Fixed-Parameter Approximability of Boolean MinCSPs*

Édouard Bonnet[1], László Egri[†2], and Dániel Marx[3]

1 **Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI), Budapest, Hungary**
bonnet.edouard@sztaki.mta.hu
2 **School of Computing Science, Simon Fraser University, Burnaby, Canada**
laszlo.egri@mail.mcgill.ca
3 **Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI), Budapest, Hungary**
dmarx@cs.bme.hu

Abstract

The minimum unsatisfiability version of a constraint satisfaction problem (MinCSP) asks for an assignment where the number of unsatisfied constraints is minimum possible, or equivalently, asks for a minimum-size set of constraints whose deletion makes the instance satisfiable. For a finite set Γ of constraints, we denote by MinCSP(Γ) the restriction of the problem where each constraint is from Γ. The polynomial-time solvability and the polynomial-time approximability of MinCSP(Γ) were fully characterized by Khanna et al. [33]. Here we study the fixed-parameter (FP-) approximability of the problem: given an instance and an integer k, one has to find a solution of size at most $g(k)$ in time $f(k) \cdot n^{O(1)}$ if a solution of size at most k exists. We especially focus on the case of constant-factor FP-approximability. Our main result classifies each finite constraint language Γ into one of three classes: (1) MinCSP(Γ) has a constant-factor FP-approximation; (2) MinCSP(Γ) has a (constant-factor) FP-approximation if and only if Nearest Codeword has a (constant-factor) FP-approximation; (3) MinCSP(Γ) has no FP-approximation, unless FPT = W[P]. We show that problems in the second class do not have constant-factor FP-approximations if both the Exponential-Time Hypothesis (ETH) and the Linear PCP Conjecture (LPC) hold. We also show that such an approximation would imply the existence of an FP-approximation for the k-Densest Subgraph problem with ratio $1 - \epsilon$ for any $\epsilon > 0$.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases constraint satisfaction problems, approximability, fixed-parameter tractability

Digital Object Identifier 10.4230/LIPIcs.ESA.2016.18

1 Introduction

Satisfiability problems and, more generally, Boolean constraint satisfaction problems (CSPs) are basic algorithmic problems arising in various theoretical and applied contexts. An instance of a Boolean CSP consists of a set of Boolean variables and a set of constraints; each

* This work was supported by the European Research Council (ERC) starting grant "PARAMTIGHT: Parameterized complexity and the search for tight complexity results" (reference 280152) and OTKA grant NK105645.
† The second author was supported by NSERC.

24th Annual European Symposium on Algorithms (ESA 2016).
Editors: Piotr Sankowski and Christos Zaroliagis; Article No. 18; pp. 18:1–18:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

constraint restricts the allowed combination of values that can appear on a certain subset of variables. In the decision version of the problem, the goal is to find an assignment that simultaneously satisfies every constraint. One can also define optimization versions of CSPs: the goal can be to find an assignment that maximizes the number of satisfied constraints, minimizes the number of unsatisfied constraints, maximizes/minimizes the weight (number of 1s) of the assignment, etc. [19].

Since these problems are usually NP-hard in their full generality, a well-established line of research is to investigate how the complexity of the problem changes for restricted versions of the problem. A large body of research deals with language-based restrictions: given any finite set Γ of Boolean constraints, one can consider the special case where each constraint is restricted to be a member of Γ. The ultimate research goal of this approach is to prove a *dichotomy theorem*: a complete classification result that specifies for each finite constraint set Γ whether the restriction to Γ yields and easy or hard problem. Numerous classification theorems of this form have been proved for various decision and optimization versions for Boolean and non-Boolean CSPs [46, 13, 10, 11, 9, 12, 8, 26, 32, 34, 47, 38]. In particular, for MinCSP(Γ), which is the optimization problem asking for an assignment minimizing the number of unsatisfied constraints, Creignou et al. [19] obtained a classification of the approximability for every finite Boolean constraint language Γ. The goal of this paper is to characterize the approximability of Boolean MinCSP(Γ) with respect to the more relaxed notion of fixed-parameter approximability.

Parameterized complexity [27, 29, 23] analyzes the running time of a computational problem not as a univariate function of the input size n, but as a function of both the input size n and a relevant parameter k of the input. For example, given a MinCSP instance of size n where we are looking for a solution satisfying all but k of the constraints, it is natural to analyze the running time of the problem as a function of both n and k. We say that a problem with parameter k is *fixed-parameter tractable (FPT)* if it can be solved in time $f(k) \cdot n^{O(1)}$ for some computable function f depending only on k. Intuitively, even if f is, say, an exponential function, this means that problem instances with "small" k can be solved efficiently, as the combinatorial explosion can be confined to the parameter k. This can be contrasted with algorithms with running time of the form $n^{O(k)}$ that are highly inefficient even for small values of k. There are hundreds of parameterized problems where brute force gives trivial $n^{O(k)}$ algorithms, but the problem can be shown to be FPT using nontrivial techniques; see the recent textbooks by Downey and Fellows [27] and by Cygan et al. [23]. In particular, there are fixed-parameter tractability results and characterization theorems for various CSPs [38, 13, 35, 36].

The notion of fixed-parameter tractability has been combined with the notion of approximability [16, 17, 28, 14, 18]. Following [16, 39], we say that a minimization problem is *fixed-parameter approximable (FPA)* if there is an algorithm that, given an instance and an integer k, in time $f_1(k) \cdot n^{O(1)}$ either returns a solution of cost at most $f_2(k) \cdot k$, or correctly states that there is no solution of cost at most k. The two crucial differences compared to the usual setup of polynomial-time approximation is that (1) the running time is not polynomial, but can have an arbitrary factor $f(k)$ depending only on k and (2) the approximation ratio is defined not as a function of the input size n but as a function of k. In this paper, we mostly focus on the case of constant-factor FPA, that is, when $f_2(k) = c$ for some constant c.

Schaefer's Dichotomy Theorem [46] identified six classes of finite Boolean constraint languages (0-valid, 1-valid, Horn, dual-Horn, bijunctive, affine) for which the decision CSP is polynomial-time solvable, and shows that every language Γ outside these classes yields NP-hard problems. Therefore, one has to study MinCSP only within these six classes, as it is otherwise already NP-hard to decide if the optimum is 0 or not, making approximation or fixed-parameter tractability irrelevant. Within these classes, polynomial-time approximability and fixed-parameter tractability seem to appear in orthogonal ways: the classes where we have positive results for one approach is very different from the classes where the other approach helps. For example, 2CNF Deletion (also called Almost 2SAT) is fixed-parameter tractable [45, 37], but has no polynomial-time approximation algorithm with constant approximation ratio, assuming the Unique Games Conjecture [15]. On the other hand, if Γ consists of the three constraints (x), $(\bar{x})$, and $(a \to b) \wedge (c \to d)$, then the problem is W[1]-hard [41], but belongs to the class IHS-B[1] and hence admits a constant-factor approximation in polynomial time [33].

By investigating constant-factor FP-approximation, we are identifying a class of tractable constraints that unifies and generalizes the polynomial-time constant-factor approximable and fixed-parameter tractable cases. We observe that if each constraint in Γ can be expressed by a 2SAT formula (i.e., Γ is bijunctive), then we can treat the MinCSP instance as an instance of 2SAT Deletion, at the cost of a constant-factor loss in the approximation ratio. Thus the fixed-parameter tractability of 2SAT Deletion implies MinCSP has a constant-factor FP-approximation if the finite set Γ is bijunctive. If Γ is in IHS-B, then MinCSP is known to have a constant-factor approximation in polynomial time, which clearly gives another class of constant-factor FP-approximable constraints. Our main results show that probably these two classes cover all the easy cases with respect to FP-approximation (see Section 2 for the definitions involving properties of constraints).

▶ **Theorem 1.** *Let Γ be a finite Boolean constraint language.*
1. *If Γ is bijunctive or IHS-B, then* MinCSP(Γ) *has a constant-factor FP-approximation.*
2. *Otherwise, if Γ is affine, then* MinCSP(Γ) *has an FP-approximation (resp., constant-factor FP-approximation) if and only if* Nearest Codeword *has an FP-approximation (resp., constant-factor FP-approximation).*
3. *Otherwise,* MinCSP(Γ) *has no fixed-parameter approximation, unless* FPT = W[P].

Given a linear code over $GF[2]$ and a vector, the Nearest Codeword (NC) problem asks for a codeword in the code that has minimum Hamming distance to the given vector. There are various equivalent formulations of this problem: Odd Set is a variant of Hitting Set where one has to select at most k elements to hit each set exactly an odd number of times, and it is also possible to express the problem as finding a solution to a system of linear equations over $GF[2]$ that minimizes the number of unsatisfied equations. Arora et al. [2] showed that, assuming NP $\not\subseteq$ DTIME($n^{\text{polylog}\, n}$), it is not possible to approximate NC within ratio $2^{\log^{1-\epsilon} n}$ for any $\epsilon > 0$. In particular, this implies that a constant-factor polynomial-time approximation is unlikely. We give some evidence that even constant-factor FP-approximation is unlikely. First, we rule out this possibility under the assumption that the Linear PCP Conjecture (LPC) and the Exponential-Time Hypothesis (ETH) both hold.

▶ **Theorem 2.** *Assuming LPC and ETH, for any constant r,* NC *has no factor-r FP-approximation.*

Second, we connect the FP-approximability of NC with the k-Densest Subgraph problem, where the task is to find k vertices that induce the maximum number of edges.

