PACS 2024: Workshop on Parameterized Algorithms and Constraint Satisfaction

Open problems

July 7, 2024

1 Min-2-Lin($\mathbb{Z}_4$) parameterized by solution cost

Dabrowski et al. [1], communicated by George Osipov

In Min-2-Lin($\mathbb{Z}_4$) we are given a set of equations of the form $ax+by=c$, where x and y are variables, $a,b,c \in \mathbb{Z}_4$, and an integer k. The goal is to delete at most k equations from the set so that the remaining ones are satisfiable in $\mathbb{Z}_4$.

Question 1. *Is Min-2-Lin($\mathbb{Z}_4$) in* FPT *or* W*[1]-hard parameterized by k?*

It is known that e.g. Min-2-Lin($\mathbb{Z}_2$) is in FPT and Min-2-Lin($\mathbb{Z}_6$) is W[1]-hard. Using [1] and some simple tricks, one can reduce Min-2-Lin($\mathbb{Z}_4$) to the following version in FPT-time: the input is a set of equations with a distinguished variable z and an integer k, with each equation of the form $x=2y$ or $x=3y$; the goal is to delete at most k equations so that the remaining ones are satisfiable in $\mathbb{Z}_4$ by an assignment that maps z to 1.

References

[1] K. K. Dabrowski, P. Jonsson, S. Ordyniak, G. Osipov, and M. Wahlström. Almost consistent systems of linear equations. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3179–3217. SIAM, 2023.

2 Approximating Negative Directed Feedback Arc Set

Communicated by George Osipov

In Negative DFAS we are given a directed graph with rational labels on the arcs, and the goal is to delete at most k arcs from the graph so that no directed negative cycle remains. When parameterized by k, this problem generalizes DFAS (special case with all arcs labelled -1) and Subset DFAS (special case with arcs labelled 0 or -1), which are in FPT. On the other hand, Negative DFAS even with labels -1, 0 and 1 is W[1]-hard [2] (see also [1] for results under structural and combined parameterizations).

Question 2. *Can Negative DFAS parameterized by k be approximated within a constant factor in* FPT *time?*

Note that constant-factor approximation in polynomial time is already ruled out for DFAS under the Unique Games Conjecture. This problem can be formulated as MinCSP on domain $\mathbb{Q}$ with constraints of the form $x-y \leq a$ for $a \in \mathbb{Q}$. The case with $a \in \{-1,0,1\}$ is sufficiently general when the arc labels are given in unary.

References

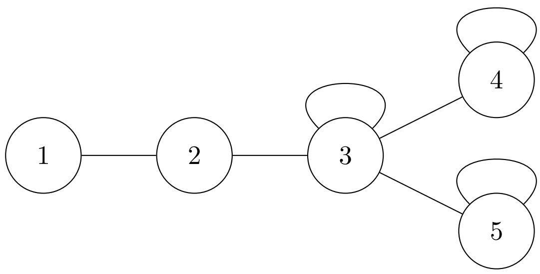
[1] K. Bérczi, A. Göke, L. M. Mendoza-Cadena, and M. Mnich. Resolving infeasibility of linear systems: A parameterized approach. *arXiv preprint arXiv:2209.02017*, 2022.

[2] A. Göke, L. M. Mendoza Cadena, and M. Mnich. Resolving infeasibility of linear systems: A parameterized approach. In *14th International Symposium on Parameterized and Exact Computation, IPEC 2019*, 2019.

3 List Homomorphism by Edge Deletion

Communicated by George Osipov

Let H be an undirected graph with loops allowed. In List Homomorphism by Edge Deletion for H (LHomED(H)), our input is a graph G, a function $L : V(G) \to 2^{V(H)}$ and an integer k. The question is whether there exists a set of edges $X \subseteq E(G)$ of size at most k such that $G - X$ admits a homomorphism f to H such that $f(v) \in L(v)$ for all $v \in V(G)$. We are interested in parameterized complexity of this problem with k as the parameter, and obtaining a dichotomy depending on the host graphs H. For irreflexive graphs H, the dichotomy follows by combining [1] with [2]. It states that, for irreflexive graphs H, LHomED(H) is in FPT whenever the decision version LHom(H) (i.e. the version with $k = 0$) is in P. For reflexive graphs, Okrasa, Pilipczuk, and Rzążewski have an unpublished algorithm that extends the statement above to purely reflexive graphs. What about mixed graphs, i.e. those containing both reflexive and irreflexive vertices? Consider the following graph:



Question 3. *What is the complexity of LHomED(H) for the graph H above?*

For some mixed graphs, the same algorithmic ideas as from the irreflexive and reflexive cases can be used (by unpublished results with Okrasa and Rzążewski). This graph is a minimal unknown case.

