J. ACM*, 55(5), 2008.

[2] M. Cygan, L. Kowalik, and M. Pilipczuk. Open problems from update meeting on graph separation problems, 2013. http://worker2013.mimuw.edu.pl/slides/update-opl.pdf.

20 Framework for refuting Turing kernels

Long-standing, appeared, e.g., in [2, 1].

One of the most important open problems in kernelization is to provide a framework for refuting Turing kernels. Currently, we know that there is a large group of problems equivalently (un)likely to have Turing kernels [3]. An interesting example of a Turing kernel appears in [4].

References

[1] M. Cygan, L. Kowalik, and M. Pilipczuk. Open problems from workshop on kernels, 2013. http://worker2013.mimuw.edu.pl/slides/worker-opl.pdf.

[2] M. R. Fellows, J. Guo, D. Marx, and S. Saurabh. Data Reduction and Problem Kernels (Dagstuhl Seminar 12241). *Dagstuhl Reports*, 2(6):26–50, 2012.

[3] D. Hermelin, S. Kratsch, K. Soltys, M. Wahlström, and X. Wu. A completeness theory for polynomial (turing) kernelization. *Algorithmica*, 71(3):702–730, 2015.

[4] B. M. P. Jansen. Turing kernelization for finding long paths and cycles in restricted graph classes. In A. S. Schulz and D. Wagner, editors, *Algorithms - ESA 2014 - 22th Annual European Symposium, Wroclaw, Poland, September 8-10, 2014. Proceedings*, volume 8737 of *Lecture Notes in Computer Science*, pages 579–591. Springer, 2014.

21 Tight bounds for kernels for Vertex Cover

Long-standing; appeared, e.g., in [3].

It seems reasonable to believe that the $2k$-vertex kernel for Vertex Cover [6] is optimal, as a $(2-\varepsilon)$-approximation algorithm for Vertex Cover would violate the Unique Games Conjecture [5], and it is hard to imagine a $(2-\varepsilon)k$-vertex kernel that would not yield a $(2-\varepsilon')$-approximation algorithm for Vertex

Cover. However, the aforementioned argumentation is informal, and there exists an example of a problem with a polynomial kernel, but without matching approximation algorithm [4]. Can we prove a matching lower bound for the $2k$-vertex kernel, assuming some widely-believed complexity assumption?
A similar question can be considered in the case of planar graphs. Here, no approximation arguments restrict us, as Vertex Cover admits a PTAS in planar graphs (via the classical Baker's approach [1]). Note that the $4k$-vertex kernel for Independent Set in planar graphs yields an $(\frac{4}{3} - \varepsilon)k$-vertex lower bound for a Vertex Cover kernel in planar graphs [2]. However, still the best known upper bound is the $2k$-vertex kernel inherited from general graphs.

References

[1] B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. ACM*, 41(1):153–180, 1994.

[2] J. Chen, H. Fernau, I. A. Kanj, and G. Xia. Parametric duality and kernelization: Lower bounds and upper bounds on kernel size. *SIAM J. Comput.*, 37(4):1077–1106, 2007.

[3] M. Cygan, L. Kowalik, and M. Pilipczuk. Open problems from workshop on kernels, 2013. http://worker2013.mimuw.edu.pl/slides/worker-opl.pdf.

[4] A. C. Giannopoulou, D. Lokshtanov, S. Saurabh, and O. Suchý. Tree deletion set has a polynomial kernel (but no $OPT^{O(1)}$ approximation). In V. Raman and S. P. Suresh, editors, *34th International Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2014, December 15-17, 2014, New Delhi, India*, volume 29 of *LIPIcs*, pages 85–96. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014.

[5] S. Khot and O. Regev. Vertex cover might be hard to approximate to within 2-ε. *J. Comput. Syst. Sci.*, 74(3):335–349, 2008.

[6] G. L. Nemhauser and L. E. Trotter. Vertex packings: Structural properties and algorithms. *Math. Program.*, 8:232–248, 1975.

22 A linear element-kernel for d-Hitting Set

Appeared in [2].

The Vertex Cover problem admits a $2k$-vertex kernel [4], but is unlikely to admit a $\mathcal{O}(k^{2-\varepsilon})$-edge kernel [3]. More generally, we know that the d-Hitting Set admits a kernel with $\mathcal{O}(k^d)$ sets and $\mathcal{O}(k^{d-1})$ elements [1], and a matching lower bound for the number of sets is known [3]. However, it remains open whether we can further reduce the number of elements in the kernel. In particular, does d-Hitting Set admit a kernel with $f(d)k$ vertices?