▶ **Theorem 3.** *If* NC *has a factor-r FP-approximation for some constant r, then for every $\epsilon > 0$, there is a factor-$(1-\epsilon)$ FP-approximation for k-*Densest Subgraph.

[1] IHS-B stands for Implicative Hitting Set-Bounded, see definition in Section 2.

Thus a constant-factor FP-approximation for NC implies that k-Densest Subgraph can be approximated arbitrarily well, which seems unlikely. Note that Theorems 2 and 3 remain valid for the other equivalent versions of NC, such as Odd Set. These theorems form the technically more involved parts of the paper.

Post's lattice is a very useful tool for classifying the complexity of Boolean CSPs (see e.g., [1, 20, 3]). A (possibly infinite) set Γ of constraints is a co-clone if it is closed under pp-definitions, that is, whenever a relation R can be expressed by relations in Γ using only equality, conjunctions, and projections, then relation R is already in Γ. Post's co-clone lattice characterizes every possible co-clone of Boolean constraints. From the complexity-theoretic point of view, Post's lattice becomes very relevant if the complexity of the CSP problem under study does not change by adding new pp-definable relations to the set Γ of allowed relations. For example, this is true for the decision version of Boolean CSP. In this case, it is sufficient to determine the complexity for each co-clone in the lattice, and a complete classification for every finite set Γ of constraints follows. For MinCSP, neither the polynomial-time solvability nor the fixed-parameter tractability of the problem is closed under pp-definitions, hence Post's lattice cannot be used directly to obtain a complexity classification. However, as observed by Khanna et al. [33] and subsequently exploited by Dalmau et al. [24, 25], the constant-factor approximability of MinCSP is closed under pp-definitions (modulo a small technicality related to equality constraints). We observe that the same holds for constant-factor FP-approximability and hence Post's lattice can be used for our purposes. Thus, the classification result amounts to identifying the maximal easy and the minimal hard co-clones.

The paper is organized as follows. Sections 2 and 3 contain preliminaries on CSPs, approximability, Post's lattice, and reductions. A more technical restatement of Theorem 1 in terms of co-clones is stated at the end of Section 3. Section 4 gives FPA algorithms, Section 5 establishes the equivalence of some CSPs with Odd Set, and Section 6 proves inapproximability results for CSPs. Section 7 proves Theorems 2 and 3, the conditional hardness results for Odd Set. Due to space restrictions, less difficult proofs appear only in the arxiv version [6].

2 Preliminaries

A subset R of $\{0,1\}^n$ is called an n-ary *Boolean relation.* If $n = 2$, relation R is *binary.* In this paper, a *constraint language* Γ is a finite collection of finitary Boolean relations. When a constraint language Γ contains only a single relation R, i.e., $\Gamma = \{R\}$, we write R instead of $\{R\}$. The decision version of CSP, restricted to finite constraint language Γ is defined as:

CSP(Γ)
Input: A pair $\langle V, \mathcal{C}\rangle$, where
- V is a set of variables,
- $\mathcal{C}$ is a multiset of constraints $\{C_1, \ldots, C_q\}$, i.e., $C_i = \langle s_i, R_i\rangle$, where s_i is a tuple of variables of length n_i, and $R_i \in \Gamma$ is an n_i-ary relation.

Question: Does there exist a solution, that is, a function $\varphi : V \to \{0,1\}$ such that for each constraint $\langle s, R\rangle \in \mathcal{C}$, with $s = \langle v_1, \ldots, v_n\rangle$, the tuple $\varphi(v_1), \ldots, \varphi(v_n)$ belongs to R?

Note that we can alternatively look at a constraint as a Boolean function $f : \{0,1\}^n \to \{0,1\}$, where n is a non-negative integer called the arity of f. We say that f is satisfied by an assignment $s \in \{0,1\}^n$ if $f(s) = 1$. For example, if $f(x,y) = x + y \mod 2$, then the corresponding relation is $\{(0,1),(1,0)\}$; we also denote addition modulo 2 with $x \oplus y$.

We recall the definition of a few well-known classes of constraint languages. A Boolean constraint language Γ is:

- *0-valid* (*1-valid*), if each $R \in \Gamma$ contains a tuple in which all entries are 0 (1);
- *k-IHS-B+* (*k-IHS-B–*), where $k \in \mathbb{Z}^+$, if each $R \in \Gamma$ can be expressed by a conjunction of clauses of the form $\neg x$, $\neg x \vee y$, or $x_1 \vee \cdots \vee x_k$ (x, $\neg x \vee y$, $\neg x_1 \vee \cdots \vee \neg x_k$); *IHS-B+* (*IHS-B–*) stands for k-IHS-B+ (k-IHS-B–) for some k; *IHS-B* stands for IHS-B+ or IHS-B–;
- *bijunctive*, if each $R \in \Gamma$ can be expressed by a conjunction of binary clauses;
- *Horn* (*dual-Horn*), if each $R \in \Gamma$ can be expressed by a conjunction of Horn (dual-Horn) clauses, i.e., clauses that have at most one positive (negative) literal;
- *affine*, if each relation $R \in \Gamma$ can be expressed by a conjunction of relations defined by equations of the form $x_1 \oplus \cdots \oplus x_n = c$, where $c \in \{0, 1\}$;
- *self-dual* if for each relation $R \in \Gamma$, $(a_1, \ldots, a_n) \in R \Rightarrow (\neg a_1, \ldots, \neg a_n) \in R$.

MinCSP(Γ)
Input: An instance $\langle V, \mathcal{C} \rangle$ of CSP(Γ), and an integer k.
Question: Is there a deletion set $W \subseteq \mathcal{C}$ such that $|W| \leq k$, and the CSP(Γ)-instance $\langle V, \mathcal{C} \setminus W \rangle$ has a solution?

MinCSP*(Γ)
Input: An instance $\langle V, \mathcal{C} \rangle$ of CSP(Γ), a subset $\mathcal{C}^* \subseteq \mathcal{C}$ of undeletable constraints, and an integer k.
Question: Is there a deletion set $W \subseteq \mathcal{C} \setminus \mathcal{C}^*$ such that $|W| \leq k$ and the CSP(Γ)-instance $\langle V, \mathcal{C} \setminus W \rangle$ has a solution?

For every finite constraint language Γ, we consider the problem MinCSP above. For technical reasons, it will be convenient to work with a slight generalization of the problem, MinCSP*(defined above), where we can specify that certain constraints are "undeletable." For these two problems, a set of potentially more than k constraints whose removal yields a satisfiable instance is called a *feasible solution*. Note that, contrary to MinCSP for which removing all the constraints constitute a trivially feasible solution, it is possible that an instance of MinCSP* has no feasible solution. A *feasible instance* is an instance that admits at least one feasible solution. We will use two types of reductions to connect the approximability of optimization problems. The first type perfectly preserves the optimum value (or cost) of instances.

▶ **Definition 4.** An optimization problem A has a *cost-preserving reduction* to problem B if there are two polynomial-time computable functions F and G such that

1. For any feasible instance I of A, $F(I)$ is a feasible instance of B having the same optimum cost as I.
2. For any feasible instance I of A, if S' is a feasible solution for $F(I)$, then $G(I, S')$ is a feasible solution of I having cost at most the cost of $F(I)$.

The following easy lemma shows that the existence of undeletable constraints does not make the problem significantly more general. Note that, in the previous definition, if instance I has no feasible solution, then the behavior of F on I is not defined.

▶ **Lemma 5.** *There is a cost-preserving reduction from* MinCSP* *to* MinCSP*.*

The second type of reduction that we use is the standard notion of A-reductions [21], which preserve approximation ratios up to constant factors. We slightly deviate from the standard definition by not requiring any specific behavior of F when I has no feasible solution.

▶ **Definition 6.** A minimization problem A is *A-reducible* to problem B if there are two polynomial-time computable functions F and G and a constant α such that

1. For any feasible instance I of A, $F(I)$ is a feasible instance of B.
2. For any feasible instance I of A, and any feasible solution S' of $F(I)$, $G(I, S')$ is a feasible solution for I.
3. For any feasible instance I of A, and any $r \geq 1$, if S' is an r-approximate feasible solution for $F(I)$, then $G(I, S')$ is an (αr)-approximate feasible solution for I.

▶ **Proposition 7.** *If optimization problem A is A-reducible to optimization problem B and B admits a constant-factor FPA algorithm, then A also has a constant-factor FPA algorithm.*

3 Post's lattice, co-clone lattice, and a simple reduction

A *clone* is a set of Boolean functions that contains all projections (that is, the functions $f(a_1, \ldots, a_n) = a_k$ for $1 \leq k \leq n$) and is closed under arbitrary composition. *All* clones of Boolean functions were identified by Post [44], and he also described their inclusion structure, hence the name Post's lattice. To make use of this lattice for CSPs, Post's lattice can be transformed to another lattice whose elements are not sets of functions closed under composition, but sets of relations closed under the following notion of definability.

▶ **Definition 8.** Let Γ be a constraint language over some domain A. We say that a relation R is *pp-definable* from Γ if there exists a (primitive positive) formula $\varphi(x_1, \ldots, x_k) \equiv \exists y_1, \ldots, y_l \psi(x_1, \ldots, x_k, y_1, \ldots, y_l)$, where ψ is a conjunction of atomic formulas with relations in Γ and EQ_A (the binary relation $\{(a, a) : a \in A\}$) such that for every $(a_1, \ldots, a_k) \in A^k$ $(a_1, \ldots, a_k) \in R$ if and only if $\varphi(a_1, \ldots, a_k)$ holds. If ψ does not contain EQ_A, then we say that R is *pp-definable* from Γ *without equality*. For brevity, we often write "$\exists\wedge$-definable" instead of "pp-definable without equality". If S is a set of relations, S is *pp-definable* ($\exists\wedge$-definable) from Γ if every relation in S is *pp*-definable ($\exists\wedge$-definable) from Γ.

