

# Completely inapproximable monotone and antimonotone parameterized problems

Dániel Marx

Tel Aviv University  
Tel Aviv, Israel  
dmarx@cs.bme.hu

## Abstract

We prove that weighted monotone/antimonotone circuit satisfiability has no fixed-parameter tractable approximation algorithm with any approximation ratio function  $\rho$ , unless  $\text{FPT} \neq \text{W}[1]$ . 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  $\rho$ .

## I. Introduction

Recently, there have been increased interest in investigating approximation in the context of parameterized complexity [3]–[5], [8], [14]. 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  $\text{W}[1]$ -hardness of other problems, such as MAXIMUM CLIQUE and MINIMUM DOMINATING SET give 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 [3]–[5], see also the survey [14]. We follow here the notation of [4]. An *fpt approximation algorithm with ratio  $\rho$*  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 maximization problems. Note that the approximation ratio  $\rho$  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  $\rho$ . 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 [4], we say that an algorithm is an *fpt cost approximation* with ratio  $\rho$  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  $\rho$ .

On the positive side, there are a couple of nontrivial fpt approximation algorithms. Grohe and Grüber [12] presented an fpt  $\rho$ -approximation algorithm for MAXIMUM DISJOINT CYCLES in directed graphs, with some unspecified function  $\rho$ . Marx and Razgon [15] gave an fpt 2-approximation algorithm for EDGE MULTICUT. Fellows (unpublished result) showed that TOPOLOGICAL BANDWIDTH has an fpt approximation algorithm with ratio  $k$  (see also [14]). On the negative side, it is known that

WEIGHTED CIRCUIT SATISFIABILITY [4] and INDEPENDENT DOMINATING SET [7] are not fpt-approximable for any function  $\rho$  (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 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, Alekhovich 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. [8] 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  $\rho$ . Our first result confirms this conjecture. The proof is completely different and much simpler than the inapproximability proofs of [1], [8]. 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 [8] 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. [8] raised the question if the maximization problem WEIGHTED ANTIMONOTONE 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  $FPT \neq W[1]$ . The proof is somewhat similar to the inapproximability results of [1], [8] 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 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 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. [9] gave a polynomial-time  $O(\sqrt{\log \text{OPT}})$  approximation for TREEWIDTH and Gupta [13] gave a polynomial-time  $O(\text{OPT})$  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 [12], [14]: if a problem has an fpt  $\rho$ -approximation algorithm for some function  $\rho$ , then there is a polynomial-time approximation algorithm with approximation ratio  $\rho'(\text{OPT})$  for some *other* function  $\rho'$  (we sketch the proof in Section II). This means that if we want to show for example that there is no polynomial-time algorithm finding a clique of size  $f(\text{OPT})$  for any function  $f$ , then we are also showing that the problem is not fpt-approximable. Therefore, it seems that ruling out such polynomial-time

algorithms is essentially a problem belonging to parameterized complexity and requires the understanding of fpt-approximability.

## II. Preliminaries

**Parameterized approximation.** An optimization problem is described by a tuple  $(I, \text{sol}, \text{cost}, \text{goal})$ , where  $I$  is the set of instances,  $\text{sol}(x)$  is the set of feasible solutions for instance  $x$ , the positive integer  $\text{cost}(x, y)$  is the cost of solution  $y$  for instance  $x$ , and  $\text{goal}$  is either  $\min$  or  $\max$ . We follow the notation Chen et al. [4] for the definitions of fpt-approximation:

**Definition 1.** Let  $O = (I, \text{sol}, \text{cost}, \text{goal})$  be an optimization problem (over some alphabet  $\Sigma$ ) and let  $\rho : \mathbb{N} \rightarrow \mathbb{R}$  such that  $\rho(k) \geq 1$  for every  $k \geq 1$  and

$$\begin{cases} k \cdot \rho(k) \text{ is nondecreasing} & \text{if goal} = \min, \\ k/\rho(k) \text{ is unbounded and nondecreasing} & \text{if goal} = \max. \end{cases}$$

An fpt-approximation algorithm with approximation ratio  $\rho$  for  $O$  is an algorithm  $\mathbb{A}$  that, given an input  $(x, k) \in \Sigma^* \times \mathbb{N}$  satisfying  $\text{sol}(x) \neq \emptyset$  and

$$\begin{cases} \text{opt}(x) \leq k & \text{if goal} = \min, \\ \text{opt}(x) \geq k & \text{if goal} = \max, \end{cases} \quad (*)$$

computes a  $y \in \text{sol}(x)$  such that

$$\begin{cases} \text{cost}(x, y) \leq k \cdot \rho(k) & \text{if goal} = \min, \\ \text{cost}(x, y) \geq k/\rho(k) & \text{if goal} = \max. \end{cases}$$

For inputs not satisfying condition (\*), the output can be arbitrary; in particular, this is the case if there is no solution for  $x$ . Furthermore, the running time of  $\mathbb{A}$  on input  $(x, k)$  is  $f(k) \cdot |x|^{O(1)}$  for some computable function  $f$ .