References

[1] F. N. Abu-Khzam. A kernelization algorithm for d-hitting set. *J. Comput. Syst. Sci.*, 76(7):524–531, 2010.

[2] M. Cygan, L. Kowalik, and M. Pilipczuk. Open problems from workshop on kernels, 2013. http://worker2013.mimuw.edu.pl/slides/worker-opl.pdf.

[3] H. Dell and D. van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *J. ACM*, 61(4):23:1–23:27, 2014.

[4] G. L. Nemhauser and L. E. Trotter. Vertex packings: Structural properties and algorithms. *Math. Program.*, 8:232–248, 1975.

23 Polynomial kernel for Directed FVS

Long-standing; appeared, e.g., in [3, 2].

One of the longstanding open problems is the question of an existence of a polynomial kernel for DIRECTED FEEDBACK VERTEX SET, parameterized by the size of the deletion set. The FPT algorithm is known since 2008 [1].

REFERENCES

[1] J. Chen, Y. Liu, S. Lu, B. O'Sullivan, and I. Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *J. ACM*, 55(5), 2008.

[2] M. Cygan, L. Kowalik, and M. Pilipczuk. Open problems from workshop on kernels, 2013. http://worker2013.mimuw.edu.pl/slides/worker-opl.pdf.

[3] M. R. Fellows, J. Guo, D. Marx, and S. Saurabh. Data Reduction and Problem Kernels (Dagstuhl Seminar 12241). *Dagstuhl Reports*, 2(6):26–50, 2012.

24 Polynomial kernel for Multiway Cut

Long-standing; appeared, e.g., in [1].

The recent applications of matroid techniques to kernelization resulted in a $\mathcal{O}(k^{t+1})$-vertex kernel for MULTIWAY CUT with t terminals and k being the bound on the size of the cutset [2]. Can the dependency on t be removed from the exponent? The problem remains open even in the (easier) edge-deletion variant of MULTIWAY CUT.

The question has been resolved positively in the planar case [3], but with extremely big exponent.

REFERENCES

[1] M. Cygan, L. Kowalik, and M. Pilipczuk. Open problems from workshop on kernels, 2013. http://worker2013.mimuw.edu.pl/slides/worker-opl.pdf.

[2] S. Kratsch and M. Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. In T. Roughgarden, editor, *FOCS*, pages 450–459. IEEE Computer Society, 2012.

[3] M. Pilipczuk, M. Pilipczuk, P. Sankowski, and E. J. van Leeuwen. Network sparsification for steiner problems on planar and bounded-genus graphs. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 276–285. IEEE Computer Society, 2014.

25 Polynomial kernel for Multicut in DAGs

From [3, 2], appeared also in [1].

In [2] the authors refute the existence of polynomial kernels for most graph separation problems in directed graphs, as DIRECTED MULTIWAY CUT with 2 terminals is OR-compositional. The remaining case is the MULTICUT problem in directed acyclic graphs (shown to be FPT in [3]). Does it admit a polynomial kernel, when parameterized by the size of the cutset and the number of terminal pairs? Or when parameterized by the size of the cutset, with constant number of terminal pairs?

REFERENCES

[1] M. Cygan, L. Kowalik, and M. Pilipczuk. Open problems from workshop on kernels, 2013. http://worker2013.mimuw.edu.pl/slides/worker-opl.pdf.

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

[3] S. Kratsch, M. Pilipczuk, M. Pilipczuk, and M. Wahlström. Fixed-parameter tractability of multicut in directed acyclic graphs. *SIAM J. Discrete Math.*, 29(1):122–144, 2015.

26 Implications of ETH

asked in Mohan's third talk at the boot camp

Recall that s_d for $d \in \mathbb{N}$, as defined in [2], denotes the infimum over all δ such that the satisfiability problem for d-CNF formulas can be solved in time $2^{\delta n}$. We set $s_\infty = \lim_{d\to\infty} s_d$.

- Assuming ETH or another suitable assumption, can we show a specific lower bound on s_3, the exponent for 3-SAT? This is also asked in [1, Section 5] under the assumption of SETH.
- Assuming ETH or another suitable assumption, can we show that $s_\infty = 1$? (That is, SETH.)