For a set of relations Γ, we denote by $\langle\Gamma\rangle$ the set of all relations that can be pp-defined over Γ. We refer to $\langle\Gamma\rangle$ as the *co-clone* generated by Γ. The set of all co-clones forms a lattice. To give an idea about the connection between Post's lattice and the co-clone lattice, we briefly mention the following theorem, and refer the reader to, for example, [5] for more information. Roughly speaking, the following theorem says that the co-clone lattice is essentially Post's lattice turned upside down, i.e., the inclusion between neighboring nodes are inverted.

▶ **Theorem 9** ([43], Theorem 3.1.3)**.** *The lattices of Boolean clones and Boolean co-clones are anti-isomorphic.*

Using the above comments, it can be seen (and it is well known) that the lattice of Boolean co-clones has the structure shown in Figure 1. In the figure, if co-clone C_2 is above co-clone C_1, then $C_2 \supset C_1$. The names of the co-clones are indicated in the nodes[2], where we follow the notation of Böhler et al [5].

[2] If the name of a clone is L_3, for example, then the corresponding co-clone is $\mathrm{Inv}(L_3)$ (Inv is defined, for example, in [5]), which is denoted by IL_3.

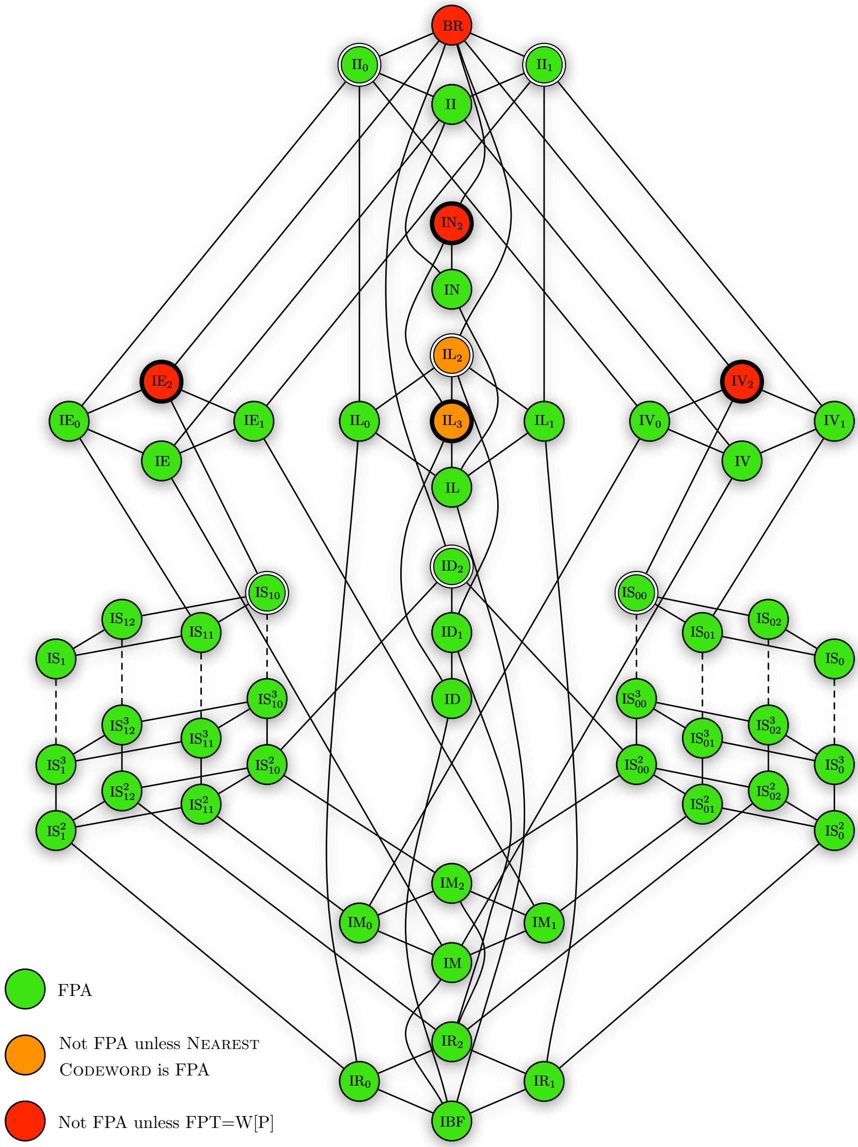

Figure 1 Classification of Boolean CSPs according to constant ratio fixed-parameter approximability. (*We thank Heribert Vollmer and Yuichi Yoshida for giving us access to their Post's lattice diagrams.*)

For a co-clone C we say that a set of relations Γ is a *base* for C if $C = \langle\Gamma\rangle$, that is, any relation in C can be pp-defined using relations in Γ. Böhler et al. give bases for all co-clones in [5], and the reader can consult this paper for details. We reproduce this list in Table 1.[3]

It is well-known that pp-definitions preserve the complexity of the decision version of CSP: if $\Gamma_2 \subseteq \langle\Gamma_1\rangle$ for two finite languages Γ_1 and Γ_2, then there is a natural polynomial-time

[3] We note that EVEN^4 can be pp-defined using DUP^3. Therefore the base $\{\text{DUP}^3, \text{EVEN}^4, x \oplus y\}$ given by Böhler et al. [5] for IN_2 can be actually simplified to $\{\text{DUP}^3, x \oplus y\}$.

Table 1 Bases for all Boolean co-clones. (See [5] for a complete definition of relations that appear.) The order of a co-clone is the minimum over all bases of the maximum arity of a relation in the base. The order is defined to be infinite if there is no finite base for that co-clone.

Co-clone	Order	Base	Co-clone	Order	Base
IBF	0	$\{=\}$, $\{\emptyset\}$	IS_{10}	∞	$\{\mathrm{NAND}^m \mid m \geq 2\} \cup \{x, \bar{x}, x \to y\}$
IR_0	1	$\{\bar{x}\}$	ID	2	$\{x \oplus y\}$
IR_1	1	$\{x\}$	ID_1	2	$\{x \oplus y, x\}$, every $R \in \{\{(a_1, a_2, a_3), (b_1, b_2, b_3)\} \mid \exists c \in \{1, 2\}$ such that $\sum_{i=1}^{3} a_i = \sum_{i=i}^{3} b_i = c\}$
IR_2	1	$\{x, \bar{x}\}, \{x\bar{x}\}$	ID_2	2	$\{x \oplus y, x \to y\}, \{x\bar{y}, \bar{x}yz\}$
IM	2	$\{x \to y\}$	IL	4	$\{\mathrm{EVEN}^4\}$
IM_1	2	$\{x \to y, x\}, \{x \wedge (y \to z)\}$	IL_0	3	$\{\mathrm{EVEN}^4, \bar{x}\}, \{\mathrm{EVEN}^3\}$
IM_0	2	$\{x \to y, \bar{x}\}, \{\bar{x} \wedge (y \to z)\}$	IL_1	3	$\{\mathrm{EVEN}^4, x\}, \{\mathrm{ODD}^3\}$
IM_2	2	$\{x \to y, x, \bar{x}\}, \{x \to y, \overline{x \to y}\}, \{x\bar{y} \wedge (u \to v)\}$	IL_2	3	$\{\mathrm{EVEN}^4, x, \bar{x}\}$, every $\{\mathrm{EVEN}^n, x\}$ where $n \geq 3$ is odd
IS_0^m	m	$\{\mathrm{OR}^m\}$	IL_3	4	$\{\mathrm{EVEN}^4, x \oplus y\}, \{\mathrm{ODD}^4\}$
IS_1^m	m	$\{\mathrm{NAND}^m\}$	IV	3	$\{x \vee y \vee \bar{z}\}$
IS_0	∞	$\{\mathrm{OR}^m \mid m \geq 2\}$	IV_0	3	$\{x \vee y \vee \bar{z}, \bar{x}\}$
IS_1	∞	$\{\mathrm{NAND}^m \mid m \geq 2\}$	IV_1	3	$\{x \vee y \vee \bar{z}, x\}$
IS_{02}^m	m	$\{\mathrm{OR}^m, x, \bar{x}\}$	IV_2	3	$\{x \vee y \vee \bar{z}, x, \bar{x}\}$
IS_{02}	∞	$\{\mathrm{OR}^m \mid m \geq 2\} \cup \{x, \bar{x}\}$	IE	3	$\{\bar{x} \vee \bar{y} \vee z\}$
IS_{01}^m	m	$\{\mathrm{OR}^m, x \to y\}$	IE_1	3	$\{\bar{x} \vee \bar{y} \vee z, x\}$
IS_{01}	∞	$\{\mathrm{OR}^m \mid m \geq 2\} \cup \{x \to y\}$	IE_0	3	$\{\bar{x} \vee \bar{y} \vee z, \bar{x}\}$
IS_{00}^m	m	$\{\mathrm{OR}^m, x, \bar{x}, x \to y\}$	IE_2	3	$\{\bar{x} \vee \bar{y} \vee z, x, \bar{x}\}$
IS_{00}	∞	$\{\mathrm{OR}^m \mid m \geq 2\} \cup \{x, \bar{x}, x \to y\}$	IN	3	$\{\mathrm{DUP}^3\}$
IS_{12}^m	m	$\{\mathrm{NAND}^m, x, \bar{x}\}$	IN_2	3	$\{\mathrm{DUP}^3, x \oplus y\}, \{\mathrm{NAE}^3\}$
IS_{12}	∞	$\{\mathrm{NAND}^m \mid m \geq 2\} \cup \{x, \bar{x}\}$	II	3	$\{\mathrm{EVEN}^4, x \to y\}$
IS_{11}^m	m	$\{\mathrm{NAND}^m, x \to y\}$	II_0	3	$\{\mathrm{EVEN}^4, x \to y, \bar{x}\}, \{\mathrm{DUP}^3, x \to y\}$
IS_{11}	∞	$\{\mathrm{NAND}^m \mid m \geq 2\} \cup \{x \to y\}$	II_1	3	$\{\mathrm{EVEN}^4, x \to y, x\}, \{x \vee (x \oplus z)\}$
IS_{10}^m	m	$\{\mathrm{NAND}^m, x, \bar{x}, x \to y\}$	BR	3	$\{\mathrm{EVEN}^4, x \to y, x, \bar{x}\}$, $\{\text{1-IN-3}\}, \{x \vee (x \oplus z)\}$