As mentioned in the introduction, an fpt  $\rho$ -approximation algorithm implies that there is polynomial-time approximation algorithm with ratio  $\rho'(\text{OPT})$  for some function  $\rho'$  (see [12], [14]). We sketch the proof here for minimization problems, the proof for maximization problems is analogous. The proof requires the technical condition that we can always find a trivial feasible solution (if exists) in polynomial time. Note that we defined  $\text{cost}(x, y)$  (and hence  $\text{opt}(x)$ ) as a positive integer. Therefore, if some maximization problem  $O$  has an fpt  $\rho$ -approximation for some function  $\rho$ , then running the algorithm with  $k = 1$  produces a feasible solution in polynomial time (if it exists). For minimization problems, the existence of an fpt  $\rho$ -approximation algorithm does not imply that it is always possible to find a feasible solution in polynomial time.

**Theorem 2.** Let  $O$  be a minimization problem such that a feasible solution can be found in polynomial time (if it

exists). If  $O$  has an fpt  $\rho$ -approximation algorithm  $\mathbb{A}$  for some function  $\rho$ , then there is a polynomial-time algorithm  $\mathbb{B}$  and a nondecreasing function  $\rho'$  such that algorithm  $\mathbb{B}$ , given an instance  $x$  of  $O$  with  $\text{sol}(x) \neq \emptyset$ , outputs a solution  $y$  of  $x$  with  $\text{cost}(x, y) \leq \text{opt}(x)\rho'(\text{opt}(x))$ .

*Proof:* Suppose that the running time of  $\mathbb{A}$  can be bounded by  $f(k)|x|^c$  for some function  $f$  and constant  $c$ . Given an instance  $x$ , algorithm  $\mathbb{B}$  does the following. First, it finds a feasible solution  $y_x$  of  $x$  in polynomial time. Then for every  $i = 1, \dots, n$ , algorithm  $\mathbb{B}$  simulates algorithm  $\mathbb{A}$  with input  $(x, i)$  for at most  $|x|^{c+1}$  steps. If the simulation terminates within  $|x|^{c+1}$  steps, then we check if the output is a feasible solution. Let  $y$  be the best feasible solution among the at most  $n$  outputs of the simulations and the feasible solution  $y_x$ .

We claim  $\text{cost}(x, y) \leq \text{opt}(x)\rho'(\text{opt}(x))$  for some function  $\rho'$ . Let  $k := \text{opt}(x)$ . If  $f(k) \leq n$  and  $k \leq n$ , then the simulation of  $\mathbb{A}$  on  $(x, k)$  terminates in  $f(k)|x|^c \leq |x|^{c+1}$  steps with a solution of cost at most  $k \cdot \rho(k)$ , which means that  $\text{cost}(x, y) \leq \text{opt}(x)\rho(\text{opt}(x))$ . Otherwise, let  $\tau(k)$  be the maximum of  $\text{cost}(x, y_x)/\text{opt}(x)$ , taken over all instances of size at most  $\max\{f(k), k\}$ . Note that this is well defined, as there are only a finite number of such instances. Therefore, if  $n \leq \max\{f(k), k\}$ , then  $\text{cost}(x, y) \leq \text{cost}(x, y_x) \leq \text{opt}(x)\tau(\text{opt}(x))$ . Thus the function  $\rho'(k) = \max\{\rho(k), \tau(k)\}$  satisfies the requirements. ■

Chen et al. [4] defined a weaker notion of approximability, which is a decision algorithm solving the gap version of the decision problem associated with the optimization problem. Similarly to [8], we consider this weaker notion in our inapproximability results (thus making the results slightly stronger).