References

[1] R. Chitnis, L. Egri, and D. Marx. List H-coloring a graph by removing few vertices. *Algorithmica*, 78:110–146, 2017.

[2] E. J. Kim, S. Kratsch, M. Pilipczuk, and M. Wahlström. Directed flow-augmentation. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 938–947, 2022.

4 Non-trivial exponential-time running bounds for symmetric Boolean languages

Lagerkvist and Wahlström [1], communicated by Magnus Wahlström

In [1], a project was initiated to investigate the exact exponential time complexity of CSP(Γ) for NP-hard CSPs, depending on the language Γ. In particular, for which languages Γ can the CSP over Γ be solved in time $O(c^n)$ for some $c < 2$? (We are assuming Boolean languages for simplicity here; the problem is plenty complex enough with this restriction.) This was undertaken under an algebraic framework with some promising partial results.

Unfortunately, the project runs into annoying issues of representation of constraints (and of having a very difficult structure to get a handle of).

To avoid this issue, a simpler question that should be much more amenable to abstract analysis is to assume that the language is *symmetric*, i.e., a relation R in the language accepts a tuple $t = (x_1, \ldots, x_r) \in D^r$ only based on the number of times each element from the domain occurs in t.

To be more precise, let $D = \{0, 1\}$ be the boolean domain and let $S \subseteq \{0, \ldots, r\}$ for some $r \in \mathbb{N}$. Then S and the arity r define a symmetric relation $R_S \subseteq \{0, 1\}^r$ as

$$(x_1, \ldots, x_r) \in R_S \Leftrightarrow \sum_{i=1}^{t} x_i \in S.$$

For example, for $r = 3$ and $S = \{1, 2, 3\}$, R_S defines the positive 3-clause. Say that a language Γ is symmetric if every relation in Γ is symmetric.

Question 4. *For which symmetric Boolean languages Γ can CSP(Γ) be solved in time $O(c^n)$ for some $c < 2$ (e.g., assuming SETH for lower bounds)?*

If this is too much of an open-ended project, let me pick up a specific restriction where the question is open. This is provably a language with restricted algebraic structure, so that it cannot trivially be used in a proof of SETH-hardness, but an algorithm seems somewhat elusive.

Question 5. *A* Sidon set *is a set $S \subseteq \mathbb{N}$ such that all sums $a + b$ for $a, b \in S$ are distinct (i.e. for $a, b, c, d \in S$, if $a + b = c + d$ then $\{a, b\} = \{c, d\}$). Can CSP(Γ) be solved in time $O(c^n)$, $c < 2$ if the relations in Γ are symmetric relations defined via Sidon sets?*

For both of the above questions, a non-uniform algorithm is fine. The purpose is to investigate the toolbox of non-trivial exponential-time algorithms, not to wrestle with questions about representations and oracle models.

References

[1] V. Lagerkvist and M. Wahlström. The (coarse) fine-grained structure of NP-hard SAT and CSP problems. *ACM Trans. Comput. Theory*, 14(1):2:1–2:54, 2022.

5 Running times of inv(p)-SAT

Communicated by Magnus Wahlström

This problem asks two specific questions about the running time of SAT problems defined only via the existence of a single partial polymorphism p, referred to as INV(p)-SAT in [1]. This problem class was designed to ask questions about limits on the possible running times for NP-hard problems based on SETH.

