

Fixed-parameter Approximability of Boolean MinCSPs*

Édouard Bonnet¹, László Egri², Bingkai Lin³, and Dániel Marx¹

- 1 Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI), Budapest, Hungary
bonnet.edouard@sztaki.mta.hu, dmarx@cs.bme.hu
- 2 Department of Mathematics and Computer Science, Indiana State University, Terre Haute, United States
laszlo.egri@mail.mcgill.ca
- 3 National Institute of Informatics
lin@nii.ac.jp

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 $\text{MINCSP}(\Gamma)$ the restriction of the problem where each constraint is from Γ . The polynomial-time solvability and the polynomial-time approximability of $\text{MINCSP}(\Gamma)$ were fully characterized by Khanna et al. [30]. 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. We show the following dichotomy: for each finite constraint language Γ ,

- either we exhibit a constant-factor FP-approximation for $\text{MINCSP}(\Gamma)$;
- or we prove that $\text{MINCSP}(\Gamma)$ has no constant-factor FP-approximation unless $\text{FPT} = \text{W}[1]$.

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.2016.246

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 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. [17].

* 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.



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 an easy or hard problem.¹ Numerous classification theorems of this form have been proved for various decision and optimization versions for Boolean and non-Boolean CSPs [43, 11, 8, 9, 7, 10, 6, 24, 29, 31, 44, 36]. In particular, for $\text{MINCSP}(\Gamma)$, which is the optimization problem asking for an assignment minimizing the number of unsatisfied constraints, Creignou et al. [17] obtained a classification of the polynomial-time approximability for every finite Boolean constraint language Γ . The goal of this paper is to characterize the approximability of Boolean $\text{MINCSP}(\Gamma)$ with respect to the more relaxed notion of fixed-parameter approximability.

Parameterized complexity [25, 27, 21] 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 search 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 [25] and by Cygan et al. [21]. In particular, there are fixed-parameter tractability results and characterization theorems for various CSPs [36, 11, 32, 33].

The notion of fixed-parameter tractability has been combined with the notion of approximability [14, 15, 26, 12, 16]. Following [14, 37], 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 $f_2(k) \cdot k$ (where the function $f_2(k) \cdot k$ is non-decreasing), 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 [43] 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

¹ We note that Dmitriy Zhuk has recently announced a proof of the dichotomy conjecture.

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 [42, 35], but has no polynomial-time approximation algorithm with constant approximation ratio, assuming the Unique Games Conjecture [13]. On the other hand, if Γ consists of the three constraints (x) , (\bar{x}) , and $(a \rightarrow b) \wedge (c \rightarrow d)$, then the problem is W[1]-hard [39], but belongs to the class IHS-B and hence admits a constant-factor approximation in polynomial time [30].²

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 these two classes are very likely to 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 $\text{MINCSP}(\Gamma)$ has a constant-factor FP-approximation.*
2. *Otherwise, $\text{MINCSP}(\Gamma)$ has no constant-factor FP-approximation, unless $\text{FPT} = \text{W}[1]$.*

Moreover, in the second case (when Γ is neither bijunctive nor IHS-B), if Γ is also *not* affine, we can show a stronger inapproximability result; namely, that $\text{MINCSP}(\Gamma)$ has no FP-approximation for any function of the optimum value, unless $\text{FPT} = \text{W}[P]$. Note that this result is stronger in two different ways: it rules out not only constant-factor but any ratio of approximation, and it relies on a weaker assumption.

Given a linear code over \mathbb{F}_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 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 \mathbb{F}_2 that minimizes the number of unsatisfied equations. Arora et al. [2] showed that, assuming $\text{NP} \not\subseteq \text{DTIME}(n^{\text{poly} \log n})$, it is not possible to approximate NEAREST CODEWORD within ratio $2^{\log^{1-\epsilon} n}$ for any $\epsilon > 0$ in polynomial time. In particular, this implies that a constant-factor polynomial-time approximation is unlikely. However, this does not give any evidence against constant-factor FP-approximation. Building on the work of Lin [34] proving hardness for BICLIQUE and related problems, we are able to show that even polylogarithmic FP-approximation is unlikely for ODD SET.

► **Theorem 2.** *ODD SET has no ratio $\log^{O(1)} k$ FP-approximation, unless $\text{FPT} = \text{W}[1]$.*

This theorem is the most technically involved part of the paper, as well as the most interesting contribution.

Post's lattice is a very useful tool for classifying the complexity of Boolean CSPs (see e.g., [1, 18, 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

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

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. [30] and subsequently exploited by Dalmau et al. [22, 23], 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, Section 6 shows the hardness result for ODD SET (Theorem 2), and Section 7 proves inapproximability results for the remaining boolean MINCSP's.

2 Preliminaries

Constraint Satisfaction Problems (CSPs). 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 $\{C_1, \dots, C_q\}$, 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 $\varphi : V \rightarrow \{0, 1\}$ such that for each constraint $\langle s, R \rangle \in \mathcal{C}$, with $s = \langle v_1, \dots, v_n \rangle$, the tuple $\langle \varphi(v_1), \dots, \varphi(v_n) \rangle$ belongs to R ?

Note that we can alternatively look at a constraint as a Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$, 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 $f(x, y) = x + y \pmod 2$, 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* (resp. *1-valid*), if each $R \in \Gamma$ contains a tuple in which all entries are 0 (resp. 1);
- *k-IHS-B+* (resp. *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 $x_1 \vee \dots \vee x_k$ (resp. x , $\neg x \vee y$, or $\neg x_1 \vee \dots \vee \neg x_k$); *IHS-B+* (resp. *IHS-B-*) stands for *k-IHS-B+* (resp. *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 $x_1 \oplus \dots \oplus x_n = c$, where $c \in \{0, 1\}$;
- *self-dual* if for each relation $R \in \Gamma$, $(a_1, \dots, a_n) \in R \Rightarrow (\neg a_1, \dots, \neg a_n) \in R$.

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.

Reductions. 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 3.** 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 4.** *There is a cost-preserving reduction from MINCSP^{*} to MINCSP.*

Proof. The function F on a feasible instance I of MINCSP^{*} is defined the following way. Let m be the number of constraints. We construct $F(I)$ by replacing each undeletable constraint with $m + 1$ copies. If I is a feasible instance of MINCSP^{*}, then I has a solution with at most m deletions, which gives a solution of $F(I)$ as well, showing that $OPT(F(I)) \leq OPT(I) \leq m$. Conversely, $OPT(F(I)) \leq m$ implies that an optimum solution of $F(I)$ uses only the deletable constraints of I , otherwise it would need to delete all $m + 1$ copies of an undeletable constraints. Thus $OPT(I) \leq OPT(F(I))$ and hence $OPT(I) = OPT(F(I))$ follows.

The function $G(I, S')$ on a feasible instance I of MINCSP^* and a feasible solution S' of $F(I)$ is defined the following way. If S' deletes only the deletable constraints of I , then $G(I, S') = S'$ is also a feasible solution of I with the same cost. Otherwise, if S' deletes at least one undeletable constraint, then it has cost at least $m + 1$, as it has to delete all $m + 1$ copies of the constraint. Now we define $G(I, S')$ to be the set of all (at most m) deletable constraints; by assumption, I is a feasible instance of MINCSP^* , hence $G(I, S')$ is a feasible solution of cost at most $m + 1$ \blacktriangleleft

The second type of reduction that we use is the standard notion of A-reductions [19], 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 5.** 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 6.** 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 $f(a_1, \dots, a_n) = a_k$ for $1 \leq k \leq n$) and is closed under arbitrary composition. All clones of Boolean functions were identified by Post [41], 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 7.** 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 $\varphi(x_1, \dots, x_k) \equiv \exists y_1, \dots, y_\ell \psi(x_1, \dots, x_k, y_1, \dots, y_\ell)$, 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 $(a_1, \dots, a_k) \in A^k$ $(a_1, \dots, a_k) \in R$ if and only if $\varphi(a_1, \dots, a_k)$ 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* (resp. $\exists\wedge$ -definable) from Γ if every relation in S is pp-definable (resp. $\exists\wedge$ -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 8** ([40], Theorem 3.1.3). *The lattices of Boolean clones and Boolean co-clones are anti-isomorphic.*