**Definition 3.** Let  $O$  and  $\rho$  be as in Definition 1. A decision algorithm  $\mathbb{A}$  is an fpt cost approximation algorithm for  $O$  with approximation ratio  $\rho$  if for every input  $(x, k) \in \Sigma^* \times \mathbb{N}$  with  $\text{sol}(x) \neq \emptyset$ , its output satisfies the following conditions:

- 1) If  $\begin{cases} k < \text{opt}(x) \text{ and goal} = \min, \\ k > \text{opt}(x) \text{ and goal} = \max, \end{cases}$  then  $\mathbb{A}$  rejects  $(x, k)$ .
- 2) If  $\begin{cases} k \geq \text{opt}(x) \cdot \rho(\text{opt}(x)) \text{ and goal} = \min, \\ k \leq \text{opt}(x)/\rho(\text{opt}(x)) \text{ and goal} = \max, \end{cases}$  then  $\mathbb{A}$  accepts  $(x, k)$ .

Furthermore, the running time of  $\mathbb{A}$  on input  $(x, k)$  is  $f(k) \cdot |x|^{O(1)}$  for some computable function  $f$ .

Clearly, an fpt approximation algorithm with ratio  $\rho$  implies that there is an fpt cost approximation algorithm with the same ratio.

**Circuits.** A Boolean circuit is a directed acyclic graph, where each node with indegree  $> 1$  is labeled as either an AND node or as an OR node, each node of indegree 1 is

labeled as a negation node, and each node of indegree 0 is an input node. Furthermore, there is a node with outdegree 0 that is the output node. Given an assignment  $a$  from the input nodes of circuit  $C$  to  $\{0, 1\}$ , we say that assignment  $a$  satisfies  $C$  if the value of the output node (computed in the obvious way) is 1. The *weight* of an assignment is the number of input nodes with value 1. Circuit  $C$  is  $k$ -satisfiable if there is a weight- $k$  assignment satisfying  $C$ .

We denote by  $|C|$  the number of nodes in the circuit. The *depth* of circuit  $C$  is the maximum length of a directed path from an input node to the output node. The *weft* of a circuit is the maximum number of nodes with indegree  $> 2$  on any path from an input node to the output. Note that any circuit can be transformed into an equivalent circuit of weft 0 by replacing each large node with a sequence of nodes with indegree 2. Thus bounding the weft is meaningful only if we simultaneously bound the depth as well. The notion of weft plays an important role in parameterized complexity and in defining the classes of the W-hierarchy. For the definitions of the classes  $W[1]$ ,  $W[P]$ , etc., the reader is referred to standard texts such as [6], [11]. Let us mention here briefly that a parameterized problem  $Q$  is in the class  $W[t]$  if there is a constant  $d$  such that there is a parameterized reduction from  $Q$  to the satisfiability of circuits with depth  $d$  and weft  $t$ , while  $Q$  is in  $W[P]$  if there is a parameterized reduction from  $Q$  to the circuit satisfiability problem without any restriction on depth and weft. The most important property of a parameterized reduction is that the parameter of the constructed instance is bounded by a function of the original instance.

In the present paper, we investigate weft only to see how restricted the classes of circuits are for which we manage to prove inapproximability and to see what the exact parameterized complexity assumptions are that we need for the results. If the reader is not interested in these issues, then these discussions can be ignored.

A circuit is *monotone* if it contains no negation nodes. A circuit is *antimonotone* if the unique inneighbor of each negation node is an input node, and every outneighbor of an input node is a negation node. We define the following two optimization problems:

<b>MONOTONE CIRCUIT SATISFIABILITY</b>	
<i>Input:</i>	A monotone circuit $C$
<i>Solutions:</i>	All satisfying assignments $a$ of $C$
<i>Cost:</i>	The weight of satisfying assignment $a$ .
<i>Goal:</i>	min.

<b>ANTIMONOTONE CIRCUIT SATISFIABILITY</b>	
<i>Input:</i>	An antimonotone circuit $C$
<i>Solutions:</i>	All satisfying assignments $a$ of $C$
<i>Cost:</i>	The weight of satisfying assignment $a$ .
<i>Goal:</i>	max.

It is known that standard parameterizations of MONOTONE CIRCUIT SATISFIABILITY and ANTIMONOTONE CIRCUIT SATISFIABILITY are W[P]-complete [6], [11].

### III. Monotone problems

We prove our main result on monotone circuits in this section:

**Theorem 4.** MONOTONE CIRCUIT SATISFIABILITY is not fpt cost approximable, unless  $\text{FPT} = \text{W[P]}$ .

*Proof:* Suppose that there is an fpt cost approximation algorithm  $\mathbb{A}$  for MONOTONE CIRCUIT SATISFIABILITY with approximation ratio  $\rho$ . We show that this algorithm  $\mathbb{A}$  can be used to solve the standard parameterization of MONOTONE CIRCUIT SATISFIABILITY in fpt-time, implying  $\text{FPT} = \text{W[P]}$ .