reduction from $\mathrm{CSP}(\Gamma_2)$ to $\mathrm{CSP}(\Gamma_1)$. The same is not true for MinCSP: the approximation ratio can change in the reduction. However, it has been observed that this change of the approximation ratio is at most a constant (depending on Γ_1 and Γ_2) [33, 24, 25]; we show the same here in the context of parameterized reductions.

▶ **Lemma 10.** *Let Γ be a constraint language, and R be a relation that is pp-definable over Γ without equality. Then there is an A-reduction from* MinCSP$(\Gamma \cup \{R\})$ *to* MinCSP(Γ).

By repeated applications of Lemma 10, the following corollary establishes that we need to provide approximation algorithms only for a few MinCSPs, and these algorithms can be used for other MinCSPs associated with the same co-clone.

▶ **Corollary 11.** *Let C be a co-clone and B be a base for C. If the equality relation can be $\exists\wedge$-defined from B, then for any finite $\Gamma \subseteq C$, there is an A-reduction from* MinCSP(Γ) *to* MinCSP(B).

For hardness results, we wish to argue that if a co-clone C is hard, then any constraint language Γ generating the co-clone is hard. However, there are two technical issues. First, co-clones are infinite and our constraint languages are finite. Therefore, we formulate this requirement instead by saying that a finite base B of the co-clone C is hard. Second, pp-definitions require equality relations, which may not be expressible by Γ. However, as the following theorem shows, this is an issue only if B contains relations where the coordinates are always equal (which will not be the case in our proofs). A k-ary relation R is *irredundant* if for every two different coordinates $1 \leq i < j \leq k$, R contains a tuple $(a_1, \ldots, a_k)$ with $a_i \neq a_j$. A set of relations S is *irredundant* if any relation in S is irredundant.

▶ **Theorem 12** ([30, 4]). *If $S \subseteq \langle \Gamma \rangle$ and S is irredundant, then S is $\exists\wedge$-definable from Γ.*

Thus, considering an irredundant base B of co-clone C, we can formulate the following result.

▶ **Corollary 13.** *Let B be an irredundant base for some co-clone C. If Γ is a finite constraint language with $C \subseteq \langle \Gamma \rangle$, then there is an A-reduction from* MinCSP(B) *to* MinCSP(Γ).

By the following lemma, if the constraint language is self-dual, then we can assume that it also contains the constant relations.

▶ **Lemma 14.** *Let Γ be a self-dual constraint language. Assume that $x \oplus y \in \Gamma$. Then there is a cost-preserving reduction from* MinCSP($\Gamma \cup \{x, \bar{x}\}$) *to* MinCSP($\Gamma$).

The following theorem states our trichotomy classification in terms of co-clones.

▶ **Theorem 15.** *Let Γ be a finite set of Boolean relations.*

1. *If $\langle \Gamma \rangle \subseteq C$ (equivalently, if $\Gamma \subseteq C$), with $C \in \{\mathrm{II}_0, \mathrm{II}_1, \mathrm{IS}_{00}, \mathrm{IS}_{10}, \mathrm{ID}_2\}$, then* MinCSP($\Gamma$) *has a constant-factor* FPA *algorithm. (Note in these cases Γ is* 0*-valid,* 1*-valid, IHS-B+, IHS-B–, or bijunctive, respectively.)*
2. *If $\langle \Gamma \rangle \in \{\mathrm{IL}_2, \mathrm{IL}_3\}$, then* MinCSP($\Gamma$) *is equivalent to* Nearest Codeword *and to* Odd Set *under A-reductions (Note that these constraint languages are affine.)*
3. *If $C \subseteq \langle \Gamma \rangle$, where $C \in \{\mathrm{IE}_2, \mathrm{IV}_2, \mathrm{IN}_2\}$, then* MinCSP($\Gamma$) *does not have a constant-factor* FPA *algorithm unless* FPT = W[P]. *(Note that in these cases Γ can $\exists\wedge$-define either arbitrary Horn relations, or arbitrary dual Horn relations, or the relation* $\mathrm{NAE}^3 = \{0,1\}^3 \setminus \{(0,0,0),(1,1,1)\}$*.)*

Looking at the co-clone lattice, it is easy to see that Theorem 15 covers all cases. It is also easy to check that Theorem 1 formulated in the introduction follows from Theorem 15. Theorem 15 is proved the following way. Statement 1 is proved in Section 4 (Lemma 16, and Corollaries 18 and 21). Statement 2 is proved in Section 5 (Theorem 23). Statement 3 is proved in Sections 6 (Corollary 27 and Lemma 28).

4 CSPs with FPA algorithms

We prove the first statement of Theorem 15 by going through co-clones one by one. As every relation of a 0-valid MinCSP is always satisfied by the all 0 assignment, and every relation of a 1-valid MinCSP is always satisfied by the all 1 assignment, we have a trivial algorithm for these problems.

▶ **Lemma 16.** *If $\langle \Gamma \rangle \subseteq \mathrm{II}_0$ or $\langle \Gamma \rangle \subseteq \mathrm{II}_1$, then* MinCSP($\Gamma$) *is polynomial-time solvable.*

Consider now the co-clone ID_2. Almost 2-SAT is defined as MinCSP(Γ(2-SAT)), where $\Gamma(\text{2-SAT}) = \{x \vee y, x \vee \neg y, \neg x \vee \neg y\}$.

▶ **Theorem 17** ([45]). Almost 2-SAT *is fixed-parameter tractable.*

Since every bijunctive relation can be pp-defined by 2-SAT, the constant-factor FP-approximability of bijunctive languages easily follows from the FPT algorithm for Almost 2-SAT and from Corollary 11.

▶ **Corollary 18.** *If* $\langle\Gamma\rangle \subseteq \mathrm{ID}_2$*, then* MinCSP($\Gamma$) *has a constant-factor* FPA *algorithm.*

Proof. We check in Table 1 that $B = \{x \oplus y, x \to y\}$ is a base for the co-clone ID_2. Relations in B (and equality) can be $\exists\wedge$-defined over Γ(2-SAT), so the result follows from Corollary 11. ◀

We consider now IS_{00} and IS_{10}. We first note that if $\langle\Gamma\rangle$ is in IS_{00} or IS_{10}, then the language is k-IHS-B+ or k-IHS– for some $k \geq 2$.

▶ **Lemma 19.** *If* $\langle\Gamma\rangle \subseteq \mathrm{IS}_{00}$*, then there is an integer* $k \geq 2$ *such that* Γ *is* k*-IHS-B+. If* $\langle\Gamma\rangle \subseteq \mathrm{IS}_{10}$*, then there is an integer* $k \geq 2$ *such that* Γ *is* k*-IHS-B–.*

By Lemma 19, if $\langle\Gamma\rangle \subseteq \mathrm{IS}_{00}$, then Γ is generated by the relations $\neg x, x \to y, x_1 \vee \cdots \vee x_k$ for some $k \geq 2$. The MinCSP problem for this set of relations is known to admit a constant-factor approximation.

▶ **Theorem 20** ([19], Lemma 7.29). MinCSP($\neg x, x \to y, x_1 \vee \cdots \vee x_k$) *has a* $(k+1)$*-factor approximation algorithm (and hence has a constant-factor FPA algorithm).*

Now Theorem 20 and Corollary 11 imply that there is a constant-factor FPA algorithm for MinCSP(Γ) whenever $\langle\Gamma\rangle$ is in the co-clone IS_{00} or IS_{10} (note that equality can be $\exists\wedge$-defined using $x \to y$). In fact, the resulting algorithm is a polynomial-time approximation algorithm: Theorem 20 gives a polynomial-time algorithm and this is preserved by Corollary 11.

▶ **Corollary 21.** *If* $\langle\Gamma\rangle \subseteq \mathrm{IS}_{00}$ *or* $\langle\Gamma\rangle \subseteq \mathrm{IS}_{10}$*, then* MinCSP($\Gamma$) *has a constant-factor* FPA *algorithm.*

Note that Theorem 7.25 in [19] gives a complete classification of Boolean MinCSPs with respect to constant-factor approximability. As mentioned, these MinCSPs also admit a constant-factor approximation algorithm. The reason we need Corollary 21 is to have the characterization in terms of the co-clone lattice.