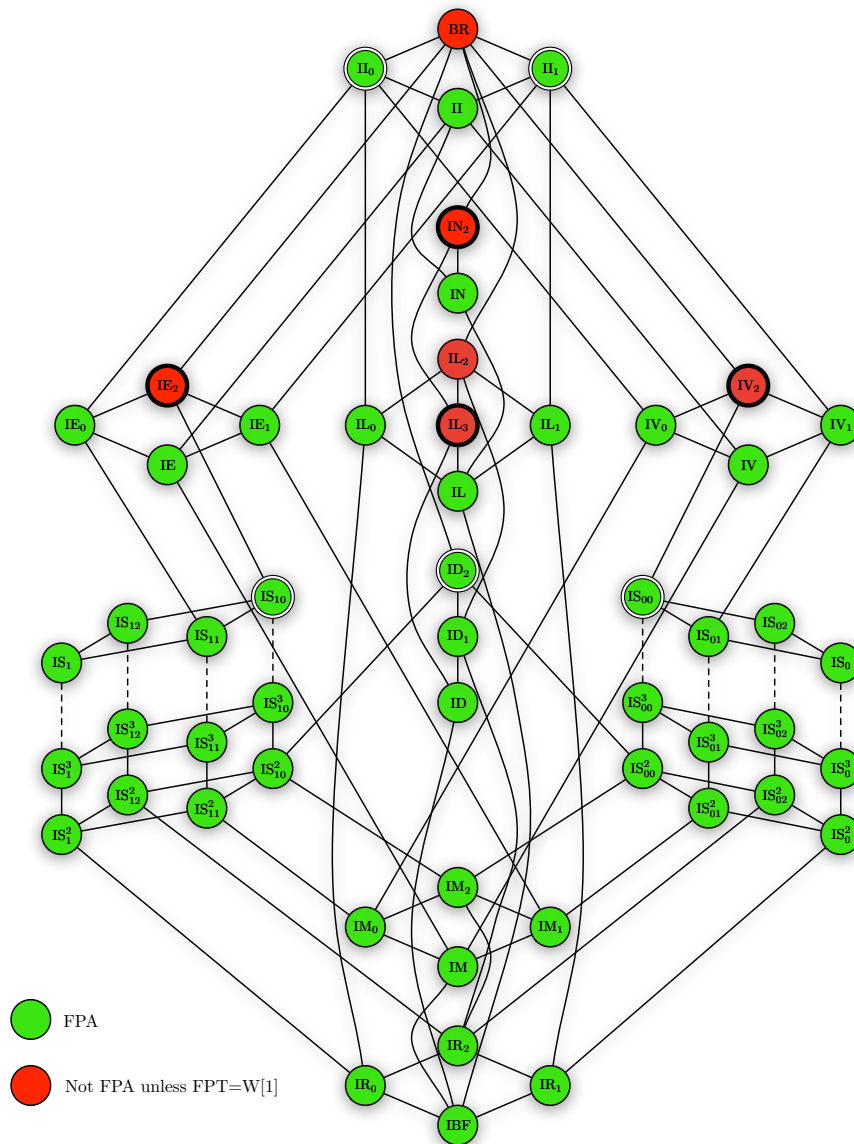


Figure 1 Classification of Boolean CSPs according to constant ratio fixed-parameter approximability.

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 1.³ 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 nodes⁴, where we follow the notation of Böhler et al [5].

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

³ We thank Heribert Vollmer and Yuichi Yoshida for giving us access to their Post’s lattice diagrams.

⁴ If the name of a clone is L_3 , for example, then the corresponding co-clone is $\text{Inv}(L_3)$ (Inv is defined, for example, in [5]), which is denoted by IL_3 .

Co-clone	Order	Base	Co-clone	Order	Base
IBF	0	$\{=\}, \{\emptyset\}$	IS ₁₀	∞	$\{\text{NAND}^m m \geq 2\} \cup \{x, \bar{x}, x \rightarrow y\}$
IR ₀	1	$\{\bar{x}\}$	ID	2	$\{x \oplus y\}$
IR ₁	1	$\{x\}$	ID ₁	2	$\{x \oplus y, x\}$, every $R \in \{(a_1, a_2, a_3), (b_1, b_2, b_3) \exists c \in \{1, 2\} \text{ such that } \sum_{i=1}^3 a_i = \sum_{i=1}^3 b_i = c\}$
IR ₂	1	$\{x, \bar{x}\}, \{x\bar{x}\}$	ID ₂	2	$\{x \oplus y, x \rightarrow y\}, \{x\bar{y}, \bar{x}yz\}$
IM	2	$\{x \rightarrow y\}$	IL	4	$\{\text{EVEN}^4\}$
IM ₁	2	$\{x \rightarrow y, x\}, \{x \wedge (y \rightarrow z)\}$	IL ₀	3	$\{\text{EVEN}^4, \bar{x}\}, \{\text{EVEN}^3\}$
IM ₀	2	$\{x \rightarrow y, \bar{x}\}, \{\bar{x} \wedge (y \rightarrow z)\}$	IL ₁	3	$\{\text{EVEN}^4, x\}, \{\text{ODD}^3\}$
IM ₂	2	$\{x \rightarrow y, x, \bar{x}\}, \{x \rightarrow y, \bar{x} \rightarrow \bar{y}\}, \{x\bar{y} \wedge (u \rightarrow v)\}$	IL ₂	3	$\{\text{EVEN}^4, x, \bar{x}\}$, every $\{\text{EVEN}^n, x\}$ where $n \geq 3$ is odd
IS ₀ ^m	m	$\{\text{OR}^m\}$	IL ₃	4	$\{\text{EVEN}^4, x \oplus y\}, \{\text{ODD}^4\}$
IS ₁ ^m	m	$\{\text{NAND}^m\}$	IV	3	$\{x \vee y \vee \bar{z}\}$
IS ₀	∞	$\{\text{OR}^m m \geq 2\}$	IV ₀	3	$\{x \vee y \vee \bar{z}, \bar{x}\}$
IS ₁	∞	$\{\text{NAND}^m m \geq 2\}$	IV ₁	3	$\{x \vee y \vee \bar{z}, x\}$
IS ₀₂ ^m	m	$\{\text{OR}^m, x, \bar{x}\}$	IV ₂	3	$\{x \vee y \vee \bar{z}, x, \bar{x}\}$
IS ₀₂	∞	$\{\text{OR}^m m \geq 2\} \cup \{x, \bar{x}\}$	IE	3	$\{\bar{x} \vee \bar{y} \vee z\}$
IS ₀₁ ^m	m	$\{\text{OR}^m, x \rightarrow y\}$	IE ₁	3	$\{\bar{x} \vee \bar{y} \vee z, x\}$
IS ₀₁	∞	$\{\text{OR}^m m \geq 2\} \cup \{x \rightarrow y\}$	IE ₀	3	$\{\bar{x} \vee \bar{y} \vee z, \bar{x}\}$
IS ₀₀ ^m	m	$\{\text{OR}^m, x, \bar{x}, x \rightarrow y\}$	IE ₂	3	$\{\bar{x} \vee \bar{y} \vee z, x, \bar{x}\}$
IS ₀₀	∞	$\{\text{OR}^m m \geq 2\} \cup \{x, \bar{x}, x \rightarrow y\}$	IN	3	$\{\text{DUP}^3\}$
IS ₁₂ ^m	m	$\{\text{NAND}^m, x, \bar{x}\}$	IN ₂	3	$\{\text{DUP}^3, x \oplus y\}, \{\text{NAE}^3\}$
IS ₁₂	∞	$\{\text{NAND}^m m \geq 2\} \cup \{x, \bar{x}\}$	II	3	$\{\text{EVEN}^4, x \rightarrow y\}$
IS ₁₁ ^m	m	$\{\text{NAND}^m, x \rightarrow y\}$	II ₀	3	$\{\text{EVEN}^4, x \rightarrow y, \bar{x}\}, \{\text{DUP}^3, x \rightarrow y\}$
IS ₁₁	∞	$\{\text{NAND}^m m \geq 2\} \cup \{x \rightarrow y\}$	II ₁	3	$\{\text{EVEN}^4, x \rightarrow y, x\}, \{x \vee (x \oplus z)\}$
IS ₁₀ ^m	m	$\{\text{NAND}^m, x, \bar{x}, x \rightarrow y\}$	BR	3	$\{\text{EVEN}^4, x \rightarrow y, x, \bar{x}\}, \{1\text{-IN-3}\}, \{x \vee (x \oplus z)\}$

■ **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.

in [5], and the reader can consult this paper for details. We reproduce this list in Table 1.⁵

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 reduction from $\text{CSP}(\Gamma_2)$ to $\text{CSP}(\Gamma_1)$. 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) [30, 22, 23]; we show the same here in the context of parameterized reductions.

► **Lemma 9.** *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 $\text{MINCSP}(\Gamma \cup \{R\})$ to $\text{MINCSP}(\Gamma)$.*

Proof. Let I be an instance of $\text{MINCSP}(\Gamma \cup \{R\})$. Let $\varphi(x_1, \dots, x_k)$ be a primitive positive formula defining R from Γ . Then φ is of the form $\exists y_1, \dots, y_\ell \psi(x_1, \dots, x_k, y_1, \dots, y_\ell)$, where ψ is the quantifier-free part of φ . The key and well-known (in similar contexts) observation is that ψ can be alternatively seen as an instance of $\text{MINCSP}(\Gamma)$. More precisely, we define the instance associated to ψ , I_ψ , as the instance that has variables $x_1, \dots, x_k, y_1, \dots, y_\ell$ and contains for every atomic formula $S(v_1, \dots, v_r)$ in ψ , the constraint $\langle (v_1, \dots, v_r), S \rangle$. It

⁵ 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 $\{\text{DUP}^3, x \oplus y\}$.

follows that for any assignment $s : x_1, \dots, x_k, y_1, \dots, y_\ell \rightarrow A$, s is a solution of I_ψ if and only if $\psi(s(x_1), \dots, s(x_k), s(y_1), \dots, s(y_\ell))$ holds.

We obtain an instance I' of $\text{MINCSP}(\Gamma)$ from I using the following replacement. For each constraint $C = \langle (u_1, \dots, u_q), R \rangle$ in I , we identify the quantifier-free part $\psi_C(u_1, \dots, u_q, y_{C,1}, \dots, y_{C,l})$ of the formula corresponding to C , and then replace C with the set of constraints of the instance I_{ψ_C} , where $y_{C,1}, \dots, y_{C,l}$ are newly introduced variables. We leave the rest of the constraints intact.

Any deletion set X_I for I is translated to a deletion set $X_{I'}$ of I' as follows. If $C = \langle (u_1, \dots, u_q), P \rangle \in X_I$ and $P \neq R$, then we place $\langle (u_1, \dots, u_q), P \rangle \in X_{I'}$. If $P = R$, then we place all the constraints that replaced C into $X_{I'}$. Since the number of these constraints is bounded by a constant, we obtain only a constant blow-up in the solution size. The converse can be shown similarly. ◀

By repeated applications of Lemma 9, the following corollary establishes that we need to provide approximation algorithms only for a few MINCSP s, and these algorithms can be used for other MINCSP s associated with the same co-clone.

► **Corollary 10.** *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 $\text{MINCSP}(\Gamma)$ to $\text{MINCSP}(B)$.*

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, \dots, a_k) with $a_i \neq a_j$. A set of relations S is *irredundant* if any relation in S is irredundant.