References

[1] M. Cygan, H. Dell, D. Lokshtanov, D. Marx, J. Nederlof, Y. Okamoto, R. Paturi, S. Saurabh, and M. Wahlström. On problems as hard as CNF-SAT. In *Proceedings of the 27th Conference on Computational Complexity, CCC 2012, Porto, Portugal, June 26-29, 2012*, pages 74–84. IEEE, 2012.

[2] R. Impagliazzo and R. Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

27 Probabilistic algorithms with one-sided error

asked in Mohan's third talk at the boot camp

An OPP algorithm is a one-sided error probabilistic polynomial-time algorithm, as defined in [1].

- Are there better non-OPP algorithms for k-SAT or Circuit SAT?
- Does there exist an OPP algorithm with success probability c^{-n} for Hamiltonian path?

References

[1] R. Paturi and P. Pudlák. On the complexity of circuit satisfiability. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 241–250, 2010.

28 Circuit minimization problem

asked in Ryan's first talk at the boot camp, see also [1]

The circuit minimization problem MCSP is defined as follows: On input $f : \{0,1\}^n \to \{0,1\}$ and $s \in \mathbb{N}$ (given in binary), decide whether the minimum size of a circuit computing f is at most s. Can we find interesting problems that are solvable in polynomial time with an oracle for MCSP? For instance, can we produce a minimum-size circuit in polynomial time when given f and an oracle for MCSP?
Kabanets and Cai [1] prove some consequences that would ensue if MCSP were in P.

References

[1] V. Kabanets and J. Cai. Circuit minimization problem. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 73–79, 2000.

29 Counting vs. decision for circuit lower bounds

asked in Ryan's fourth talk at the boot camp

Counting the number of solutions is harder than deciding their existence, as witnessed by results such as Toda's theorem. From a circuit complexity viewpoint, if we were able to count satisfying assignments of circuits faster, would this also imply stronger lower bounds?

30 Batch evaluation of formulas

asked in Ryan's fourth talk at the boot camp

It was shown in [1] that the truth tables of ACC circuits can be produced faster than by brute force. More precisely, an ACC circuit of size s with n inputs can be evaluated on all assignments $x \in \{0,1\}^n$ in time

$$2^n \mathrm{poly}(n) + 2^{\mathrm{poly}(\log s)}.$$

Can Boolean formulas of size s on n variables be evaluated on all assignments in time

$$2^n \mathrm{poly}(n) + \mathrm{poly}(s)?$$

References

[1] R. Williams. Nonuniform ACC circuit lower bounds. *J. ACM*, 61(1):2:1–2:32, 2014.

31 Graph diameter

asked in Virginia's second talk at the boot camp

The diameter of a graph G is the longest distance between any two vertices of G.

- Prove any hardness result for the diameter in dense graphs. Are there other equivalent graph problems?
- Can we relate the sparse and dense cases of graph diameter to each other? That is, does an $n^{1.9}$ time algorithm for the sparse case imply an $n^{2.9}$ algorithm for the dense case?

32 Lower bounds for polynomial-time approximations

asked in Virginia's second talk at the boot camp

Most of the known fine-grained reductions between polynomial-time solvable problems do not preserve approximability.

- Can we prove approximation hardness for the graph diameter on dense graphs?
- Are the known running times for 3/2-approximations of the diameter/radius optimal?
- Can we show hardness of approximation for the longest common subsequence or the edit distance in subquadratic time? In practice, linear-time approximations and heuristics are used. Furthermore, a near-linear-time poly-logarithmic approximation algorithm is known [1].

References

[1] A. Andoni, R. Krauthgamer, and K. Onak. Polylogarithmic approximation for edit distance and the asymmetric query complexity. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 377–386, 2010.

33 Implications of a "polynomial-time Rice's theorem" conjecture.

asked by Russell at the boot camp, who offers up to $200 for a solution

Let $\mathcal{P}$ be a property of Boolean functions defined on $\{0,1\}^n$ (for each n). We say that $\mathcal{P}$ is "white-box" solvable if there is a polynomial-time algorithm which when given a circuit C, decides if the function computed by C has the property $\mathcal{P}$ or not. The property is "black-box" solvable if there is a polynomial-time algorithm which, given an oracle for a Boolean function f and a parameter $s \in \mathbb{N}$ such that f has a circuit of size at most s, decides whether f has the property $\mathcal{P}$ or not. The general form of the polynomial-time Rice's theorem conjecture is that any white-box solvable property is also black-box solvable. This problem appears in [1].
The conjecture implies that $\mathrm{P} \neq \mathrm{NP}$ and is possibly out of range of current techniques. However, the truth or falsity of the conjecture might have implications. Show that if the conjecture is *false*, then circuit lower bounds follow.