Let us recall some definitions from [1]. Let $R(X)$ with $R \subseteq \{0, 1\}^n$ be a constraint on a tuple of variables $X = (x_1, \ldots, x_n)$. A *partial assignment* to X is an assignment $f \colon X_f \to \{0, 1\}$ for some $X_f \subseteq X$. An *extension oracle* for $R(X)$ is an oracle that, given a partial assignment f to X, reveals whether there exists a tuple $t \in R$ such that $t_i = f(x_i)$ for every $x_i \in X_f$. Then INV(p)-SAT in the extension oracle model is the SAT problem where every constraint is provided only via an extension oracle, with the promise that every relation R used in a constraint $R(X)$ in the input is preserved by the partial polymorphism p.

We review two specific examples. The *partial Maltsev operation* over D is the partial operation p on D such that for all $x, y \in D$,

$$p(x, x, y) = p(y, x, x) = y,$$

and for all other inputs $x, y, z \in D$, $p(x, y, z)$ is undefined. Then INV(p)-SAT over captures precisely the SAT problems over domain $\{0, 1\}$ that can be solved via the meet-in-the-middle strategy, and correspondingly (with a tiny bit of care) INV(p)-SAT can be solved in time for $O^*(2^{n/2})$ extension oracle queries on inputs with n variables.

In particular, note that SUBSET SUM corresponds to an instance of INV(p)-SAT with a single constraint, where we do not have an efficient extension oracle. Via an appropriate branching strategy, this can be reduced to an instance of $\textsc{inv}(p)$-SAT with multiple constraints, each of which is provided via an extension oracle with a running time of $2^{o(n)}$ per query [1]. Hence, the algorithm for $\textsc{inv}(p)$-SAT can be used to solve Subset Sum in time $O(2^{n/2+o(n)})$ and if $\textsc{inv}(p)$-SAT can be solved in time $O^*(c^n)$ for any $c < \sqrt{2}$ then so can Subset Sum.

The latter is a major open problem, and the natural approaches tend to involve heavy use of number theory.

Can we exclude such an algorithm in the very restricted extension oracle model? We note that $\textsc{inv}(p)$-SAT for the partial Maltsev operation cannot be solved in time $O(2^{(1/4.82-\varepsilon)n})$ for any $\varepsilon > 0$ assuming randomized SETH [1], but there is clearly quite a gap between the upper and lower bounds here.

Question 6. *Can $\textsc{inv}(p)$-SAT where p is the partial Maltsev operation be solved in $O^*(2^{cn})$ time for some $c < 1/2$ in the extension oracle, or can this be excluded e.g. under SETH?*

Next, consider the k-SAT problem. For $k \geq 3$, let s_k be the infimum over all s such that k-SAT can be solved in time $O^*(2^{sn})$. Significant work (and multiple conjectures) has gone into investigating s_k. ETH (the exponential time hypothesis) is equivalent to $s_3 > 0$ and SETH (the strong exponential time hypothesis) is equivalent to $\lim_{k\to\infty} s_k = 1$. The fastest known algorithms for k-SAT bound $s_k = 1 - \Omega(1/k)$. An even more drastic conjecture, the Super Strong Exponential Time Hypothesis (SSETH) states that this is optimal, i.e., $s_k = 1 - \Theta(1/k)$ [4, 3]. This has been confirmed to be the convergence for specific algorithms [2] but appears somewhat contentious (cf. [3]).

The *partial k-NU* operation, or *partial k-near unanimity* operation, over a domain D is the k-ary partial operation defined as

$$\mathrm{nu}_k(x, x, \ldots, y) = \ldots = \mathrm{nu}_k(y, x, \ldots, x) = x$$

for all $x, y \in D$, and otherwise undefined. Then, again restricting to domain $\{0, 1\}$, k-SAT for $k \geq 3$ is preserved by nu_{k+1} but not by nu_k (and $\mathrm{nu}_{k+!}$ is a pretty natural property of k-SAT). Let c_k be the infimum over all c such that $\textsc{inv}(\textsc{nu}_k)$-SAT can be solved in time $O^*(2^{cn})$ in the extension oracle model. Clearly this is a much broader task than simply solving k-SAT.

The best lower bounds under SETH bound c_k as $c_k = 1 - \Omega(\log k/k)$ [1], which incidentally matches the convergence of the algorithm of Vyas and Williams [3] on random k-SAT formulas.