► **Theorem 11** ([28, 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 12.** *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 $\text{MINCSP}(B)$ to $\text{MINCSP}(\Gamma)$.*

Proof. Since $B \subseteq \langle \Gamma \rangle$ and B is irredundant, B can be $\exists\wedge$ -defined over Γ using Theorem 11. Then repeated applications of Lemma 9 shows the existence of the reduction from $\text{MINCSP}(B)$ to $\text{MINCSP}(\Gamma)$. ◀

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

► **Lemma 13.** *Let Γ be a self-dual constraint language. Assume that $x \oplus y \in \Gamma$. Then there is a cost-preserving reduction from $\text{MINCSP}(\Gamma \cup \{x, \bar{x}\})$ to $\text{MINCSP}(\Gamma)$.*

Proof. Let I be an instance of $\text{MINCSP}(\Gamma \cup \{x, \bar{x}\})$. We construct an instance J of $\text{MINCSP}^*(\Gamma)$ such that a deletion set of size k of I corresponds to a deletion set of size k of J ; then the result for MINCSP follows from the fact that there is a cost-preserving reduction from MINCSP^* to MINCSP (Lemma 4). Every constraint of the form $R(x_1, \dots, x_r)$ in I is placed into J . We introduce an undeletable constraint $x \oplus y$ to J , where x and y are new

variables. For any constraint $v = 0$ in I , we add a constraint $v \oplus y$, and for any constraint $v = 1$ in I , we add a constraint $v \oplus x$ in J .

Let W_I be a deletion set for I . To obtain a deletion set W_J of the same size for J , any constraint that is not of the form $v = 0$ or $v = 1$ is placed into W_J . For any constraint $v = 0 \in W_I$, we place the constraint $v \oplus y$ into W_J , and for any constraint $v = 1 \in W_I$, we place the constraint $v \oplus x$ into W_J . Then assigning 0 to x and 1 to y , and for the remaining variables of J using the assignment for the variables of I , we obtain a satisfying assignment for J .

The converse can be done by essentially reversing the argument, except that we might need to use the complement of the satisfying assignment for J to satisfy constraints of the form $v = 0$ and $v = 1$. ◀

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

► **Theorem 14.** *Let Γ be a finite set of Boolean relations.*

1. *If $\langle \Gamma \rangle \subseteq C$ (equivalently, if $\Gamma \subseteq C$), with $C \in \{\text{II}_0, \text{II}_1, \text{IS}_{00}, \text{IS}_{10}, \text{ID}_2\}$, then $\text{MINCSP}(\Gamma)$ 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 $\langle \Gamma \rangle \in \{\text{IL}_2, \text{IL}_3\}$, then $\text{MINCSP}(\Gamma)$ is equivalent to NEAREST CODEWORD and to ODD SET under A-reductions (note that these constraint languages are affine) and has no constant-factor FP-approximation, unless $\text{FPT} = \text{W}[1]$.*
3. *If $C \subseteq \langle \Gamma \rangle$, where $C \in \{\text{IE}_2, \text{IV}_2, \text{IN}_2\}$, then $\text{MINCSP}(\Gamma)$ does not have an FPA algorithm, unless $\text{FPT} = \text{W}[P]$. (Note that in these cases Γ can $\exists\wedge$ -define either arbitrary Horn relations, or arbitrary dual Horn relations, or the relation $\text{NAE}^3 = \{0, 1\}^3 \setminus \{(0, 0, 0), (1, 1, 1)\}$.)*

Looking at the co-clone lattice, it is easy to see that Theorem 14 covers all cases. It is also easy to check that Theorem 1 formulated in the introduction follows from Theorem 14. Theorem 14 is proved the following way. Statement 1 is proved in Section 4 (Lemma 15, and Corollaries 17 and 20). Statement 2 is proved in Section 5 (Theorem 22) and in Section 6. Statement 3 is proved in Section 7 (Corollary 35 and Lemma 36).

4 CSPs with FPA algorithms

We prove the first statement of Theorem 14 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 15.** *If $\langle \Gamma \rangle \subseteq \text{II}_0$ or $\langle \Gamma \rangle \subseteq \text{II}_1$, then $\text{MINCSP}(\Gamma)$ is polynomial-time solvable.*

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

► **Theorem 16** ([42]). *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 10.

► **Corollary 17.** *If $\langle \Gamma \rangle \subseteq \text{ID}_2$, then $\text{MINCSP}(\Gamma)$ has a constant-factor FPA algorithm.*

Proof. We check in Table 1 that $B = \{x \oplus y, x \rightarrow y\}$ is a base for the co-clone ID_2 . Relations in B (and equality) can be $\exists\wedge$ -defined over $\Gamma(2\text{-SAT})$, so the result follows from Corollary 10. ◀

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 $k\text{-IHS-B+}$ or $k\text{-IHS-}$ for some $k \geq 2$.

► **Lemma 18.** *If $\langle\Gamma\rangle \subseteq \text{IS}_{00}$, then there is an integer $k \geq 2$ such that Γ is $k\text{-IHS-B+}$. If $\langle\Gamma\rangle \subseteq \text{IS}_{10}$, then there is an integer $k \geq 2$ such that Γ is $k\text{-IHS-B-}$.*

Proof. We note that there is no finite base for IS_{00} , thus $\langle\Gamma\rangle$ is a proper subset of IS_{00} (see Table 1). There is no proper base for IS_{01} , IS_{02} , or IS_0 either. It follows from the structure of the co-clone lattice (Figure 1) that there exists a finite $k \geq 2$ such that $\langle\Gamma\rangle \subseteq \text{IS}_{00}^k$, which implies that Γ is IHS-B+ . The proof of the second statement is analogous. ◀

By Lemma 18, if $\langle\Gamma\rangle \subseteq \text{IS}_{00}$, then Γ is generated by the relations $\neg x, x \rightarrow y, x_1 \vee \dots \vee x_k$ for some $k \geq 2$. The MINCSP problem for this set of relations is known to admit a constant-factor approximation.

► **Theorem 19** ([17], Lemma 7.29). *$\text{MINCSP}(\neg x, x \rightarrow y, x_1 \vee \dots \vee x_k)$ has a $(k+1)$ -factor approximation algorithm (and hence has a constant-factor FPA algorithm).*

Now Theorem 19 and Corollary 10 imply that there is a constant-factor FPA algorithm for $\text{MINCSP}(\Gamma)$ whenever $\langle\Gamma\rangle$ is in the co-clone IS_{00} or IS_{10} (note that equality can be $\exists\wedge$ -defined using $x \rightarrow y$). In fact, the resulting algorithm is a polynomial-time approximation algorithm: Theorem 19 gives a polynomial-time algorithm and this is preserved by Corollary 10.

► **Corollary 20.** *If $\langle\Gamma\rangle \subseteq \text{IS}_{00}$ or $\langle\Gamma\rangle \subseteq \text{IS}_{10}$, then $\text{MINCSP}(\Gamma)$ has a constant-factor FPA algorithm.*

Note that Theorem 7.25 in [17] gives a complete classification of Boolean MINCSP s with respect to constant-factor approximability. As mentioned, these MINCSP s also admit a constant-factor approximation algorithm. The reason we need Corollary 20 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 $\text{NEAREST CODEWORD (NC)}$ 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 $\mathcal{S} = \{S_1, S_2, \dots, S_m\}$ 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 for each set we can specify whether it 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 21.** *There is a cost-preserving reduction from EVEN/ODD SET to ODD SET .*

Proof. Let I be the instance of EVEN/ODD SET. If all sets in I are even sets, then the empty set is an optimal solution. Otherwise, fix an arbitrary odd set S_o in I . We obtain an instance I' of ODD SET by introducing every odd set of I into I' , and for each even set S_e of I , we introduce the set $S_e \Delta S_o$ into I' , where Δ denotes the symmetric difference of two sets. This completes the reduction.

Let W be a solution of size k for I . We claim that W is also a solution of I' . Then those sets of I' that correspond to odd sets of I are obviously hit an odd number of times by W . We have to show that the remaining sets of I' are also hit an odd number of times. Let $T = S_e \Delta S_o$ be such a set for some even set S_e of I and the fixed set S_o . Let $A = S_o \setminus S_e$, $B = S_e \cap S_o$, and $C = S_o \setminus S_e$. If W hits A an even number of times, then B must be hit an odd number of times (since $A \cup B = S_o$), and therefore C is hit an odd number of times (as $B \cup C = S_e$). Since $T = S_e \Delta S_o = A \cup C$, T is hit an odd number of times. If W hits A an odd number of times, then B must be hit an even number of times, therefore C must be hit an even number of times, and therefore T is hit an odd number of times.

