



Minimal Distance of Propositional Models

Mike Behrisch¹ · Miki Hermann² · Stefan Mengel³ · Gernot Salzer¹

Published online: 20 December 2018
© The Author(s) 2018

Abstract

We investigate the complexity of three optimization problems in Boolean propositional logic related to information theory: Given a conjunctive formula over a set of relations, find a satisfying assignment with minimal Hamming distance to a given assignment that satisfies the formula (NearestOtherSolution, NOSol) or that does not need to satisfy it (NearestSolution, NSol). The third problem asks for two satisfying assignments with a minimal Hamming distance among all such assignments (MinSolutionDistance, MSD). For all three problems we give complete classifications with respect to the relations admitted in the formula. We give polynomial time algorithms for several classes of constraint languages. For all other cases we prove hardness or completeness regarding APX, poly-APX, or equivalence to well-known hard optimization problems.

Keywords Constraint satisfaction problem · Hamming distance · Optimization problems · Approximation.

This paper is the final version of a trilogy of conference papers *Give Me Another One!* [6] (ISAAC 2015), *As Close As It Gets* [7] (WALCOM 2016), and *The Next Whisky Bar* [8] (CSR 2016).

✉ Gernot Salzer
salzer@logic.at

Mike Behrisch
behrisch@logic.at

Miki Hermann
hermann@lix.polytechnique.fr

Stefan Mengel
mengel@cril.fr

¹ Technische Universität Wien, Vienna, Austria

² LIX (UMR CNRS 7161), École Polytechnique, Palaiseau, France

³ CRIL (UMR CNRS 8188), Université d'Artois, Lens, France

1 Introduction

We investigate the solution spaces of Boolean constraint satisfaction problems built from atomic constraints by means of conjunction and variable identification. We study three minimization problems in connection with Hamming distance: Given an instance of a constraint satisfaction problem in the form of a generalized conjunctive formula over a set of atomic constraints, the first problem asks to find a satisfying assignment with minimal Hamming distance to a given assignment (NearestSolution, NSol). Note that for this problem we assume neither that the given assignment satisfies the formula nor that the solution is different from the assignment. The second problem is similar to the first one, but this time the given assignment has to satisfy the formula and we look for another solution with minimal Hamming distance (NearestOtherSolution, NOSol). The third problem is to find two satisfying assignments with minimal Hamming distance among all satisfying assignments (MinSolutionDistance, MSD). Note that the dual problem MaxHammingDistance has been studied in [14].

The NSol problem appears in several guises throughout literature. E.g., a common problem in Artificial Intelligence is to find solutions of constraints close to an initial configuration; our problem is an abstraction of this setting for the Boolean domain. Bailleux and Marquis [4] describe such applications in detail and introduce the decision problem DistanceSAT: Given a propositional formula φ , a partial interpretation I , and a bound k , is there a satisfying assignment differing from I in no more than k variables? It is straightforward to show that DistanceSAT corresponds to the decision variant of our problem with existential quantification (called NSol_{pp}^d later on). While [4] investigates the complexity of DistanceSAT for a few relevant classes of formulas and empirically evaluates two algorithms, we analyze the decision and the optimization problem for arbitrary semantic restrictions on the formulas.

Hamming distance also plays an important role in belief revision. The result of revising/updating a formula φ by another formula ψ is characterized by the set of models of ψ that are closest to the models of φ . Dalal [15] selects the models of ψ having a minimal Hamming distance to models of φ to be the models that result from the change.

As is common, we analyze the complexity of our optimization problems modulo a parameter that specifies the atomic constraints allowed to occur in the constraint satisfaction problem. We give a complete classification of the approximation complexity with respect to this parameterization. It turns out that our problems can either be solved in polynomial time, or they are complete for a well-known optimization class, or else they are equivalent to well-known hard optimization problems.

Our study can be understood as a continuation of the minimization problems investigated by Khanna et al. in [22], especially that of MinOnes. The MinOnes optimization problem asks for a solution of a constraint satisfaction problem with the minimal Hamming weight, i.e., minimal Hamming distance to the 0-vector. Our work generalizes these results by allowing the given vector to be also different from zero.

Our work can also be seen as a generalization of questions in coding theory. In fact, our problem MSD restricted to affine relations is the well-known problem MinDistance of computing the minimum distance of a linear code. This quantity is of central

importance in coding theory, because it determines the number of errors that the code can detect and correct. Moreover, our problem NSol restricted to affine relations is the problem NearestCodeword of finding the nearest codeword to a given word, which is the basic operation when decoding messages received through a noisy channel. Thus our work can be seen as a generalization of these well-known problems from affine to general relations.

In the case of NearestSolution we are able to apply methods from clone theory, even though the problem turns out to be more intricate than pure satisfiability. The other two problems, however, cannot be shown to be compatible with existential quantification easily, which makes classical clone theory inapplicable. Therefore we have to resort to weak co-clones that require only closure under conjunction and equality. In this connection, we apply the theory developed in [28, 29] as well as the minimal weak bases of Boolean co-clones from [23].

This paper is structured as follows. Section 2 recalls basic definitions and notions. Section 3 introduces the trilogy of optimization problems studied in this paper, namely Nearest Solution (denoted by NSol), Nearest Other Solution (denoted by NOSol), and Minimum Solution Distance (denoted by MSD), as well as their decision versions. It also states our three main results, i.e., a complete classification of complexity for these optimization problems, depicted in Figs. 1, 2, and 3. Section 4 investigates the (non-)applicability of clone theory to our problems. It also provides a duality result for the constraint languages used as parameters. Section 5 contains the proofs of complexity classification results for NearestSolution, Section 6 for NearestOtherSolution, and Section 7 for MinSolutionDistance. Finally, the concluding remarks in Section 8 compare our theorems to previously existing similar results and put our results into perspective.

2 Preliminaries

2.1 Boolean Relations and Relational Clones

An n -ary *Boolean relation* R is a subset of $\{0, 1\}^n$; its elements (b_1, \dots, b_n) are also written as $b_1 \cdots b_n$. Let V be a set of variables. An *atomic constraint*, or an *atom*, is an expression $R(x)$, where R is an n -ary relation and x is an n -tuple of variables from V . Let Γ be a non-empty finite set of Boolean relations, also called a *constraint language*. A (conjunctive) Γ -*formula* is a finite conjunction of atoms $R_1(x_1) \wedge \cdots \wedge R_k(x_k)$, where the R_i are relations from Γ and the x_i are variable tuples of suitable arity. For technical reasons in connection with reductions we also allow empty conjunctions ($k = 0$) here. Such formulas elegantly take care of certain marginal cases at the cost of adding only one additional trivial problem instance.

An *assignment* is a mapping $m: V \rightarrow \{0, 1\}$ assigning a Boolean value $m(x)$ to each variable $x \in V$. In a given context we can assume V to be finite, by restricting it e.g. to the variables occurring in a formula. If we impose an arbitrary but fixed order on the variables, say x_1, \dots, x_n , then the assignments can be identified with elements from $\{0, 1\}^n$. The i -th component of a tuple $m \in \{0, 1\}^n$ is denoted by $m[i]$ and corresponds to the value of the i -th variable, i.e., $m[i] = m(x_i)$. The

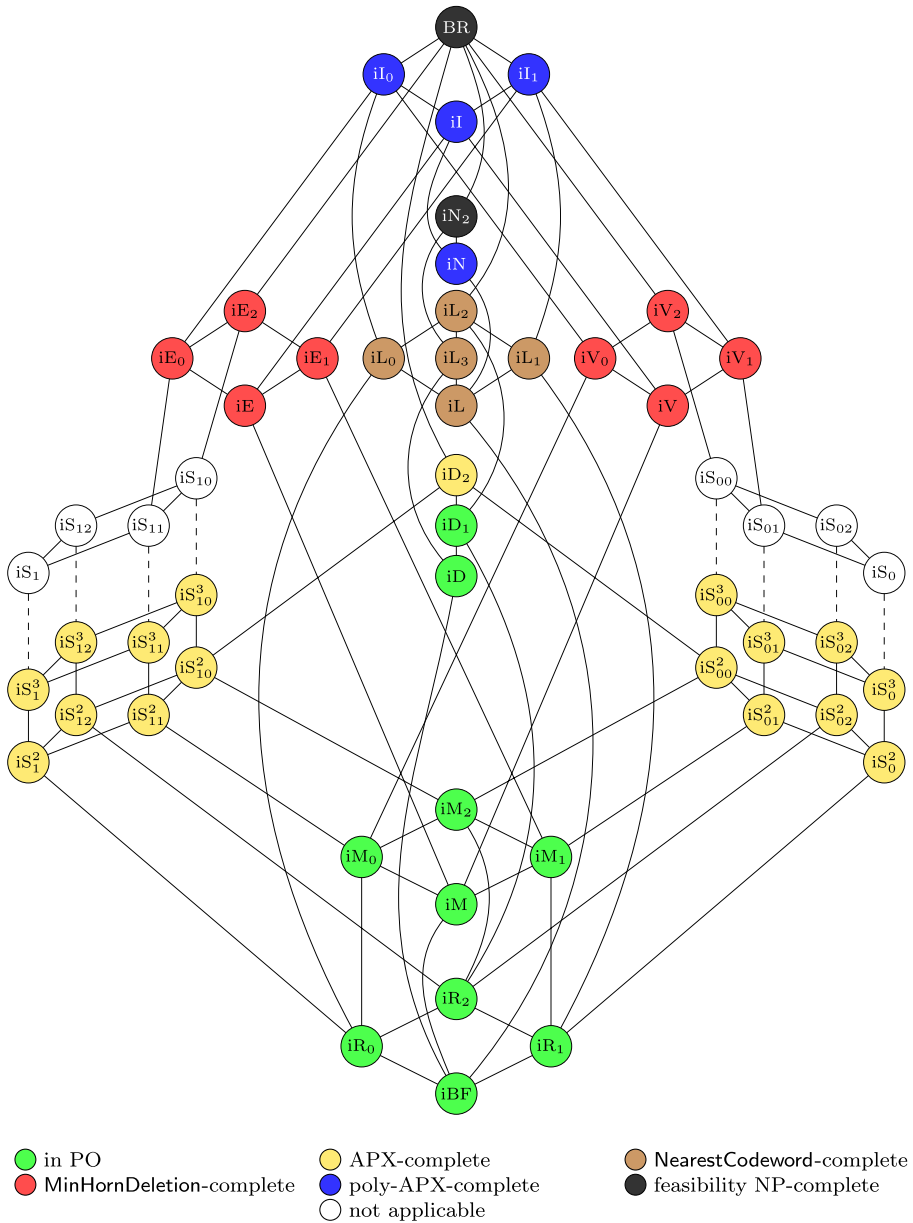


Fig. 1 Lattice of co-clones with complexity classification for NSol

Hamming weight $hw(m) = |\{i|m[i] = 1\}|$ of m is the number of 1s in the tuple m . The *Hamming distance* $hd(m, m') = |\{i|m[i] \neq m'[i]\}|$ of m and m' is the number of coordinates on which the tuples disagree. The complement \bar{m} of a tuple m is its pointwise complement, $\bar{m}[i] = 1 - m[i]$.

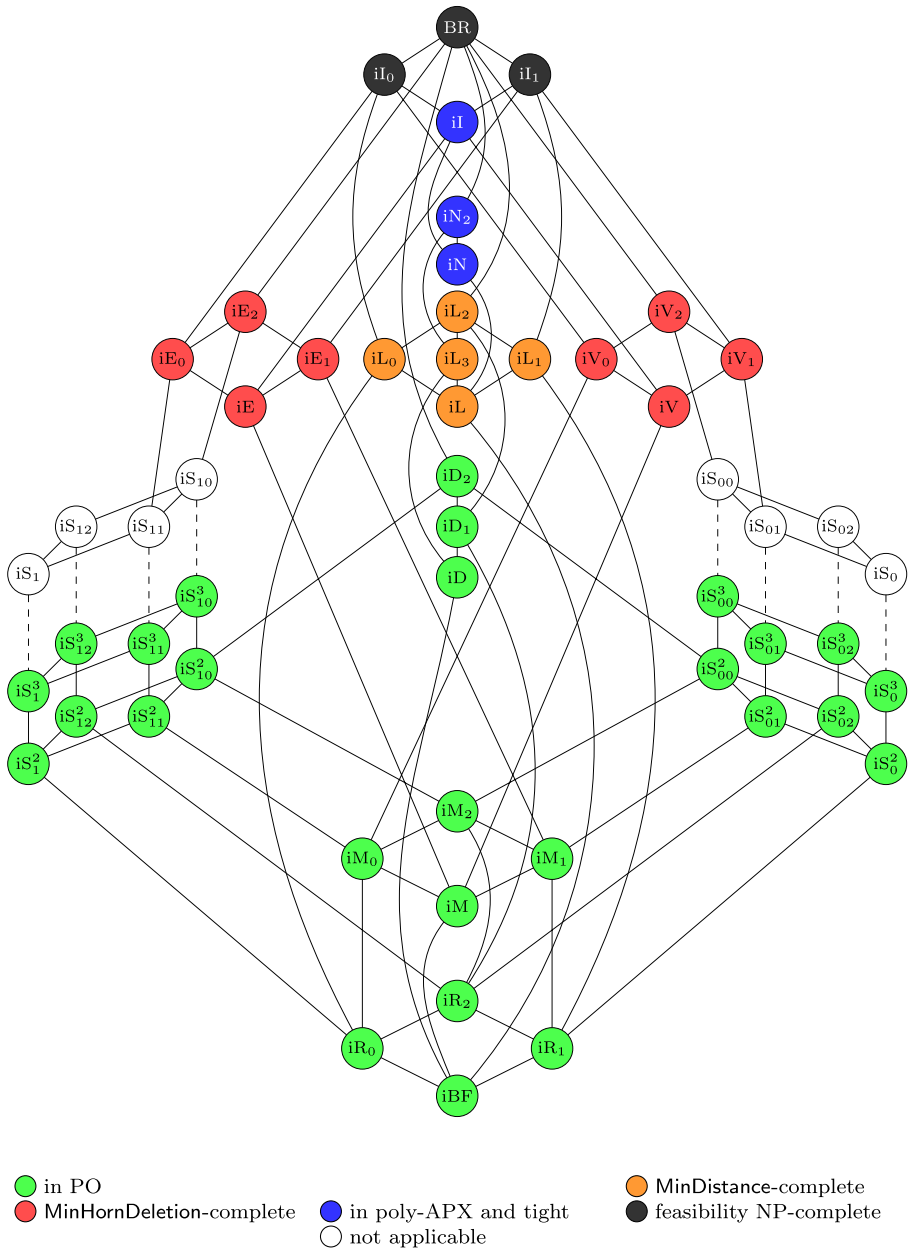


Fig. 2 Lattice of co-clones with complexity classification for NOSol

An assignment m satisfies a constraint $R(x_1, \dots, x_n)$ if $(m(x_1), \dots, m(x_n)) \in R$ holds. It satisfies the formula φ if it satisfies all its atoms; m is said to be a *model* or *solution* of φ in this case. We use $[\varphi]$ to denote the set of models of φ . For a term t ,

relations represented this way we usually omit the brackets. A *literal* is a variable v , or its negation $\neg v$. Assignments are extended to literals by defining $m(\neg v) = 1 - m(v)$.

Table 1 defines Boolean functions and relations needed later on, in particular exclusive or $[x \oplus y]$, not-all-equal nae^3 , k -ary disjunction or^k , and k -ary negated conjunction nand^k .

Throughout the text we refer to different types of Boolean constraint relations following Schaefer’s terminology [27] (see also the monograph [11] and the survey [9]). A Boolean relation R is (1) *1-valid* if $1 \cdots 1 \in R$ and *0-valid* if $0 \cdots 0 \in R$, (2) *Horn (dual Horn)* if R can be represented by a formula in conjunctive normal form (CNF) with at most one unnegated (negated) variable per clause, (3) *monotone* if it is both Horn and dual Horn, (4) *bijunctive* if it can be represented by a CNF formula with at most two literals per clause, (5) *affine* if it can be represented by an affine system of equations $Ax = b$ over \mathbb{Z}_2 , (6) *complementive* if for each $m \in R$ also $\bar{m} \in R$, (7) *implicative hitting set-bounded+* with bound k (denoted by $k\text{-IHS-B}^+$) if R can be represented by a CNF formula with clauses of the form $(x_1 \vee \cdots \vee x_k)$, $(\neg x \vee y)$, x , and $\neg x$, (8) *implicative hitting set-bounded-* with bound k (denoted by $k\text{-IHS-B}^-$) if R can be represented by a CNF formula with clauses of the form $(\neg x_1 \vee \cdots \vee \neg x_k)$, $(\neg x \vee y)$, x , and $\neg x$. A set Γ of Boolean relations is called *0-valid (1-valid, Horn, dual Horn, monotone, affine, bijunctive, complementive, $k\text{-IHS-B}^+$, $k\text{-IHS-B}^-$)* if every relation in Γ is 0-valid (1-valid, Horn, dual Horn, monotone, affine, bijunctive, complementive, $k\text{-IHS-B}^+$, $k\text{-IHS-B}^-$).