Let  $C$  be a monotone circuit with  $n$  inputs. There is a natural correspondence between the assignments to the  $n$  inputs and the subsets of  $[n]$  (as usual,  $[n]$  denotes  $\{1, \dots, n\}$ ). Thus we can interpret  $C$  as a Boolean function  $C(S)$  defined over the subsets  $S \subseteq [n]$ .

Let  $\mathcal{H}$  be a family of functions from  $[n]$  to  $[k']$ . We say that  $\mathcal{H}$  is a  $k'$ -perfect family of hash functions if for every  $k'$ -element set  $S \subseteq [n]$ , there is an  $h \in \mathcal{H}$  such that  $h$  is one-to-one on  $S$ , i.e.,  $h(s) \neq h(s')$  for every  $s, s' \in S$ ,  $s \neq s'$ . By Alon et al. [2], it is possible to construct a  $k'$ -perfect family  $\mathcal{H}$  of size  $2^{O(k')} \log n$  in time that is polynomial in  $n$  and  $|\mathcal{H}|$ .

Let monotone circuit  $C$  and integer  $k$  be the input of a MONOTONE CIRCUIT SATISFIABILITY instance. Let  $k' := \lceil \rho(k) \cdot k \rceil$  and let  $\mathcal{H}$  be a  $k'$ -perfect family of hash functions from  $[n]$  to  $[k']$ . We define the following function:

$$C'(S) := \bigwedge_{h \in \mathcal{H}} \bigvee_{T \in \binom{[k']}{\leq k}} C(S \cap h^{-1}(T)),$$

where  $\binom{[k']}{\leq k}$  denotes the subsets of  $[k']$  of size at most  $k$  and  $h^{-1}(T) = \{i \in [n] \mid h(i) \in T\}$ . It is clear that we can construct a monotone circuit  $C'$  expressing the function  $C'(S)$  in time  $g(k)|C|^{O(1)}$ . We claim that

- (1) if  $C$  is  $k$ -satisfiable, then  $C'$  is also  $k$ -satisfiable, and
- (2) if  $C$  is not  $k$ -satisfiable, then  $C'$  is not  $k'$ -satisfiable.

Let us run  $\mathbb{A}$  with input  $(C', k')$ ; as the size of  $C'$  is  $g(k)|C|^{O(1)}$  and  $k'$  is a function of  $k$ , the running time of  $\mathbb{A}$  is  $f(k)|C|^{O(1)}$  for some function  $f(k)$ . If  $C$  is  $k$ -satisfiable, then  $\text{opt}(C') \leq k$  and  $k' \geq k \cdot \rho(k) \geq \text{opt}(C') \cdot \rho(\text{opt}(C'))$ , thus  $\mathbb{A}$  accepts. On the other hand, if  $C$  is not  $k$ -satisfiable, then  $\text{opt}(C') > k'$  and  $\mathbb{A}$  rejects. This means that we can solve the standard parameterization of MONOTONE CIRCUIT SATISFIABILITY using algorithm  $\mathbb{A}$ , implying  $\text{FPT} = \text{W[P]}$ .

To prove (1), we show that any weight- $k$  satisfying assignment of  $C$  satisfies  $C'$  as well. Let  $S \subseteq [n]$  be a weight- $k$  satisfying assignment of  $C(S)$ . We have to show that the disjunction in  $C'$  is true for every  $h \in \mathcal{H}$ . Let  $T := \{h(s) : s \in S\}$ ; clearly,  $|T| \leq k$ . By definition,  $S \subseteq h^{-1}(T)$ , thus  $C(S \cap h^{-1}(T)) = C(S) = 1$ . Thus the disjunction is satisfied by the term corresponding to  $T$ .

To prove (2), let  $S$  be a weight- $k'$  satisfying assignment of  $C'$ . Let  $h \in \mathcal{H}$  be a hash function that maps  $S$  one-to-one; since  $\mathcal{H}$  is  $k'$ -perfect, at least one such function exists. We claim that the disjunction corresponding to this  $h$  is not satisfied. To see this, observe that for every  $T \in \binom{[k']}{\leq k}$ , we have  $|S \cap h^{-1}(T)| \leq k$ : for every  $t \in T$ , function  $h$  maps at most one element of  $S$  to  $t$ . Thus if  $f(S \cap h^{-1}(T)) = 1$  for some  $T \in \binom{[k']}{\leq k}$ , then  $S \cap h^{-1}(T)$  is a satisfying assignment of weight at most  $k$  for  $C$ , a contradiction. ■

Inspection of the proof shows that if circuit  $C$  has depth  $d$  and weft  $w$ , then we can construct  $C'$  such that it has depth  $d+2$  and weft  $w+2$ . Since the  $W[2]$ -complete HITTING SET problem can be expressed as a monotone circuit having depth 2, we get the following version of Theorem 4:

**Corollary 5.** *If MONOTONE CIRCUIT SATISFIABILITY is fpt cost approximable for circuits with depth 4, then  $FPT = W[2]$ .*

Note that this corollary shows the inapproximability of a more restricted problem, but the assumption is somewhat stronger than in Theorem 4.