Conversely, let W' be a solution of size k for I' . We claim that W' is also a solution for I . Odd sets of I are clearly hit an odd number of times by W' . Let S_e be an even set of I . We show that S_e is hit an even number of times. The fixed set S_o (also in the instance I') and $S_e \Delta S_o$ are both hit an odd number of times by W' . Define sets A, B, C as above. If W' hits C an odd number of times, then A must be hit an even number of times, as $A \cup C = S_e \Delta S_o$ is hit an odd number of times. Since $S_o = B \cup A$ is hit an odd number of times, B is hit an odd number of times. Since $S_e = C \cup B$, S_e is hit an even number of times. The case when W' hits C an even number of times can be analyzed is similarly. \blacktriangleleft

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

► Theorem 22. ⁶ *The following problems are equivalent under cost-preserving reductions: (1) NEAREST CODEWORD, (2) ODD SET, (3) MINCSP(B_2), and (4) MINCSP(B_3).*

Proof. (1) \Rightarrow (2): Let A be an $m \times n$ generator matrix for the NEAREST CODEWORD problem, and c be an m -dimensional vector such that we want to find a codeword of Hamming distance at most k from c . Let C be the set of all codewords generated by A , i.e., vectors in the column space of A . Let A^\perp be the $\ell \times m$ matrix whose rows form a basis for the subspace perpendicular to the column space of A . Then $w \in C$ if and only if $A^\perp w = 0$. Assume now that z is a vector that differs from c at most in k positions. Then we can write $z = z' + c$, where the weight of z' is the distance between z and z' . To find such a z , we write $A^\perp(z' + c) = 0$, and now we wish to find a solution that minimizes the weight of z' . Observe that $A^\perp z' = A^\perp c$ (since we are working in $\text{GF}(2)$). This can be encoded as a problem where we have a ground set $U = \{1, \dots, m\}$, and sets S_i , $1 \leq i \leq \ell$, defined as follows. Element j is in S_i if $A^\perp(i, j) = 1$. We want to find a subset $W \subseteq U$ of size at most k such that S_i is hit an even number of times if the i -th element of the vector $A^\perp c$ is 0, and an odd number of times if it is 1. This is an instance of the EVEN/ODD SET problem. Using Lemma 21, we can further reduce this problem to ODD SET, and we are done.

(2) \Rightarrow (3): We show that there is a cost-preserving reduction from ODD SET to $\text{MINCSP}^*(B_2)$ (and hence to $\text{MINCSP}(B_2)$ by Lemma 4). First we $\exists \wedge$ -express the relation ODD^n using B_2 .

⁶ Note that Lemma 1 in [20] can be adapted to obtain the reduction from ODD SET to $\text{MINCSP}(B_2)$.

We induct on n . For $n = 2$, we have that $\text{ODD}^2(x_1, x_2) = \text{EVEN}^4(x_1, x_2, 0, 1)$. Assume we have a formula that defines ODD^n . Then observe that

$$\text{ODD}^{n+1}(x_1, \dots, x_{n+1}) = \exists u \text{ODD}^n(x_1, \dots, x_{n-1}, u) \wedge \text{EVEN}^3(u, x_n, x_{n+1}).$$

The variables of the MINCSP instance J are the elements of the ground set of the ODD SET instance I . For each set $\{y_1, \dots, y_s\}$ of I , we add an undeletable constraint $\text{ODD}^s(y_1, \dots, y_s)$ to J . Finally, for each variable y that appears in a constraint, we add the constrain \bar{y} . It is easy to see that a hitting set of size k for I (consisting of constraints of the form \bar{y}) corresponds to a deletion set of size k for I .

(3) \Rightarrow (4): This follows from Lemma 13.

(4) \Rightarrow (1): Let I be the MINCSP(B_3) instance, and assume it has n variables and m constraints. We define a NEAREST CODEWORD instance J as follows. The matrix A has dimension $m \times n$, and columns are indexed by the variables of I . If the i -th constraint of I is $\text{EVEN}^4(x_{j_1}, x_{j_2}, x_{j_3}, x_{j_4})$, then the i -th row of A has 1-s in positions j_1, j_2, j_3, j_4 , and the i -th entry of vector b is 0. If the i -th constraint of I is $x_{k_1} \oplus x_{k_2}$, then the i -th row of A has 1-s in positions k_1 and k_2 , and the i -th entry of b is 1. Clearly, a deletion set of size k for I corresponds to a solution of J having distance k from vector b . \blacktriangleleft

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 23.** *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 NEAREST CODEWORD [2]. Let $\mathcal{S} = \{S_1, \dots, S_m\}$ be any instance over universe $U = \{x_1, \dots, x_n\}$. Let $\varepsilon > 0$ be any real positive value and k be the size of an optimal solution. We can assume that $k \geq \frac{1}{\varepsilon}$ since one can find an optimal solution by exhaustive search in time $O(n^{1+\frac{1}{\varepsilon}}m)$. We build the set-system $\mathcal{S}' = \mathcal{S} \cup \bigcup_{i \in [n], j \in [m]} S_j^i \cup \{\{e\}\}$ over universe $U' = U \cup \bigcup_{i, h \in [n]} \{x_h^i\} \cup \{e\}$, where e is a new element, such that $S_j^i = \{e, x_i\} \cup \{x_h^i \mid x_h \in S_j\}$. 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 $T' = \{e\} \cup T \cup \{x_h^i \mid x_i, x_h \in T\}$ 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_j^i 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_j^i with $j \in [m]$ is hit exactly twice by e and x_i . Thus, one has to select a subset of $\{x_1^i, \dots, x_n^i\}$ to hit each set of the family $\{S_1^i, \dots, S_m^i\}$ 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 . Finally, T has size smaller than $\sqrt{r(1 + k + k^2)} \leq \sqrt{r}(k + 1) = (1 + \frac{1}{k})\sqrt{r}k \leq (1 + \varepsilon)\sqrt{r}k$. ◀

Repeated application of the self-improvement property in Lemma 23 shows that any constant-ratio approximation implies the existence of $(1 + \varepsilon)$ -approximation for arbitrary small $\varepsilon > 0$. In a similar way, we can show that polylogarithmic approximation implies the existence of logarithmic approximation.

- **Corollary 24.** 1. If ODD SET admits an FPA algorithm with some ratio $r \geq 1$, then, for any $\varepsilon > 0$, it also admits an FPA algorithm with ratio $1 + \varepsilon$ and
2. If ODD SET admits an FPA algorithm with ratio $\log^c k$, then it also admits an FPA algorithm with ratio $\log k$.

Proof. We observe that for any $r' > 1$, there exists an $\varepsilon > 0$ such that $(1 + \varepsilon)\sqrt{(1 + \varepsilon)\sqrt{r'}} \leq \sqrt{r'}$. Thus, applying twice the reduction of Lemma 23, we can improve any fixed-parameter r' -approximation to a fixed-parameter $\sqrt{r'}$ -approximation. Therefore, starting with an r -approximation, we can repeatedly apply the self-improvement property a constant number of times to obtain an FPA algorithm with ratio arbitrarily close to 1. Similarly, starting with a $\log^c k$ -approximation, applying the self-improvement property a constant number of times gives a $\log k$ -approximation. Note that we are repeating the reduction of Lemma 23 a constant number of times, hence the reduction is still a polynomial-time reduction. ◀

6 Hardness of Odd Set

In this section, we show that ODD SET has no constant-factor FP-approximation unless $W[1] = FPT$. Under the same hypothesis, we even rule out FP-approximation with ratio smaller than $\log k$.

The proofs in this section use linear algebra. We will need the following notation. If n, m, d, k are positive integers and q is a prime power, then \mathbb{F}_q^d denotes the d -dimension vector space over \mathbb{F}_q . Each vector $\mathbf{v} \in \mathbb{F}_q^d$ can be written as $\mathbf{v} = (v_1, v_2, \dots, v_d)$ with $v_i \in \mathbb{F}_q$ for all $i \in [d]$. We will denote by $\mathbf{v}[i]$ the i -th coordinate of \mathbf{v} . Let $\mathbf{1}_d := (1, 1, \dots, 1) \in \mathbb{F}_q^d$ and $\mathbf{0}_d := (0, 0, \dots, 0) \in \mathbb{F}_q^d$. Suppose $\mathbf{a} = (a_1, a_2, \dots, a_n) \in \mathbb{F}_q^n$ and $\mathbf{b} = (b_1, b_2, \dots, b_m) \in \mathbb{F}_q^m$, we write $\mathbf{a} \circ \mathbf{b} := (a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m) \in \mathbb{F}_q^{n+m}$ for the concatenation of these two vectors.

6.1 One side gap for the biclique problem

Our inapproximability result for ODD SET builds on the recent $W[1]$ -hardness and inapproximability results for BICLIQUE by Lin [34]. The decision version of BICLIQUE problem asks for a complete bipartite subgraph with k vertices on each side. We consider the approximation version of the problem where one side is fixed and the other side has to be maximized. Formally, we define the following gap version of the problem.

GAP-BICLIQUE(s, ℓ, h)

Instance: A bipartite graph $G = (L \cup R, E)$ with n vertices and $s, \ell, h \in \mathbb{N}$ with $\ell < h$.

Parameter: s .

Problem: Distinguish between the following cases:

(yes) There exist s vertices in L with h common neighbors.

(no) Any s vertices in L have at most ℓ common neighbors.

The following theorem is the main result of Lin [34].

► **Theorem 25** ([34, Theorem 1.3]). *There is a polynomial time algorithm \mathbb{A} such that for every graph G with n vertices and $k \in \mathbb{N}$ with $\lceil n^{\frac{6}{k+6}} \rceil > (k+6)!$ and $6 \mid k+1$ the algorithm \mathbb{A} constructs a bipartite graph $H = (A \dot{\cup} B, E)$ satisfying:*

1. *if G contains a clique of size k , i.e., $K_k \subseteq G$, then there are s vertices in A with at least $\lceil n^{\frac{6}{k+1}} \rceil$ common neighbors in B ;*
2. *otherwise $K_k \not\subseteq G$, any s vertices in A have at most $(k+1)!$ common neighbors in B , where $s = \binom{k}{2}$.*

Our goal is to reduce GAP-BICLIQUE to the following gap version of ODD SET.

GAP-ODD-SET(k_1, k_2)	
<i>Instance:</i>	A set $\mathcal{W} = \{w_1, w_2, \dots, w_n\}$ of n vectors from the vector space \mathbb{F}_2^m ($m = n^{O(1)}$) and $k_1, k_2 \in \mathbb{N}$ with $k_1 \leq k_2$.
<i>Parameter:</i>	k_1 .
<i>Problem:</i>	Distinguish between the following two cases: (yes) There exists a set $I \subseteq [n]$ with $ I \leq k_1$ s.t. $\sum_{i \in I} w_i = \mathbf{1}_m$. (no) For all non-empty set $I \subseteq [n]$ with $ I \leq k_2$, $\sum_{i \in I} w_i \neq \mathbf{1}_m$.