A formula constructed from atoms by conjunction, variable identification, and existential quantification is called a *primitive positive formula (pp-formula)*. If φ is such a formula, we write again $[\varphi]$ for its set of models, i.e., the Boolean relation defined by φ . As above the coordinates of this relation are understood to be the variables of φ in lexicographic order, unless otherwise stated by explicit enumeration. We denote by $\langle \Gamma \rangle$ the set of all relations that can be expressed using relations from $\Gamma \cup \{\approx\}$, conjunction, variable identification (and permutation), cylindrification, and existential quantification, i.e., the set of all relations that are primitive positively definable from Γ and equality. The set $\langle \Gamma \rangle$ is called the *co-clone* generated by Γ . A *base* of a co-clone \mathcal{B} is a set of relations Γ such that $\langle \Gamma \rangle = \mathcal{B}$, i.e., just a generating set with regard to primitive positive definability including equality. Note that traditionally (e.g. [18]), the notion of *base* also involves minimality with respect to set inclusion. Our use of the term *base* is in accordance with [10], where finite bases

Table 1 List of some Boolean functions and relations

$x \oplus y = x + y \pmod{2}$	$\text{or}^k = \{0, 1\}^k \setminus \{0 \cdots 0\}$ for $k \geq 1$
$\approx = \{00, 11\}$	$\text{nand}^k = \{0, 1\}^k \setminus \{1 \cdots 1\}$ for $k \geq 1$
$\text{dup}^3 = \{0, 1\}^3 \setminus \{010, 101\}$	$\text{even}^k = \{(a_1, \dots, a_k) \in \{0, 1\}^k \mid \sum_{i=1}^k a_i \text{ even}\}$
$\text{nae}^3 = \{0, 1\}^3 \setminus \{000, 111\}$	$\text{odd}^k = \{(a_1, \dots, a_k) \in \{0, 1\}^k \mid \sum_{i=1}^k a_i \text{ odd}\}$
$S_0 = [(x_1 \wedge x_4) \approx (x_2 \wedge x_3)]$	$S_1 = [S_0(\neg x_1, \neg x_2, \neg x_3, \neg x_1)]$
$S_2 = [(\neg x_1 \vee \neg x_2) \rightarrow \neg x_3]$	
$\text{even}_{k \neq}^k = \{(a_1, \dots, a_{2k}) \in \{0, 1\}^{2k} \mid \text{even}^k(a_1, \dots, a_k) \wedge \bigwedge_{i=1}^k (a_{k+i} \approx \neg a_i)\}$	

Table 2 Some relevant Boolean co-clones with bases

iS_0^k {or ^k }	iL {even ⁴ }	iN {dup ³ }
iS_1^k {nand ^k }	iL ₂ {even ⁴ , ¬x, x}	iN ₂ {nae ³ }
iS_{00}^k {or ^k , x → y, ¬x, x}	iV {x ∨ y ∨ ¬z}	iI {even ⁴ , x → y}
iS_{10}^k {nand ^k , ¬x, x, x → y}	iV ₂ {x ∨ y ∨ ¬z, ¬x, x}	iI ₀ {even ⁴ , x → y, ¬x}
iD ₁ {x ⊕ y, x}	iE {¬x ∨ ¬y ∨ z}	iI ₁ {even ⁴ , x → y, x}
iD ₂ {x ⊕ y, x → y}	iE ₂ {¬x ∨ ¬y ∨ z, ¬x, x}	iM ₂ {x → y, ¬x, x}

for all Boolean co-clones have been determined. Some of these are listed in Table 2. The sets of relations being 0-valid, 1-valid, complementive, Horn, dual Horn, affine, bijunctive, 2affine (both bijunctive and affine), monotone, *k*-IHS-B⁺, and *k*-IHS-B[−] each form a co-clone denoted by iI₀, iI₁, iN₂, iE₂, iV₂, iL₂, iD₂, iD₁, iM₂, iS^k₀₀, and iS^k₁₀, respectively; see Table 3.

We will also use a weaker closure than ⟨Γ⟩, called *conjunctive closure* and denoted by ⟨Γ⟩_∧, where the constraint language Γ is closed under conjunctive definitions, but not under existential quantification or addition of explicit equality constraints.

Sets of relations of the form $W = \langle W \cup \{\approx\} \rangle_{\wedge}$ are called *weak systems* and are in a one-to-one correspondence with so-called strong partial clones [26]. It is a well-known consequence of the Galois theory developed in [26] that for every co-clone ⟨Γ'⟩ whose corresponding clone is finitely generated (this presents no restriction in the Boolean case), there is a largest partial clone whose total part coincides with that clone, cf. [24, Theorem 20.7.2] or see [28, Theorems 4.6, 4.7, 4.11] for a proof in the Boolean case. This largest partial clone even is a strong partial clone, and hence, there is a least weak system *W* under inclusion such that $\langle W \rangle = \langle \Gamma' \rangle$. Any *finite* weak generating set Γ of this weak system *W*, i.e., $W = \langle \Gamma \cup \{\approx\} \rangle_{\wedge}$, is called a *weak base* of ⟨Γ'⟩, see [28, Definition 4.2]. Such a set Γ, in particular, is a finite base of the co-clone ⟨Γ'⟩. Finally, to get from the closure operator $\langle \Gamma \cup \{\approx\} \rangle_{\wedge}$ (which is hard to handle in the context of our problems) to ⟨Γ⟩_∧ (which is easy to handle), one needs the notion of *irredundancy*. A relation *R* is called *irredundant*, if it has neither duplicate nor fictitious coordinates. It can be observed from the proofs of Proposition 5.2 and Corollary 5.6 in [28] or from [29, Proposition 3.11], that $R \in \langle \Gamma \cup \{\approx\} \rangle_{\wedge}$ implies $R \in \langle \Gamma \rangle_{\wedge}$ for any irredundant relation *R*. Following Schnoor [29, p. 30], we call a weak base of ⟨Γ'⟩ consisting exclusively of irredundant relations an *irredundant weak base*. Thus, if Γ is an irredundant weak base of ⟨Γ'⟩, then the

Table 3 Sets of Boolean relations with their names determined by co-clone inclusions

$\Gamma \subseteq iI_0 \Leftrightarrow \Gamma$ is 0-valid	$\Gamma \subseteq iI_1 \Leftrightarrow \Gamma$ is 1-valid
$\Gamma \subseteq iE_2 \Leftrightarrow \Gamma$ is Horn	$\Gamma \subseteq iV_2 \Leftrightarrow \Gamma$ is dual Horn
$\Gamma \subseteq iM_2 \Leftrightarrow \Gamma$ is monotone	$\Gamma \subseteq iD_2 \Leftrightarrow \Gamma$ is bijunctive
$\Gamma \subseteq iL_2 \Leftrightarrow \Gamma$ is affine	$\Gamma \subseteq iD_1 \Leftrightarrow \Gamma$ is 2affine
$\Gamma \subseteq iN_2 \Leftrightarrow \Gamma$ is complementive	$\Gamma \subseteq iI \Leftrightarrow \Gamma$ is both 0- and 1-valid
$\Gamma \subseteq iS_{00}^k \Leftrightarrow \Gamma$ is <i>k</i> -IHS-B ⁺	$\Gamma \subseteq iS_{10}^k \Leftrightarrow \Gamma$ is <i>k</i> -IHS-B [−]

minimality of the weak system $W = \langle \Gamma \cup \{\approx\} \rangle_{\wedge}$ implies that $\Gamma \subseteq W \subseteq \langle \Gamma' \cup \{\approx\} \rangle_{\wedge}$ (cf. [28, Corollary 4.3]), and thus $\Gamma \subseteq \langle \Gamma' \rangle_{\wedge}$ because of irredundancy. Hence, we obtain the following useful tool.

Theorem 1 (Schnoor [29, Corollary 3.12]) *If Γ is an irredundant weak base of a co-clone iC , e.g. a minimal weak base of iC , then $\Gamma \subseteq \langle \Gamma' \rangle_{\wedge}$ holds for any base Γ' of iC .*

According to Lagerkvist [23], a *minimal weak base* is an irredundant weak base satisfying an additional minimality property that ensures small cardinality. The utility of Theorem 1 comes in particular from the fact that Lagerkvist determined minimal weak bases for all finitely generated Boolean co-clones in [23]. For our purposes we note that each of the co-clones iV , iV_0 , iV_1 , iV_2 , iN , iN_2 , and iI is generated by a minimal weak base consisting of a single relation (Table 4).

Another source of weak base relations without duplicate coordinates comes from the following construction: let χ_n be the 2^n -ary relation that is given by the value tables (in some chosen enumeration) of the n distinct n -ary projection functions. More formally, let $\beta: 2^n \rightarrow \{0, 1\}^n$ be the reader’s preferred bijection between the index set $2^n = \{0, \dots, 2^n - 1\}$ and the set of all arguments of an n -ary Boolean function—often lexicographic enumeration is chosen here for presentational purposes, but the order of enumeration of the n -tuples does not matter as long as it remains fixed. Then $\chi_n = \{e_i \circ \beta \mid 1 \leq i \leq n\}$ where $e_i: \{0, 1\}^n \rightarrow \{0, 1\}$ denotes the projection function onto the i -th coordinate. Let C be a clone with corresponding co-clone iC . Since iC is closed with respect to intersection of relations of identical arity, for any k -ary relation R , there is a least k -ary relation in iC containing R , scilicet $C \circ \langle R \rangle := \bigcap \{R' \in iC \mid R' \supseteq R, R' \text{ } k\text{-ary}\}$. Traditionally, e.g. [24, Sect. 2.8, p. 134] or [25, Definition 1.1.16, p. 48], this relation is denoted by $\Gamma_C(R)$, but here we have chosen a different notation to avoid confusion with constraint languages. It is well known, e.g. [25, Satz 1.1.19(i), p. 50], and easy to see that $C \circ \langle R \rangle$ is completely determined by the ℓ -ary part of C whenever $\ell \geq |R|$: given any enumeration of $\emptyset \neq R = \{r_1, \dots, r_\ell\}$ (for technical reasons we have to exclude the case $\ell = 0$ in this presentation because we do not consider clones with nullary operations here) we have $C \circ \langle R \rangle = \{f \circ (r_1, \dots, r_\ell) \mid f \in C, f \text{ } \ell\text{-ary}\}$, where $f \circ (r_1, \dots, r_\ell)$ denotes the row-wise application of f to a matrix whose columns are formed by the tuples r_1, \dots, r_ℓ . Relations of the form $C \circ \langle \chi_n \rangle$ represent the n -ary part of the clone C

Table 4 Minimal weak bases for some co-clones

$R_{iI} = \text{even}^4$	$R_{iV} = (S_1 \times \{0, 1\}) \cap (\{0, 1\} \times S_2)$
$R_{iI_0} = \text{even}^3 \times \{0\}$	$R_{iV_0} = S_1 \times \{0\}$
$R_{iI_1} = \text{odd}^3 \times \{1\}$	$R_{iV_1} = R_{iV} \times \{1\}$
$R_{iI_2} = \text{even}^3_{3 \neq} \times \{0\} \times \{1\}$	$R_{iV_2} = S_1 \times \{0\} \times \{1\}$
$R_{iI_3} = \text{even}^4_{4 \neq}$	$R_{iN_2} = [R_{iN}(x_1, \dots, x_4) \wedge \bigwedge_{i=1}^4 x_{i+4} \approx \neg x_i]$
$R_{iN} = \text{even}^4 \cap S_0$	$R_{iI} = [S_1(\neg x_1, \neg x_2, \neg x_3) \wedge S_1(x_4, x_2, x_3)]$

as a 2^n -ary relation and are called n -th *graphic* of C (cf. e.g. [24, p. 133 and Theorem 2.8.1(b)]). Indeed, the previous characterization of $C \circ \langle \chi_n \rangle$ yields $C \circ \langle \chi_n \rangle = \{f \circ (e_1 \circ \beta, \dots, e_n \circ \beta) \mid f \in C, f \text{ } n\text{-ary}\} = \{f \circ (e_1, \dots, e_n) \circ \beta \mid f \in C, f \text{ } n\text{-ary}\} = \{f \circ \beta \mid f \in C, f \text{ } n\text{-ary}\}$. With the help of this description of $C \circ \langle \chi_n \rangle$ and standard clone theoretic manipulations, one can easily verify the following result, identifying possible candidates for irredundant singleton weak bases.

Theorem 2 ([28, Theorem 4.11]) *Let C be a clone and $R = C \circ \langle \{r_1, \dots, r_n\} \rangle$ with $n \geq 1$, then $C \circ \langle \chi_n \rangle$ gives a singleton weak base of $\langle \{R\} \rangle$ without duplicate coordinates.*

2.2 Approximability, Reductions, and Completeness

We assume that the reader has a basic knowledge of approximation algorithms and complexity theory. We recall some basic notions of approximation algorithms and complexity theory; for details see the monographs [3, 11].

A combinatorial optimization problem \mathcal{P} is a quadruple $(I, \text{sol}, \text{obj}, \text{goal})$, where:

- I is the set of admissible instances of \mathcal{P} .
- $\text{sol}(x)$ denotes the set of feasible solutions for every instance $x \in I$.
- $\text{obj}(x, y)$ denotes the non-negative integer measure of y for every instance $x \in I$ and every feasible solution $y \in \text{sol}(x)$; obj is also called *objective function*.
- $\text{goal} \in \{\min, \max\}$ denotes the *optimization goal* for \mathcal{P} .

A combinatorial optimization problem is said to be an NP-*optimization problem* (NPO-problem) if

- the instances and solutions are recognizable in polynomial time,
- the size of the solutions in $\text{sol}(x)$ is polynomially bounded in the size of x , and
- the objective function obj is computable in polynomial time.

The optimal value of the objective function for the solutions of an instance x is denoted by $\text{OPT}(x)$. In our case the optimization goal will always be minimization, i.e., $\text{OPT}(x)$ will be the minimum.

Given an instance $x \in I$ with a feasible solution $y \in \text{sol}(x)$ and a real number $r \geq 1$, we say that y is r -*approximate* if $\text{obj}(x, y) \leq r \text{OPT}(x)$ holds and our goal is minimization, or $\text{obj}(x, y) \geq \text{OPT}(x)/r$ and we consider a maximization problem.

Let A be an algorithm that for any instance x of \mathcal{P} such that $\text{sol}(x) \neq \emptyset$ returns a feasible solution $A(x) \in \text{sol}(x)$. Given an arbitrary function $r: \mathbb{N} \rightarrow [1, \infty)$, we say that A is an $r(n)$ -approximate algorithm for \mathcal{P} if for any instance $x \in I$ having feasible solutions the algorithm returns an $r(|x|)$ -approximate solution, where $|x|$ is the size of x . If an NPO problem \mathcal{P} admits an $r(n)$ -approximate polynomial-time algorithm, we say that \mathcal{P} is approximable within $r(n)$.

An NPO problem \mathcal{P} is in the class PO if the optimum is computable in polynomial time (i.e. if \mathcal{P} admits a 1-approximate polynomial-time algorithm). \mathcal{P} is in the class APX (poly-APX) if it is approximable within a constant (polynomial) function in the size of the instance x . NPO is the class of all NPO problems and NPOPB is the class of all NPO problems where the objective function is polynomially bounded. The

following inclusions hold for these approximation complexity classes: $\text{PO} \subseteq \text{APX} \subseteq \text{poly-APX} \subseteq \text{NPO}$. All inclusions are strict unless $\text{P} = \text{NP}$.

For reductions among decision problems we use the polynomial-time many-one reduction denoted by \leq_m . Many-one equivalence between decision problems is denoted by \equiv_m . For reductions among optimization problems we use approximation preserving reductions, also called AP-reductions, denoted by \leq_{AP} . AP-equivalence between optimization problems is denoted by \equiv_{AP} .

We say that an optimization problem \mathcal{P} AP-reduces to another optimization problem \mathcal{Q} , denoted $\mathcal{P} \leq_{\text{AP}} \mathcal{Q}$, if there are two polynomial-time computable functions f and g and a real constant $\alpha \geq 1$ such that for all $r > 1$ and all \mathcal{P} -instances x the following conditions hold.

- $f(x)$ is a \mathcal{Q} -instance or the generic unsolvable instance \perp (which is not part of \mathcal{Q}).
- If x admits feasible solutions, then $f(x)$ is different from \perp and also admits feasible solutions.
- For any feasible solution y' of $f(x)$, $g(x, y')$ is a feasible solution of x .
- If y' is an r -approximate solution of the \mathcal{Q} -instance $f(x)$, then $g(x, y')$ is an $(1 + (r - 1)\alpha + o(1))$ -approximate solution of the \mathcal{P} -instance x , where $o(1)$ refers to the size of x .

Our definition of AP-reducibility slightly extends the one in [3] by introducing a generic unsolvable instance \perp . This extension allows us to reduce problems with unsolvable instances to such without as long as the unsolvable instances can be detected in polynomial time, by making f map the unsolvable instances to \perp . This practice has been implicit in previous work, e.g. [22].

We also need a slightly non-standard variation of AP-reductions. We say that an optimization problem \mathcal{P} AP-Turing-reduces to another optimization problem \mathcal{Q} if there is a polynomial-time oracle algorithm A and a constant $\alpha \geq 1$ such that for all $r > 1$ on any input x for \mathcal{P}

- if all oracle calls with a \mathcal{Q} -instance x' are answered with a feasible \mathcal{Q} -solution y for x' , then A outputs a feasible \mathcal{P} -solution for x , and
- if for every call the oracle answers with an r -approximate solution, then A computes a $(1 + (r - 1)\alpha + o(1))$ -approximate solution for the \mathcal{P} -instance x .

It is straightforward to check that AP-Turing-reductions are transitive. Moreover, if \mathcal{P} AP-Turing-reduces to \mathcal{Q} with constant α and \mathcal{Q} has an $r(n)$ -approximation algorithm, then there is an $\alpha r(n)$ -approximation algorithm for \mathcal{P} .