The weft of a depth-4 circuit is clearly at most 4. We can decrease the weft at the cost of increasing the depth as follows. First, the disjunction in  $C'$  can be implemented without increasing the weft by using a composition of OR nodes with indegree two. This increases the depth, but this increase is bounded by a function of  $k$ . Second, instead of starting with a weft-2 circuit  $C$ , we can start with a weft-1 circuit:

**Proposition 6.** *There is a function  $d(k)$  such that the standard parameterization of MONOTONE CIRCUIT SATISFIABILITY is  $W[1]$ -hard for instances with weft 1 and depth bounded by  $d(k)$ .*

*Proof:* We reduce from the  $W[1]$ -hard MULTICOLORED CLIQUE problem [10]: given a graph  $G$  and a proper  $k$ -coloring of the vertices, find a clique that contains exactly one vertex from each color class. Let  $V_1, \dots, V_k$  be the  $k$  color classes. We construct a monotone circuit  $C$  where the inputs  $x_1, \dots, x_n$  correspond to the vertices of  $G$  and  $C$  expresses the following function:

$$\bigwedge_{1 \leq i < j \leq k} \bigvee_{\substack{x_a \in V_i, x_b \in V_j \\ x_a, x_b \text{ are adjacent}}} (x_a \wedge x_b).$$

It is easy to see that every weight- $k$  satisfying assignment

of  $C$  corresponds to a  $k$ -clique of  $G$ . Note that in a weight- $k$  satisfying assignment, in each color class exactly one variable is 1. If the first conjunction is implemented with AND nodes of indegree 2, then the weft of  $C$  is 1 and the depth is bounded by a function  $k$ . ■

Putting together, we obtain:

**Corollary 7.** *There is a function  $d(k)$  such that if MONOTONE CIRCUIT SATISFIABILITY is fpt cost approximable for instances with weft 2 and depth at most  $d(k)$ , then  $FPT = W[1]$ .*

## IV. Antimonotone problems

The main result of this section is that ANTIMONOTONE CIRCUIT SATISFIABILITY is not fpt approximable. We prove the result by showing inapproximability for the following combinatorial problem:

THRESHOLD SET	
<i>Input:</i>	A collection $\mathcal{S}$ of subsets of $U$ with a positive integer weight $w(S)$ for each set $S \in \mathcal{S}$ .
<i>Solutions:</i>	A set $T \subseteq U$ such that $ T \cap S  \leq w(S)$ for every $S \in \mathcal{S}$ .
<i>Cost:</i>	$ T $
<i>Goal:</i>	max.

It is not difficult to express THRESHOLD SET as an antimonotone circuit. In particular, we discuss at the end of this section how to express the THRESHOLD SET instances constructed in the inapproximability proof.

The reduction showing the inapproximability of THRESHOLD SET uses Reed-Solomon codes to construct instances and relies on the erasure correction properties of such codes to argue that finding even an approximate solution is as hard as finding an optimum solution. Let us recall some basic facts about Reed-Solomon codes (no more background is required for understanding the current paper). Let  $F_q$  be the  $q$ -element finite field. For some  $k \leq D < q$ , the Reed-Solomon code is a function  $RS : F_q^k \rightarrow F_q^D$  defined as follows. Let us pick arbitrary distinct nonzero elements  $\alpha_1, \dots, \alpha_D$  from  $F$ . For every  $m = (m_1, \dots, m_k) \in F_q^k$  and  $1 \leq i \leq D$ , we define  $RS(m)$  such that its  $i$ -th component is  $RS(m)[i] := \sum_{j=1}^k \alpha_j^i m_j$ . It is well known that the Reed-Solomon code can correct  $D - k$  erasures, or in other words, the original vector  $m$  can be recovered from any  $k$  components of  $RS(m)$ . We can state this formally as follows:

**Proposition 8.** *For every  $a, b \in F_q^k$  and  $I \subseteq \{1, \dots, D\}$  with  $|I| = k$ , if  $RS(a)[i] = RS(b)[i]$  for every  $i \in I$ , then  $a = b$ .*