References

[1] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. *J. ACM*, 59(2):6, 2012.

34 The exact and FPT complexities of succinct problems.

asked by Russell at the boot camp

It follows from the NEXP-completeness of Succinct-3-SAT that many NP-complete graph problems are NEXP-complete on succinctly described graphs [1]. For instance, the graph could be of the form $H \otimes G_n$ where H is an explicitly given graph and G_n could be an $n \times n$ grid specified by just the parameter n in binary; in such cases, the description of the graph is exponentially smaller than the size of the graph).
Can we say something interesting about the exact or FPT complexity of such graph problems?

References

[1] C. H. Papadimitriou and M. Yannakakis. A note on succinct representations of graphs. *Information and Control*, 71(3):181–185, 1986.

35 PIT-hard problems.

asked by Russell at the boot camp

The low-PIT problem is defined to be the polynomial identity testing problem restricted to circuits (over the rationals) which involve constants $-1, 0, 1$ and compute only low-degree polynomial functions of the leaf gates (this ensures that both the degree of the polynomial computed as well as the constants appearing during the computation remain relatively small).

What other interesting PIT-equivalent problems are there? A recent example due to Kopparty, Saraf and Shpilka [2] shows that the problem of factorizing polynomials given by circuits can be solved with access to an oracle for low-PIT (and is also PIT-hard).
Russell suggested the problem of approximating the volume of a convex region given by a membership oracle [1] which is given by a circuit. This problem also seems to have a meta-algorithmic component since it takes a circuit as input, and hence may be a good candidate.

References

[1] M. E. Dyer, A. M. Frieze, and R. Kannan. A random polynomial time algorithm for approximating the volume of convex bodies. *J. ACM*, 38(1):1–17, 1991.

[2] S. Kopparty, S. Saraf, and A. Shpilka. Equivalence of polynomial identity testing and polynomial factorization. *Computational Complexity*, 24(2):295–331, 2015.

36 Tight lower bounds for subgraph isomorphism

asked in Daniel's third talk at the boot camp, see slide 19

The subgraph isomorphism problem asks, given a pattern graph H and a host graph G, to decide whether G contains a subgraph that is isomorphic to H. Write $n = |V(G)|$ and $k = |V(H)|$. Then the problem can clearly be solved in time $f(k)n^{k+\mathcal{O}(1)}$. Furthermore, a lower bound of $f(k)n^{\Omega(k)}$ is known for the k-clique problem [1], a special case of subgraph isomorphism.
However, when parameterizing by $\ell = |E(H)|$, no such tight lower bounds are known: The above result implies a lower bound of $f(\ell)n^{\Omega(\sqrt{\ell})}$ under ETH, which is far from tight, and the best known lower bound is $f(\ell)n^{\Omega(\ell/\log \ell)}$ [2]. Does the subgraph isomorphism problem admit a tight lower bound of $f(\ell)n^{\Omega(\ell)}$?

References

[1] J. Chen, X. Huang, I. A. Kanj, and G. Xia. Strong computational lower bounds via parameterized complexity. *J. Comput. Syst. Sci.*, 72(8):1346–1367, 2006.

[2] D. Marx. Can you beat treewidth? *Theory of Computing*, 6(1):85–112, 2010.

37 Set cover under SETH

asked in Daniel's fourth talk at the boot camp

In the set cover problem, we are given m sets over a universe with n elements, and a number $t \in \mathbb{N}$. The task is to determine whether we can choose t of these sets to cover the entire universe.
For several problems, we have no tight lower bounds under SETH, but rather under the assumption that the set cover problem cannot be solved in time $2^{\epsilon n}m^{\mathcal{O}(1)}$ for $\epsilon < 1$. These include the Steiner tree problem, the set partitioning problem, subset sum, and connected vertex cover, see also Figure 1 in [1] (published version: [2]).
Can we rule out a $2^{\epsilon n}m^{\mathcal{O}(1)}$ time algorithm for the set cover problem under SETH? This would imply tight lower bounds for the problems mentioned above.

References

[1] M. Cygan, H. Dell, D. Lokshtanov, D. Marx, J. Nederlof, Y. Okamoto, R. Paturi, S. Saurabh, and M. Wahlström. On problems as hard as CNFSAT. *CoRR*, abs/1112.2275, 2011.