We will relate our problems to well-known optimization problems, by calling the problem \mathcal{P} under investigation \mathcal{Q} -complete if $\mathcal{P} \equiv_{\text{AP}} \mathcal{Q}$. This notion of completeness is stricter than the one in [22], since the latter relies on A-reductions. For \mathcal{Q} , we will consider the following optimization problems analyzed in [22].

Problem MinOnes(Γ)

Input: A conjunctive formula φ over relations from Γ .

Solution: An assignment m satisfying φ .

Objective: Minimum Hamming weight $\text{hw}(m)$.

Problem WeightedMinOnes(Γ)

Input: A conjunctive formula φ over relations from Γ and a weight function $w: V \rightarrow \mathbb{N}$ assigning non-negative integer weights to the variables of φ .

Solution: An assignment m satisfying φ .

Objective: Minimum value $\sum_{x:m(x)=1} w(x)$.

We now define some well-studied problems to which we will relate our problems. Note that these problems do not depend on any parameter.

Problem NearestCodeword

Input: A matrix $A \in \mathbb{Z}_2^{k \times l}$ and a vector $m \in \mathbb{Z}_2^l$.

Solution: A vector $x \in \mathbb{Z}_2^k$.

Objective: Minimum Hamming distance $\text{hd}(xA, m)$.

Problem MinDistance

Input: A matrix $A \in \mathbb{Z}_2^{k \times l}$.

Solution: A non-zero vector $x \in \mathbb{Z}_2^k$ with $Ax = 0$.

Objective: Minimum Hamming weight $\text{hw}(x)$.

Problem MinHornDeletion

Input: A conjunctive formula φ over relations from $\{x \vee y \vee \neg z, x, \neg x\}$.

Solution: An assignment m to φ .

Objective: Minimum number of unsatisfied conjuncts of φ .

NearestCodeword, MinDistance and MinHornDeletion are known to be NP-hard to approximate within a factor $2^{\Omega(\log^{1-\varepsilon}(n))}$ for every $\varepsilon > 0$ [1, 16, 22]. Thus if a problem \mathcal{P} is equivalent to any of these problems, it follows that $\mathcal{P} \notin \text{APX}$ unless $\text{P} = \text{NP}$.

2.3 Satisfiability

We also use the classic problem $\text{SAT}(\Gamma)$ asking for the satisfiability of a given conjunctive formula over a constraint language Γ . Schaefer [27] completely classified its complexity. $\text{SAT}(\Gamma)$ is polynomial-time decidable if Γ is 0-valid ($\Gamma \subseteq \text{iI}_0$), 1-valid ($\Gamma \subseteq \text{iI}_1$), Horn ($\Gamma \subseteq \text{iE}_2$), dual Horn ($\Gamma \subseteq \text{iV}_2$), bijunctive ($\Gamma \subseteq \text{iD}_2$), or affine ($\Gamma \subseteq \text{iL}_2$); otherwise it is NP-complete. Moreover, we need the decision problem $\text{AnotherSAT}(\Gamma)$: Given a conjunctive formula over Γ and a satisfying assignment m , is there another satisfying assignment m' different from m ? The complexity of this problem was completely classified by Juban [20]. $\text{AnotherSAT}(\Gamma)$ is polynomial-time decidable if Γ is both 0- and 1-valid ($\Gamma \subseteq \text{iI}$), complementive ($\Gamma \subseteq \text{iN}_2$), Horn ($\Gamma \subseteq \text{iE}_2$), dual Horn ($\Gamma \subseteq \text{iV}_2$), bijunctive ($\Gamma \subseteq \text{iD}_2$), or affine ($\Gamma \subseteq \text{iL}_2$); otherwise it is NP-complete.

2.4 Linear and Integer Programming

A *unimodular matrix* is a square integer matrix having determinant $+1$ or -1 . A *totally unimodular matrix* is a matrix for which every square non-singular submatrix is unimodular. A totally unimodular matrix need not be square itself. Any totally

unimodular matrix has only 0, +1 or −1 entries. If A is a totally unimodular matrix and \mathbf{b} is an integral vector, then for any given linear functional f such that the linear program $\min\{f(\mathbf{x}) \mid A\mathbf{x} \geq \mathbf{b}\}$ has a real minimum \mathbf{x} , it also has an integral minimum point \mathbf{x} . That is, the feasible region $\{\mathbf{x} \mid A\mathbf{x} \geq \mathbf{b}\}$ is an integral polyhedron. For this reason, linear programming methods can be used to obtain the solutions for integer linear programs in this case. Linear programs can be solved in polynomial time, hence so can integer programs with totally unimodular matrices. For details see the monograph by Schrijver [30].

3 Results

This section presents the problems we consider and our results; the proofs follow in subsequent sections. The input to all our problems is a conjunctive formula over a constraint language. The satisfying assignments of the formula, i.e. its models or solutions, form a Boolean relation that can be understood as an associated generalized binary code. As for linear codes, the minimization target is always the Hamming distance between the codewords or models. Our three problems differ in the information additionally available for computing the required Hamming distance.

Given a formula and an arbitrary assignment, the first problem asks for a solution closest to the given assignment.

Problem NearestSolution(Γ), NSol(Γ)

Input: A conjunctive formula φ over relations from Γ and an assignment m to the variables occurring in φ , which is not required to satisfy φ .

Solution: An assignment m' satisfying φ (i.e. a codeword of the code described by φ).

Objective: Minimum Hamming distance $\text{hd}(m, m')$.

Note that the problem generalizes the MinOnes problem from [22]. Indeed, if we take the all-zero assignment $m = 0 \cdots 0$ as part of the input, we get exactly the MinOnes problem as a special case.

Theorem 3 (illustrated in Fig. 1) *For a given Boolean constraint language Γ the optimization problem NSol(Γ) is*

- (i) *in PO if Γ is*
 - (a) *2affine ($\Gamma \subseteq \text{iD}_1$) or*
 - (b) *monotone ($\Gamma \subseteq \text{iM}_2$);*
- (ii) *APX-complete if*
 - (a) *Γ generates iD_2 ($\langle \Gamma \rangle = \text{iD}_2$), or*
 - (b) *$[x \vee y] \in \langle \Gamma \rangle$ and Γ is k -IHS- B^+ ($\text{iS}_0^2 \subseteq \langle \Gamma \rangle \subseteq \text{iS}_{00}^k$) for some $k \in \mathbb{N}$, $k \geq 2$, or*
 - (c) *$[\neg x \vee \neg y] \in \langle \Gamma \rangle$ and Γ is k -IHS- B^- ($\text{iS}_1^2 \subseteq \langle \Gamma \rangle \subseteq \text{iS}_{10}^k$) for some $k \in \mathbb{N}$, $k \geq 2$;*

- (iii) NearestCodeword-complete if Γ is exactly affine ($iL \subseteq \langle \Gamma \rangle \subseteq iL_2$);
- (iv) MinHornDeletion-complete if Γ is
 - (a) exactly Horn ($iE \subseteq \langle \Gamma \rangle \subseteq iE_2$) or
 - (b) exactly dual Horn ($iV \subseteq \langle \Gamma \rangle \subseteq iV_2$);
- (v) poly-APX-complete if Γ does not contain an affine relation and it is
 - (a) 0-valid ($iN \subseteq \langle \Gamma \rangle \subseteq iI_0$) or
 - (b) 1-valid ($iN \subseteq \langle \Gamma \rangle \subseteq iI_1$); and
- (vi) otherwise ($iN_2 \subseteq \langle \Gamma \rangle$) it is NP-complete to decide whether a feasible solution for $\text{NSol}(\Gamma)$ exists.

Proof The proof is split into several propositions presented in Section 5.

- (i) See Propositions 17 and 18.
- (ii) See Propositions 20, 21, and 22.
- (iii) See Corollary 25 and Proposition 26.
- (iv) See Propositions 29 and 30.
- (v) See Proposition 31.
- (vi) See Proposition 19. □

Given a constraint and one of its solutions, the second problem asks for another solution closest to the given one.

Problem NearestOtherSolution(Γ), $\text{NOSol}(\Gamma)$

Input: A conjunctive formula φ over relations from Γ and a satisfying assignment m (to the variables mentioned in φ).

Solution: An assignment $m' \neq m$ satisfying φ .

Objective: Minimum Hamming distance $\text{hd}(m, m')$.

The difference between the problems NearestSolution and NearestOtherSolution is the knowledge, or its absence, whether the input assignment satisfies the constraint. Moreover, for NearestSolution we may output the given assignment if it satisfies the formula while for NearestOtherSolution we have to output an assignment different from the one given as the input.

Theorem 4 (illustrated in Fig. 2) *For every constraint language Γ the optimization problem $\text{NOSol}(\Gamma)$ is*

- (i) in PO if
 - (a) Γ is bijunctive ($\Gamma \subseteq iD_2$) or
 - (b) Γ is k -IHS- B^+ ($\Gamma \subseteq iS_{00}^k$) for some $k \in \mathbb{N}$, $k \geq 2$ or
 - (c) Γ is k -IHS- B^- ($\Gamma \subseteq iS_{10}^k$) for some $k \in \mathbb{N}$, $k \geq 2$;
- (ii) MinDistance-complete if Γ is exactly affine ($iL \subseteq \langle \Gamma \rangle \subseteq iL_2$);
- (iii) MinHornDeletion-complete under AP-Turing-reductions if Γ is
 - (a) exactly Horn ($iE \subseteq \langle \Gamma \rangle \subseteq iE_2$) or
 - (b) exactly dual Horn ($iV \subseteq \langle \Gamma \rangle \subseteq iV_2$);

- (iv) *in poly-APX if Γ is*
 - (a) *exactly both 0-valid and 1-valid ($\langle \Gamma \rangle = \mathbf{iI}$) or*
 - (b) *exactly complementive ($\mathbf{iN} \subseteq \langle \Gamma \rangle \subseteq \mathbf{iN}_2$),*

where $\text{NOSol}(\Gamma)$ is n -approximable but not $(n^{1-\varepsilon})$ -approximable for any $\varepsilon > 0$ unless $\text{P} = \text{NP}$;
- (v) *and otherwise ($\mathbf{iI}_0 \subseteq \langle \Gamma \rangle$ or $\mathbf{iI}_1 \subseteq \langle \Gamma \rangle$) it is NP-complete to decide whether a feasible solution for $\text{NOSol}(\Gamma)$ exists.*

Proof The proof is split into several propositions presented in Section 6.

- (i) See Propositions 33 and 34.
- (ii) See Proposition 44.
- (iii) See Corollary 47.
- (iv) See Propositions 35 and 39.
- (v) See Proposition 35. □

The third problem does not take any assignments as input, but asks for two solutions which are as close to each other as possible. We optimize once more the Hamming distance between the solutions.

Problem $\text{MinSolutionDistance}(\Gamma)$, $\text{MSD}(\Gamma)$

Input: A conjunctive formula φ over relations from Γ .

Solution: Two satisfying truth assignments $m \neq m'$ to the variables occurring in φ .

Objective: Minimum Hamming distance $\text{hd}(m, m')$.

The $\text{MinSolutionDistance}$ problem enlarges the notion of minimum distance of an error correcting code. The following theorem is a more fine-grained analysis of the result published by Vardy in [31], extended to an optimization problem.

Theorem 5 (illustrated in Fig. 3) *For any constraint language Γ the optimization problem $\text{MSD}(\Gamma)$ is*

- (i) *in PO if Γ is*
 - (a) *bijunctive ($\Gamma \subseteq \mathbf{iD}_2$) or*
 - (b) *Horn ($\Gamma \subseteq \mathbf{iE}_2$) or*
 - (c) *dual Horn ($\Gamma \subseteq \mathbf{iV}_2$);*
- (ii) *MinDistance-complete if Γ is exactly affine ($\mathbf{iL} \subseteq \langle \Gamma \rangle \subseteq \mathbf{iL}_2$);*
- (iii) *in poly-APX if $\text{dup}^3 \in \langle \Gamma \rangle$ and Γ is both 0-valid and 1-valid ($\mathbf{iN} \subseteq \langle \Gamma \rangle \subseteq \mathbf{iI}$), where $\text{MSD}(\Gamma)$ is n -approximable but not $(n^{1-\varepsilon})$ -approximable for any $\varepsilon > 0$ unless $\text{P} = \text{NP}$; and*
- (iv) *otherwise ($\mathbf{iN}_2 \subseteq \langle \Gamma \rangle$ or $\mathbf{iI}_0 \subseteq \langle \Gamma \rangle$ or $\mathbf{iI}_1 \subseteq \langle \Gamma \rangle$) it is NP-complete to decide whether a feasible solution for $\text{MSD}(\Gamma)$ exists.*

Proof The proof is split into several propositions presented in Section 7.

- (i) See Propositions 48 and 49.
- (ii) See Proposition 55.

- (iii) For $\Gamma \subseteq \text{iI}$, every formula φ over Γ has at least two solutions since it is both 0-valid and 1-valid. Thus $\text{TwoSolutionSAT}(\Gamma)$ is in P, and Proposition 54 yields that $\text{MSD}(\Gamma)$ is n -approximable. By Proposition 56 this approximation is indeed tight.
- (iv) According to [20], $\text{AnotherSAT}(\Gamma)$ is NP-hard for $\text{iI}_0 \subseteq \langle \Gamma \rangle$, or $\text{iI}_1 \subseteq \langle \Gamma \rangle$. By Lemma 51 it follows that $\text{TwoSolutionSAT}(\Gamma)$ is NP-hard, too. For $\text{iN}_2 \subseteq \langle \Gamma \rangle$ we can reduce the NP-hard problem $\text{SAT}(\Gamma)$ to $\text{TwoSolutionSAT}(\Gamma)$. Hence it is NP-complete to decide whether a feasible solution for $\text{MSD}(\Gamma)$ exists in all three cases. \square

The three optimization problems can be transformed into decision problems in the usual way. We add an integer bound k to the input and ask if the Hamming distance satisfies the inequality $\text{hd}(m, m') \leq k$. This way we obtain the corresponding decision problems NOSol^d , NSol^d , and MSD^d , respectively. Their complexity follows immediately from the theorems above. All cases in PO become polynomial-time decidable, whereas the other cases, which are APX-hard, become NP-complete. This way we obtain dichotomy theorems classifying the decision problems as polynomial or NP-complete for all sets Γ of relations. We obtain the following dichotomies for each of the respective decision problems.

Corollary 6 *For each constraint language Γ*

- $\text{NSol}^d(\Gamma)$ is in P if Γ is 2affine or monotone, and it is NP-complete otherwise.
- $\text{NOSol}^d(\Gamma)$ is in P if Γ is bijunctive, k -IHS- B^+ , or k -IHS- B^- , and it is NP-complete otherwise.
- $\text{MSD}^d(\Gamma)$ is in P if Γ is bijunctive, Horn, or dual-Horn, and it is NP-complete otherwise.

4 Applicability of Clone Theory and Duality

We show that clone theory is applicable to the problem NSol , as well as a possibility to exploit inner symmetries between co-clones, which shortens several proofs in the following sections.

4.1 Nearest Solution

There are two natural versions of $\text{NSol}(\Gamma)$. In one version the formula φ is quantifier free while in the other one we do allow existential quantification. We call the former version $\text{NSol}(\Gamma)$ and the latter $\text{NSol}_{\text{pp}}^d(\Gamma)$ and show that both versions are equivalent.

Let $\text{NSol}^d(\Gamma)$ and $\text{NSol}_{\text{pp}}^d(\Gamma)$ be the decision problems corresponding to $\text{NSol}(\Gamma)$ and $\text{NSol}_{\text{pp}}(\Gamma)$, asking whether there is a satisfying assignment within a given bound.

Proposition 7 *For any constraint language Γ , we have $\text{NSol}^d(\Gamma) \equiv_m \text{NSol}_{\text{pp}}^d(\Gamma)$ and $\text{NSol}(\Gamma) \equiv_{\text{AP}} \text{NSol}_{\text{pp}}(\Gamma)$.*

Proof The reduction from left to right is trivial in both cases. For the other direction, consider first an instance of $\text{NSol}_{\text{pp}}^{\text{d}}(\Gamma)$ with formula φ , assignment m , and bound k . Let x_1, \dots, x_n be the free variables of φ and let y_1, \dots, y_ℓ be the existentially quantified ones, which we can assume to be disjoint. By discarding variables y_i while not changing $[\varphi]$, we can assume that each variable y_i occurs in at least one atom of φ . We construct a quantifier-free formula φ' , where the non-quantified variables of φ get duplicated by a factor $\lambda := (n + \ell + 1)^2$ such that the effect of quantified variables becomes negligible. For each variable z we define the set $B(z)$ as follows:

$$B(z) = \begin{cases} \{x_i^1, \dots, x_i^\lambda\} & \text{if } z = x_i \text{ for some } i \in \{1, \dots, n\}, \\ \{y_i\} & \text{if } z = y_i \text{ for some } i \in \{1, \dots, \ell\}. \end{cases}$$

For every atom $R(z_1, \dots, z_s)$ in φ , the quantifier-free formula φ' over the variables $\bigcup_{i=1}^n B(x_i) \cup \bigcup_{i=1}^\ell B(y_i)$ contains the atom $R(z'_1, \dots, z'_s)$ for every (z'_1, \dots, z'_s) from $B(z_1) \times \dots \times B(z_s)$. Moreover, we construct an assignment $B(m)$ of φ' by assigning to every variable x_i^j the value $m(x_i)$ and to y_i the value 0. Note that because there is an upper bound on the arities of relations from Γ , this is a polynomial time construction.

