Journal of Computer and System Sciences 79 (2013) 144–151

Contents lists available at SciVerse ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Completely inapproximable monotone and antimonotone parameterized problems ☆

Dániel Marx

Computer and Automation Research Institute, Hungarian Academy of Sciences (MTA SZTAKI), Budapest, Hungary

ARTICLE INFO

Article history:
Received 10 September 2011
Received in revised form 14 August 2012
Accepted 4 September 2012
Available online 13 September 2012

Keywords:
Inapproximability
Fixed-parameter tractability
Circuits
Circuit satisfiability

ABSTRACT

We prove that weighted circuit satisfiability for monotone or antimonotone circuits has no fixed-parameter tractable approximation algorithm with *any* approximation ratio function ρ, unless FPT $\neq W[1]$. In particular, not having such an fpt-approximation algorithm implies that these problems have no polynomial-time approximation algorithms with ratio $\rho(\text{OPT})$ for any nontrivial function ρ.

1. Introduction

Recently, there has been increased interest in investigating approximation in the context of parameterized complexity [5,4,6,15,9]. Recall that a *parameterization* of a problem is a polynomial-time computable function that assigns an integer k to each problem instance x. An *fpt-algorithm* for a parameterized problem is an algorithm with running time $f(k) \cdot |x|^{O(1)}$, where k is the parameter of the input instance x and f is an arbitrary computable function. A decision problem is *fixed-parameter tractable* (FPT) with parameter k if it can be solved by an fpt-algorithm. The parameter can be any well-defined function of the input instance x; for example, the required number of vertices in the solution, or the maximum degree of the input graph. The standard way of turning an optimization problem into a decision problem is to add a value k to the input instance and ask if there is a solution with cost at most/at least k. Taking this value k appearing in the input as the parameter is called the *standard parameterization* of the optimization problem. For a large number of NP-hard optimization problems, the standard parameterization is FPT, for example, this is the case for Minimum Vertex Cover, Longest Path, Directed Feedback Vertex Set, and Multiway Cut. Intuitively, these results show that the problems can be solved efficiently if the optimum is small. On the other hand, the $W[1]$-hardness of other problems, such as Maximum Clique and Minimum Dominating Set gives evidence that these problems are not fixed-parameter tractable, and probably there is no significantly better algorithms than solving the problem in time $n^{O(k)}$ by brute force.

FPT approximation algorithms were introduced by three independent papers [5,6,4], see also the survey [15]. We follow here the notation of [5]. *An fpt-approximation algorithm with ratio ρ* for a minimization problem P is an fpt-algorithm that, given an instance x of P and a positive integer k, computes a solution of cost at most $k \cdot \rho(k)$ if a solution of cost at most k exists; if there is no solution of cost at most k, then the output can be arbitrary. The definition can be adapted to

☆ A preliminary version of the paper appeared in the proceedings of the 25th Annual IEEE Conference on Computational Complexity (CCC 2010). Research supported by the European Research Council (ERC) grant "PARAMTIGHT: Parameterized complexity and the search for tight complexity results," reference 280152.

E-mail address: dmarx@cs.bme.hu.

0022-0000/$ – see front matter
http://dx.doi.org/10.1016/j.jcss.2012.09.001

maximization problems. Note that the approximation ratio ρ is a function of k, not the input size: intuitively, if k is small, then $k \cdot \rho(k)$ can be still considered small. We say that problem is *fpt-approximable* if it has an fpt-approximation algorithm for some function ρ. As we are proving hardness results in this paper, it will be convenient to prove hardness result for a decision problem associated with approximation. Following [5], we say that an algorithm is an *fpt cost approximation* with ratio ρ if it distinguishes (in fpt-time) between instances of optimum value at most k and more than $k \cdot \rho(k)$ (see Definition 3). It is clear that it is sufficient to prove hardness results for this decision variant to rule out the possibility of fpt-approximation with ratio ρ.

On the positive side, there are a couple of nontrivial fpt-approximation algorithms. Seymour et al. [18] proved a relation between the packing and the covering number of directed cycles, which was subsequently made algorithmic by Grohe and Grüber [13] in order to obtain an fpt ρ-approximation algorithm for Maximum Disjoint Cycles in directed graphs, with some unspecified function ρ. Marx and Razgon [16] gave an fpt 2-approximation algorithm for Edge Multicut. However, later it was shown that Edge Multicut (as well as Vertex Multicut) is actually fixed-parameter tractable [17,3]. Fellows (unpublished result) showed that Topological Bandwidth has an fpt-approximation algorithm with ratio k (see also [15]). On the negative side, it is known that Weighted Circuit Satisfiability [5] and Independent Dominating Set [8] are not fpt-approximable for *any* function ρ (under standard complexity assumptions). However, these results are not very enlightening as they use in an essential way that the considered problems are not (anti)monotone: it is not true that every subset or every superset of a solution is a solution. Therefore, it is very well possible that every feasible solution of an instance is of the same size, in which case any approximation algorithm has to actually find an optimum solution. We call a minimization problem *monotone* if any superset of a solution is also a solution. Similarly, a maximization problem is *antimonotone,* if any subset of the solution is also a solution. Inapproximability is usually a more meaningful question for such problems.