We prove that unless $W[1] = \text{FPT}$, GAP-ODD-SET(k, ck) has no FPT-algorithm for any constant $c \geq 1$. Our reduction takes two steps. First we show that the following problem GAP-LINEAR-DEPENDENT-SET $_q(k_1, k_2)$ is $W[1]$ -hard for $k_2 = k_1 \log k_1$ and $q = n^{O(1)}$. As finding dependent sets is a fairly natural algorithmic problem in linear algebra, the FP-inapproximability of this problem is interesting on its own right.

GAP-LINEAR-DEPENDENT-SET$_q(k_1, k_2)$	
<i>Instance:</i>	A set $\mathcal{W} = \{w_1, w_2, \dots, w_n\}$ of n vectors from the vector space \mathbb{F}_q^m ($m = n^{O(1)}$) and $k_1, k_2 \in \mathbb{N}$ with $k_1 \leq k_2$.
<i>Parameter:</i>	k_1 .
<i>Problem:</i>	Distinguish between the following cases: (yes) There exist k_1 vectors in \mathcal{W} that are linearly dependent. (no) There are no k_2 vectors in \mathcal{W} that are linearly dependent.

Then we provide a gap-preserving reduction from GAP-LINEAR-DEPENDENT-SET $_q(k_1, k_2)$ to GAP-ODD-SET(k, ck).

6.2 From Biclique to Linear Dependent Set

The following theorem states our first reduction, which transfers the inapproximability of BICLIQUE to LINEAR DEPENDENT SET.

► **Theorem 26.** *Given an instance $G = (L \dot{\cup} R, E)$ of GAP-BICLIQUE(s, ℓ, h), there is an algorithm that constructs a set \mathcal{W} of vectors of \mathbb{F}_q^m in time $(|L| + |R|)^{O(1)}$ with q and m being bounded by $(|L| + |R|)^{O(1)}$, such that*

- *(Completeness) if G is a yes-instance of GAP-BICLIQUE(s, ℓ, h), then \mathcal{W} contains at most sh vectors that are linearly dependent,*
- *(Soundness) if G is a no-instance of GAP-BICLIQUE(s, ℓ, h), then \mathcal{W} contains no $h \sqrt{\frac{h}{\ell}} - 1$ vectors or less that are linearly dependent.*

Before proving Theorem 26, let us show how it can be used to prove an inapproximability result for GAP-LINEAR-DEPENDENT-SET.

► **Theorem 27.** *There is no FPT-algorithm for GAP-LINEAR-DEPENDENT-SET_{2^{O(log n)}}(k₁, k₂) with k₂ = k₁ log k₁, unless W[1] = FPT.*

Proof. We show how *k*-CLIQUE can be solved using an algorithm for GAP-LINEAR-DEPENDENT-SET_{2^{O(log n)}}(k₁, k₂). Let (G, k) be an input instance of *k*-CLIQUE. Without loss of generality, we can assume that $\lceil n^{\frac{6}{k+6}} \rceil > (k+6)!$ and $6 \mid k+1$. Let $s = \binom{k}{2}$, $\ell = (k+1)!$ and $h = k^{2k^3}$. For sufficiently large *n*, we have $(k+1)! \leq \ell < h \leq n^{6/(k+1)}$. Applying Theorem 25 to *G* and *k*, we obtain a bipartite graph *H* in $|G|^{O(1)}$ -time such that *G* contains a *k*-clique if and only if (H, s, ℓ, h) is a yes-instance of GAP-BICLIQUE(s, ℓ, h).

Applying Theorem 26 to (H, s, ℓ, h), we obtain a set \mathcal{W} of vectors of \mathbb{F}_q^m for some *q* and *m* bounded by $|G|^{O(1)}$. Let $k_1 = hs$ and $k_2 = k_1 \log k_1$. Note that for $k \geq 4$

$$\sqrt[s]{\frac{h}{\ell}} = \sqrt[s]{\frac{k^{2k^3}}{(k+1)!}} \geq \sqrt[k^2]{\frac{k^{2k^3}}{k^{2k}}} \geq k^{2k-2/k} \geq k^7 \geq 2k^5 \log k + 2k^2 \log k \geq s \log(hs).$$

Thus $k_1 \log k_1 = hs \log(hs) \leq h \sqrt[s]{\frac{h}{\ell}}$. By the completeness and soundness of Theorem 26, we have (W, k₁, k₂) is a yes-instance of GAP-LINEAR-DEPENDENT-SET_q(k₁, k₂) if and only if *G* contains a *k*-clique. ◀

It will be convenient to work with a colored version of GAP-LINEAR-DEPENDENT-SET.

GAP-LINEAR-DEPENDENT-SET_q^{col}(k₁, k₂)

Instance: A set $\mathcal{W} = \{w_1, w_2, \dots, w_n\}$ of *n* vectors from the vector space \mathbb{F}_q^m ($m = n^{O(1)}$), $k_1, k_2 \in \mathbb{N}$ with $k_1 \leq k_2$ and a coloring $c : \mathcal{W} \rightarrow [k_1]$.

Parameter: k_1 .

Problem: Distinguish between the following cases:

- (yes) There exist at most k_1 vectors in \mathcal{W} with distinct colors under *c* that are linearly dependent.
- (no) There are no k_2 vectors or less in \mathcal{W} that are linearly dependent.

With a standard application of the color-coding technique (see, e.g., [21]), we can reduce an instance of GAP-LINEAR-DEPENDENT-SET_q(k₁, k₂) to $2^{O(k_1)} \log n$ instances of GAP-LINEAR-DEPENDENT-SET_q^{col}(k₁, k₂), showing the hardness of the latter problem.

► **Theorem 28.** *There is no FPT-algorithm for GAP-LINEAR-DEPENDENT-SET_q^{col}(k₁, k₂) with k₂ = k₁ log k₁ and q = n^{O(1)}, unless W[1] = FPT.*

Let us prove now Theorem 26, the reduction from GAP-BICLIQUE to GAP-LINEAR-DEPENDENT-SET.

Proof (of Theorem 26). Assume that an instance $G = (L \dot{\cup} R, E)$ of GAP-BICLIQUE(s, ℓ, h) is given. We construct a set \mathcal{W} of vectors and prove the required completeness and soundness.

Construction of \mathcal{W} . Let $q := 2^{\lceil \log(|L|+|R|) \rceil} \geq |L| + |R|$. We identify *L* and *R* with disjoint subsets of \mathbb{F}_q . Let $B := \max\{h, s\} - 1$. We first define a function $\iota : L \cup R \rightarrow \mathbb{F}_q^B$ as follows.

- for each $v \in R$, $\iota(v) := (1, v, \dots, v^{h-2}) \circ \mathbf{0}_{B-h+1}$,
- for each $u \in L$, $\iota(u) := (1, u, \dots, u^{s-2}) \circ \mathbf{0}_{B-s+1}$.

Using well-known properties of Vandermonde matrices, we can see that any $h - 1$ vectors in $\iota(R)$ are linearly independent and any h vectors from $\iota(R)$ are linearly dependent. To summarize, R and ι satisfies the following conditions:

- (R1) for all $I \in \binom{R}{h}$, the vectors $\{\iota(v) : v \in I\}$ are linearly dependent.
- (R2) for all $I \in \binom{R}{h-1}$, the vectors $\{\iota(v) : v \in I\}$ are linearly independent.

Similarly, we also have that

- (L1) for all $I \in \binom{L}{s}$, the vectors $\{\iota(u) : u \in I\}$ are linearly dependent.
- (L2) for all $I \in \binom{L}{s-1}$, the vectors $\{\iota(u) : u \in I\}$ are linearly independent.

Then we let $m = qB$ and consider vectors from $\mathbb{F}_q^m = \mathbb{F}_q^{qB}$, which can be seen as the concatenation of q blocks, each of B coordinates. For $x \in \mathbb{F}_q^m$, we use the notation $x^{(i)}$ to refer to the i -block, which is the B -dimensional vector given by coordinates $\{(i-1)B+1, (i-1)B+2, \dots, (i-1)B+B\}$. For each $u \in L$ and $v \in R$ with $\{u, v\} = e \in E$, we introduce a vector $w_e \in \mathbb{F}_q^{qB}$ such that

- (W1) for all $i \in [q] \setminus \{v, u\}$, $w_e^{(i)} = \mathbf{0}_B$,
- (W2) $w_e(v) = \iota(u)$,
- (W3) $w_e(u) = \iota(v)$.

That is, we can imagine w_e as being partitioned q blocks of B coordinates, with the representation of u appearing in the v -th block and the representation of v appearing in the u -th block. Note the use of u and v in the definition: the v -th block on its own describes both v (by its position) and u (by its content), and similarly the u -th block also describes both endpoints of e . Finally, let

$$\mathcal{W} := \{w_e : e \in E\}.$$

Obviously, \mathcal{W} can be computed in time $(|L| + |R|)^{O(1)}$.