We claim that φ has a solution m' with $\text{hd}(m, m') \leq k$ if and only if φ' has a solution m'' with $\text{hd}(B(m), m'') \leq k\lambda + \ell$. First, observe that if m' with the desired properties exists, then there is an extension m'_e of m' to the y_i that satisfies all atoms. Define m'' by setting $m''(x_i^j) := m'(x_i)$ and $m''(y_i) := m'_e(y_i)$ for all i and j . Then m'' is clearly a satisfying assignment of φ' . Moreover, m'' and $B(m)$ differ in at most $k\lambda$ variables among the x_i^j . Since there are only ℓ other variables y_i , we get $\text{hd}(m'', B(m)) \leq k\lambda + \ell$ as desired.

Now suppose m'' satisfies φ' with $\text{hd}(B(m), m'') \leq k\lambda + \ell$. We may assume for each i that $m''(x_i^1) = \dots = m''(x_i^\lambda)$. Indeed, if this is not the case, then setting all x_i^j to $B(m)(x_i^j) = m(x_i)$ will result in a satisfying assignment closer to $B(m)$. After at most n iterations we get some m'' as desired. Now define an assignment m' for φ by setting $m'(x_i) := m''(x_i^1)$. Then m' satisfies φ , because the variables y_i can be assigned values as in m'' . Moreover, whenever $m(x_i)$ differs from $m'(x_i)$, the inequality $B(m)(x_i^j) \neq m''(x_i^j)$ holds for every j . Thus we obtain $\lambda \text{hd}(m, m') \leq \text{hd}(B(m), m'') \leq k\lambda + \ell$. Therefore, we have the inequality $\text{hd}(m, m') \leq k + \ell/\lambda$ and hence $\text{hd}(m, m') \leq k$, since $\ell/\lambda < 1$. This completes the many-one reduction.

To see that the construction above is also an AP-reduction, let m'' be an r -approximation for φ' and $B(m)$, i.e., $\text{hd}(B(m), m'') \leq r \cdot \text{OPT}(\varphi', B(m))$. Construct m' as before, so $\lambda \text{hd}(m, m') \leq \text{hd}(B(m), m'') \leq r \cdot \text{OPT}(\varphi', B(m))$. Since $\text{OPT}(\varphi', B(m)) \leq \lambda \text{OPT}(\varphi, m) + \ell$ as above, we get $\lambda \text{hd}(m, m') \leq r(\lambda \text{OPT}(\varphi, m) + \ell)$. This implies $\text{hd}(m, m') \leq r \cdot \text{OPT}(\varphi, m) + r \cdot \ell/\lambda = (r + o(1)) \cdot \text{OPT}(\varphi, m)$ and shows that the construction is an AP-reduction with $\alpha = 1$. \square

Remark 8 Note that in the reduction from $\text{NSol}_{\text{pp}}^{\text{d}}(\Gamma)$ to $\text{NSol}^{\text{d}}(\Gamma)$ we construct the assignment $B(m)$ as an extension of m by setting all new variables to 0. In particular, if m is the constant 0-assignment, then so is $B(m)$. We use this observation as we continue.

The following four technical results are the missing theoretical backbone of [8], which had to be omitted from [8] due to page limitations. The first of these lemmas allows us to consider constraints with disjoint variables independently.

Lemma 9 *Let $\varphi(x, y) = \psi(x) \wedge \chi(y)$ be a Γ -formula over a constraint language Γ and m an assignment over disjoint variable blocks x and y . Let (φ, m) be an instance of $\text{NSol}(\Gamma)$. Then $\text{OPT}(\varphi, m) = \text{OPT}(\psi, m \upharpoonright_x) + \text{OPT}(\chi, m \upharpoonright_y)$.*

Proof If $s \in [\varphi]$, then $s \upharpoonright_x \in [\psi]$ and $s \upharpoonright_y \in [\chi]$. Conversely, if $s_\psi \in [\psi]$ and $s_\chi \in [\chi]$, then $s := s_\psi \cup s_\chi$ is a model of φ . If $s \in [\varphi]$ is optimal, i.e. $\text{hd}(s, m) = \text{OPT}(\varphi, m)$, then

$$\begin{aligned} \text{OPT}(\varphi, m) = \text{hd}(s, m) &= \text{hd}(s \upharpoonright_x, m \upharpoonright_x) + \text{hd}(s \upharpoonright_y, m \upharpoonright_y) \\ &\geq \text{OPT}(\psi, m \upharpoonright_x) + \text{OPT}(\chi, m \upharpoonright_y). \end{aligned}$$

Conversely, if $s_\psi \in [\psi]$ and $s_\chi \in [\chi]$ are optimal solutions for their respective problems, then $s := s_\psi \cup s_\chi$ satisfies

$$\begin{aligned} \text{OPT}(\varphi, m) \leq \text{hd}(s, m) &= \text{hd}(s \upharpoonright_x, m \upharpoonright_x) + \text{hd}(s \upharpoonright_y, m \upharpoonright_y) \\ &= \text{OPT}(\psi, m \upharpoonright_x) + \text{OPT}(\chi, m \upharpoonright_y). \quad \square \end{aligned}$$

We can also show that introducing explicit equality constraints does not change the complexity of our problem. We need two introductory lemmas. The first one deals with equalities that do not interfere with the other atoms of the given formula.

Lemma 10 *For constraint languages Γ , $\text{NSol}(\Gamma \cup \{\approx\})$ and $\text{NSol}^d(\Gamma \cup \{\approx\})$ reduce to particular cases of the respective problem, where for each constraint $x \approx y$ in the given formula φ at least one of x, y occurs also in some Γ -atom of φ .*

Proof Let (φ, m) be an instance of $\text{NSol}(\Gamma \cup \{\approx\})$. Without loss of generality we assume φ to be of the form $\psi \wedge \varepsilon$, where ψ is a Γ -formula and ε is a $\{\approx\}$ -formula. Let $(V_i)_{i \in I}$ be the unique finest partition of the variables in ε satisfying that variables x, y are in the same partition class if $x \approx y$ occurs in ε .

For each index $i \in I$ we designate a specific variable $x_i \in V_i$. Let ψ' be the formula obtained from ψ by substituting all occurrences of variables $y \in V_i$ by x_i . Moreover, let I' be the set of indices $i \in I$ such that x_i actually occurs in ψ' , and let $I'' := I \setminus I'$ be the set of indices without this property. We set $\varepsilon' := \bigwedge_{i \in I'} \varepsilon_i$ and $\varepsilon'' := \bigwedge_{i \in I''} \varepsilon_i$, where the formula $\varepsilon_i := \bigwedge_{y \in V_i} (x_i \approx y)$ expresses the equivalence of the variables in V_i . Note that the formulas $\psi \wedge \varepsilon$ and $\chi := \psi' \wedge \varepsilon' \wedge \varepsilon''$ contain the same variables and have identical sets of models.

Now consider the formula $\varphi' := \psi' \wedge \varepsilon'$ and the assignment $m' := m \upharpoonright_{V'}$, where V' is the set of variables occurring in φ' . The pair (φ', m') is an $\text{NSol}(\Gamma \cup \{\approx\})$ -instance with the additional properties stated in the lemma. By construction we have $\chi = \varphi' \wedge \varepsilon''$, where the set V' of variables in φ' and the set V'' of variables in ε'' are disjoint. By Lemma 9 we obtain $\text{OPT}(\varphi, m) = \text{OPT}(\chi, m) = \text{OPT}(\varphi', m') + \text{OPT}(\varepsilon'', m \upharpoonright_{V''})$.

An optimal solution $s_{\varepsilon''}$ of ε'' and the optimal value $d := \text{OPT}(\varepsilon'', m \upharpoonright_{V''})$ can obviously be computed in polynomial time. Therefore the instance (φ, m, k) of

$\text{NSol}^d(\Gamma \cup \{\approx\})$ corresponds to the instance $(\varphi', m', k - d)$ of the restricted decision problem in the polynomial-time many-one reduction.

Moreover, if s' is an r -approximate solution of (φ', m') for some $r \geq 1$, then $s := s' \cup s_{\varepsilon''}$ is a solution of φ , and we have

$$\text{hd}(s, m) = \text{hd}(s', m') + d \leq r \text{OPT}(\varphi', m') + d \leq r \text{OPT}(\varphi', m') + rd = r \text{OPT}(\varphi, m),$$

so the constructed solution s of φ is also r -approximate. This concludes the proof of the AP-reduction with factor $\alpha = 1$. □

When dealing with $\text{NSol}(\Gamma \cup \{\approx\})$, the previous lemma enables us to concentrate on instances where the formula φ has the form

$$\psi(z_1, \dots, z_n, x_1, \dots, x_t) \wedge \bigwedge_{i=1}^t \bigwedge_{x \in V_i} (x_i \approx x),$$

where V_1, \dots, V_t are disjoint sets of variables, being also disjoint from the variables of the Γ -formula ψ . For each $1 \leq i \leq t$ the given assignment m can have equal distance to the zero vector and the all-ones vector on the variables in $V_i \cup \{x_i\}$, or it can be closer to one of the constant vectors. It is convenient to group the equality constraints according to these three cases. The following lemma discusses how to remove those equality constraints, on whose variables m is not equidistant from $\mathbf{0}$ and $\mathbf{1}$.

Lemma 11 *Let Γ be a constraint language and*

$$\psi(z_1, \dots, z_n, x_1, \dots, x_\alpha, v_1, \dots, v_\beta, w_1, \dots, w_\gamma)$$

be any Γ -formula containing precisely the distinct variables $z_1, \dots, z_n, x_1, \dots, x_\alpha, v_1, \dots, v_\beta$ and w_1, \dots, w_γ . Consider a formula

$$\varphi := \psi \wedge \bigwedge_{a=1}^\alpha \bigwedge_{x \in I'_a} (x_a \approx x) \wedge \bigwedge_{b=1}^\beta \bigwedge_{x \in J'_b} (v_b \approx x) \wedge \bigwedge_{c=1}^\gamma \bigwedge_{x \in K'_c} (w_c \approx x)$$

where $I'_1, \dots, I'_\alpha, J'_1, \dots, J'_\beta$ and K'_1, \dots, K'_γ are non-empty sets of variables that are pairwise disjoint and disjoint from the variables in ψ . For $1 \leq a \leq \alpha, 1 \leq b \leq \beta$ and $1 \leq c \leq \gamma$ we put $I_a := I'_a \cup \{x_a\}, J_b := J'_b \cup \{v_b\}$ and $K_c := K'_c \cup \{w_c\}$. Moreover, let m be an assignment for φ , such that for $1 \leq a \leq \alpha, 1 \leq b \leq \beta$ and $1 \leq c \leq \gamma$

$$\begin{aligned} d_{0,I_a} &:= \text{hd}(m \upharpoonright_{I_a}, \mathbf{0}), & d_{1,I_a} &:= \text{hd}(m \upharpoonright_{I_a}, \mathbf{1}), & d_{1,I_a} - d_{0,I_a} &= 0 \\ d_{0,J_b} &:= \text{hd}(m \upharpoonright_{J_b}, \mathbf{0}), & d_{1,J_b} &:= \text{hd}(m \upharpoonright_{J_b}, \mathbf{1}), & \text{satisfy } e_b &:= d_{1,J_b} - d_{0,J_b} > 0 \\ d_{0,K_c} &:= \text{hd}(m \upharpoonright_{K_c}, \mathbf{0}), & d_{1,K_c} &:= \text{hd}(m \upharpoonright_{K_c}, \mathbf{1}), & f_c &:= d_{0,K_c} - d_{1,K_c} > 0 \end{aligned}$$

It is possible to construct a formula ψ' , whose size is polynomial in the size of φ , and an assignment M for $\varphi' := \psi' \wedge \bigwedge_{a=1}^\alpha \bigwedge_{x \in I'_a} (x_a \approx x)$ such that the following holds

- ψ, φ, φ' and ψ' are equisatisfiable;

- if ψ is satisfiable, then $\text{OPT}(\varphi, m) = \text{OPT}(\varphi', M) + d$ where $d = \sum_{b=1}^{\beta} d_{0, J_b} + \sum_{c=1}^{\gamma} d_{1, K_c}$;
- for every $r \in [1, \infty)$, one can produce an (r -approximate) solution of (φ, m) from any (r -approximate) solution of (φ', M) in polynomial time.

Proof First, we describe how to construct the formula ψ' . In the following we use the abbreviations $Z := \{z_1, \dots, z_n, x_1, \dots, x_a\}$, $Z' := Z \cup \bigcup_{a=1}^{\alpha} I_a$, $V := \{v_1, \dots, v_{\beta}\}$ and $W := \{w_1, \dots, w_{\gamma}\}$. For every variable $u \in Z \cup V \cup W$ define a set $B(u)$ of variables as follows:

$$B(u) = \begin{cases} \{u\} & \text{if } u \in Z \\ \{u^1, \dots, u^{e_b}\} & \text{if } u = v_b \in V \\ \{u^1, \dots, u^{f_c}\} & \text{if } u = w_c \in W. \end{cases}$$

For each atom $R(u_1, \dots, u_q)$ of ψ define a set of atoms $\{R(u'_1, \dots, u'_q) \mid (u'_1, \dots, u'_q) \in \prod_{i=1}^q B(u_i)\}$, take the union over all these sets and define ψ' as the conjunction of all its members, giving a formula over $Z \cup V' \cup W'$ where $V' = \bigcup_{u \in V} B(u)$ and $W' = \bigcup_{u \in W} B(u)$. Adding again the equality constraints, where m has equal distance from $\mathbf{0}$ and $\mathbf{1}$ we get $\varphi' = \psi' \wedge \bigwedge_{a=1}^{\alpha} \bigwedge_{x \in I'_a} (x_a \approx x)$ over $Z' \cup V' \cup W'$. This is a polynomial time construction since the arities of relations in Γ are bounded.

Moreover, we define an assignment M to the variables u of φ' as follows:

$$M(u) = \begin{cases} m(u) & \text{if } u \in Z' \\ 0 & \text{if } u \in V' \\ 1 & \text{if } u \in W'. \end{cases}$$

Let S' be a solution of (φ', M) . If S' is constant on $B(u)$, for each $u \in V \cup W$, then put $S'' := S'$. Otherwise, by letting $S''(u) := S'(u)$ for $u \in Z'$ and for $u \in B(u')$ where $u' \in V \cup W$ is such that S' is constant on $B(u')$, and by defining $S''(u) := M(u) = 0$ for the remaining variables $u \in V'$ and $S''(u) := M(u) = 1$ for the remaining variables $u \in W'$, we obtain a model S'' of φ' satisfying $\text{hd}(S'', M) \leq \text{hd}(S', M)$ and being constant on $B(u)$ for each $u \in V \cup W$. From S'' we construct an assignment S of φ by defining $S(u) := S''(u)$ for $u \in Z'$, $S(u) := S''(v_b^1)$ for $u \in J_b$ and $1 \leq b \leq \beta$, and $S(u) := S''(w_c^1)$ for $u \in K_c$ and $1 \leq c \leq \gamma$. It satisfies φ as $e_b, f_c > 0$ for $1 \leq b \leq \beta$ and $1 \leq c \leq \gamma$. From these definitions, it follows

$$\begin{aligned} \text{hd}(S'', M) &= \\ &= \text{hd}(S'' \upharpoonright_{Z'}, M \upharpoonright_{Z'}) + \sum_{b=1}^{\beta} \text{hd}(S'' \upharpoonright_{B(v_b)}, M \upharpoonright_{B(v_b)}) + \sum_{c=1}^{\gamma} \text{hd}(S'' \upharpoonright_{B(w_c)}, M \upharpoonright_{B(w_c)}) \\ &= \text{hd}(S'' \upharpoonright_{Z'}, m \upharpoonright_{Z'}) + \sum_{b=1}^{\beta} S''(v_b^1) \cdot e_b + \sum_{c=1}^{\gamma} (1 - S''(w_c^1)) \cdot f_c, \end{aligned}$$

because S'' is constant on $B(u)$ for $u \in V \cup W$ and $|B(v_b)| = e_b$, $|B(w_c)| = f_c$ for $1 \leq b \leq \beta$ and $1 \leq c \leq \gamma$; and

$$\begin{aligned} \text{hd}(S, m) &= \\ &= \text{hd}(S \upharpoonright_{Z'}, m \upharpoonright_{Z'}) + \sum_{b=1}^{\beta} \text{hd}(S \upharpoonright_{J_b}, m \upharpoonright_{J_b}) + \sum_{c=1}^{\gamma} \text{hd}(S \upharpoonright_{K_c}, m \upharpoonright_{K_c}) \\ &= \text{hd}(S'' \upharpoonright_{Z'}, m \upharpoonright_{Z'}) + \sum_{b=1}^{\beta} \left(S''(v_b^1) \cdot e_b + d_{0,J_b} \right) + \sum_{c=1}^{\gamma} \left((1 - S''(w_c^1)) \cdot f_c + d_{1,K_c} \right). \end{aligned}$$

Consequently, $\text{hd}(S, m) = \text{hd}(S'', M) + d$, where $d = \sum_{b=1}^{\beta} d_{0,J_b} + \sum_{c=1}^{\gamma} d_{1,K_c}$.