[2] M. Cygan, H. Dell, D. Lokshtanov, D. Marx, J. Nederlof, Y. Okamoto, R. Paturi, S. Saurabh, and M. Wahlström. On problems as hard as CNF-SAT. In *Proceedings of the 27th Conference on Computational Complexity, CCC 2012, Porto, Portugal, June 26-29, 2012*, pages 74–84. IEEE, 2012.

Open problems from open problem session on 2015-09-23

38 A question on block-dense distributions related to communication complexity of boolean functions

asked by Raghu Meka during open problem session on 2015-09-23, based on [1]

For a fixed *gadget* $g : \{0,1\}^b \times \{0,1\}^b \to \{0,1\}$, we are interested in the communication complexity of computing a function $f : \{0,1\}^n \to \{0,1\}$ in the following manner: both Alice and Bob have n blocks $(\alpha_i)_{i=1}^n$ of b bits each, and they want to compute $f(g(\alpha_i, \beta_i)_{i=1}^n)$. It is conjectured that the communication complexity of a function f (in a fixed model, like randomized, non-deterministic, etc.) should be related to some measure of f like decision tree complexity (in a related model).
Basing on [1], a positive answer to the following conjecture would imply such a result in the non-deterministic setting. Henceforth we focus on gadget g being the inner product of α and β in $\mathbb{F}_2$. For a distribution X on $(\{0,1\}^b)^n$ (we think it as of n *blocks*, each of b bits), we say that it is *blockwise-dense* if for every index set $I \subseteq [n]$, the min-entropy of X restricted to blocks with indices in I it at least $0.9b|I|$.
The conjecture asserts that there exists a constant b such that if X and Y are two independent blockwise-dense distributions that never set any block to all-zeroes, then $g^n(X, Y)$ has full support.
In [1] the above is proven for $b = 2\log n$.

References

[1] M. Göös, S. Lovett, R. Meka, T. Watson, and D. Zuckerman. Rectangles are nonnegative juntas. In R. A. Servedio and R. Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 257–266. ACM, 2015.

39 Online matrix-vector multiplication: complexity and relation to other popular conjectures

asked by Sebastian Krinninger during open problem session on 2015-09-23, based on [1]

In this problem all computations are over $\mathbb{F}_2$. The online matrix-vector multiplication problem is as follows. First we are given an $n \times n$ matrix M, which we can process in polynomial time. Then a sequence of n-dimensional vectors v is revealed one-by-one, and we are to compute Mv before the next vector comes. The online matrix-vector multiplication conjecture asserts that one cannot process a sequence of n queries in $\mathcal{O}(n^{3-\varepsilon})$ time for any $\varepsilon > 0$. The (weaker) one-round version asserts that one cannot process a single query in $\mathcal{O}(n^{2-\varepsilon})$ time for any $\varepsilon > 0$. In [2] it is shown that one can process a single query in $\mathcal{O}(n^2/(\varepsilon \log n)^2)$ time after preprocessing M in time $\mathcal{O}(n^{2+\varepsilon})$. As shown in [1], this conjecture explains many hardness barriers in online problems.

- Can we relate this conjecture to other conjectures, such as 3-SUM, SETH, APSP?
- Can we improve the upper bound given by [2]?

References

[1] M. Henzinger, S. Krinninger, D. Nanongkai, and T. Saranurak. Unifying and strengthening hardness for dynamic problems via the online matrix-vector multiplication conjecture. In R. A. Servedio and R. Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 21–30. ACM, 2015.

[2] R. Williams. Matrix-vector multiplication in sub-quadratic time: (some preprocessing required). In N. Bansal, K. Pruhs, and C. Stein, editors, *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, pages 995–1001. SIAM, 2007.

40 Lower bounds for dynamic problems, where the incremental and decremental upper bounds differ significantly

asked by Sebastian Krinninger during open problem session on 2015-09-23, based on [3]

Current techniques in conditional lower bounds for dynamic problems [1, 3] do not seem to be easily applicable to problems where the currently known upper bounds for the incremental and decremental scenarios differ significantly. Can we show some lower bounds in such cases? An example of such problem is the directed single source reachability (see [2] for more on this problem).

References

[1] A. Abboud and V. V. Williams. Popular conjectures imply strong lower bounds for dynamic problems. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 434–443, 2014.