*Proof:* Let  $w_i \in F_q^k$  be a column vector whose  $j$ -th component is  $\alpha_j^i$ , which means that  $RS(m)[i] = m \cdot w_i$ . Note

that the vectors  $\{w_i \mid i \in I\}$  are linearly independent by the well-known properties of Vandermonde matrices, hence they form a basis of  $F_q^k$ . Therefore, if  $a$  and  $b$  have the same inner product with every vector in this basis, then  $a = b$ . ■

Now we are ready to prove the inapproximability result for THRESHOLD SET:

**Theorem 9.** THRESHOLD SET is not fpt cost approximable, unless  $\text{FPT} = \text{W}[1]$ .

*Proof:* Suppose that there is an fpt cost approximation algorithm  $\mathbb{A}$  for THRESHOLD SET with approximation ratio  $\rho$ . We show how to solve MAXIMUM CLIQUE in fpt-time using  $\mathbb{A}$ . Let  $G(V, E)$  be a graph where we have to decide if a clique of size  $k$  exists. Let  $D$  be the smallest integer such that  $D/\rho(D) \geq k$ . Note that such a  $D$  exists and can be computed in time depending only on  $k$ . Let  $n$  be the number of vertices of  $G$ . By adding isolated vertices, we can assume that  $n > D^k$  and  $n = 2^{k\ell}$  for some integer  $\ell \geq 1$ ; these additional vertices can increase the size of the graph only by a factor of at most  $D^k \cdot 2^k$ , which is a function of  $k$  only. Let  $q = 2^\ell$  and let  $F_q$  be the  $q$ -element finite field. We identify the vertices of  $G$  with  $F_q^k$ , i.e., we will consider vertices as  $k$ -dimensional row vectors over  $F_q$ . For a vector  $g$ , we denote by  $g[i]$  its  $i$ -th component.

We construct an instance of THRESHOLD SET as follows. The set  $U$  consists of  $n \cdot D$  elements:  $U = \{s_{d,g} : 1 \leq d \leq D, g \in F_q^k\}$ . Let us define the collection  $\mathcal{S}$  of subsets in the input. First, for every  $1 \leq d \leq D$ ,  $\mathcal{S}$  contains the set  $S_d := \{s_{d,g} : g \in F_q^k\}$ . By setting  $w(S_d) = 1$  for every  $1 \leq d \leq D$ , we ensure that the solution  $T$  contains at most one element with first index  $d$ . Thus a solution can be interpreted as a collection of at most  $D$  vectors from  $F_q^k$ . Equivalently, every solution can be interpreted as a table of size  $k \times D$ , where each entry is either empty or contains an element of  $F$ . More precisely, if the solution contains element  $s_{d,g}$  of  $S_d$ , then the  $d$ -th column contains components of the  $k$ -dimensional vector  $g \in F_q^k$ ; if the solution contains no element of  $S_d$ , then the  $d$ -th column of the table is empty. This table will be interpreted as an encoding of the  $k$  vertices of the clique: the  $D$ -dimensional vector in the  $i$ -th row is interpreted as the encoding of the  $i$ -th vertex of the clique by a Reed-Solomon code  $F_q^k \rightarrow F_q^D$ . (Note that the  $k$ -dimensional column vectors of this table could be also interpreted as vertices, but this is not what we are doing here.) By the properties of the Reed-Solomon code, any  $k$  full columns already describe a clique of size  $k$ . Note that  $n = 2^{k\ell}$  and  $n > D^k$  implies  $D < q$ , as required by the Reed-Solomon code.

In order to enforce this interpretation, we add further sets to  $\mathcal{S}$  as follows. Let  $(X, a, b, u, v)$  be such that

- $X$  is a  $k$ -element subset of  $\{1, \dots, D\}$ ,
- $1 \leq a < b \leq k$ ,

- $u, v \in F_q^k$  are nonadjacent vertices of  $G$  (including the possibility  $u = v$ ).

For each such 5-tuple, we add the following set to  $\mathcal{S}$ :

$$S_{X,a,b,u,v} = \{s_{j,g} : j \in X, g \in F_q^k, g[a] = RS(u)[j], g[b] = RS(v)[j]\}.$$

The weight of each such set is  $k - 1$ . If we interpret a solution as a table of size  $k \times D$  (as described above), then weight  $k - 1$  of the set  $S_{X,a,b,u,v}$  ensures that it is not possible that the entries in row  $a$  and columns  $X$  agree with  $RS(u)$ , while the entries in row  $b$  and columns  $X$  agree with  $RS(v)$ . As we want the  $a$ -th and  $b$ -th vertex of the clique to be adjacent, we require this for every nonadjacent  $u$  and  $v$ . This completes the construction of the THRESHOLD SET instance  $x$ . Note that the size of instance  $x$  is  $g(k)n^{O(1)}$  for some function  $g(k)$  depending only on  $k$ .