5 CSPs equivalent to Odd Set

In this section we show the equivalence of several problems under A-reductions. We identify CSPs that are equivalent to the following well-known combinatorial problems. In the Nearest Codeword (NC) problem, the input is an $m \times n$ matrix A, and an m-dimensional vector b. The output is an n dimensional vector x that minimizes the Hamming distance between Ax and b. In the Odd Set problem, the input is a set-system $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ over universe U. The output is a subset $T \subseteq U$ of minimum size such that every set of $\mathcal{S}$ is hit an odd number of times by T, that is, $\forall i \in [m]$, $|S_i \cap T|$ is odd.

Even/Odd Set is the same problem as Odd Set, except that we can specify whether a set should be hit an even or odd number of times (the objective is the same as in Odd Set: find a subset of minimum size satisfying the requirements). We show that there is a parameter preserving reduction from Even/Odd Set to Odd Set.

▶ **Lemma 22.** *There is a cost-preserving reduction from* Even/Odd Set *to* Odd Set.

We define the relations $\mathrm{EVEN}^m = \{(a_1, \ldots, a_m) \in \{0,1\}^m : \sum_{i=1}^m a_i \text{ is even}\}$, $\mathrm{ODD}^m = \{(a_1, \ldots, a_m) \in \{0,1\}^m : \sum_{i=1}^m a_i \text{ is odd}\}$, and the languages $B_2 = \{\mathrm{EVEN}^4, x, \bar{x}\}$, $B_3 = \{\mathrm{EVEN}^4, x \oplus y\}$. Note that B_2 and B_3 are bases for the co-clones IL_2 and IL_3, respectively.

▶ **Theorem 23.** [4] *The following problems are equivalent under cost-preserving reductions:* (1) Nearest Codeword, (2) Odd Set, (3) MinCSP(B_2), *and* (4) MinCSP(B_3).

6 Hard CSPs: Horn ($\mathbf{IV_2}$), dual-Horn ($\mathbf{IE_2}$) and $\mathbf{IN_2}$

In this section, we establish statement 3 of Theorem 15 by proving the inapproximability of MinCSP(Γ) if Γ generates one of the co-clones IE_2, IV_2, or IN_2. The inapproximability proof uses previous results on the inapproximability of circuit satisfiability problems.

A *Boolean circuit* is a directed acyclic graph, where each node with in-degree at least 2 is labeled as either an AND node or as an OR node, each node of in-degree 1 is labeled as a negation node, and each node of in-degree 0 is an input node. Furthermore, there is a node with out-degree 0 that is the output node. Given an assignment φ from the input nodes of circuit C to $\{0, 1\}$, we say that assignment φ 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. A circuit is *monotone* if it contains no negation gates. The problem Monotone Circuit Satisfiability (MCS) takes as input a monotone circuit C and an integer k, and the task is to decide if there is a satisfying assignment of weight at most k. The following theorem is a restatement of a result of Marx [40]. We use this to show that Horn-CSPs are hard.

▶ **Theorem 24** ([40]). Monotone Circuit Satisfiability *does not have an* FPA *algorithm, unless* FPT = W[P].

▶ **Corollary 25.** Monotone Circuit Satisfiability*, where circuits are restricted to have gates of in-degree at most* 2*, does not have an FPA algorithm, unless* FPT = W[P].

We use Corollary 25 to establish the inapproximability of Horn-SAT and dual-Horn-SAT, assuming that FPT $\neq$ W[P]. Using the co-clone lattice, this will show hardness of approximability of MinCSP(Γ) if $\langle\Gamma\rangle \in \{\mathrm{IV}_2, \mathrm{IE}_2\}$.

▶ **Lemma 26.** *If there is an* FPA *algorithm for* MinCSP($\{x \vee y \vee \bar{z}, x, \bar{x}\}$) *or* MinCSP($\{\bar{x} \vee \bar{y} \vee z, x, \bar{x}\}$) *with constant approximation ratio, then* FPT = W[P].

Proof. We prove that there is a parameter preserving polynomial-time reduction from Monotone Circuit Satisfiability to MinCSP*($\{x \vee y \vee \bar{z}, x, \bar{x}\}$). This is sufficient by Corollary 25. Let C be the MCS instance. We produce an instance I of MinCSP* as follows. We think of inputs of C as gates, and we refer to these as "input gates". This will simplify the discussion. For each gate of C, we introduce a new variable into I, and we let f denote the natural bijection from the gates and inputs of C to the variables of the instance I.

We add constraints to simulate each AND gate of C as follows. Observe first that the implication relation $x \to y$ can be expressed as $y \vee y \vee \bar{x}$. For each AND gate $G_\wedge$ such that G_1 and G_2 are the gates feeding into $G_\wedge$ (note that G_1 and G_2 are allowed to be input gates), we add two constraints to I as follows. Let $y = f(G_\wedge)$, $x_1 = f(G_1)$, and $x_2 = f(G_2)$. We place the constraints $y \to x_1, y \to x_2$ into I. We observe that the only way variable y could take on value 1 is if both x_1 and x_2 are assigned 1. (In this case, note that y could also be assigned 0 but that will be easy to fix.)

Similarly, we add constraints to simulate each OR gate of C as follows. For each OR gate $G_\vee$ such that G_1 and G_2 are the gates feeding into $G_\vee$, we add a constraint to I, we add

[4] Note that Lemma 1 in [22] can be adapted to obtain the reduction from Odd Set to MinCSP(B_2).

the constraint $x_1 \vee x_2 \vee \bar{y}$ to I, where $y = f(G_\vee)$, $x_1 = f(G_1)$, and $x_2 = f(G_2)$. Note that if both x_1 and x_2 are 0, than y is forced to have value 0. (Otherwise y can take on either value 0 or 1, but again, this difference between an OR gate and our gadget will be easy to handle.)

In addition, we add a constraint $x_o = 1$, where x_o is the variable such that $x_o = f(G)$, where G is the output gate. We define all constraints that appeared until now to be undeletable, so that they cannot be removed in solution of the MinCSP* instance. To finish the construction, for each variable x such that $x = f(G)$ where G is an input gate, we add a constraint $x = 0$ to I. We call these constraints *input constraints*. Note that only input constraints can be removed.

If there is a satisfying assignment φ_C of C (from gates of C to $\{0, 1\}$) of weight k, then we remove the input constraints $x = 0$ of I such that $\varphi_C(G) = 1$, where $f(G) = x$. Clearly, the map $\varphi_C \circ f^{-1}$ is a satisfying assignment for I, where we needed k deletions.

For the other direction, assume that we have a satisfying assignment φ_I for I after removing some k input constraints (note that if any other constraints are removed, we can simply ignore those deletions). We repeatedly change φ_I as long as either of the following conditions apply. If x_1, x_2 and y are such that $f^{-1}(x_1)$ and $f^{-1}(x_2)$ are gates feeding into gate $f^{-1}(y)$ where $f^{-1}(y)$ is an AND gate, and $\varphi_I(x_1) = 1, \varphi_I(x_2) = 1, \varphi_I(y) = 0$, then we change $\varphi_I(y)$ to 1. Similarly, if $f^{-1}(y)$ is an OR gate, $1 \in \{\varphi_I(x_1), \varphi_I(x_2)\}, \varphi_I(y) = 0$, then we change $\varphi_I(y)$ to 1. It follows form the definition of the constraints we introduced for AND and OR gates that once we finished modifying φ_I, the resulting assignment φ_I' is still a satisfying assignment. Now it follows that $\varphi_I' \circ f$ is a weight k satisfying assignment for C.

To show the inapproximability of MinCSP$(\{\bar{x} \vee \bar{y} \vee z, x, \bar{x}\})$, we note that there is a parameter preserving bijection between instances of MinCSP$(\{\bar{x} \vee \bar{y} \vee z, x, \bar{x}\})$ and MinCSP$(\{x \vee y \vee \bar{z}, x, \bar{x}\})$: given an instance I of either problem, we obtain an equivalent instance of the other problem by replacing every literal ℓ with $\neg\ell$. Satisfying assignments are converted by replacing 0-s with 1-s and vice versa. ◀

As $\{x \vee y \vee \bar{z}, x, \bar{x}\}$ (resp., $\{\bar{x} \vee \bar{y} \vee z, x, \bar{x}\}$) is an irredundant base of IV_2 (resp., IE_2), Corollary 13 implies hardness if $\langle\Gamma\rangle$ contains IV_2 or IE_2.

▶ **Corollary 27.** *If Γ is a (finite) constraint language with $\mathrm{IV}_2 \subseteq \langle\Gamma\rangle$ or $\mathrm{IE}_2 \subseteq \langle\Gamma\rangle$, then* MinCSP$(\Gamma)$ *is not FP-approximable, unless* FPT = W[P].

▶ **Lemma 28.** *If Γ is a (finite) constraint language with $\mathrm{IN}_2 \subseteq \langle\Gamma\rangle$ then* MinCSP(Γ) *is not FP-approximable, unless* P = NP.

7 Odd Set is probably hard

We provide evidence that problems equivalent to NC and Odd Set (in particular, problems in Theorem 15(2)) are hard, i.e., they are unlikely to have a constant-factor FPA algorithm.

In the k-Densest Subgraph problem, we are given a graph $G = (V, E)$ and an integer k; the task is to find a set S of k vertices that maximizes the number of edges in the induced subgraph $G[S]$. Note that an exact algorithm for k-Densest Subgraph would imply an exact algorithm for Clique. Due to its similarity to Clique, it is reasonable to assume that k-Densest Subgraph is even hard to approximate. We formulate the following specific hardness assumption.