[2] M. Henzinger, S. Krinninger, and D. Nanongkai. Sublinear-time decremental algorithms for single-source reachability and shortest paths on directed graphs. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 674–683, 2014.

[3] M. Henzinger, S. Krinninger, D. Nanongkai, and T. Saranurak. Unifying and strengthening hardness for dynamic problems via the online matrix-vector multiplication conjecture. In R. A. Servedio and R. Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 21–30. ACM, 2015.

41 Worst-case bounds for fully-dynamic minimum spanning forest

asked by Sebastian Krinninger during open problem session on 2015-09-23

A fully-dynamic algorithm for minimum spanning forest with amortized polylog(n) time for update is known for a few years [1], and a worst-case polylog(n) bound is known for the easier connectivity problem (i.e., the unweighted case) [2]. Does there exists a fully-dynamic algorithm for minimum spanning forest with worst-case polylog(n) time for update?

References

[1] J. Holm, K. de Lichtenberg, and M. Thorup. Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. *J. ACM*, 48(4):723–760, 2001.

[2] B. M. Kapron, V. King, and B. Mountjoy. Dynamic graph connectivity in polylogarithmic worst case time. In S. Khanna, editor, *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1131–1142. SIAM, 2013.

42 Theory of representations and their computational complexity

asked by Russell Impagliazzo during open problem session on 2015-09-23

This is a vague and open-ended question, discussing some generic theory of efficient representations of databases and queries.

Consider the following situation: we have a large relational database, and we have a sequence of queries coming in. The queries can be expressed in different ways: as regular expressions, DFAs, or NFAs. Depending on the representation, their evaluation has different computational complexity: for DFA it is PSPACE-complete, for NFA it is EXPSPACE-complete. However, to offset this complexity gap, it may be possible that the same question can be asked in an NFA form in a much more succinct way that is is done by a DFA.

From the other side, if one wants to represent a database — say, a subset $S \subseteq \{0,1\}^n$ — so that one can later process queries efficiently, there are different ways to do it: as an ILP, as a CNF formula describing it, as ROBDD, etc.

In a different world, one could think of different ways of representing abelian groups: either as a product of groups of the type $\mathbb{Z}_{p^k}$, or as a pair (N, g), where the group in question is the group generated by g in $\mathbb{Z}_N^*$.

This leads to the generic question: what is the best representation of the query (and maybe the database) depending on the purpose? Can one develop any high-level theory of comparing representations?

43 Approximating structure of the optimum solution

asked by Antonina Kolokolova during open problem session on 2015-09-23

Consider the following approximation paradigm: we are to solve an instance of some optimization problem, but we are not only interested in the quality of the solution (as in the classic approximation algorithms), but also how the solution we computed is similar to the actual optimum in structural sense. For example, in the Vertex Cover problem, we may be interested not only in the size of the computed solution, but also (or even only!) in the size of the symmetric difference between the computed solution and some optimum solution.

The natural measures of similarity between the computed solution and the optimum solution are Hamming distance (for example, between NP witnesses) or generally some L_1 or L_p distances.

This paradigm has been introduced in [2]. In [1] the distances between NP witnesses are discussed, and, among other results, it is proven that little can be done in the Vertex Cover case: the existence of an algorithm that guarantees that the symmetric difference is at most $n/2 - n^{\varepsilon}$ for any $\varepsilon > 0$ is unlikely.

Another example from a different area is the complexity of finding approximate Nash equilibria: while we can find quite efficiently an approximate equilibrium where no player can improve its payoff by more than ε, such an equilibrium may be structurally very different from the actual Nash equilibrium.

The question is to extend this theory; in particular, does there exist an interesting positive result, where such a structural approximation is possible?

References

[1] D. Sheldon and N. E. Young. Hamming approximation of NP witnesses. *Theory of Computing*, 9:685–702, 2013.

[2] I. van Rooij, M. Hamilton, M. Müller, and T. Wareham. Approximating solution structure. In E. D. Demaine, G. Gutin, D. Marx, and U. Stege, editors, *Structure Theory and FPT Algorithmics for Graphs, Digraphs and Hypergraphs, 08.07. - 13.07.2007*, volume 07281 of *Dagstuhl Seminar Proceedings*. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2007.

Change log

23 Sep 2015	problems from open problem session on 2015-09-23
23 Sep 2015	all problems from boot camp
10 Sep 2015	split references between problems
28 Aug 2015	enhanced entry on Problem 10
25 Aug 2015	open problems from FPT and exponential-time algorithms