Using this, we shall prove below that $\text{OPT}(\varphi', M) + d = \text{OPT}(\varphi, m)$. Thus, if S' now takes the role of an r -approximate solution of (φ', M) for some $r \geq 1$, then it follows that

$$\begin{aligned} \text{hd}(S, m) = \text{hd}(S'', M) + d &\leq \text{hd}(S', M) + d \leq r \text{OPT}(\varphi', M) + d \\ &\leq r \text{OPT}(\varphi', M) + rd = r \text{OPT}(\varphi, m). \end{aligned}$$

Let subsequently S' be such that $\text{OPT}(\varphi', M) = \text{hd}(S', M)$, and let s be a model of φ . Construct a model s' of φ' by putting $s'(u) := s(u)$ for $u \in Z'$ and $s'(u) := s(u')$ for $u \in B(u')$ and $u' \in V \cup W$. As above we get $\text{hd}(s, m) = \text{hd}(s', M) + d$ because the definitions imply

$$\begin{aligned} \text{hd}(s', M) &= \\ &= \text{hd}(s' \upharpoonright_{Z'}, M \upharpoonright_{Z'}) + \sum_{b=1}^{\beta} \text{hd}(s' \upharpoonright_{B(v_b)}, M \upharpoonright_{B(v_b)}) + \sum_{c=1}^{\gamma} \text{hd}(s' \upharpoonright_{B(w_c)}, M \upharpoonright_{B(w_c)}) \\ &= \text{hd}(s \upharpoonright_{Z'}, m \upharpoonright_{Z'}) + \sum_{b=1}^{\beta} s(v_b) \cdot e_b + \sum_{c=1}^{\gamma} (1 - s(w_c)) \cdot f_c ; \\ \text{hd}(s, m) &= \\ &= \text{hd}(s \upharpoonright_{Z'}, m \upharpoonright_{Z'}) + \sum_{b=1}^{\beta} \text{hd}(s \upharpoonright_{J_b}, m \upharpoonright_{J_b}) + \sum_{c=1}^{\gamma} \text{hd}(s \upharpoonright_{K_c}, m \upharpoonright_{K_c}) \\ &= \text{hd}(s \upharpoonright_{Z'}, m \upharpoonright_{Z'}) + \sum_{b=1}^{\beta} (s(v_b) \cdot e_b + d_{0,J_b}) + \sum_{c=1}^{\gamma} ((1 - s(w_c)) \cdot f_c + d_{1,K_c}). \end{aligned}$$

By minimality of S' , we obtain $\text{hd}(S'', M) \leq \text{hd}(S', M) \leq \text{hd}(s', M)$. If we additionally require that s be an optimal solution of (φ, m) , then $\text{hd}(s', M) = \text{hd}(s, m) - d \leq \text{hd}(S, m) - d = \text{hd}(S'', M)$. Thus, the distances $\text{hd}(S'', M)$, $\text{hd}(S', M)$ and $\text{hd}(s', M)$ coincide, which implies the desired equality $\text{OPT}(\varphi, m) = \text{hd}(s, m) = \text{hd}(s', M) + d = \text{hd}(S', M) + d = \text{OPT}(\varphi', M) + d$. \square

The previous lemma, in fact, describes an AP-reduction from the specialized version of the problem $\text{NSol}(\Gamma \cup \{\approx\})$ discussed in Lemma 10 to an even more specialized variant (the analogous statement is true for the decision version—instances

(φ, m, k) can be decided by considering $(\varphi', M, k - d)$ instead): namely all equality constraints touch variables in Γ -atoms and the given assignment has equal distance from the constant tuples on each variable block connected by equalities. In the next result we show how to remove also these equality constraints.

Proposition 12 *For constraint languages Γ , we have $\text{NSol}^d(\Gamma) \equiv_m \text{NSol}^d(\Gamma \cup \{\approx\})$ and $\text{NSol}(\Gamma) \equiv_{\text{AP}} \text{NSol}(\Gamma \cup \{\approx\})$.*

Proof The reduction from left to right is trivial. For the other direction, consider first an instance of $\text{NSol}^d(\Gamma \cup \{\approx\})$ with formula φ , assignment m , and bound k . Applying the reductions indicated in Lemmas 10 and 11, we can assume (also for $\text{NSol}(\Gamma \cup \{\approx\})$) that φ is of the form $\psi \wedge \bigwedge_{a=1}^\alpha \bigwedge_{x \in I'_a} (x_a \approx x)$ with a Γ -formula ψ containing the distinct variables $z_1, \dots, z_n, x_1, \dots, x_\alpha$ ($n \geq 0, \alpha \geq 1$) and non-empty disjoint (from each other and from ψ) variable sets I'_a for $1 \leq a \leq \alpha$. Moreover, we can suppose that $\text{hd}(m \upharpoonright_{I_a}, \mathbf{0}) = \text{hd}(m \upharpoonright_{I_a}, \mathbf{1}) =: c_a$ for all $1 \leq a \leq \alpha$, where I_a denotes the set $I'_a \cup \{x_a\}$.

We define $c := \sum_{a=1}^\alpha c_a$, and we choose some ℓ -element index set I such that $\alpha/\ell < 1$, that is, $\ell \geq \alpha + 1$ (we shall place another condition on ℓ at the end). We construct a formula φ' as follows: For each atom $R(u_1, \dots, u_q)$ of ψ we introduce the set $\{R(u_1^i, \dots, u_q^i) \mid (i_1, \dots, i_q) \in I^q\}$ of atoms where for $1 \leq v \leq q$ and $i \in I$ we let $u_v^i := z_{j,i}$ if $u_v = z_j$ for some $1 \leq j \leq n$ and $u_v^i = x_a$ if else $u_v \in \{x_1, \dots, x_\alpha\}$. Take the union over all these sets and let φ' be the conjunction of all atoms in this union. This construction can be carried out in polynomial time since there is a bound on the arities of relations in Γ . Define an assignment M by $M(x_a) := m(x_a)$ for $1 \leq a \leq \alpha$ and $M(z_{j,i}) := m(z_j)$ for $1 \leq j \leq n$ and $i \in I$. We claim that existence of solutions for (φ, m, k) can be decided by checking for solutions of $(\varphi', M, \ell(k - c) + \alpha)$. The argument is similar to that of Proposition 7: ψ is (un)satisfiable if and only if φ and φ' are, so we have a correct answer in the unsatisfiable case. Otherwise, consider a solution s to (φ, m, k) . Letting $Z := \{z_1, \dots, z_n\}$, we have

$$\begin{aligned} \text{hd}(s, m) &= \text{hd}(s \upharpoonright_Z, m \upharpoonright_Z) + \sum_{a=1}^\alpha \text{hd}(s \upharpoonright_{I_a}, m \upharpoonright_{I_a}) = \text{hd}(s \upharpoonright_Z, m \upharpoonright_Z) + \sum_{a=1}^\alpha c_a \\ &= \text{hd}(s \upharpoonright_Z, m \upharpoonright_Z) + c, \end{aligned}$$

i.e. $\text{hd}(s \upharpoonright_Z, m \upharpoonright_Z) \leq k - c$. Putting $s'(x_a) := s(x_a)$ for $1 \leq a \leq \alpha$ and $s'(z_{j,i}) := s(z_j)$ for $1 \leq j \leq n$ and $i \in I$ we get a model of φ' , and it follows that $\text{hd}(s' \upharpoonright_{Z'}, M \upharpoonright_{Z'}) = \ell \cdot \text{hd}(s \upharpoonright_Z, m \upharpoonright_Z) \leq \ell \cdot (k - c)$, where $Z' := \{z_{j,i} \mid 1 \leq j \leq n, i \in I\}$. Therefore, abbreviating $X := \{x_1, \dots, x_\alpha\}$, we obtain $\text{hd}(s', M) = \text{hd}(s' \upharpoonright_{Z'}, M \upharpoonright_{Z'}) + \text{hd}(s' \upharpoonright_X, M \upharpoonright_X) \leq \ell \cdot (k - c) + \alpha$.

Conversely, let S' be a solution of $(\varphi', M, \ell(k - c) + \alpha)$. As in Proposition 7 we can construct a solution S'' being constant on $\{z_{j,i} \mid i \in I\}$ for each $1 \leq j \leq n$. Letting $S(x) := S''(x_a)$ for $x \in I_a$ and $1 \leq a \leq \alpha$ and $S(z_j) := S''(z_{j,i})$ for some fixed index $i \in I$ and all $1 \leq j \leq n$, one obtains a model of φ . If $S(z_j) \neq m(z_j)$ for some $1 \leq j \leq n$, then we have $S''(z_{j,i}) = S(z_j) \neq m(z_j) = M(z_{j,i})$ for all $i \in I$. Hence, we have $\ell \cdot \text{hd}(S \upharpoonright_Z, m \upharpoonright_Z) \leq \text{hd}(S'' \upharpoonright_{Z'}, M \upharpoonright_{Z'}) \leq \text{hd}(S'', M) \leq \text{hd}(S', M)$. Division

by ℓ implies $\text{hd}(S \upharpoonright_Z, m \upharpoonright_Z) \leq \text{hd}(S', M)/\ell \leq k - c + \alpha/\ell < k - c + 1$, i.e. $\text{hd}(S \upharpoonright_Z, m \upharpoonright_Z) \leq k - c$. From this we finally infer that $\text{hd}(S, m) = \text{hd}(S \upharpoonright_Z, m \upharpoonright_Z) + c \leq k$.

Suppose now that S' is an r -approximate solution for (φ', M) for some $r \geq 1$, i.e. we have $\text{hd}(S', M) \leq r \text{OPT}(\varphi', M)$. Constructing a model S of φ as before, we obtain $\ell \text{hd}(S \upharpoonright_Z, m \upharpoonright_Z) \leq \text{hd}(S', M) \leq r \text{OPT}(\varphi', M)$. Furthermore, from an optimal solution of φ , we get a model s' of φ' satisfying

$$\begin{aligned} \text{OPT}(\varphi', M) &\leq \text{hd}(s', M) = \text{hd}(s' \upharpoonright_{Z'}, M \upharpoonright_{Z'}) + \text{hd}(s' \upharpoonright_X, M \upharpoonright_X) \\ &= \ell(\text{OPT}(\varphi, m) - c) + \text{hd}(s' \upharpoonright_X, M \upharpoonright_X) \\ &\leq \ell(\text{OPT}(\varphi, m) - c) + \alpha. \end{aligned}$$

Multiplying this inequality by r , combining it with previous inequalities and dividing by ℓ we thus have $\text{hd}(S \upharpoonright_Z, m \upharpoonright_Z) \leq r \text{OPT}(\varphi, m) - rc + r\alpha/\ell$. Note that $\text{OPT}(\varphi, m) > 0$, because if $\text{OPT}(\varphi, m) = 0$, then we would have a unique optimal model of φ , namely m . Then $m \upharpoonright_{I_1}$ would have to be constant, implying $\text{hd}(m \upharpoonright_{I_1}, \mathbf{0}) \neq \text{hd}(m \upharpoonright_{I_1}, \mathbf{1})$, as one distance would be zero and the other one $|I_1| > 0$. Therefore, for $\ell \in \Omega(|\varphi|^2)$ we have $\text{hd}(S, m) = \text{hd}(S \upharpoonright_Z, m \upharpoonright_Z) + c \leq \text{hd}(S \upharpoonright_Z, m \upharpoonright_Z) + rc \leq r \text{OPT}(\varphi, m) + r\alpha/\ell \leq \text{OPT}(\varphi, m)(r + r\alpha/\ell) = \text{OPT}(\varphi, m)(r + o(1))$. This demonstrates an AP-reduction with factor 1. □

Propositions 7 and 12 allow us to switch freely between formulas with quantifiers and equality and those without. Hence we may derive upper bounds in the setting without quantifiers and equality while using the latter in hardness reductions. In particular, we can use pp-definability when implementing a constraint language Γ by another constraint language Γ' . Hence it suffices to consider Post’s lattice of co-clones to characterize the complexity of $\text{NSol}(\Gamma)$ for every finite constraint language Γ .

Corollary 13 *For constraint languages Γ and Γ' , for which the inclusion $\Gamma' \subseteq \langle \Gamma \rangle$ holds, we have the reductions $\text{NSol}^d(\Gamma') \leq_m \text{NSol}^d(\Gamma)$ and $\text{NSol}(\Gamma') \leq_{\text{AP}} \text{NSol}(\Gamma)$. Thus, if $\langle \Gamma' \rangle = \langle \Gamma \rangle$ is satisfied, then the equivalences $\text{NSol}^d(\Gamma) \equiv_m \text{NSol}^d(\Gamma')$ and $\text{NSol}(\Gamma) \equiv_{\text{AP}} \text{NSol}(\Gamma')$ hold.*

Next we prove that, in certain cases, unit clauses in the formula do not change the complexity of NSol .

Proposition 14 *Let Γ be a constraint language such that feasible solutions of $\text{NSol}(\Gamma)$ can be found in polynomial time. Then we have $\text{NSol}(\Gamma) \equiv_{\text{AP}} \text{NSol}(\Gamma \cup \{\{x\}, [\neg x]\})$.*

Proof The direction from left to right is obvious. For the other direction, we give an AP-reduction from $\text{NSol}(\Gamma \cup \{\{x\}, [\neg x]\})$ to $\text{NSol}(\Gamma \cup \{\approx\})$. The latter is AP-equivalent to $\text{NSol}(\Gamma)$ by Proposition 12.

The idea of the construction is to introduce two sets of variables y_1, \dots, y_{n^2} and z_1, \dots, z_{n^2} such that in any feasible solution all y_i and all z_i take the same value. By setting $m(y_i) = 1$ and $m(z_i) = 0$ for each i , any feasible solution m' of small Hamming distance to m will have $m'(y_i) = 1$ and $m'(z_i) = 0$ for all i as well,

because deviating from this would be prohibitively expensive. Finally, we simulate the unary relations x and $\neg x$ by $x \approx y_1$ and $x \approx z_1$, respectively. We now describe the reduction formally.

Consider a formula φ over $\Gamma \cup \{[x], [\neg x]\}$ with the variables x_1, \dots, x_n and an assignment m . If (φ, m) fails to have feasible solutions, i.e., if φ is unsatisfiable, we can detect this in polynomial time by the assumption of the lemma and return the generic unsatisfiable instance \perp . Otherwise, we construct a $(\Gamma \cup \{\approx\})$ -formula φ' over the variables $x_1, \dots, x_n, y_1, \dots, y_{n^2}, z_1, \dots, z_{n^2}$ and an assignment m' . We obtain φ' from φ by replacing every occurrence of a constraint $[x]$ by $x \approx y_1$ and every occurrence of $[\neg x]$ by $x \approx z_1$. Finally, we add the atoms $y_i \approx y_1$ and $z_i \approx z_1$ for all $i \in \{2, \dots, n^2\}$. Let m' be the assignment of the variables of φ' given by $m'(x_i) = m(x_i)$ for each $i \in \{1, \dots, n\}$, and $m'(y_i) = 1$ and $m'(z_i) = 0$ for all $i \in \{1, \dots, n^2\}$. To any feasible solution m'' of φ' we assign $g(\varphi, m, m'')$ as follows.

1. If φ is satisfied by m , we define $g(\varphi, m, m'')$ to be equal to m .
2. Else if $m''(y_i) = 0$ holds for all $i \in \{1, \dots, n^2\}$ or $m''(z_i) = 1$ for all $i \in \{1, \dots, n^2\}$, we define $g(\varphi, m, m'')$ to be any satisfying assignment of φ .
3. Otherwise, we have $m''(y_i) = 1$ and $m''(z_i) = 0$ for all $i \in \{1, \dots, n^2\}$. In this case we define $g(\varphi, m, m'')$ to be the restriction of m'' onto x_1, \dots, x_n .

Observe that all variables y_i and all z_i are forced to take the same value in any feasible solution, respectively, so $g(\varphi, m, m'')$ is always well-defined. The construction is an AP-reduction. Assume that m'' is an r -approximate solution. We will show that $g(\varphi, m, m'')$ is also an r -approximate solution.

Case 1 $g(\varphi, m, m'')$ computes the optimal solution, so there is nothing to show.

Case 2 Observe first that φ has a solution because otherwise it would have been mapped to \perp and m'' would not exist. Thus, $g(\varphi, m, m'')$ is well-defined and feasible by construction. Observe that m' and m'' disagree on all y_i or on all z_i , so we have $\text{hd}(m', m'') \geq n^2$. Moreover, since φ has a feasible solution, it follows that $\text{OPT}(\varphi', m') \leq n$. Since m'' is an r -approximate solution, we have that

$$n \text{OPT}(\varphi', m') \leq n^2 \leq \text{hd}(m', m'') \leq r \text{OPT}(\varphi', m') .$$

If $\text{OPT}(\varphi', m') = 0$, then m' would have to be a model of φ' , and so would be its restriction to the x_i , i.e. m , a model of φ . This is handled in the first case, which is disjoint from the current one; hence, we infer $n \leq r$. Consequently, the distance $\text{hd}(m, g(\varphi, m, m''))$ is bounded above by $n \leq r \leq r \cdot \text{OPT}(\varphi, m)$, where the last inequality holds because φ is not satisfied by m and thus the distance of any optimal solution from m is at least 1.

Case 3 The variables x_i , for which $[x_i]$ is a constraint, all satisfy $g(\varphi, m, m'')(x_i) = 1$ by construction. Moreover, we have $g(\varphi, m, m'')(x_i) = 0$ for all x_i for which $[\neg x_i]$ is a constraint of φ . Consequently, $g(\varphi, m, m'')$ is feasible. Again, $\text{OPT}(\varphi', m') \leq n$, so any optimal solution to (φ', m') must set all variables y_i to 1 and all z_i to 0. It follows that $\text{OPT}(\varphi, m) = \text{OPT}(\varphi', m')$. Thus we get

$$\text{hd}(m, g(\varphi, m, m'')) = \text{hd}(m', m'') \leq r \cdot \text{OPT}(\varphi', m') = r \cdot \text{OPT}(\varphi, m),$$

which completes the proof. □

4.2 Inapplicability of Clone Closure

Corollary 13 shows that the complexity of NSol is not affected by existential quantification by giving an explicit reduction from NSol_{pp} to NSol. It does not seem possible to prove the same for NOSol and MSD. However, similar results hold for the conjunctive closure; thus we resort to minimal or irredundant weak bases of co-clones instead of usual bases.