▶ **Assumption 29.** *There is an $\varepsilon > 0$ such that for any function f, one cannot approximate k-*Densest Subgraph *within ratio $1 - \varepsilon$ in time $f(k) \cdot n^{O(1)}$.*

It will be more convenient to work with a slightly different version of k-Densest Subgraph. In the Multicolored k-Densest Subgraph problem, we are given a graph $G = (V, E)$ whose vertex-set V is partitioned into k classes $C_1, \ldots, C_k$, and the goal is to find a set $S = \{v_1, \ldots, v_k\}$ of k vertices satisfying $v_i \in C_i$ for each $i \in [k]$, and maximizing the number of edges in the induced subgraph $G[S]$. We argue in the arxiv version that Assumption 29 implies Assumption 30 [6].

▶ **Assumption 30.** *There is an $\varepsilon > 0$ such that for any function f, one cannot approximate* Multicolored k-Densest Subgraph *within ratio $1 - \varepsilon$ in time $f(k) \cdot n^{O(1)}$.*

Odd Set has the so-called *self-improvement* property. Informally, a polynomial time (resp. fixed-parameter time) approximation within some ratio r can be turned into a polynomial time (resp. fixed-parameter time) approximation within some ratio close to $\sqrt{r}$.

▶ **Lemma 31.** *If there is an r-approximation for* Odd Set *running in time $f(n, m, k)$ where n is the size of the universe, m the number of sets, and k the size of an optimal solution, then for any $\varepsilon > 0$, there is a $(1+\varepsilon)\sqrt{r}$-approximation running in time* $\max(f(1+n+n^2, 1+m+nm, 1+k+k^2), O(n^{1+\frac{1}{\varepsilon}} m))$.

Proof. The following reduction is inspired by the one showing the self-improvement property of NC [2]. Let $\mathcal{S} = \{S_1, \ldots, S_m\}$ be any instance over universe $U = \{x_1, \ldots, x_n\}$. Let $\varepsilon > 0$ be any real positive value and k be the size of an optimal solution. We can assume that $k \geqslant \frac{1}{\varepsilon}$ since one can find an optimal solution by exhaustive search in time $O(n^{1+\frac{1}{\varepsilon}} m)$. We build the set-system $\mathcal{S}' = \mathcal{S} \cup \bigcup_{i \in [n], j \in [m]} S^i_j \cup \{\{e\}\}$ over universe $U' = U \cup \bigcup_{i,h \in [n]} \{x^i_h\} \cup \{e\}$ such that $S^i_j = \{e, x_i\} \cup \{x^i_h \mid x_h \in S_j\}$. Note that the size of the new instance is squared. We show that there is a solution of size at most k to instance $\mathcal{S}$ if and only if there is a solution of size at most $1 + k + k^2$ to instance $\mathcal{S}'$.

If T is a solution to $\mathcal{S}$, then $T' = \{e\} \cup T \cup \{x^i_h \mid x_i, x_h \in T\}$ is a solution to $\mathcal{S}'$. Indeed, sets in $\mathcal{S} \cup \{\{e\}\}$ are obviously hit an odd number of times. And, for any $i \in [n]$ and $j \in [m]$, set S^i_j is hit exactly once (by e) if $x_i \notin T$, and is hit by e, x_i, plus as many elements as S_j is hit by T; so again an odd number of times. Finally, $|T'| = 1 + |T| + |T|^2$.

Conversely, any solution to $\mathcal{S}'$ should contain element e (to hit $\{e\}$), and should intersect U in a subset T hitting an odd number of times each set S_i ($\forall i \in [m]$). Then, for each $x_i \in T$, each set S^i_j with $j \in [m]$ is hit exactly twice by e and x_i. Thus, one has to select a subset of $\{x^i_1, \ldots, x^i_n\}$ to hit each set of the family $\{S^i_1, \ldots, S^i_m\}$ an odd number of times. Again, this needs as many elements as a solution to $\mathcal{S}$ needs. So, if there is a solution to $\mathcal{S}'$ of size at most $1 + k + k^2$, then there is a solution to $\mathcal{S}$ of size at most k. In fact, we will only use the weaker property that if there is a solution to $\mathcal{S}'$ of size at most k, then there is a solution to $\mathcal{S}$ of size at most $\sqrt{k}$.

Now, assuming there is an r-approximation for Odd Set running in time $f(n, m, k)$, we run that algorithm on the instance $\mathcal{S}'$ produced from $\mathcal{S}$. This takes time $f(1 + n + n^2, 1 + m + nm, 1 + k + k^2)$ and produces a solution of size $r(1 + k + k^2)$. From that solution, we can extract a solution T to $\mathcal{S}$ by taking its intersection with U. And T has size smaller than $\sqrt{r(1 + k + k^2)} \leqslant \sqrt{r}(k + 1) = (1 + \frac{1}{k})\sqrt{r}k \leqslant (1 + \varepsilon)\sqrt{r}k$. ◀

Repeated application of the self-improvement in Lemma 31 shows that any constant-ratio approximation implies the existence of $(1 + \varepsilon)$-approximation for arbitrary small $\varepsilon > 0$.

▶ **Corollary 32.** *If* Odd Set *admits an FPA algorithm with some ratio r, then, for any $\varepsilon > 0$, it also admits an FPA algorithm with ratio $1 + \varepsilon$.*

Now we show that an approximation for Odd Set with ratio $1 + \frac{\varepsilon}{3}$ implies the existence of a $(1 - \epsilon)$-approximation for k-Densest Subgraph. In light of Corollary 32, this means that any constant-factor approximation for Odd Set would violate Assumption 29.

▶ **Theorem 33.** *For any ratio r,* Odd Set *does not have an FPA algorithm with ratio r, unless Assumption 29 fails.*

Proof. Let $\varepsilon > 0$ be such that k-Densest Subgraph and therefore Multicolored k-Densest Subgraph do not admit a fixed-parameter $(1 - \varepsilon)$-approximation. We show that an FPA algorithm with ratio $1 + \frac{\varepsilon}{3}$ for Odd Set would contradict Assumption 29, and we conclude with Corollary 32. Let $G = (V = C_1 \uplus \ldots \uplus C_k, E)$ be an instance of Multicolored k-Densest Subgraph, and let X be an optimal solution inducing m edges. For any $\{i, j\} \in \binom{[k]}{2}$, we let $E_{\{i,j\}}$ be the set of edges between C_i and C_j.

We build $2^{\binom{k}{2}}$ instances of Odd Set: one for each subset of $\binom{[k]}{2}$. One such subset is $\mathcal{P} := \{\{i, j\} \mid E_{\{i,j\}} \cap E(X) \neq \emptyset\}$. In words, $\mathcal{P}$ is a correct guess of which $E_{\{i,j\}}$ are inhabited by the edges induced by the optimal solution X. Let $\mathcal{V}$ be the subset of indices $i \in [k]$ such that i appears in at least one pair of $\mathcal{P}$, and let $k' = |\mathcal{V}|$. Informally, $\mathcal{V}$ corresponds to the color classes of the vertices which are not isolated in the subgraph induced by X.

The universe U consists of an element x_v per vertex v of C_i such that $i \in \mathcal{V}$ and an element x_e per edge e in $E_{\{i,j\}}$ such that $\{i, j\} \in \mathcal{P}$. For any vertex $u \in C_i$ and any $j \in [k]$ such that $\{i, j\} \in \mathcal{P}$, we set $S_{u,j} = \{x_v \mid v \in C_i \text{ and } v \neq u\} \cup \{x_{vw} \mid vw \in E_{\{i,j\}} \text{ and } u \in \{v, w\}\}$, and for each $i \in \mathcal{V}$, $S_i = \{x_v \mid v \in C_i\}$. The set-system is $\mathcal{I} = (U, \mathcal{S} = \bigcup_{u \in C_i, \{i,j\} \in \mathcal{P}} S_{u,j} \uplus \bigcup_{i \in \mathcal{V}} S_i)$.

First, we show that the instance of Odd Set built for subset $\mathcal{P}$ admits a solution of size $k' + m$. Let $X' = \{a_1, \ldots, a_{k'}\} \subseteq X$ be the k' vertices which are not isolated in $G[X]$. We claim that $Z = \{x_{a_i}\}_{i=1}^{k'} \cup \{x_e\}_{e \in E(X')}$ is an odd set of $\mathcal{I}$. Each S_i with $i \in \mathcal{V}$ is hit by exactly one element of Z since no two a_p's can come from the same color class. Each $S_{u,j}$ with $u \in C_i$, $\{i, j\} \in \mathcal{P}$, and $u \notin X'$ is hit exactly once by x_{a_p} where the color class of a_p is C_i. Each $S_{u,j}$ with $u = a_p \in C_i \cap X'$, $\{i, j\} \in \mathcal{P}$ is hit exactly once by $x_{a_p a_q}$ where the color class of a_q is C_j. Finally, Z has the desired size $|X'| + |E(X')| = k' + |E(X)| = k' + m$.

Since we have established that $\mathcal{I}$ has an odd set of size $k' + m$, our supposed $1 + \frac{\varepsilon}{3}$-approximation would return a solution Z of at most $(1 + \frac{\varepsilon}{3})(k' + m)$ elements. We now show how to obtain a good approximation for Multicolored k-Densest Subgraph from such a solution to Odd Set. By construction, for each $i \in \mathcal{V}$, the set S_i should be hit an odd number of times, that is $|Z \cup S_i|$ is odd. In particular, $Z \cup S_i$ is non-empty. So, we can build a set $\{x_{u_i} | i \in \mathcal{V}\}$ where x_{u_i} is an arbitrary element of $Z \cup S_i$.