Completeness. Suppose G is a yes-instance of $\text{GAP-BICLIQUE}(s, \ell, h)$. There exist a set $X \in \binom{L}{s}$ and a set $Y \in \binom{R}{h}$ such that for all $u \in X$ and $v \in Y$, $\{u, v\} \in E$. Suppose $X = \{u_1, u_2, \dots, u_s\}$ and $Y = \{v_1, v_2, \dots, v_h\}$. By (R1) and (L1), there exists $a_i \in \mathbb{F}_q$ for each $i \in [s]$ and $b_j \in \mathbb{F}_q$ for each $j \in [h]$ such that

$$\sum_{i \in [s]} a_i \iota(u_i) = \mathbf{0}_B \text{ and } \sum_{j \in [h]} b_j \iota(v_j) = \mathbf{0}_B.$$

We prove that \mathcal{W} contains a set of sh dependent vectors by showing

$$\sum_{i \in [s], j \in [h]} a_i b_j w_{\{u_i, v_j\}} = \mathbf{0}_m.$$

Let $w = \sum_{i \in [s], j \in [h]} a_i b_j w_{\{u_i, v_j\}}$. It is easy to check that

- by (W1), for every $z \in [q] \setminus (X \cup Y)$, $w^{(z)} = \mathbf{0}_B$,
- by (W2), for every $v_j \in Y$, $w^{(v_j)} = \sum_{i \in [s]} a_i b_j \iota(u_i) = b_j \sum_{i \in [s]} a_i \iota(u_i) = \mathbf{0}_B$,
- by (W3), for every $u_i \in X$, $w^{(u_i)} = \sum_{j \in [h]} a_i b_j \iota(v_j) = a_i \sum_{j \in [h]} b_j \iota(v_j) = \mathbf{0}_B$.

It follows that $\{w_{\{u,v\}} : u \in X \text{ and } v \in Y\}$ is a set of sh linearly dependent vectors from \mathcal{W} .

Soundness. Suppose G is a no-instance of $\text{GAP-BICLIQUE}(s, \ell, h)$. Let $W \subseteq \mathcal{W}$ be a set of vectors that are linearly dependent. We define two vertex sets and their edge set as follows. Let

$$X := \{u \in L : \text{there exists } v \in R \text{ such that } w_{\{u,v\}} \in W\},$$

$$Y := \{v \in R : \text{there exists } u \in L \text{ such that } w_{\{u,v\}} \in W\},$$

and

$$E_W := \{\{u, v\} : w_{\{u,v\}} \in W\}.$$

First, we note that X and Y are not empty because W is non-empty. By (R2) and (W3), for every $u \in X$, there exists at least h vertices in Y that are adjacent to u , i.e. $|N(u) \cap Y| \geq h$. Similarly, by (L2) and (W2), for every $v \in Y$, we have $|N(v) \cap X| \geq s$.

► **Claim 29.** For $s, \ell, h \in \mathbb{N}^+$ and two non-empty sets X and Y . Let $G_W := (X \cup Y, E_W)$ be a bipartite graph such that

- (i) every vertex in X has at least h neighbors,
 - (ii) every vertex in Y has at least s neighbors,
 - (iii) every s -vertex set of X has at most ℓ common neighbors,
- then $|E_W| \geq (\frac{h}{\ell})^{1/s} h$.

Proof of Claim 29. Because X is not empty, there exists a vertex $u \in X$. By (i), u has at least h neighbors in Y , so $|Y| \geq h$. By (ii), for every $v \in Y$, v has at least s neighbors in X . If $\binom{|X|}{s} \ell < |Y|$, then there must exist a s -vertex set in X which has more than ℓ common neighbors in Y . Thus we must have that

$$|X|^s \geq \binom{|X|}{s} \ell \geq \frac{|Y|}{\ell} \geq \frac{h}{\ell}.$$

By (i) again, we conclude that $|E_W| \geq h|X| \geq (\frac{h}{\ell})^{1/s} h$, what we had to show. ◻

As E_W satisfies the conditions of Claim 29, we have $|W| = |E_W| \geq h \sqrt[s]{\frac{h}{\ell}}$. ◀

6.3 From Linear Dependent Set to Odd Set

We present the following reduction as a warm up. Together with Theorem 28, it shows that there is no better than 3-approximation for ODD SET, unless $\text{W}[1] = \text{FPT}$.

► **Theorem 30.** Given an instance \mathcal{W} of $\text{GAP-LINEAR-DEPENDENT-SET}_{2^d}^{\text{col}}(k, k \log k)$ with $\mathcal{W} \subseteq \mathbb{F}_2^{m \times d}$ one can construct an instance \mathcal{W}' of $\text{GAP-ODD-SET}(k+1, 3k+1)$ with $\mathcal{W}' \subseteq \mathbb{F}_2^{1+m \times d+k}$ in polynomial time such that \mathcal{W} is a yes-instance if and only if \mathcal{W}' is a yes-instance.

Proof. Suppose \mathcal{W} is an instance of $\text{GAP-LINEAR-DEPENDENT-SET}_{2^d}^{\text{col}}(k, k \log k)$ with the coloring c and $k \geq 3$. For the definition of the set \mathcal{W}' , we need to introduce some notation. Let η_i be the k -dimensional vector with 1 at the i -th position and 0 everywhere else. We can view the finite field \mathbb{F}_2^d as a d -dimensional vector space of \mathbb{F}_2 , hence there exist d elements e_1, e_2, \dots, e_d in \mathbb{F}_2^d such that every $v \in \mathbb{F}_2^d$ can be expressed, in a unique way, as the sum of a subset of the e_i 's. Fix d such elements e_1, e_2, \dots, e_d and let $f : \mathbb{F}_2^d \rightarrow \mathbb{F}_2^d$ be a function that for each $v = \sum_{i \in [d]} c_i e_i \in \mathbb{F}_2^d$, $f(v) = (c_1, c_2, \dots, c_d) \in \mathbb{F}_2^d$. Similarly, we can define a

function $F : \mathbb{F}_{2^d}^m \rightarrow \mathbb{F}_{2^d}^{md}$ with $F(v) = f(v[1]) \circ \dots \circ f(v[m])$. Observe that for any $x \in \mathbb{F}_{2^d}^m$, we have $F(x) = \mathbf{0}_{md}$ if and only if $x = \mathbf{0}_m$, and $F(x + y) = F(x) + F(y)$ holds for any $x, y \in \mathbb{F}_{2^d}^m$.

The set \mathcal{W}' is defined as follows.

$$\mathcal{W}' := \{\mathbf{0}_1 \circ F(aw) \circ \eta_{c(w)} : w \in \mathcal{W}, a \in \mathbb{F}_{2^d}^+\} \cup \{\mathbf{1}_1 \circ \mathbf{1}_{md} \circ \mathbf{0}_k\}.$$

- (Completeness) If \mathcal{W} is a yes-instance of $\text{GAP-LINEAR-DEPENDENT-SET}_{2^d}^{col}(k, k \log k)$, there must exist $w_1, w_2, \dots, w_k \in \mathcal{W}$ and $a_1, a_2, \dots, a_k \in \mathbb{F}_{2^d}^+$ such that for all $i \in [k]$ $c(w_i) = i$ and

$$\sum_{i \in [k]} a_i w_i = \mathbf{0}_m.$$

It is easy to verify that

$$\mathbf{1}_1 \circ \mathbf{1}_{md} \circ \mathbf{0}_k + \sum_{i \in [k]} (\mathbf{0}_1 \circ F(a_i w_i) \circ \eta_i) = \mathbf{1}_{1+md+k}.$$

- (Soundness) Suppose \mathcal{W} is a no-instance and $W' \subseteq \mathcal{W}'$ is a set of vectors whose sum w' is equal to $\mathbf{1}_{1+md+k}$. First, W' must contain the vector $\mathbf{1}_1 \circ \mathbf{1}_{md} \circ \mathbf{0}_k$, otherwise w' has 0 in its first coordinate. For each $i \in [k]$, let

$$X_i := \{(w, a) : w \in \mathcal{W} \text{ and } a \in \mathbb{F}_{2^d}^+ \text{ such that } c(w) = i \text{ and } \mathbf{0}_1 \circ F(aw) \circ \eta_i \in W'\}.$$

It follows that for each $i \in [k]$, $|X_i|$ must be odd, otherwise the $(1 + md + i)$ -th element of w' is not equal to 1. If for all $i \in [k]$, $|X_i| \geq 3$, then $|W'| \geq 3k + 1$ and we are done. Otherwise suppose for some $i \in [k]$, $X_i = \{(x^*, a^*)\}$. For each $w \in \mathcal{W}$, let $a_w := \sum_{a \in \mathbb{F}_{2^d}^+, (w, a) \in X_{c(w)}} a$. Then we define

$$X := \{w : w \in \mathcal{W}, a_w \neq 0\}.$$

Note that $x^* \in X$ so X is not empty. Let y be the sum of vectors in $W' \setminus \{\mathbf{1}_1 \circ \mathbf{1}_{md} \circ \mathbf{0}_k\}$. Thus $y = w' - \mathbf{1}_1 \circ \mathbf{1}_{md} \circ \mathbf{0}_k = \mathbf{1}_{1+md+k} - \mathbf{1}_1 \circ \mathbf{1}_{md} \circ \mathbf{0}_k = \mathbf{0}_1 \circ \mathbf{0}_{md} \circ \mathbf{1}_k$. This means that $\sum_{x \in X} F(a_x \cdot x) = \mathbf{0}_{md}$, which is only possible if $\sum_{x \in X} a_x \cdot x = \mathbf{0}_m$. Since X is a non-empty set of linearly dependent vectors in \mathcal{W} , we must have $k \log k \leq |X| \leq |W'| - 1$. Therefore $|W'| \geq 1 + k \log k \geq 1 + 3k$. ◀

The self-improvement property of ODD SET (Corollary 24(1)) shows that a c -approximation for ODD SET implies the existence of a polynomial-time c' -approximation for every $c' > 0$. This means that the reduction in Theorems 30 and 28 already show that there is no constant-factor FP-approximation for ODD SET , unless $\text{W}[1] = \text{FPT}$. However, with a slight change in the reduction, we can improve the inapproximability to $\log k$ and then Corollary 24(2) can improve this further to polylogarithmic inapproximability, proving Theorem 2.