Proposition 15 *Let Γ and Γ' be constraint languages. If $\Gamma' \subseteq \langle \Gamma \rangle_{\wedge}$ holds then we have the reductions $\text{NOSol}^d(\Gamma') \leq_m \text{NOSol}^d(\Gamma)$ and $\text{NOSol}(\Gamma') \leq_{\text{AP}} \text{NOSol}(\Gamma)$, as well as $\text{MSD}^d(\Gamma') \leq_m \text{MSD}^d(\Gamma)$ and $\text{MSD}(\Gamma') \leq_{\text{AP}} \text{MSD}(\Gamma)$.*

Proof We prove only the part that $\Gamma' \subseteq \langle \Gamma \rangle_{\wedge}$ implies $\text{NOSol}(\Gamma') \leq_{\text{AP}} \text{NOSol}(\Gamma)$. The other results will be clear from that reduction since the proof is generic and therefore holds for both NOSol and MSD, as well as for their decision variants.

Let a Γ' -formula φ be an instance of $\text{NOSol}(\Gamma')$. Since $\Gamma' \subseteq \langle \Gamma \rangle_{\wedge}$, every constraint $R(x_1, \dots, x_k)$ of φ can be written as a conjunction of constraints over relations from Γ . Substitute the latter into φ , obtaining φ' . Now φ' is an instance of $\text{NOSol}(\Gamma)$, where φ' is only polynomially larger than φ . As φ and φ' have the same variables and hence the same models, also the closest distinct models of φ and φ' are the same. \square

4.3 Duality

Given a relation $R \subseteq \{0, 1\}^n$, its *dual* relation is $\text{dual}(R) = \{\overline{m} \mid m \in R\}$, i.e., the relation containing the complements of tuples from R . Duality naturally extends to sets of relations and co-clones. We define $\text{dual}(\Gamma) = \{\text{dual}(R) \mid R \in \Gamma\}$ as the set of dual relations to Γ . Since taking complements is involutive, duality is a symmetric relation. If a relation R' (a set of relations Γ') is a dual relation to R (a set of dual relations to Γ), then R (Γ) is also dual to R' (to Γ'). By a simple inspection of the bases of co-clones in Table 2, we can easily see that many co-clones are dual to each other. For instance iE_2 is dual to iV_2 . The following proposition shows that it is sufficient to consider only one half of Post’s lattice of co-clones.

Proposition 16 *For any constraint language Γ we have*

$$\begin{aligned} \text{NSol}^d(\Gamma) \equiv_m \text{NSol}^d(\text{dual}(\Gamma)) \quad \text{and} \quad \text{NSol}(\Gamma) \equiv_{\text{AP}} \text{NSol}(\text{dual}(\Gamma)), \\ \text{NOSol}^d(\Gamma) \equiv_m \text{NOSol}^d(\text{dual}(\Gamma)) \quad \text{and} \quad \text{NOSol}(\Gamma) \equiv_{\text{AP}} \text{NOSol}(\text{dual}(\Gamma)), \end{aligned}$$

as well as

$$\text{MSD}^d(\Gamma) \equiv_m \text{MSD}^d(\text{dual}(\Gamma)) \quad \text{and} \quad \text{MSD}(\Gamma) \equiv_{\text{AP}} \text{MSD}(\text{dual}(\Gamma)).$$

Proof Let φ be a Γ -formula and m an assignment to φ . We construct a $\text{dual}(\Gamma)$ -formula φ' by substitution of every atom $R(\mathbf{x})$ by $\text{dual}(R)(\mathbf{x})$. The assignment m satisfies φ if and only if \overline{m} satisfies φ' , where \overline{m} is the pointwise complement of m . Moreover, $\text{hd}(m, m') = \text{hd}(\overline{m}, \overline{m}')$. \square

5 Finding the Nearest Solution

This section contains the proof of Theorem 3. We first consider the polynomial-time cases followed by the cases of higher complexity.

5.1 Polynomial-Time Cases

Proposition 17 *If a constraint language Γ is both bijunctive and affine ($\Gamma \subseteq \text{iD}_1$), then $\text{NSol}(\Gamma)$ can be solved in polynomial time.*

Proof Since $\Gamma \subseteq \text{iD}_1 = \langle \Gamma' \rangle$ with $\Gamma' := \{[x \oplus y], [x]\}$, we have $\text{NSol}(\Gamma) \leq_{\text{AP}} \text{NSol}(\Gamma')$ by Corollary 13. Every Γ' -formula φ is equivalent to a linear system of equations over the Boolean ring \mathbb{Z}_2 of type $x \oplus y = 1$ and $x = 1$. Substitute the fixed values $x = 1$ into the equations of the type $x \oplus y = 1$ and propagate. If a contradiction is found thereby, reject the input. After an exhaustive application of this rule only equations of the form $x \oplus y = 1$ remain. For each of them put an edge $\{x, y\}$ into E , defining an undirected graph $G = (V, E)$, whose vertices V are the unassigned variables. If G is not bipartite, then φ has no solutions, so we can reject the input. Otherwise, compute a bipartition $V = L \dot{\cup} R$. We assume that G is connected; if not perform the following algorithm for each connected component (cf. Lemma 9). Assign the value 0 to each variable in L and the value 1 to each variable in R , giving the satisfying assignment m_1 . Swapping the roles of 0 and 1 w.r.t. L and R we get a model m_2 . Return $\min\{\text{hd}(m, m_1), \text{hd}(m, m_2)\}$. \square

Proposition 18 *If a constraint language Γ is monotone ($\Gamma \subseteq \text{iM}_2$), then the problem $\text{NSol}(\Gamma)$ can be solved in polynomial time.*

Proof We have $\text{iM}_2 = \langle \Gamma' \rangle$ where $\Gamma' := \{[x \rightarrow y], [\neg x], [x]\}$. Thus Corollary 13 and $\Gamma \subseteq \langle \Gamma' \rangle$ imply $\text{NSol}(\Gamma) \leq_{\text{AP}} \text{NSol}(\Gamma')$. The relations $[\neg x]$ and $[x]$ determine a unique value for the respective variable, therefore we can eliminate unit clauses and propagate the values. If a contradiction occurs, we reject the input. It thus remains to consider formulas φ containing only binary implicative clauses of type $x \rightarrow y$.

Let V be the set of variables in φ , and for $i \in \{0, 1\}$ let $V_i = \{x \in V \mid m(x) = i\}$ be the variables mapped to value i by assignment m . We transform the formula φ to a linear programming problem as follows. For each clause $x \rightarrow y$ we add the inequality $y \geq x$, and for each variable $x \in V$ we add the constraints $x \geq 0$ and $x \leq 1$. As linear objective function we use $f(\mathbf{x}) = \sum_{x \in V_0} x + \sum_{x \in V_1} (1 - x)$. For an arbitrary solution m' , it returns the number of variables that change their parity between m and m' , i.e., $f(m') = \text{hd}(m, m')$. This way we obtain the (integer) linear programming problem $(f, A\mathbf{x} \geq \mathbf{b})$, where A is a totally unimodular matrix and \mathbf{b} is an integral column vector.

The rows of A consist of the left-hand sides of inequalities $y - x \geq 0$, $x \geq 0$, and $-x \geq -1$, which constitute the system $A\mathbf{x} \geq \mathbf{b}$. Every entry in A is 0, +1, or -1. Every row of A has at most two non-zero entries. For the rows with two entries, one entry is +1, the other is -1. According to Condition (iv) in Theorem 19.3 in [30], this is a sufficient condition for A being totally unimodular. As A is totally unimodular

and \mathbf{b} is an integral vector, f has integral minimum points, and one of them can be computed in polynomial time (see e.g. [30, Chapter 19]). \square

5.2 Hard Cases

We start off with an easy corollary of Schaefer’s dichotomy.

Proposition 19 *Let Γ be a finite set of Boolean relations. If $iN_2 \subseteq \langle \Gamma \rangle$, then it is NP-complete to decide whether a feasible solution exists for $NSol(\Gamma)$; otherwise, $NSol(\Gamma) \in poly\text{-APX}$.*

Proof If $iN_2 \subseteq \langle \Gamma \rangle$ holds, checking the existence of feasible solutions for $NSol(\Gamma)$ -instances is NP-hard by Schaefer’s theorem [27].

Let (φ, m) be an instance of $NSol(\Gamma)$. We give an n -approximate algorithm for the other cases, where n denotes the number of variables in φ . If m satisfies φ , return m . Otherwise compute an arbitrary solution m' of φ , which can be done in polynomial time by Schaefer’s theorem. This algorithm is n -approximate: If m satisfies φ , the algorithm returns the optimal solution; otherwise we have $OPT(\varphi, m) \geq 1$ and $hd(m, m') \leq n$, hence the answer m' of the algorithm is n -approximate. \square

5.2.1 APX-Complete Cases

We start with reductions from the optimization version of vertex cover. Since the relation $[x \vee y]$ is a straightforward Boolean encoding of vertex cover, we immediately get the following result.

Proposition 20 *$NSol(\Gamma)$ is APX-hard for every constraint language Γ satisfying $iS_0^2 \subseteq \langle \Gamma \rangle$ or $iS_1^2 \subseteq \langle \Gamma \rangle$.*

Proof We have $iS_0^2 = \langle \{[x \vee y]\} \rangle$ and $iS_1^2 = \langle \{[\neg x \vee \neg y]\} \rangle$. We discuss the former case, the latter one being symmetric and provable from the first one by Proposition 16.

We encode **VertexCover** into $NSol(\{[x \vee y]\})$. For each edge $\{x, y\} \in E$ of a graph $G = (V, E)$ we add the clause $(x \vee y)$ to the formula φ_G . Every model m' of φ_G yields a vertex cover $\{v \in V \mid m'(v) = 1\}$, and conversely, the characteristic function of any vertex cover satisfies φ_G . Moreover, we choose $m = \mathbf{0}$. Then $hd(\mathbf{0}, m')$ is minimal if and only if the number of 1s in m' is minimal, i.e., if m' is a minimal model of φ_G , i.e., if m' represents a minimal vertex cover of G . Since **VertexCover** is APX-complete (see e.g. [3]) and $NSol(\{[x \vee y]\}) \leq_{AP} NSol(\Gamma)$ (see Corollary 13), the result follows. \square

Proposition 21 *We have $NSol(\Gamma) \in APX$ for constraint languages $\Gamma \subseteq iD_2$.*

Proof $\Gamma' := \{[x \oplus y], [x \rightarrow y]\}$ is a base of iD_2 . By Corollary 13 it suffices to show that $NSol(\Gamma')$ is in APX. Let (φ, m) be an instance of this problem. Feasibility for φ can be encoded as an integer program as follows: Every constraint $x \oplus y$ induces an

equation $x + y = 1$, every constraint $x \rightarrow y$ an inequality $x \leq y$. If we restrict all variables to $\{0, 1\}$ by the appropriate inequalities, it is clear that an assignment m' satisfies φ if it satisfies the linear system with inequality side conditions. As objective function we use $f(x) := \sum_{x \in V_0} x + \sum_{x \in V_1} (1 - x)$, where V_i is the set of variables mapped to i by m . Clearly, for every solution m' we have $f(m') = \text{hd}(m, m')$. The 2-approximation algorithm from [17] for integer linear programs, where every inequality contains at most two variables, completes the proof. \square

Proposition 22 *We have $\text{NSol}(\Gamma) \in \text{APX}$ for constraint languages $\Gamma \subseteq \text{iS}_{00}^\ell$ with $\ell \geq 2$.*

Proof $\Gamma' := \{[x_1 \vee \dots \vee x_\ell], [x \rightarrow y], [\neg x], [x]\}$ is a base of iS_{00}^ℓ . By Corollary 13 it suffices to show that $\text{NSol}(\Gamma')$ is in APX. Let (φ, m) be an instance of this problem. We use an approach similar to the one for the corresponding case in [22], again writing φ as an integer program. We write constraints $x_1 \vee \dots \vee x_\ell$ as inequalities $x_1 + \dots + x_\ell \geq 1$, constraints $x \rightarrow y$ as $x \leq y$, $\neg x$ as $x = 0$, and x as $x = 1$. Moreover, we add $x \geq 0$ and $x \leq 1$ for each variable x . It is easy to check that the feasible Boolean solutions of φ and of the linear system coincide. As objective function we use $f(x) := \sum_{x \in V_0} x + \sum_{x \in V_1} (1 - x)$, where V_i is the set of variables mapped to i by m . Clearly, for every solution m' we have $f(m') = \text{hd}(m, m')$. Therefore it suffices to approximate the optimal solution for the integer linear program.

To this end, let m'' be a (generally non-integer) solution to the relaxation of the linear program, which can be computed in polynomial time. We construct m' by setting $m'(x) = 0$ if $m''(x) < 1/\ell$ and $m'(x) = 1$ if $m''(x) \geq 1/\ell$. As $\ell \geq 2$, we get $\text{hd}(m, m') = f(m') \leq \ell f(m'') \leq \ell \cdot \text{OPT}(\varphi, m)$. It is easy to check that m' is a feasible solution, which completes the proof. \square

5.2.2 NearestCodeword-Complete Cases

This section essentially uses the facts that MinOnes is NearestCodeword -complete for the co-clone iL_2 and that it is a special case of NSol . The following result was stated by Khanna et al. for completeness via A-reductions [22, Theorem 2.14]. A closer look at the proof reveals that it also holds for the stricter notion of completeness via AP-reductions that we use. In this respect the proofs of Propositions 23 and 27 spell out the missing details from [8, Propositions 15 and 18].

Proposition 23 *$\text{MinOnes}(\Gamma)$ is NearestCodeword -complete by AP-reductions for constraint languages Γ satisfying $\langle \Gamma \rangle = \text{iL}_2$.*

Proof According to [22, Lemma 8.13], $\text{MinOnes}(\Gamma)$ is NearestCodeword -hard for $\text{iL} \subseteq \langle \Gamma \rangle$. This proof uses AP-reductions, i.e., $\text{NearestCodeword} \leq_{\text{AP}} \text{MinOnes}(\Gamma)$.

Regarding the other direction, $\text{MinOnes}(\Gamma) \leq_{\text{AP}} \text{NearestCodeword}$, we first observe that $\text{odd}^3 = \{(a_1, a_2, a_3) \in \{0, 1\}^3 \mid \sum_i a_i \text{ odd}\}$ and $\text{even}^3 = \{(a_1, a_2, a_3) \in \{0, 1\}^3 \mid \sum_i a_i \text{ even}\}$ perfectly implement every constraint in iL_2 , i.e., $\langle \{\text{odd}^3, \text{even}^3\} \rangle = \text{iL}_2$ as shown in [22, Lemma 7.6]. Therefore, for $\Gamma \subseteq \text{iL}_2$, the problem $\text{WeightedMinOnes}(\Gamma)$ AP-reduces to

WeightedMinOnes($\{\text{odd}^3, \text{even}^3\}$) [22, Lemma 3.9]. The latter problem AP-reduces to WeightedMinCSP($\{\text{odd}^3, \text{even}^3, [\neg x]\}$) [22, Lemma 8.1], which further AP-reduces to WeightedMinCSP($\{\text{odd}^3, \text{even}^3\}$) because of $[\neg x] = [\text{even}^3(x, x, x)]$. In total we thus have that the problem WeightedMinOnes(Γ) AP-reduces to the problem WeightedMinCSP($\{\text{odd}^3, \text{even}^3\}$). We conclude by observing that MinOnes is a particular case of the WeightedMinOnes problem and that NearestCodeword is the same as WeightedMinCSP($\{\text{odd}^3, \text{even}^3\}$), yielding $\text{MinOnes}(\Gamma) \leq_{\text{AP}} \text{NearestCodeword}$. \square

Lemma 24 *We have $\text{MinOnes}(\Gamma) \leq_{\text{AP}} \text{NSol}(\Gamma)$ for any constraint language Γ .*

Proof $\text{MinOnes}(\Gamma)$ is a special case of $\text{NSol}(\Gamma)$ where m is the $\mathbf{0}$ -assignment. \square

Corollary 25 *$\text{NSol}(\Gamma)$ is NearestCodeword-hard for constraint languages Γ satisfying $iL \subseteq \langle \Gamma \rangle$.*

Proof $\Gamma' := \{\text{even}^4, [x], [\neg x]\}$ is a base of iL_2 . By Proposition 23, $\text{MinOnes}(\Gamma')$ is NearestCodeword-complete. By Lemma 24, $\text{MinOnes}(\Gamma')$ reduces to $\text{NSol}(\Gamma')$. By Proposition 14, $\text{NSol}(\Gamma')$ is AP-equivalent to $\text{NSol}(\{\text{even}^4\})$. Finally, because of $\text{even}^4 \in iL \subseteq \langle \Gamma \rangle$ and Corollary 13, $\text{NSol}(\{\text{even}^4\})$ reduces to $\text{NSol}(\Gamma)$. \square

Proposition 26 *We have $\text{NSol}(\Gamma) \leq_{\text{AP}} \text{MinOnes}(\{\text{even}^4, [\neg x], [x]\})$ for constraint languages $\Gamma \subseteq iL_2$.*

Proof $\Gamma' := \{\text{even}^4, [\neg x], [x]\}$ is a base of iL_2 . By Corollary 13 it suffices to show $\text{NSol}(\Gamma') \leq_{\text{AP}} \text{MinOnes}(\Gamma')$.