Question 7. *Can we bound $c_k = 1 - O(1/k)$, e.g. under SETH?*

References

[1] V. Lagerkvist and M. Wahlström. The (coarse) fine-grained structure of NP-hard SAT and CSP problems. *ACM Trans. Comput. Theory*, 14(1):2:1–2:54, 2022.

[2] D. Scheder and N. Talebanfard. Super strong eth is true for ppsz with small resolution width. In *35th Computational Complexity Conference (CCC 2020)*, volume 169, page 3. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[3] N. Vyas and R. Williams. On super strong eth. *Journal of Artificial Intelligence Research*, 70:473–495, 2021.

[4] R. Williams. Circuit analysis algorithms. Talk at Simons Institute for Theory of Computing, 2015. Available at `https://youtu.be/adJvi7tL-qM?t=925`.

6 Languages with linear non-redundancy

Asked repeatedly [3, 2, 1], communicated by Magnus Wahlström.

A language Γ has *non-redundancy* $f(n)$ if every formula F over Γ with n variables contains a subformula F' on at most $f(n)$ constraints such that F and F' have identical solution sets. This is closely related to the notion of a *sparsification* of $\mathrm{CSP}(\Gamma)$, i.e. a kernelization of $\mathrm{CSP}(\Gamma)$ parameterized by n. If every relation in Γ has arity at most r, then Γ trivially has non-redundancy $f(n) = O(n^r)$, but for many cases significantly better upper bounds are possible.

The languages where the trivial bound is tight have been characterized by Carbonnel [1] (see also Chen et al. [2] over the Boolean domain). But what about the other end of the scale? The following is asked by Carbonnel [1], but relates closely to sparsification questions asked previously [3, 2].

Question 8. *Which languages Γ have non-redundancy $f(n) = O(n)$?*

Over the Boolean domain, there is a plausible candidate due to Chen et al. [2], who characterize languages that can be encoded over integer rings and show that every such language has sparsification $f(n) = O(n)$. However, even over the Boolean domain it is not known whether this covers all cases, and over larger domains the question is wide open.

References

[1] C. Carbonnel. On redundancy in constraint satisfaction problems. In *28th International Conference on Principles and Practice of Constraint Programming (CP 2022)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2022.

[2] H. Chen, B. M. Jansen, and A. Pieterse. Best-case and worst-case sparsifiability of Boolean CSPs. *Algorithmica*, 82(8):2200–2242, 2020.

[3] V. Lagerkvist and M. Wahlström. Sparsification of SAT and CSP problems via tractable extensions. *ACM Transactions on Computation Theory (TOCT)*, 12(2):1–29, 2020.

7 Polynomial kernelization of Boolean MinCSP

Communicated by Magnus Wahlström

With the result of Kim et al. [1], we finally know which Boolean MinCSPs that are FPT and W[1]-hard under the natural parameter. But for which languages does the problem have a polynomial kernel? More precisely, the question is the following.

Question 9. *For which Boolean languages Γ does MinCSP(Γ) have a polynomial kernel parameterized by the number of unsatisfied constraints (under the usual assumptions, e.g. the polynomial hierarchy does not collapse)?*

The problem domain includes Almost 2-SAT as one item, which, famously, has a randomized polynomial kernel via matroid methods [2] but is open for a deterministic kernel. Hence randomized kernelizations are fine. It also, to the best of our knowledge, does not cover any of the intractable "dragons" of polynomial kernelization such as Multiway Cut and Directed Feedback Vertex Set.

References

[1] E. J. Kim, S. Kratsch, M. Pilipczuk, and M. Wahlström. Flow-augmentation III: complexity dichotomy for Boolean CSPs parameterized by the number of unsatisfied constraints. In *SODA*, pages 3218–3228. SIAM, 2023.

[2] S. Kratsch and M. Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. *J. ACM*, 67(3):16:1–16:50, 2020.