We claim that

- (1) if  $G$  has a  $k$ -clique, then  $x$  has a solution  $T$  of size  $D$  and
- (2) if  $G$  has no  $k$ -clique, then every solution of  $x$  has size less than  $k$ .

If these claims are true, then we can decide whether  $G$  has a  $k$ -clique by running  $\mathbb{A}$  on  $(x, k)$ . If  $G$  has a  $k$ -clique, then we have  $\text{opt}(x) \geq D$ , which means that  $\text{opt}(x)/\rho(\text{opt}(x)) \geq D/\rho(D) \geq k$ , and  $\mathbb{A}$  accepts. On the other hand, if there is no  $k$ -clique, then  $\text{opt}(x) < k$ , and  $\mathbb{A}$  rejects. As the size of  $x$  is  $g(k)n^{O(1)}$ , the running time of  $\mathbb{A}$  can be bounded as  $f(k)n^{O(1)}$  for some computable function  $f(k)$ . It follows that the construction of the THRESHOLD SET instance and running algorithm  $\mathbb{A}$  is an fpt-time algorithm for solving the MAXIMUM CLIQUE problem, implying  $\text{FPT} = \text{W}[1]$ .

To prove (1), suppose that  $G$  has a  $k$ -clique  $v_1, \dots, v_k$ . For every  $1 \leq j \leq D$ , we define the vector  $g_j \in F_q^k$  such that  $g_j[i] = RS(v_i)[j]$  for every  $1 \leq i \leq k$ . We claim that the  $D$ -element set  $T := \{s_{j,g_j} : 1 \leq j \leq D\}$  is a feasible solution. It is clear that the edge  $S_j$  contains exactly one element of  $T$ . Let us verify that every set  $S_{X,a,b,u,v}$  of  $\mathcal{S}$  contains at most  $k - 1$  elements of  $T$ . Since  $u$  and  $v$  are not adjacent and  $v_a$  and  $v_b$  are adjacent, at least one of  $v_a \neq u$  or  $v_b \neq v$  holds. Suppose that  $v_a \neq u$  (the case  $v_b \neq v$  is similar). By Prop. 8, there has to be a  $j \in X$  such that  $RS(v_a)[j] \neq RS(u)[j]$ . This implies that  $s_{j,g_j} \in T$  is not in  $S_{X,a,b,u,v}$ : we have  $g_j[a] = RS(v_a)[j]$  by the definition of  $g_j$ , while  $s_{j,g} \in S_{X,a,b,u,v}$  only if  $g[a] = RS(u)[j]$ . Therefore,  $S_{X,a,b,u,v} \cap T$  contains no element from  $S_j$ , implying that the intersection has size at most  $k - 1$ .

To prove (2), suppose now that there is a solution  $T$  of size at least  $k$ . Define  $J \subseteq \{1, \dots, D\}$  such that  $j \in J$  if and only if  $S_j \cap T \neq \emptyset$ . Since  $|S_j \cap T| \leq 1$  for every  $1 \leq j \leq D$ , we have  $|J| \geq k$ . Let  $X$  be a  $k$ -element subset of  $J$ . For every  $j \in X$ , there is a unique value  $g_j$  such that  $s_{j,g_j} \in T$ . For  $1 \leq i \leq k$ , let  $v_i$  be the unique vertex

satisfying  $RS(v_i)[j] = g_j[i]$  for every  $j \in X$  (the existence and the uniqueness of  $v_i$  follows from Prop. 8). We claim that  $v_1, \dots, v_k$  form a clique in  $G$ . Suppose that  $v_a$  and  $v_b$  are not adjacent. Then the set  $S_{X,a,b,v_a,v_b}$  is a member of  $\mathcal{S}$ . It is easy to see that  $v_{j,g_j}$  is in this set for every  $j \in X$ : by the definition of  $v_a$  and  $v_b$ , we have  $g_j[a] = RS(v_a)[j]$  and  $g_j[b] = RS(v_b)[j]$  for every  $a, b \in X$ . However, the set  $S_{X,a,b,v_a,v_b}$  has weight  $k - 1 < |X|$ , a contradiction. ■