We proceed by reducing $\text{NSol}(\Gamma')$ to a subproblem of $\text{NSol}_{\text{pp}}(\Gamma')$, where only instances $(\varphi, \mathbf{0})$ are considered. Then, using Proposition 7 and Remark 8, this reduces to a subproblem of $\text{NSol}(\Gamma')$ with the same restriction on the assignments, which is exactly $\text{MinOnes}(\Gamma')$. Note that $[x \oplus y]$ is equal to $[\exists z \exists z' (\text{even}^4(x, y, z, z') \wedge \neg z \wedge z')]$ so we can freely use $[x \oplus y]$ in any Γ' -formula. Let formula φ and assignment m be an instance of $\text{NSol}(\Gamma')$. We copy all clauses of φ to φ' . For each variable x of φ for which $m(x) = 1$, we take a new variable x' and add the constraint $x \oplus x'$ to φ' . Moreover, we existentially quantify x . Clearly, there is a bijection I between the satisfying assignments of φ and those of φ' : For every solution s of φ we get a solution $I(s)$ of φ' by setting for each x' introduced in the construction of φ' the value $I(s)(x')$ to the complement of $s(x)$. Moreover, we have that $\text{hd}(m, s) = \text{hd}(\mathbf{0}, I(s))$. This yields a trivial AP-reduction with $\alpha = 1$. \square

5.2.3 MinHornDeletion-Complete Cases

Proposition 27 (Khanna et al. [22]) *We have MinHornDeletion-completeness for the problems $\text{MinOnes}(\{x \vee y \vee \neg z, x, \neg x\})$ and $\text{WeightedMinOnes}(\{x \vee y \vee \neg z, x \vee y\})$ via AP-reductions.*

Proof These results are stated in [22, Theorem 2.14] for completeness via A-reductions. The actual proof in [22, Lemma 8.7 and Lemma 8.14], however, uses

AP-reductions, hence the results also hold for our stricter notion of completeness. \square

Lemma 28 $\text{NSol}(\{x \vee y \vee \neg z\}) \leq_{\text{AP}} \text{WeightedMinOnes}(\{x \vee y \vee \neg z, x \vee y\})$.

Proof Let formula φ and assignment m be an instance of $\text{NSol}(\{x \vee y \vee \neg z\})$ over the variables x_1, \dots, x_n . Let V_1 be the set of variables x_i with $m(x_i) = 1$. We construct a $\{x \vee y \vee \neg z, x \vee y\}$ -formula φ' by adding to φ for each $x_i \in V_1$ the constraint $x_i \vee x'_i$ where x'_i is a new variable. We set the weights of the variables of φ' as follows. For $x_i \in V_1$ we set $w(x_i) = 0$, all other variables get weight 1. To each satisfying assignment m' of φ' we construct the assignment m'' which is the restriction of m' to the variables of φ . This construction is an AP-reduction.

Note that m'' is feasible if m' is. Let m' be an r -approximation of $\text{OPT}(\varphi')$. Note that whenever for $x_i \in V_1$ we have $m'(x_i) = 0$ then $m'(x'_i) = 1$. The other way round, we may assume that whenever $m'(x_i) = 1$ for $x_i \in V_1$ then $m'(x'_i) = 0$. If this is not the case, then we can change m' accordingly, decreasing the weight that way. It follows that $w(m') = n_0 + n_1$ where we have

$$\begin{aligned} n_0 &= |\{i \mid x_i \in V_1, m'(x_i) = 0\}| = |\{i \mid x_i \in V_1, m'(x_i) \neq m(x_i)\}| \\ n_1 &= |\{i \mid x_i \notin V_1, m'(x_i) = 1\}| = |\{i \mid x_i \notin V_1, m'(x_i) \neq m(x_i)\}|, \end{aligned}$$

which means that $w(m')$ equals $\text{hd}(m, m')$. Analogously, any model $s \in [\varphi]$ can be extended to a model $m' \in [\varphi']$ by putting $m'(x'_i) = 1$ if $x_i \in V_1$ and $s(x_i) = 0$, and $m'(x'_i) = 0$ for the remaining $x_i \in V_1$; thereby $w(m') = \text{hd}(m, s)$. Consequently, the optima in both problems correspond, that is, we get $\text{OPT}(\varphi') = \text{OPT}(\varphi, m)$. Hence we deduce $\text{hd}(m, m') = w(m') \leq r \text{OPT}(\varphi') = r \text{OPT}(\varphi, m)$. \square

Proposition 29 For every dual Horn constraint language $\Gamma \subseteq \text{iV}_2$ we have the reduction $\text{NSol}(\Gamma) \leq_{\text{AP}} \text{WeightedMinOnes}(\{x \vee y \vee \neg z, x \vee y\})$.

Proof Since $\{x \vee y \vee \neg z, x, \neg x\}$ is a base of iV_2 , by Corollary 13 it suffices to prove the reduction $\text{NSol}(\{x \vee y \vee \neg z, x, \neg x\}) \leq_{\text{AP}} \text{WeightedMinOnes}(\{x \vee y \vee \neg z, x \vee y\})$. To this end, first reduce $\text{NSol}(\{x \vee y \vee \neg z, x, \neg x\})$ to $\text{NSol}(x \vee y \vee \neg z)$ by Proposition 14 and then use Lemma 28. \square

Proposition 30 $\text{NSol}(\Gamma)$ is MinHornDeletion-hard for finite Γ with $\text{iV}_2 \subseteq \langle \Gamma \rangle$.

Proof For $\Gamma' := \{x \vee y \vee \neg z, x, \neg x\}$ we have $\text{MinHornDeletion} \equiv_{\text{AP}} \text{MinOnes}(\Gamma')$ by Proposition 27. Now it follows $\text{MinOnes}(\Gamma') \leq_{\text{AP}} \text{NSol}(\Gamma') \leq_{\text{AP}} \text{NSol}(\Gamma)$ using Lemma 24 and Corollary 13 on the assumption $\Gamma' \subseteq \text{iV}_2 \subseteq \langle \Gamma \rangle$. \square

5.2.4 Poly-APX-Hardness

Proposition 31 The problem $\text{NSol}(\Gamma)$ is poly-APX-hard for constraint languages Γ satisfying $\text{iN} \subseteq \langle \Gamma \rangle \subseteq \text{iI}_0$ or $\text{iN} \subseteq \langle \Gamma \rangle \subseteq \text{iI}_1$.

Proof The constraint language $\Gamma_1 := \{\text{even}^4, [x \rightarrow y], [x]\}$ is a base of iI_1 . $\text{MinOnes}(\Gamma_1)$ is poly-APX-hard by Theorem 2.14 of [22] and reduces to $\text{NSol}(\Gamma_1)$ by Lemma 24. Since $[x \rightarrow y] = [\text{dup}^3(x, y, 1)] = [\exists z(\text{dup}^3(x, y, z) \wedge z)]$, as well as $\langle\{\text{even}^4\}\rangle = iL$, $\langle\{\text{dup}^3\}\rangle = iN$, and $iL \subseteq iN$, we have the reductions

$$\text{NSol}(\Gamma_1) \leq_{\text{AP}} \text{NSol}(\Gamma_1 \cup \{\text{dup}^3\}) \leq_{\text{AP}} \text{NSol}(\{\text{even}^4, \text{dup}^3, x\}) \equiv_{\text{AP}} \text{NSol}(\{\text{dup}^3, x\})$$

by Corollary 13. The problem of finding feasible solutions of $\text{NSol}(\Gamma)$, where $iN \subseteq \langle\Gamma\rangle \subseteq iI_0$ or $iN \subseteq \langle\Gamma\rangle \subseteq iI_1$, is polynomial-time decidable. Indeed, such Γ is 0- or 1-valid, therefore the all-zero or all-one tuple is always guaranteed to exist as a feasible solution. Therefore Proposition 14 implies $\text{NSol}(\{\text{dup}^3, x\}) \equiv_{\text{AP}} \text{NSol}(\{\text{dup}^3\})$; the latter problem reduces to $\text{NSol}(\Gamma)$ because $\text{dup}^3 \in iN \subseteq \langle\Gamma\rangle$ and Corollary 13. \square

6 Finding Another Solution Closest to the Given One

In this section we study the optimization problem *NearestOtherSolution*. We first consider the polynomial-time cases and then the cases of higher complexity.

6.1 Polynomial-Time Cases

Since we cannot take advantage of clone closure, we must proceed differently. We use the following result based on a theorem by Baker and Pixley [5].

Proposition 32 (Jeavons et al. [19]) *Every bijunctive constraint $R(x_1, \dots, x_n)$ is equivalent to the conjunction $\bigwedge_{1 \leq i < j} R_{ij}(x_i, x_j)$, where R_{ij} is the projection of R to the coordinates i and j .*

Proposition 33 *If Γ is bijunctive ($\Gamma \subseteq iD_2$) then $\text{NOSol}(\Gamma)$ is in PO.*

Proof According to Proposition 32 we may assume that the formula φ is a conjunction of atoms $R(x, y)$ or a unary constraint $R(x, x)$ of the form $[x]$ or $[\neg x]$.

Unary constraints fix the value of the constrained variable and can be eliminated by propagating the value to the other clauses. For each of the remaining variables, x , we attempt to construct a model m_x of φ with $m_x(x) \neq m(x)$ such that $\text{hd}(m_x, m)$ is minimal among all models with this property. This can be done in polynomial time as described below. If the construction of m_x fails for every variable x , then m is the sole model of φ and the problem is not solvable. Otherwise choose one of the variables x for which $\text{hd}(m_x, m)$ is minimal and return m_x as second solution m' .

It remains to describe the computation of m_x . Initially we set $m_x(x)$ to $1 - m(x)$ and $m_x(y) := m(y)$ for all variables $y \neq x$, and mark x as flipped. If m_x satisfies all atoms we are done. Otherwise let $R(u, v)$ be an atom falsified by m_x . If u and v are marked as flipped, the construction fails, a model m_x with the property $m_x(x) \neq m(x)$ does not exist. Otherwise $R(u, v)$ contains a uniquely determined variable v not marked as flipped. Set $m_x(v) := 1 - m(v)$, mark v as flipped, and repeat this step. This process terminates after flipping every variable at most once. \square

Proposition 34 *If $\Gamma \subseteq \text{iS}_{00}^k$ or $\Gamma \subseteq \text{iS}_{10}^k$ for some $k \geq 2$ then $\text{NOSol}(\Gamma)$ is in PO.*

Proof We perform the proof only for iS_{00}^k . Proposition 16 implies the same result for iS_{10}^k .

The co-clone iS_{00}^k is generated by $\Gamma' := \{\text{or}^k, [x \rightarrow y], [x], [\neg x]\}$. In fact, Γ' is even a *plain base* of iS_{00}^k [12], meaning that every relation in Γ can be expressed as a conjunctive formula over relations in Γ' , without existential quantification or explicit equalities. Hence we may assume that φ is given as a conjunction of Γ' -atoms.

Note that $x \vee y$ is a polymorphism of Γ' , i.e., for any two solutions m_1, m_2 of φ their disjunction $m_1 \vee m_2$ – defined by $(m_1 \vee m_2)(x) = m_1(x) \vee m_2(x)$ for all x – is also a solution of φ . Therefore we get the optimal solution m' of an instance (φ, m) by flipping in m either some ones to zeros or some zeros to ones, but not both. To see this, assume the optimal solution m' flips both ones and zeros. Then $m' \vee m$ is a solution of φ that is closer to m than m' , which contradicts the optimality of m' .

Unary constraints fix the value of the constrained variable and can be eliminated by propagating the value to the other clauses (including removal of disjunctions containing implied positive literals and shortening disjunctions containing implied negative literals). This propagation does not lead to contradictions since m is a model of φ . For each of the remaining variables, x , we attempt to construct a model m_x of φ with $m_x(x) \neq m(x)$ such that $\text{hd}(m_x, m)$ is minimal among all models with this property. This can be done in polynomial time as described below. If the construction of m_x fails for every variable x , then m is the sole model of φ and the problem is not solvable. Otherwise choose one of the variables x for which $\text{hd}(m_x, m)$ is minimal and return m_x as second solution m' .

It remains to describe the computation of m_x . If $m(x) = 0$, we flip x to 1 and propagate this change iteratively along the implications, i.e., if $x \rightarrow y$ is a constraint of φ and $m(y) = 0$, we flip y to 1 and iterate. This kind of flip never invalidates any disjunctions, it could only lead to contradictions with conditions imposed by negative unit clauses (and since their values were propagated before such a contradiction would be immediate). For $m(x) = 1$ we proceed dually, flipping x to 0, removing x from disjunctions if applicable, and propagating this change backward along implications $y \rightarrow x$ where $m(y) = 1$. This can possibly lead to immediate inconsistencies with already inferred unit clauses, or it can produce contradictions through empty disjunctions, or it can create the necessity for further flips from 0 to 1 in order to obtain a solution (because in a disjunctive atom all variables with value 1 have been flipped, and thus removed). In all these three cases the resulting assignment does not satisfy φ , and there is no model that differs from m in x and that can be obtained by flipping in one way only. Otherwise, the resulting assignment satisfies φ , and this is the desired m_x . Our process terminates after flipping every variable at most once, since we flip only in one way (from zeros to ones or from ones to zeros). Thus, m_x is computable in polynomial time. \square

6.2 Hard Cases

Proposition 35 *Let Γ be a constraint language. If $iI_1 \subseteq \langle \Gamma \rangle$ or $iI_0 \subseteq \langle \Gamma \rangle$ holds then it is NP-complete to decide whether a feasible solution for $\text{NOSol}(\Gamma)$ exists. Otherwise, $\text{NOSol}(\Gamma) \in \text{poly-APX}$.*

Proof Finding a feasible solution to $\text{NOSol}(\Gamma)$ corresponds exactly to the decision problem $\text{AnotherSAT}(\Gamma)$ which is NP-hard if and only if $iI_1 \subseteq \langle \Gamma \rangle$ or $iI_0 \subseteq \langle \Gamma \rangle$ according to Juban [20]. If $\text{AnotherSAT}(\Gamma)$ is polynomial-time decidable, we can always find a feasible solution for $\text{NOSol}(\Gamma)$ if it exists. Obviously, every feasible solution is an n -approximation of the optimal solution, where n is the number of variables in the input. \square

6.2.1 Tightness Results

It will be convenient to consider the following decision problem asking for another solution that is not the complement, i.e., that does not have maximal distance from the given one.

Problem: $\text{AnotherSAT}_{\text{nc}}(\Gamma)$

Input: A conjunctive formula φ over relations from Γ and an assignment m satisfying φ .

Question: Is there another satisfying assignment m' of φ , different from m , such that $\text{hd}(m, m') < n$, where n is the number of variables in φ ?

Remark 36 $\text{AnotherSAT}_{\text{nc}}(\Gamma)$ is NP-complete for $iI_0 \subseteq \langle \Gamma \rangle$ and $iI_1 \subseteq \langle \Gamma \rangle$, since already $\text{AnotherSAT}(\Gamma)$ is NP-complete for these cases, as shown in [20]. Moreover, $\text{AnotherSAT}_{\text{nc}}(\Gamma)$ is polynomial-time decidable if Γ is Horn ($\Gamma \subseteq iE_2$), dual Horn ($\Gamma \subseteq iV_2$), bijunctive ($\Gamma \subseteq iD_2$), or affine ($\Gamma \subseteq iL_2$), for the same reason as for $\text{AnotherSAT}(\Gamma)$: For each variable x_i we flip the value of $m[i]$, substitute $\bar{m}(x_i)$ for x_i , and construct another satisfying assignment if it exists. Consider now the solutions which we get for every variable x_i . Either there is no solution for any variable, then $\text{AnotherSAT}_{\text{nc}}(\Gamma)$ has no solution; or there are only the solutions which are the complement of m , then $\text{AnotherSAT}_{\text{nc}}(\Gamma)$ has no solution as well; or else we get a solution m' with $\text{hd}(m, m') < n$, leading also to a solution for $\text{AnotherSAT}_{\text{nc}}(\Gamma)$. Hence, taking into account Proposition 38 below, we obtain a dichotomy result also for $\text{AnotherSAT}_{\text{nc}}(\Gamma)$.

Note that $\text{AnotherSAT}_{\text{nc}}(\Gamma)$ is not compatible with existential quantification. Let $\varphi(y, x_1, \dots, x_n)$ with model m be an instance of $\text{AnotherSAT}_{\text{nc}}(\Gamma)$ and let m' be a solution satisfying $\text{hd}(m, m') < n + 1$. Now consider the formula $\varphi_1(x_1, \dots, x_n) = \exists y \varphi(y, x_1, \dots, x_n)$, obtained by existentially quantifying the variable y , and the tuples m_1 and m'_1 obtained from m and m' by omitting the first component. Both, m_1 and m'_1 , are still solutions of φ' , but we cannot guarantee $\text{hd}(m_1, m'_1) < n$. Hence we need the equivalent of Proposition 15 for this problem, whose proof is analogous.

Proposition 37 *The reduction $\text{AnotherSAT}_{\text{nc}}(\Gamma') \leq_m \text{AnotherSAT}_{\text{nc}}(\Gamma)$ holds for all constraint languages Γ and Γ' satisfying $\Gamma' \subseteq \langle \Gamma \rangle_{\wedge}$.*

Proposition 38 *If a constraint language Γ satisfies $\langle \Gamma \rangle = \text{iI}$ or $\text{iN} \subseteq \langle \Gamma \rangle \subseteq \text{iN}_2$, then $\text{AnotherSAT}_{\text{nc}}(\Gamma)$ is NP-complete.*

Proof Containment in NP is clear, it remains to show hardness. Since the problem $\text{AnotherSAT}_{\text{nc}}$ is not compatible with existential quantification, we cannot use clone theory, but have to consider the three co-clones iN_2 , iN , and iI separately and make use of minimal weak bases.