8 Parameterised complexity of Promise CSPs

Communicated by Andrei Krokhin

Promise CSP (PCSP) is a recent generalisation of the standard CSP. It comes in two versions: decision and search, which are in general not known to be equivalent. The most prominent example of PCSP is approximate graph colouring: for fixed $k \leq c$, decide whether a given graph is k-colourable or not even c-colourable (the promise being that the graph falls into one of these two categories). The search version of that is, given a graph that is promised to be k-colourable (but the colouring is not given), find a c-colouring. The general PCSP is parameterised by two relational structures $\mathbb{A}$ and $\mathbb{B}$ such that there is a homomorphism $\mathbb{A} \to \mathbb{B}$. Note that the special case when $\mathbb{A} = \mathbb{B}$ is precisely the standard problem CSP($\mathbb{A}$).

PCSP($\mathbb{A}, \mathbb{B}$) can be solvable in polynomial time even if both CSP($\mathbb{A}$) and CSP($\mathbb{B}$) are NP-complete. One standard example of this is when the two CSPs are (monotone) 1-in-3-Sat and (monotone) Not-All-Equal-Sat: given a satisfiable instance of (monotone) 1-in-3-Sat, it is NP-hard to find a satisfying assignment for it, but if one relaxes each 1-in-3 constraint in it to allow all Not-All-Equal triples, then one can find in polynomial time a satisfying assignment for the relaxed instance.

One can investigate the parameterised complexity of many problems around PCSPs - there is practically nothing known beyond the case of standard CSPs. For a recent survey about PCSPs, see [1] and references therein.

References

[1] A. Krokhin and J. Opršal. An invitation to the promise constraint satisfaction problem. *ACM SIGLOG News*, 9(3):30–59, 2022. Avaliable at https://arxiv.org/abs/2208.13538.

9 VCSPs parameterized above lower bounds

Communicated by Magnus Wahlström

This is another less precise open question.

The immediate parameter to study the parameterized complexity of VCSPs and other optimization problems is the *natural parameter*, i.e. the solution cost. However, for some problems you get very interesting results by parameterizing by the *relaxation gap* of some suitable relaxation.

There are two classical results in this direction:

- Multiway Cut can be solved in time $O^*(4^{k-\mu})$, where k is the solution cost and μ is the optimum of the half-integral LP relaxation due to Garg et al. [1]. This implies that Multiway Cut can be solved in time $O^*(2^k)$, which for the vertex deletion variant is the fastest known algorithm parameterized by k.

- Vertex Cover can be solved in time $O^*(4^{k-\mu})$ by the same strategy, which can be improved to $O^*(2.3146^{k-\mu})$; here, μ is the optimum of the Nemhauser-Trotter LP-relaxation [4]. This does not give the fastest algorithm for Vertex Cover under the natural parameter, but it is an important result in its own right due to the close connection to this parameterization and the Almost 2-SAT problem.

However, such results are not “normally” possible – for an arbitrary LP-relaxation, it is likely to be NP-hard even to find a solution of cost μ, i.e. with gap $k - \mu = 0$. Thus, the lower bound μ must be of some special type.

Some extensions to the above for other VCSPs are known [3, 5], but the results remain relatively “isolated” and we do not have a good picture of when such a result is possible. (More exotic variants also exist [2] but may go beyond this basic question.)

Is there a good “CSP perspective” on when a VCSP is FPT parameterized by the relaxation gap $k-\mu$, depending on the language and the choice of lower bound μ? Can it for example be usefully studied as a parameterization of some PCSP variant?

References

[1] M. Cygan, M. Pilipczuk, M. Pilipczuk, and J. O. Wojtaszczyk. On multiway cut parameterized above lower bounds. *TOCT*, 5(1):3, 2013.

[2] S. Garg and G. Philip. Raising the bar for vertex cover: Fixed-parameter tractability above a higher guarantee. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1152–1166. SIAM, 2016.

[3] Y. Iwata, M. Wahlström, and Y. Yoshida. Half-integrality, LP-branching and FPT algorithms. *SIAM J. Comput.*, 45(4):1377–1411, 2016.

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

[5] F. Reidl and M. Wahlström. Parameterized algorithms for zero extension and metric labelling problems. In *ICALP*, volume 107 of *LIPIcs*, pages 94:1–94:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.

Change log

2024-06-07	problem 3 added
2024-05-02	problem 2 added
2024-02-26	problem 1 added
2024-02-26	first file