► **Theorem 31.** *Given an instance \mathcal{W} of $\text{GAP-LINEAR-DEPENDENT-SET}_{2^d}^{col}(k, k \log k)$ one can construct $(d + 1)^k$ instances of $\text{GAP-ODD-SET}(k, k \log k)$ in time $d^k n^{O(1)}$ such that \mathcal{W} is a yes-instance if and only if at least one of the instances $\text{GAP-ODD-SET}(k + 1, k \log k + 1)$ is a yes-instance.*

Proof. We use the same definitions for η , f , and F as in the proof of Theorem 30. For each $i \in [d]$, we define a subset C_i of $\mathbb{F}_{2^d}^+$ by setting

$$C_i := \{c \in \mathbb{F}_{2^d}^+ : f(c)[i] = 1 \text{ and } f(c) \text{ contains an even number of ones}\}.$$

In addition, we put $C_{d+1} := \{c \in \mathbb{F}_{2^d}^+ : f(c) \text{ contains an odd number of ones}\}$. The sets C_i 's have the following properties that are crucial to our reduction.

- (i) $\bigcup_{i \in [d+1]} C_i = \mathbb{F}_{2^d}^+$.
- (ii) for all $i \in [d+1]$ the sum of any odd numbers of elements in C_i is not equal to 0.

The main idea is the following. If there are k vectors in \mathcal{W} that can generate the zero vector with coefficients a_1, \dots, a_k , then we “guess” for each a_i one of the sets C_1, \dots, C_{d+1} that contains it. We create one instance for each of these guesses. The advantage of this approach is that we can more efficiently argue about soundness than in the proof of Theorem 30.

Formally, for every $g \in [d+1]^k$, we construct an instance \mathcal{W}_g of GAP-ODD-SET($k+1, k \log k + 1$) by setting

$$\mathcal{W}_g := \{\mathbf{0}_1 \circ F(aw) \circ \eta_{c(w)} : w \in \mathcal{W}, a \in C_{g[c(w)]}\} \cup \{\mathbf{1}_1 \circ \mathbf{1}_{md} \circ \mathbf{0}_k\}.$$

- (Completeness) If \mathcal{W} is a yes-instance of GAP-LINEAR-DEPENDENT-SET $_{2^d}^{col}(k, k \log k)$, there must exist $w_1, w_2, \dots, w_k \in \mathcal{W}$ and $a_1, a_2, \dots, a_k \in \mathbb{F}_{2^d}^+$ such that for all $i \in [k]$ $c(w_i) = i$ and

$$\sum_{i \in [k]} a_i w_i = \mathbf{0}.$$

It follows that

$$\mathbf{1}_1 \circ \mathbf{1}_{md} \circ \mathbf{0}_k + \sum_{i \in [k]} (\mathbf{0}_1 \circ F(a_i w_i) \circ \eta_i) = \mathbf{1}_{1+md+k}.$$

Next we define $g^* \in [d+1]^k$ by setting $g^*[i] := \min\{j \in [d+1] : a_i \in C_j\}$ for all $i \in [k]$. Because $\bigcup_{j \in [d+1]} C_j = \mathbb{F}_{2^d}^+$, the vector g^* is well defined. It is easy to check that for all $i \in [k]$, $a_i \in C_{g^*(i)}$, hence $\mathbf{0}_1 \circ F(a_i w_i) \circ \eta_i \in \mathcal{W}_{g^*}$. This means that \mathcal{W}_{g^*} is a yes-instance.

- (Soundness) Suppose \mathcal{W} is a no instance and $W' \subseteq \mathcal{W}_g$ for some $g \in [d+1]^k$ is a set of vectors whose sum w' is equal to $\mathbf{1}_{1+md+k}$. First, W' must contain the vector $\mathbf{1}_1 \circ \mathbf{1}_{md} \circ \mathbf{0}_k$, otherwise w' has 0 in its first coordinate. For each $i \in [k]$, let

$$X_i := \{(w, a) : w \in \mathcal{W} \text{ and } a \in \mathbb{F}_{2^d}^+ \text{ such that } c(w) = i \text{ and } \mathbf{0}_1 \circ F(aw) \circ \eta_i \in W'\}.$$

It follows that for each $i \in [k]$, $|X_i|$ must be odd, otherwise the $(1+md+i)$ -th element of w' is not equal to 1. For each $w \in \mathcal{W}$, let $a_w := \sum_{a \in \mathbb{F}_{2^d}^+, (w,a) \in X_{c(w)}} a$. Then we define

$$X := \{w : w \in \mathcal{W}, a_w \neq 0\}.$$

Because $|X_i|$ is odd for all $i \in [k]$, there exists an element $w_i \in \mathcal{W}$ such that $|\{a \in \mathbb{F}_{2^d}^+ : (w_i, a) \in X_i\}|$ is odd. Note that for all $(w, a) \in X_i$, $a \in C_{g(i)}$. By (ii), the sum of odd number of elements from $C_{g(i)}$ is non-zero. Thus $a_{w_i} = \sum_{a \in \mathbb{F}_{2^d}^+, (w_i, a) \in X_i} a \neq 0$, which implies that X is not empty.

Let y be the sum of vectors in $W' \setminus \{\mathbf{1}_1 \circ \mathbf{1}_{md} \circ \mathbf{0}_k\}$. Thus $y = w' - \mathbf{1}_1 \circ \mathbf{1}_{md} \circ \mathbf{0}_k = \mathbf{1}_{1+md+k} - \mathbf{1}_1 \circ \mathbf{1}_{md} \circ \mathbf{0}_k = \mathbf{0}_1 \circ \mathbf{0}_{md} \circ \mathbf{1}_k$. This means that $\sum_{x \in X} F(a_x \cdot x) = \mathbf{0}_{md}$, which is only possible if $\sum_{x \in X} a_x \cdot x = \mathbf{0}_m$. Since X is a non-empty set of linearly dependent vectors in \mathcal{W} , we must have $k \log k \leq |X| \leq |W'| - 1$. Therefore $|W'| \geq 1 + k \log k$. ◀

7 Hardness of Horn (IV_2), dual-Horn (IE_2) and IN_2

In this section, we establish statement 3 of Theorem 14 by proving the inapproximability of $\text{MINCSP}(\Gamma)$ 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 *monotone 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 0 is an input node, and nodes with in-degree 1 are not allowed. Furthermore, there is a node with out-degree 0 that is the output node. Let C be a monotone Boolean circuit. Given an assignment φ from the nodes of C to $\{0, 1\}$, we say that φ satisfies C if

- for any OR node G_\vee with in-neighbors G_1, \dots, G_n , $\varphi(G_\vee) = 1$ if and only if $1 \in \{\varphi(G_1), \dots, \varphi(G_n)\}$,
- for any AND node G_\wedge with in-neighbors G_1, \dots, G_n , $\varphi(G_\wedge) = 1$ if and only if $\varphi(G_1) = \dots = \varphi(G_n) = 1$, and
- $\varphi(G_o) = 1$, where G_o is the output node.

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 . 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 [38]. We use this to show that Horn-CSPs are hard.

► **Theorem 32** ([38]). **MONOTONE CIRCUIT SATISFIABILITY** *does not have an FPA algorithm, unless $\text{FPT} = \text{W[P]}$.*

The following corollary can be achieved by simply replacing nodes of in-degree larger than 2 with a circuit consisting of nodes having in-degree at most two in the standard way.

► **Corollary 33.** **MONOTONE CIRCUIT SATISFIABILITY**, *where circuits are restricted to have nodes of in-degree at most 2, does not have an FPA algorithm, unless $\text{FPT} = \text{W[P]}$.*

We use Corollary 33 to establish the inapproximability of Horn-SAT and dual-Horn-SAT, assuming that $\text{FPT} \neq \text{W[P]}$. Using the co-clone lattice, this will show hardness of approximability of $\text{MINCSP}(\Gamma)$ if $\langle \Gamma \rangle \in \{\text{IV}_2, \text{IE}_2\}$.

► **Lemma 34.** *If $\text{MINCSP}(\{x \vee y \vee \bar{z}, x, \bar{x}\})$ or $\text{MINCSP}(\{\bar{x} \vee \bar{y} \vee z, x, \bar{x}\})$ has a constant-factor FP-approximation, then $\text{FPT} = \text{W[P]}$.*

Proof. We prove that there is a parameter preserving polynomial-time reduction from **MONOTONE CIRCUIT SATISFIABILITY** to $\text{MINCSP}^*(\{x \vee y \vee \bar{z}, x, \bar{x}\})$. This is sufficient by Corollary 33. Let C be an MCS instance. We produce an instance I of MINCSP^* as follows. For each node of C , we introduce a new variable into I , and we let f denote the natural bijection from the nodes of C to the variables of the instance I .

We add constraints to simulate the computation of each AND node of C as follows. Observe first that the implication relation $x \rightarrow y$ can be expressed as $y \vee y \vee \bar{x}$. For each AND node G_\wedge such that G_1 and G_2 are the nodes feeding into G_\wedge , 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 $y \rightarrow x_1$ and $y \rightarrow x_2$ 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 the computation of each OR node of C as follows. For each OR node G_\vee such that G_1 and G_2 are the nodes feeding into G_\vee , we add

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, then y is forced to have value 0. (Otherwise y can take on either value 0 or 1, but again, this difference between an OR function 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 node. We define all constraints that appeared until now to be undeletable (recall that MINCSP^* allows undeletable constraints). To finish the construction, for each variable x such that $x = f(G)$ where G is an input node, we add a constraint $x = 0$ to I . We call these constraints *input constraints*. Note that only input constraints can be deleted in I .

If there is a satisfying assignment φ_C of C of weight k , then we delete the input constraints $x = 0$ of I such that $\varphi_C(G) = 1$, where G is a node such that $f(G) = x$. 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. We repeatedly apply the following two modifications of φ_I as long as possible. Let x_1, x_2 , and y be variables such that $f^{-1}(x_1)$ and $f^{-1}(x_2)$ are in-neighbors of node $f^{-1}(y)$.