Let $\mathcal{E} = \{S_{u_i,j} \mid \{i, j\} \in \mathcal{P}\}$. Each of the $2|\mathcal{P}|$ sets of $\mathcal{E}$ (note that if, say, $\{1, 2\} \in \mathcal{P}$, then both $S_{u_1,2}$ and $S_{u_2,1}$ become a member of $\mathcal{E}$) are hit an even number of times by $Z \cap \bigcup_{i \in \mathcal{V}} S_i$. Indeed, $|(Z \cap \bigcup_{i \in \mathcal{V}} S_i) \cap S_{u_i,j}| = |Z \cap S_i \setminus \{x_{u_i}\}| = |Z \cap S_i| - 1$ which is even. We observe that each $S_{u_i,j} \in \mathcal{E}$ intersects with only one other set of $\mathcal{E}$, namely, $S_{u_j,i}$. So, we need at least $|\mathcal{P}|$ elements to hit the sets in $\mathcal{E}$. If there is an edge between u_i and u_j, both $S_{u_i,j}$ and $S_{u_j,i}$ can be hit at the same time by including element $x_{u_i u_j}$ into the solution. Otherwise $S_{u_i,j}$ and $S_{u_j,i}$ are disjoint and at least two elements are necessary to hit them. As there are at least $\frac{k'}{2}$ edges on k' non isolated vertices, we have $y \geq \frac{k'}{2}$. The set $Z \setminus (S_1 \cup \cdots \cup S_k)$ contains at most $|Z| - k \leq (1 + \frac{\varepsilon}{3})m + \frac{\varepsilon}{3}k' \leq (1 + \frac{\varepsilon}{3})m + \frac{2\varepsilon}{3}m = (1 + \varepsilon)m$ elements and these elements hit every set in $\mathcal{E}$. Thus, it can be true only for at most εm of the m pairs in $\mathcal{P}$ that the two sets $S_{u_i,j}, S_{u_j,i} \in \mathcal{E}$ cannot be hit by a single element of Z. Equivalently, it is true for at least $(1 - \varepsilon)m$ of the m pairs in $\mathcal{P}$ that the two sets $S_{u_i,j}, S_{u_j,i} \in \mathcal{E}$ are hit by a single element of Z. As mentioned previously, that element can only be $x_{u_i u_j}$. The fact that such an element actually exists means that there is an edge between u_i and u_j. Therefore, $\{u_i\}_{i \in \mathcal{V}}$

induces at least $(1-\varepsilon)m$ edges. It follows that Z is an $(1-\varepsilon)$-approximate solution for the instance of Multicolored k-Densest Subgraph; a contradiction to Assumption 29. ◀

▶ **Assumption 34** (Linear PCP Conjecture). *There exist constants $0 < \alpha < 1$, $A, B > 0$, such that* Max 3-SAT *on n variables can be decided with completeness 1 and error α by a verifier using $\log n + A$ random bits and reading B bits of the proof.*

LPC is probably better thought of as an open problem rather than a conjecture. In previous work, LPC has almost always proved to be a necessary hypothesis in showing that a specific problem cannot admit an FPA algorithm [7]. If LPC turns out to be true, the consequence for approximation is that there is a linear reduction introducing a constant gap from 3-SAT to Max 3-SAT. Thus, if we combine this fact with the sparsification lemma of Impagliazzo et al. [31], we may observe the following result:

▶ **Lemma 35** (Lemma 2, [7]). *Under LPC and ETH, there are two constants $r < 1$ and $\delta > 0$ such that one cannot distinguish satisfiable instances of* Max 3-SAT *with m clauses from instances where at most rm clauses are satisfied in time $2^{\delta m}$.*

The previous result was in fact stated slightly more generally allowing a weaker form of LPC where the completeness is not 1 but $1-\varepsilon$. We re-stated the lemma this way since we will need perfect completeness. The state-of-the-art PCP concerning the inapproximability of Max 3-SAT only implies the following:

▶ **Theorem 36** ([42]). *Under ETH, one cannot distinguish satisfiable instances of* Max 3-SAT *from instances where at most $(\frac{7}{8}+o(1))m$ clauses are satisfied in time $2^{m^{1-o(1)}}$.*

Now, we are set for the following result:

▶ **Theorem 37.** *Under LPC and ETH, for any ratio r,* Odd Set *does not have an FPA algorithm with ratio r.*

Proof. Again, the idea is to assume an FPA algorithm with ratio $1+\varepsilon$ for Odd Set (with parameter k), and show that it would imply a too good approximation for Max 3-SAT in subexponential time, therefore contradicting Lemma 35, and then conclude with Lemma 31.

Let $\phi = \bigwedge_{1 \leqslant i \leqslant m} C_i$ be any instance of Max 3-SAT, where the C_is are 3-clauses over the set of n variables V. We partition the clauses arbitrarily into k sets $A_1, A_2, \dots, A_k$ of size roughly $\frac{m}{k}$. We denote by V_i the set of all the variables appearing in at least one clause of A_i; each V_i has size at most $\frac{3m}{k}$. Of course, while the A_i's are a partition of the clauses, the V_i's can intersect with each other. We build an instance $\mathcal{I} = (U, \mathcal{S})$ of Odd Set the following way. For each $i \in [k]$, set U_i contains one element $x(\mathcal{A}, i)$ per assignment $\mathcal{A}$ of V_i that satisfies all the clauses inside A_i. The universe U is $\bigcup_i U_i$. For each $i \neq j \in [k]$, for each variable $y \in V_i \cap V_j$, we set $S_{y,i,j} = \{x(\mathcal{A}, i) \mid y \text{ is set to true by } \mathcal{A}\} \cup \{x(\mathcal{A}, j) \mid y \text{ is set to false by } \mathcal{A}\}$. Observe that $S_{y,i,j}$ and $S_{y,j,i}$ are two different sets. Finally, $\mathcal{S} = \bigcup_{i \in [k]} \{U_i\} \cup \bigcup_{i \neq j \in [k], y \in V_i \cap V_j} S_{y,i,j}$.

If ϕ is satisfiable, we fix a (global) satisfying assignment $\mathcal{A}_g$. We claim that $S = \{x(\mathcal{A}, i) \mid \mathcal{A} \text{ agrees with } \mathcal{A}_g \text{ in the entire } V_i\}$ is a solution of size k to the Odd Set instance. Set S is of size k since for each $i \in [k]$ exactly one element $x(\mathcal{A}, i)$ can be such $\mathcal{A}$ agrees with $\mathcal{A}_g$. This also shows that each set U_i is hit exactly once by S. Finally, for each $i \neq j \in [k]$ and for each variable $y \in V_i \cap V_j$, sets $\{x(\mathcal{A}, i) \mid y \text{ is set to true by } \mathcal{A}\}$ and $\{x(\mathcal{A}, j) \mid y \text{ is set to false by } \mathcal{A}\}$ can be hit at most once. Besides, $\{x(\mathcal{A}, i) \mid y \text{ is set to true by } \mathcal{A}\}$ is hit exactly once by S if and only if $\{x(\mathcal{A}, j) \mid y \text{ is set to false by } \mathcal{A}\}$ is not hit by S, since the partial assignments mapped to the elements in S necessarily agree. Therefore, $S_{y,i,j}$ is hit exactly once by S, and S is a solution.

Now, we assume that Odd Set admits an FPA algorithm with ratio $1+\varepsilon$ for a small ε that we will fix later. If the solution S returned by this algorithm on instance $\mathcal{I}$ is of size greater than $(1+\varepsilon)k$, then we know that an optimal odd set has more than k elements, so we know that ϕ is not satisfiable. So, we can assume that $|S| \leqslant (1+\varepsilon)k$. Each U_i has to be hit at least once and U_is are pairwise disjoint, so we can arbitrarily decompose S into $P \uplus R$, where P is of size k and hits each U_i exactly once, and therefore $|R| \leqslant \varepsilon k$. Thus, at least $(1-\varepsilon)k$ sets U_is are hit exactly once by S. We denote by $\mathcal{U}$ the set of such sets U_i. Let $\mathcal{A}_g$ be the assignment of V agreeing on each assignment $\mathcal{A}$ of V_i such that $x(\mathcal{A}, i)$ is the only element hitting $U_i \in \mathcal{U}$ (and setting the potential remaining variable arbitrarily). Assignment $\mathcal{A}_g$ is well defined since if $x(\mathcal{A}, i)$ is the only element hitting $U_i \in \mathcal{U}$ and $x(\mathcal{A}', j)$ is the only element hitting $U_j \in \mathcal{U}$, and assignments $\mathcal{A}$ and $\mathcal{A}'$ disagree on a variable y, then $S_{y,i,j}$ would be hit an even number of times (0 or 2). By construction, $\mathcal{A}_g$ satisfies all the clauses in the A_is such that $U_i \in \mathcal{U}$, that is at least $(1-\varepsilon)k \times \frac{m}{k} = (1-\varepsilon)m$ clauses. Let r and δ be two constants satisfying Lemma 35. If we choose $\varepsilon = \frac{1-r}{2}$, this number of clauses exceed rm, so we would know that the instance is satisfiable.