Let us discuss how the THRESHOLD SET instances constructed in the proof can be expressed as circuits. Let the inputs  $s_{j,g}$  correspond to the elements of  $U$ . The requirement  $|T \cap S_j| \leq 1$  can be expressed by requiring  $\bar{s}_{j,g'} \vee \bar{s}_{j,g''}$  for every distinct  $g', g'' \in F^t$ . The set  $S_{X,a,b,v_a,v_b}$  (which has weight  $k - 1$ ) intersects  $S_j$  only if  $j \in X$ , that is, for  $k$  values of  $j$ . Therefore,  $|T \cap S_{X,a,b,v_a,v_b}| \leq k - 1$  is true if and only if there is a  $j \in X$  such that  $T$  and  $S_{X,a,b,u,v} \cap S_j$  are disjoint. This means that the THRESHOLD SET instance can be expressed as

$$\left( \bigwedge_{j=1}^D \bigwedge_{\substack{g', g'' \in F^t \\ g' \neq g''}} (\bar{s}_{j,g'} \vee \bar{s}_{j,g'') \right) \wedge \left( \bigwedge_{S_{X,a,b,u,v} \in \mathcal{S}} \bigvee_{j \in X} \bigwedge_{s_{j,g} \in S_{X,a,b,u,v}} \bar{s}_{j,g} \right).$$

We obtain that the instance can be implemented by an antimonotone formula of depth 3. If the first conjunction and the disjunction over  $j \in X$  are implemented by nodes of indegree 2, then we can get weft-2 circuit having depth bounded by a function of  $k$ .

**Corollary 10.** *There is a function  $d(k)$  such that if ANTIMONOTONE CIRCUIT SATISFIABILITY is fpt cost approximable for instances of depth 3, or for instances of weft 2 and depth bounded by  $d(k)$ , then  $FPT = W[1]$ .*

## Acknowledgments

I'm grateful to Martin Grohe for useful discussions and to the anonymous referees for comments that improved the presentation of the paper.

## References

- [1] M. Alekhovich and A. A. Razborov. Resolution is not automatizable unless  $w[p]$  is tractable. *SIAM J. Comput.*, 38(4):1347–1363, 2008.
- [2] N. Alon, R. Yuster, and U. Zwick. Color-coding. *J. Assoc. Comput. Mach.*, 42(4):844–856, 1995.
- [3] L. Cai and X. Huang. Fixed-parameter approximation: conceptual framework and approximability results. In *Proceedings of the International Workshop on Parameterized and Exact Computation (IWPEC 2006)*, volume 4169 of *Lecture Notes in Computer Science*, pages 96–108, 2006.
- [4] Y. Chen, M. Grohe, and M. Grüber. On parameterized approximability. In *Proceedings of the International Workshop on Parameterized and Exact Computation (IWPEC 2006)*, volume 4169 of *Lecture Notes in Computer Science*, pages 109–120, 2006.
- [5] R. Downey, M. Fellows, and C. McCartin. Parameterized approximation algorithms. In *Proceedings of the International Workshop on Parameterized and Exact Computation (IWPEC 2006)*, volume 4169 of *Lecture Notes in Computer Science*, pages 121–129, 2006.
- [6] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, New York, 1999.
- [7] R. G. Downey, M. R. Fellows, C. McCartin, and F. Rosamond. Parameterized approximation of dominating set problems. *Inform. Process. Lett.*, 109(1):68–70, 2008.
- [8] K. Eickmeyer, M. Grohe, and M. Grüber. Approximation of natural  $W[P]$ -complete minimisation problems is hard. In *IEEE Conference on Computational Complexity*, pages 8–18, 2008.
- [9] U. Feige, M. Hajiaghayi, and J. R. Lee. Improved approximation algorithms for minimum weight vertex separators. *SIAM J. Comput.*, 38(2):629–657, 2008.
- [10] M. R. Fellows, D. Hermelin, F. A. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems. *Theor. Comput. Sci.*, 410(1):53–61, 2009.
- [11] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, Berlin, 2006.
- [12] M. Grohe and M. Grüber. Parameterized approximability of the disjoint cycle problem. In *ICALP*, pages 363–374, 2007.
- [13] A. Gupta. Improved results for directed multicut. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (Baltimore, MD, 2003)*, pages 454–455, New York, 2003. ACM.
- [14] D. Marx. Parameterized complexity and approximation algorithms. *The Computer Journal*, 51(1):60–78, 2008.
- [15] D. Marx and I. Razgon. Constant ratio fixed-parameter approximation of the edge multicut problem. *Information Processing Letters*, 109(20):1161–1166, 2009.