The first inapproximability result in the fpt sense for a monotone problem was obtained earlier and independently of the study of fpt-approximation. As a key step in showing that resolution is not automatizable, Alekhnovich and Razborov [1] showed that there is no fpt 2-approximation algorithm for Weighted Monotone Circuit Satisfiability, unless every problem in the class $W[P]$ can be solved by a randomized fpt-algorithm. Eickmeyer et al. [9] improved this result in two ways: they weakened the complexity assumption by removing the word "randomized," and increased the ratio from 2 to any polylogarithmic function. They conjectured that the problem has no fpt-approximation algorithm for any function ρ. Our first result confirms this conjecture. The proof is completely different and much simpler than the inapproximability proofs of [9,1]. Instead of using expanders for gap amplification in a multi-layer circuit, our proof achieves an arbitrary large gap using a simple construction based on k-perfect hash functions. Furthermore, it is shown in [9] that appropriate gap-preserving reductions can transfer the inapproximability result from Weighted Monotone Circuit Satisfiability to other parameterized minimization problems such as Minimum Chain Reaction Closure, Minimum Generating Set, and Minimum Linear Inequality Deletion. The reductions transfer our result as well, thus it follows that these problems are not fpt-approximable either.

Eickmeyer et al. [9] raised the question if the maximization problem Weighted Antimonote Circuit Satisfiability is fpt-approximable; they conjectured that this problem is also hard to approximate. Note that finding a maximum weight solution for an antimonotone circuit is equivalent to finding a minimum weight solution for a monotone circuit, but the approximability of the two problems can be different. We prove this conjecture by showing that the problem is not fpt-approximable, unless $\mathrm{FPT} \neq W[1]$. The proof is somewhat similar to the inapproximability results of [1,9] for the monotone version: it uses simple linear algebra (Reed–Solomon codes) for error correction. However, the construction of the circuit is much simpler, since we do not have to repeat the same circuit in multiple layers to increase the gap. In fact, we prove the result for a special case of circuit satisfiability, which is a fairly natural $W[2]$-complete combinatorial problem on hypergraphs: in the Threshold Set problem, we are given a collection of subsets $\mathcal{S}$ of a universe U with a weight $w(S)$ for each set $S \in \mathcal{S}$, the goal is to select the maximum number of elements from U such that every $S \in \mathcal{S}$ contains at most $w(S)$ elements.

It would be interesting to obtain similar inapproximability results for more restricted versions of circuit satisfiability and perhaps even for natural combinatorial problems such as Independent Set and Hitting Set. The current paper is already a step in this direction: we prove inapproximability results for Threshold Set and for monotone/antimonotone circuit satisfiability with certain bounds on the depth and weft of the circuit. However, beyond a certain point, much deeper techniques would be required than the elementary methods of the present paper. In particular, the known proofs giving evidence that there is no polynomial-time constant factor approximation algorithm for Hitting Set and Independent Set all use the PCP theorem. Thus ruling out fpt-approximation for these problems would require the use of (some generalization of) the PCP theorem.

Finally, let us mention that expressing the approximation ratio as a function of the optimum (rather than as a function of the input size) makes sense also in the context of polynomial-time approximation algorithms. There are such results in the literature: for example, Feige et al. [10] gave a polynomial-time $O(\sqrt{\log \mathrm{OPT}})$ approximation for Treewidth and Gupta [14] gave a polynomial-time $O(\mathrm{OPT})$ approximation for Directed Multicut. However, there are no known inapproximability results in this direction. For example, it is not known whether there is a polynomial-time algorithm that, given a graph with maximum clique size k, always finds a clique of size at least, say, $\log\log\log k$. Showing that a certain problem is not fpt-approximable would clearly imply that there is no polynomial-time approximation algorithm with any ratio depending only on the optimum. In particular, there are no such polynomial-time approximation algorithms for the problems considered in this paper (under standard assumptions). Interestingly, the reverse implication also holds [13,15]: if a problem has an fpt