Say, the running time of the FPA algorithm is $f(k)(|U| + |\mathcal{S}|)^c$ for some constant c. We may observe that $|U| \leqslant k2^{\frac{3m}{k}}$ and $|\mathcal{S}| \leqslant k + 2\binom{k}{2}n$. Thus, the running time is $g(k)n^c 2^{\frac{3mc}{k}}$. Setting $k = \frac{6c}{\delta}$, this running time would be better than $2^{\delta m}$, contradicting LPC or ETH. ◀

References

1 Eric Allender, Michael Bauland, Neil Immerman, Henning Schnoor, and Heribert Vollmer. The complexity of satisfiability problems: Refining Schaefer's theorem. *J. Comput. Syst. Sci.*, 75(4):245–254, 2009.

2 Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. Syst. Sci.*, 54(2):317–331, 1997. doi:10.1006/jcss.1997.1472.

3 Arnab Bhattacharyya and Yuichi Yoshida. An algebraic characterization of testable boolean CSPs. In *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I*, pages 123–134, 2013.

4 V. G. Bodnarchuk, L. A. Kaluzhnin, V. N. Kotov, and B. A. Romov. Galois theory for Post algebras. *Kibernetika*, 5(3):1–10, 1969.

5 Elmar Böhler, Steffen Reith, Henning Schnoor, and Heribert Vollmer. Bases for boolean co-clones. *Inf. Process. Lett.*, 96(2):59–66, 2005.

6 Édouard Bonnet, László Egri, and Dániel Marx. Fixed-parameter approximability of boolean minCSPs. *CoRR*, abs/1601.04935, 2016. URL: http://arxiv.org/abs/1601.04935.

7 Édouard Bonnet, Bruno Escoffier, Eun Jung Kim, and Vangelis Th. Paschos. On subexponential and fpt-time inapproximability. *Algorithmica*, 71(3):541–565, 2015. doi:10.1007/s00453-014-9889-1.

8 Andrei A. Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *J. ACM*, 53(1):66–120, 2006. doi:10.1145/1120582.1120584.

9 Andrei A. Bulatov. Complexity of conservative constraint satisfaction problems. *ACM Trans. Comput. Log.*, 12(4):24, 2011. doi:10.1145/1970398.1970400.

10 Andrei A. Bulatov. The complexity of the counting constraint satisfaction problem. *J. ACM*, 60(5):34, 2013. doi:10.1145/2528400.

11 Andrei A. Bulatov, Martin E. Dyer, Leslie Ann Goldberg, Markus Jalsenius, Mark Jerrum, and David Richerby. The complexity of weighted and unweighted #CSP. *J. Comput. Syst. Sci.*, 78(2):681–688, 2012. doi:10.1016/j.jcss.2011.12.002.

12 Andrei A. Bulatov and Dániel Marx. The complexity of global cardinality constraints. *Logical Methods in Computer Science*, 6(4), 2010. doi:10.2168/LMCS-6(4:4)2010.

13 Andrei A. Bulatov and Dániel Marx. Constraint satisfaction parameterized by solution size. *SIAM J. Comput.*, 43(2):573–616, 2014. doi:10.1137/120882160.

14 Liming Cai and Xiuzhen Huang. Fixed-parameter approximation: Conceptual framework and approximability results. *Algorithmica*, 57(2):398–412, 2010. doi:10.1007/s00453-008-9223-x.

15 Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D. Sivakumar. On the hardness of approximating multicut and sparsest-cut. *Computational Complexity*, 15(2):94–114, 2006. doi:10.1007/s00037-006-0210-9.

16 Yijia Chen, Martin Grohe, and Magdalena Grüber. On parameterized approximability. In *Parameterized and Exact Computation, Second International Workshop, IWPEC 2006, Zürich, Switzerland, September 13-15, 2006, Proceedings*, pages 109–120, 2006.

17 Yijia Chen, Martin Grohe, and Magdalena Grüber. On parameterized approximability. *Electronic Colloquium on Computational Complexity (ECCC)*, 14(106), 2007. URL: http://eccc.hpi-web.de/eccc-reports/2007/TR07-106/index.html.

18 Rajesh Hemant Chitnis, MohammadTaghi Hajiaghayi, and Guy Kortsarz. Fixed-parameter and approximation algorithms: A new look. In *Parameterized and Exact Computation - 8th International Symposium, IPEC 2013, Sophia Antipolis, France, September 4-6, 2013, Revised Selected Papers*, pages 110–122, 2013. doi:10.1007/978-3-319-03898-8_11.

19 Nadia Creignou, Sanjeev Khanna, and Madhu Sudan. *Complexity Classifications of Boolean Constraint Satisfaction Problems*. Society for Industrial and Applied Mathematics, 2001.

20 Nadia Creignou and Heribert Vollmer. Parameterized complexity of weighted satisfiability problems: Decision, enumeration, counting. *Fundam. Inform.*, 136(4):297–316, 2015.

21 Pierluigi Crescenzi and Alessandro Panconesi. Completeness in approximation classes. *Inf. Comput.*, 93(2):241–262, 1991. doi:10.1016/0890-5401(91)90025-W.

22 Robert Crowston, Gregory Gutin, Mark Jones, and Anders Yeo. Parameterized complexity of satisfying almost all linear equations over $\mathbb{F}_2$. *Theory Comput. Syst.*, 52(4):719–728, 2013. doi:10.1007/s00224-012-9415-2.

23 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

24 Víctor Dalmau and Andrei A. Krokhin. Robust satisfiability for csps: Hardness and algorithmic results. *TOCT*, 5(4):15, 2013. doi:10.1145/2540090.

25 Víctor Dalmau, Andrei A. Krokhin, and Rajsekar Manokaran. Towards a characterization of constant-factor approximable min csps. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 847–857, 2015. doi:10.1137/1.9781611973730.58.

26 Vladimir G. Deineko, Peter Jonsson, Mikael Klasson, and Andrei A. Krokhin. The approximability of MAXCSP with fixed-value constraints. *J. ACM*, 55(4), 2008. doi:10.1145/1391289.1391290.

27 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

28 Rodney G. Downey, Michael R. Fellows, Catherine McCartin, and Frances Rosamond. Parameterized approximation of dominating set problems. *Inform. Process. Lett.*, 109(1):68–70, 2008. doi:10.1016/j.ipl.2008.09.017.

29 J. Flum and M. Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, Berlin, 2006.

30 D. Geiger. Closed systems of functions and predicates. *Pacific Journal of Mathematics*, 27:95–100, 1968.

31 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

32 Peter Jonsson, Mikael Klasson, and Andrei A. Krokhin. The approximability of three-valued MAXCSP. *SIAM J. Comput.*, 35(6):1329–1349, 2006. doi:10.1137/S009753970444644X.

33 Sanjeev Khanna, Madhu Sudan, Luca Trevisan, and David P. Williamson. The approximability of constraint satisfaction problems. *SIAM J. Comput.*, 30(6):1863–1920, 2000. doi:10.1137/S0097539799349948.

34 Vladimir Kolmogorov and Stanislav Zivny. The complexity of conservative valued CSPs. *J. ACM*, 60(2):10, 2013. doi:10.1145/2450142.2450146.

35 Stefan Kratsch, Dániel Marx, and Magnus Wahlström. Parameterized complexity and kernelizability of MaxOnes and ExactOnes problems. In *Mathematical Foundations of Computer Science 2010, 35th International Symposium, MFCS 2010, Brno, Czech Republic, August 23-27, 2010. Proceedings*, pages 489–500, 2010. doi:10.1007/978-3-642-15155-2_43.

36 Stefan Kratsch and Magnus Wahlström. Preprocessing of MinOnes problems: A dichotomy. In *Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part I*, pages 653–665, 2010. doi:10.1007/978-3-642-14165-2_55.

37 Daniel Lokshtanov, N. S. Narayanaswamy, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. Faster parameterized algorithms using linear programming. *ACM Transactions on Algorithms*, 11(2):15:1–15:31, 2014. doi:10.1145/2566616.

38 Dániel Marx. Parameterized complexity of constraint satisfaction problems. *Computational Complexity*, 14(2):153–183, 2005. doi:10.1007/s00037-005-0195-9.

39 Dániel Marx. Parameterized complexity and approximation algorithms. *Comput. J.*, 51(1):60–78, 2008.

40 Dániel Marx. Completely inapproximable monotone and antimonotone parameterized problems. *J. Comput. Syst. Sci.*, 79(1):144–151, 2013.

41 Dániel Marx and Igor Razgon. Constant ratio fixed-parameter approximation of the edge multicut problem. *Inf. Process. Lett.*, 109(20):1161–1166, 2009. doi:10.1016/j.ipl.2009.07.016.

42 Dana Moshkovitz and Ran Raz. Two-query PCP with subconstant error. *J. ACM*, 57(5), 2010. doi:10.1145/1754399.1754402.

43 R. Pöschel and Kalužnin. *Funktionen- und Relationenalgebren*. Deutscher Verlag der Wissenschaften, Berlin, 1979.

44 Emil L. Post. *On the Two-Valued Iterative Systems of Mathematical Logic*. Princeton University Press, 1941.

45 Igor Razgon and Barry O'Sullivan. Almost 2-SAT is fixed-parameter tractable. *J. Comput. Syst. Sci.*, 75(8):435–450, 2009.

46 Thomas J. Schaefer. The complexity of satisfiability problems. In *Conference Record of the Tenth Annual ACM Symposium on Theory of Computing (San Diego, Calif., 1978)*, pages 216–226. ACM, New York, 1978.

47 Johan Thapper and Stanislav Zivny. The complexity of finite-valued CSPs. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 695–704, 2013. doi:10.1145/2488608.2488697.