1. If $f^{-1}(y)$ is an AND node, $\varphi_I(x_1) = 1, \varphi_I(x_2) = 1$, and $\varphi_I(y) = 0$, then we change $\varphi_I(y)$ to 1.
2. If $f^{-1}(y)$ is an OR node, $1 \in \{\varphi_I(x_1), \varphi_I(x_2)\}$, and $\varphi_I(y) = 0$, then we change $\varphi_I(y)$ to 1.

It follows from the definition of the constraints we introduced for AND and OR nodes 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 $\text{MINCSP}(\{\bar{x} \vee \bar{y} \vee z, x, \bar{x}\})$, we note that there is a parameter preserving bijection between instances of $\text{MINCSP}(\{\bar{x} \vee \bar{y} \vee z, x, \bar{x}\})$ and $\text{MINCSP}(\{x \vee y \vee \bar{z}, x, \bar{x}\})$: 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 12 implies hardness if $\langle \Gamma \rangle$ contains IV_2 or IE_2 .

► **Corollary 35.** *If Γ is a (finite) constraint language with $\text{IV}_2 \subseteq \langle \Gamma \rangle$ or $\text{IE}_2 \subseteq \langle \Gamma \rangle$, then $\text{MINCSP}(\Gamma)$ is not FP-approximable, unless $\text{FPT} = \text{W[P]}$.*

Finally, we consider the co-clone IN_2 .

► **Lemma 36.** *If Γ is a (finite) constraint language with $\text{IN}_2 \subseteq \langle \Gamma \rangle$ then $\text{MINCSP}(\Gamma)$ is not FP-approximable, unless $\text{P} = \text{NP}$.*

Proof. Let $\text{NAE}^3 = \{0, 1\}^3 \setminus \{(0, 0, 0), (1, 1, 1)\}$. From Table 1, we see that NAE^3 is a base for the co-clone IN_2 of all self-dual languages. If there was a constant-factor FPA algorithm for $\text{MINCSP}(\text{NAE}^3)$, then setting the parameter to 0 would give a polynomial time decision algorithm for $\text{CSP}(\text{NAE}^3)$. But $\text{CSP}(\text{NAE}^3)$ is NP-complete [43], so there is no constant-factor FPA algorithm for $\text{MINCSP}(\text{NAE}^3)$ unless $\text{NP} = \text{P}$. ◀

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. URL: <http://dx.doi.org/10.1006/jcss.1997.1472>, 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 Andrei A. Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *J. ACM*, 53(1):66–120, 2006. URL: <http://doi.acm.org/10.1145/1120582.1120584>, doi:10.1145/1120582.1120584.
- 7 Andrei A. Bulatov. Complexity of conservative constraint satisfaction problems. *ACM Trans. Comput. Log.*, 12(4):24, 2011. URL: <http://doi.acm.org/10.1145/1970398.1970400>, doi:10.1145/1970398.1970400.
- 8 Andrei A. Bulatov. The complexity of the counting constraint satisfaction problem. *J. ACM*, 60(5):34, 2013. URL: <http://doi.acm.org/10.1145/2528400>, doi:10.1145/2528400.
- 9 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. URL: <http://dx.doi.org/10.1016/j.jcss.2011.12.002>, doi: 10.1016/j.jcss.2011.12.002.
- 10 Andrei A. Bulatov and Dániel Marx. The complexity of global cardinality constraints. *Logical Methods in Computer Science*, 6(4), 2010. URL: [http://dx.doi.org/10.2168/LMCS-6\(4:4\)2010](http://dx.doi.org/10.2168/LMCS-6(4:4)2010), doi:10.2168/LMCS-6(4:4)2010.
- 11 Andrei A. Bulatov and Dániel Marx. Constraint satisfaction parameterized by solution size. *SIAM J. Comput.*, 43(2):573–616, 2014. URL: <http://dx.doi.org/10.1137/120882160>, doi:10.1137/120882160.
- 12 Liming Cai and Xiuzhen Huang. Fixed-parameter approximation: Conceptual framework and approximability results. *Algorithmica*, 57(2):398–412, 2010. URL: <http://dx.doi.org/10.1007/s00453-008-9223-x>, doi:10.1007/s00453-008-9223-x.
- 13 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. URL: <http://dx.doi.org/10.1007/s00037-006-0210-9>, doi: 10.1007/s00037-006-0210-9.
- 14 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.
- 15 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>.
- 16 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. URL: http://dx.doi.org/10.1007/978-3-319-03898-8_11, doi:10.1007/978-3-319-03898-8_11.
- 17 Nadia Creignou, Sanjeev Khanna, and Madhu Sudan. *Complexity Classifications of Boolean Constraint Satisfaction Problems*. Society for Industrial and Applied Mathematics, 2001.

- 18 Nadia Creignou and Heribert Vollmer. Parameterized complexity of weighted satisfiability problems: Decision, enumeration, counting. *Fundam. Inform.*, 136(4):297–316, 2015.
- 19 Pierluigi Crescenzi and Alessandro Panconesi. Completeness in approximation classes. *Inf. Comput.*, 93(2):241–262, 1991. URL: [http://dx.doi.org/10.1016/0890-5401\(91\)90025-W](http://dx.doi.org/10.1016/0890-5401(91)90025-W), doi:10.1016/0890-5401(91)90025-W.
- 20 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. URL: <http://dx.doi.org/10.1007/s00224-012-9415-2>, doi:10.1007/s00224-012-9415-2.
- 21 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. URL: <http://dx.doi.org/10.1007/978-3-319-21275-3>, doi:10.1007/978-3-319-21275-3.
- 22 Víctor Dalmau and Andrei A. Krokhin. Robust satisfiability for csps: Hardness and algorithmic results. *TOCT*, 5(4):15, 2013. URL: <http://doi.acm.org/10.1145/2540090>, doi:10.1145/2540090.
- 23 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. URL: <http://dx.doi.org/10.1137/1.9781611973730.58>, doi:10.1137/1.9781611973730.58.
- 24 Vladimir G. Deineko, Peter Jonsson, Mikael Klasson, and Andrei A. Krokhin. The approximability of MAXCSP with fixed-value constraints. *J. ACM*, 55(4), 2008. URL: <http://doi.acm.org/10.1145/1391289.1391290>, doi:10.1145/1391289.1391290.
- 25 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. URL: <http://dx.doi.org/10.1007/978-1-4471-5559-1>, doi:10.1007/978-1-4471-5559-1.
- 26 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. URL: <http://dx.doi.org/10.1016/j.ipl.2008.09.017>, doi:10.1016/j.ipl.2008.09.017.
- 27 J. Flum and M. Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, Berlin, 2006.
- 28 D. Geiger. Closed systems of functions and predicates. *Pacific Journal of Mathematics*, 27:95–100, 1968.
- 29 Peter Jonsson, Mikael Klasson, and Andrei A. Krokhin. The approximability of three-valued MAXCSP. *SIAM J. Comput.*, 35(6):1329–1349, 2006. URL: <http://dx.doi.org/10.1137/S009753970444644X>, doi:10.1137/S009753970444644X.
- 30 Sanjeev Khanna, Madhu Sudan, Luca Trevisan, and David P. Williamson. The approximability of constraint satisfaction problems. *SIAM J. Comput.*, 30(6):1863–1920, 2000. URL: <http://dx.doi.org/10.1137/S0097539799349948>, doi:10.1137/S0097539799349948.
- 31 Vladimir Kolmogorov and Stanislav Zivny. The complexity of conservative valued CSPs. *J. ACM*, 60(2):10, 2013. URL: <http://doi.acm.org/10.1145/2450142.2450146>, doi:10.1145/2450142.2450146.
- 32 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. URL: http://dx.doi.org/10.1007/978-3-642-15155-2_43, doi:10.1007/978-3-642-15155-2_43.

- 33 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. URL: http://dx.doi.org/10.1007/978-3-642-14165-2_55, doi:10.1007/978-3-642-14165-2_55.
- 34 Bingkai Lin. The parameterized complexity of k -biclique. In Piotr Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 605–615. SIAM, 2015. URL: <http://dx.doi.org/10.1137/1.9781611973730.41>, doi:10.1137/1.9781611973730.41.
- 35 Daniel Lokshantov, 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. URL: <http://doi.acm.org/10.1145/2566616>, doi:10.1145/2566616.
- 36 Dániel Marx. Parameterized complexity of constraint satisfaction problems. *Computational Complexity*, 14(2):153–183, 2005. URL: <http://dx.doi.org/10.1007/s00037-005-0195-9>, doi:10.1007/s00037-005-0195-9.
- 37 Dániel Marx. Parameterized complexity and approximation algorithms. *Comput. J.*, 51(1):60–78, 2008.
- 38 Dániel Marx. Completely inapproximable monotone and antimonotone parameterized problems. *J. Comput. Syst. Sci.*, 79(1):144–151, 2013.
- 39 Dániel Marx and Igor Razgon. Constant ratio fixed-parameter approximation of the edge multicut problem. *Inf. Process. Lett.*, 109(20):1161–1166, 2009. URL: <http://dx.doi.org/10.1016/j.ipl.2009.07.016>, doi:10.1016/j.ipl.2009.07.016.
- 40 R. Pöschel and Kalužnin. *Funktionen- und Relationenalgebren*. Deutscher Verlag der Wissenschaften, Berlin, 1979.
- 41 Emil L. Post. *On the Two-Valued Iterative Systems of Mathematical Logic*. Princeton University Press, 1941.
- 42 Igor Razgon and Barry O’Sullivan. Almost 2-SAT is fixed-parameter tractable. *J. Comput. Syst. Sci.*, 75(8):435–450, 2009.
- 43 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.
- 44 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. URL: <http://doi.acm.org/10.1145/2488608.2488697>, doi:10.1145/2488608.2488697.