Case $\langle \Gamma \rangle = \text{iN}$ Putting $R := \{000, 101, 110\}$, we present a reduction from the problem $\text{AnotherSAT}(\{R\})$, which is NP-hard [20] as $\langle \{R\} \rangle = \text{iI}_0$. The problem remains NP-complete if we restrict it to instances $(\varphi, \mathbf{0})$, since R is 0-valid and any given model m other than the constant 0-assignment admits the trivial solution $m' = \mathbf{0}$. Thus we can perform a reduction from this restricted problem.

Consider the relation $R_{\text{iN}} = \{0000, 1010, 1100, 1111, 0101, 0011\}$. Given a formula φ over R , we construct a formula ψ over R_{iN} by replacing every constraint $R(x, y, z)$ with a new constraint $R_{\text{iN}}(x, y, z, w)$, where w is a new global variable. Moreover, we set m to the constant 0-assignment. This construction is a many-one reduction from the restricted version of $\text{AnotherSAT}(\{R\})$ to $\text{AnotherSAT}_{\text{nc}}(\{R_{\text{iN}}\})$.

To see this, observe that the tuples in R_{iN} that have a 0 in the last coordinate are exactly those in $R \times \{0\}$. Thus any solution of φ can be extended to a solution of ψ by assigning 0 to w . Conversely we observe that any solution m' of the $\text{AnotherSAT}_{\text{nc}}(\{R_{\text{iN}}\})$ -instance $(\psi, \mathbf{0})$ is different from $\mathbf{0}$ and $\mathbf{1}$. As R_{iN} is complementary, we may assume $m'(w) = 0$. Then m' restricted to the variables of φ solves the $\text{AnotherSAT}(\{R\})$ -instance $(\varphi, \mathbf{0})$.

Finally, observe that R_{iN} is a minimal weak base and Γ is a base of the co-clone iN , therefore we have $R_{\text{iN}} \in \langle \Gamma \rangle_{\wedge}$ by Theorem 1. Now the NP-hardness of $\text{AnotherSAT}_{\text{nc}}(\Gamma)$ follows from the one of $\text{AnotherSAT}_{\text{nc}}(\{R_{\text{iN}}\})$ by Proposition 37.

Case $\langle \Gamma \rangle = \text{iN}_2$ We give a reduction from $\text{AnotherSAT}_{\text{nc}}(\{R_{\text{iN}}\})$, which is NP-hard by the previous case. By Theorem 1, $\langle \Gamma \rangle_{\wedge}$ contains the relation $R_{\text{iN}_2} = \{m\bar{m} \mid m \in R_{\text{iN}}\}$. For an R_{iN} -formula $\varphi(x_1, \dots, x_n)$, we construct a corresponding R_{iN_2} -formula $\psi(x_1, \dots, x_n, x'_1, \dots, x'_n)$ by replacing every constraint $R_{\text{iN}}(x, y, z, w)$ with a new constraint $R_{\text{iN}_2}(x, y, z, w, x', y', z', w')$. Assignments m for φ extend to assignments M for ψ by setting $M(x') := \bar{m}(x)$. Conversely, assignments for ψ yield assignments for φ by restricting them to the variables in φ . Because every variable x_1, \dots, x_n assigned by models of φ actually occurs in some R_{iN} -atom in φ and hence in some R_{iN_2} -atom of ψ , and because of the structure of R_{iN_2} , any model of ψ distinct from M and \bar{M} restricts to a model of φ other than m or \bar{m} . Consequently, this construction is again a reduction from $\text{AnotherSAT}_{\text{nc}}(\{R_{\text{iN}}\})$ to $\text{AnotherSAT}_{\text{nc}}(\{R_{\text{iN}_2}\})$, reducing itself to $\text{AnotherSAT}_{\text{nc}}(\Gamma)$ by Proposition 37.

Case $\langle \Gamma \rangle = iI$ We proceed as in Case $\langle \Gamma \rangle = iN$, but use $R_{iI} = \{0000, 0011, 0101, 1111\}$ instead of R_{iN} , and $\{000, 011, 101\}$ for R . Note that the R_{iI} -tuples with first coordinate 0 are exactly those in $\{0\} \times R$. The relation R_{iI} is not complementive, but (as every variable assigned by any model of ψ occurs in some atomic R_{iI} -constraint) the only solution m' such that $m'(w) = 1$ is the constant 1-assignment, which is ruled out by the requirement $\text{hd}(m, m') < n$. Hence we may again assume $m'(w) = 0$. \square

Proposition 39 *For a constraint language Γ satisfying $\langle \Gamma \rangle = iI$ or $iN \subseteq \langle \Gamma \rangle \subseteq iN_2$ and any $\varepsilon > 0$ there is no polynomial-time $n^{1-\varepsilon}$ -approximation algorithm for $\text{NOSol}(\Gamma)$, unless $P = NP$.*

Proof Assume there is a constant $\varepsilon > 0$ with a polynomial-time $n^{1-\varepsilon}$ -approximation algorithm for $\text{NOSol}(\Gamma)$. We show how to use this algorithm to solve $\text{AnotherSAT}_{nc}(\Gamma)$ in polynomial time. Proposition 38 completes the proof.

Let (φ, m) be an instance of $\text{AnotherSAT}_{nc}(\Gamma)$ with n variables. If $n = 1$, then we reject the instance. Otherwise, we construct a new formula φ' and a new assignment m' as follows. Let k be the smallest integer greater than $1/\varepsilon$. Choose a variable x of φ and introduce $n^k - n$ new variables x^i for $i = 1, \dots, n^k - n$. For every $i \in \{1, \dots, n^k - n\}$ and every constraint $R(y_1, \dots, y_\ell)$ in φ , such that $x \in \{y_1, \dots, y_\ell\}$, construct a new constraint $R(z_1^i, \dots, z_\ell^i)$ by $z_j^i = x^i$ if $y_j = x$ and $z_j^i = y_j$ otherwise; add all the newly constructed constraints to φ in order to get φ' . Moreover, we extend m to a model of φ' by setting $m'(x^i) = m(x)$. Now run the $n^{1-\varepsilon}$ -approximation algorithm for $\text{NOSol}(\Gamma)$ on (φ', m') . If the answer is \bar{m}' then reject, otherwise accept.

We claim that the algorithm described above is a correct polynomial-time algorithm for the decision problem $\text{AnotherSAT}_{nc}(\Gamma)$ when Γ is complementive. Polynomial runtime is clear. It remains to show its correctness. If the only solutions to φ are m and \bar{m} , then, as $n > 1$, the only models of φ' are m' and \bar{m}' . Hence the approximation algorithm must answer \bar{m}' and the output is correct. Now assume that there is a satisfying assignment m_s different from m and \bar{m} . The relation $[\varphi]$ is complementive, hence we may assume that $m_s(x) = m(x)$. It follows that φ' has a satisfying assignment m'_s for which $0 < \text{hd}(m'_s, m') < n$ holds. But then the approximation algorithm must find a satisfying assignment m'' for φ' with $\text{hd}(m', m'') < n \cdot (n^k)^{1-\varepsilon} = n^{k(1-\varepsilon)+1}$. Since the inequality $k > 1/\varepsilon$ holds, it follows that $\text{hd}(m', m'') < n^k$. Consequently, m'' is not the complement of m' and the output of our algorithm is again correct.

When Γ is not complementive but both 0-valid and 1-valid ($\langle \Gamma \rangle = iI$), we perform the expansion algorithm described above for each variable of the formula φ and reject if the result is the complement for each run. The runtime remains polynomial. If $[\varphi] = \{m, \bar{m}\}$, then indeed every run results in the corresponding \bar{m}' , and we correctly reject. Otherwise, we have a model $m_s \in [\varphi] \setminus \{m, \bar{m}\}$, so there is a variable x of φ , where $m_s(x) \neq \bar{m}(x)$, i.e. $m_s(x) = m(x)$. For this instance

(φ', m') the approximation algorithm does not return $\overline{m'}$, wherefore we correctly accept. □

6.2.2 MinDistance-Equivalent Cases

In this section we show that affine co-clones lead to problems equivalent to MinDistance. We thereby add the missing details to the rather superficial treatment of this matter given in [6].

Lemma 40 *For affine constraint languages Γ ($\Gamma \subseteq \text{iL}_2$) we have $\text{NOSol}(\Gamma) \leq_{\text{AP}} \text{MinDistance}$.*

Proof Let the formula φ and the satisfying assignment m be an instance of $\text{NOSol}(\Gamma)$ over the variables x_1, \dots, x_n . The input φ can be written as $Ax = b$, with m being a solution of this affine system. A tuple m' is a solution of $Ax = b$ if and only if it can be written as $m' = m + m_0$ where m_0 is a solution of $Ax = 0$. The Hamming distance is invariant with respect to affine translations: namely we have $\text{hd}(m', m) = \text{hd}(m' + m'', m + m'')$ for any tuple m'' , in particular, for $m'' = -m$ we obtain $\text{hd}(m', m) = \text{hd}(m' - m, 0)$. Therefore $m' \neq m$ is a solution of $Ax = b$ with minimal Hamming distance to m if and only if $m_0 = m' - m$ is a non-zero solution of the homogeneous system $Ax = 0$ with minimum Hamming weight. Hence, the problem $\text{NOSol}(\Gamma)$ for affine languages Γ is equivalent to computing the non-trivial solutions of homogeneous systems with minimal weight, which is exactly the MinDistance problem. □

We need to express an affine sum of even number of variables by means of the minimal weak base for each of the affine co-clones. In the following lemma, the existentially quantified variables are uniquely determined, therefore the existential quantifiers serve only to hide superfluous variables and do not pose any problems as they were mentioned before.

Lemma 41 *For every $n \in \mathbb{N}$, $n \geq 1$, the constraint $x_1 \oplus x_2 \oplus \dots \oplus x_{2n} = 0$ can be equivalently expressed by each of the following formulas:*

1. $\exists y_0, \dots, y_n \left(\begin{array}{l} y_0 = 0 \wedge y_n = 0 \wedge \\ R_{\text{iL}}(y_0, x_1, x_2, y_1) \wedge \\ R_{\text{iL}}(y_1, x_3, x_4, y_2) \wedge \dots \wedge R_{\text{iL}}(y_{n-1}, x_{2n-1}, x_{2n}, y_n) \end{array} \right),$
2. $\exists y_0, \dots, y_{2n} \left(\begin{array}{l} R_{\text{iL}_0}(y_0, x_1, y_1, y_0) \wedge \\ R_{\text{iL}_0}(y_1, x_2, y_2, y_0) \wedge \dots \wedge R_{\text{iL}_0}(y_{2n-1}, x_{2n}, y_{2n}, y_{2n}) \end{array} \right),$
3. $\exists y_0, \dots, y_{2n} \left(\begin{array}{l} R_{\text{iL}_1}(y_0, x_1, y_1, y_0) \wedge \\ R_{\text{iL}_1}(y_1, x_2, y_2, y_0) \wedge \dots \wedge R_{\text{iL}_1}(y_{2n-1}, x_{2n}, y_{2n}, y_{2n}) \end{array} \right),$
4. $\exists y_0, \dots, y_n, z_0, \dots, z_n, w_1, \dots, w_{2n} \left(\begin{array}{l} y_0 = 0 \wedge y_n = 0 \wedge \\ R_{\text{iL}_3}(y_0, x_1, x_2, y_1, z_0, w_1, w_2, z_1) \wedge \\ R_{\text{iL}_3}(y_1, x_3, x_4, y_2, z_1, w_3, w_4, z_2) \wedge \dots \wedge \\ R_{\text{iL}_3}(y_{n-1}, x_{2n-1}, x_{2n}, y_n, z_{n-1}, w_{2n-1}, w_{2n}, z_n) \end{array} \right),$

$$5. \quad \exists y_0, \dots, y_{2n}, z_0, \dots, z_{2n}, w_1, \dots, w_{2n} \left(\begin{array}{l} R_{iL_2}(y_0, x_1, y_1, z_0, w_1, z_1, y_0, z_0) \wedge \\ R_{iL_2}(y_1, x_2, y_2, z_1, w_2, z_2, y_0, z_0) \wedge \dots \wedge \\ R_{iL_2}(y_{2n-1}, x_{2n}, y_{2n}, z_{2n-1}, w_{2n}, z_{2n}, y_{2n}, z_{2n}) \end{array} \right),$$

where the number of existentially quantified variables is linearly bounded in the length of the constraint. Note moreover that in each case any model of $x_1 \oplus x_2 \oplus \dots \oplus x_{2n} = 0$ uniquely determines the values of the existentially quantified variables.

Proof Write out the constraint relations following the existential quantifiers as (conjunctions of) equalities. From this uniqueness of valuations for the existentially quantified variables is easy to see, and likewise that any model of $\bigoplus_{i=1}^{2n} x_i = 0$ also satisfies each of the formulas 1. up to 5. Adding up the equalities behind the existential quantifiers shows the converse direction. \square

The following lemma shows that *MinDistance* is AP-equivalent to a restricted version, containing only constraints generating the minimal weak base, for each co-clone in the affine case.

Lemma 42 *For each co-clone $\mathcal{B} \in \{iL, iL_0, iL_1, iL_2, iL_3\}$ we have $\text{MinDistance} \leq_{\text{AP}} \text{NOSol}(\{R_{\mathcal{B}}, [\neg x]\})$.*

Proof Consider a co-clone $\mathcal{B} \in \{iL, iL_0, iL_1, iL_2, iL_3\}$ and a *MinDistance*-instance represented by a matrix $A \in \mathbb{Z}_2^{k \times l}$. If one of the columns of A , say the i -th, is zero, then the i -th unit vector is an optimal solution to this instance with optimal value 1. Hence, we assume from now on that none of the columns equals a zero vector.

Every row of A expresses the fact that a sum of $n \leq l$ variables equals zero. If n is odd, we extend this sum to one with $n + 1$ summands, thereby introducing a new variable v , which we existentially quantify and confine to zero using a unary $[\neg x]$ -constraint. Then we replace the expanded sum by the existential formula from Lemma 41 corresponding to the co-clone \mathcal{B} under consideration. This way we have introduced only linearly many new variables in l for every row, and for any feasible solution for the *MinDistance*-problem the values of the existential variables needed to encode it are uniquely determined. Thus, taking the conjunction over all these formulas we only have a linear growth in the size of the instance.

Next, we show how to deal with the existential quantifiers: First we transform the expression to prenex normal form getting a formula ψ of the form

$$\exists y_1, \dots, y_p \varphi(y_1, \dots, y_p, x_1, \dots, x_l),$$

which holds if and only if $Ax = \mathbf{0}$ for $x = (x_1, \dots, x_l)$. We use the same blow-up construction regarding x_1, \dots, x_l as in Proposition 7 and Lemma 11 to make the influence of y_1, \dots, y_p on the Hamming distance negligible. For this we put $J := \{1, \dots, t\}$ and introduce new variables x_i^j where $1 \leq i \leq l$ and $j \in J$. If u is among x_1, \dots, x_l , we define its blow-up set to be $B(u) = \{x_i^j \mid j \in J\}$, otherwise, for $u \in \{y_1, \dots, y_p\}$, we set $B(u) = \{u\}$. Now for each atom $R(u_1, \dots, u_q)$ of φ we form the set of atoms $\{R(u'_1, \dots, u'_q) \mid (u'_1, \dots, u'_q) \in \prod_{i=1}^q B(u_i)\}$, and define

the quantifier free formula φ' to be the conjunction of all atoms in the union of these sets. Note that this construction takes time polynomial in the size of ψ and hence in the size of the input MinDistance-instance whenever t is polynomial in the input size because the atomic relations in ψ are at most octonary.

If s is an assignment of values to x making $Ax = \mathbf{0}$ true, we define $s'(x_i^j) := s(x_i)$ and extend this to a model of φ' assigning the uniquely determined values to y_1, \dots, y_p . Let m' be the model arising in this way from the zero assignment m . If s' is any model of φ' , then for every $1 \leq i \leq l$, all $j \in J$ and each atom $R(u_1, \dots, u_q)$ of φ , s' satisfies, in particular, the conjunction $R(u'_1, \dots, u'_q) \wedge R(u''_1, \dots, u''_q)$ where for $u \in \{u_1, \dots, u_q\}$ we have $u' = u'' = u$ if $u \in \{y_1, \dots, y_p\}$, $u' = x_i^1, u'' = x_i^j$ if $u = x_i$, and $u' = u'' = x_k^1$ if $u = x_k$ for some $k \in \{1, \dots, l\} \setminus \{i\}$. Hence, the vectors $(s'(x_1^1), \dots, s'(x_l^1))$ and $(s'(x_1^1), \dots, s'(x_{i-1}^1), s'(x_i^j), s'(x_{i+1}^1), \dots, s'(x_l^1))$ both belong to the kernel of A and so does their difference, which is $s'(x_i^j) - s'(x_i^1)$ times the i -th unit vector. As the i -th column of A is non-zero, we must have $s'(x_i^j) = s'(x_i^1)$. This also implies that if s' is zero on x_1^1, \dots, x_n^1 , then it must be zero on all x_i^j ($1 \leq i \leq l, j \in J$) and thus it must coincide with m' . Therefore, every feasible solution to the NOSol-instance (φ', m') yields a non-zero vector $(s'(x_1^1), \dots, s'(x_l^1))$ in the kernel of A .

Further, if s' is an r -approximation to an optimal solution, i.e., if $\text{hd}(s', m') \leq r \text{OPT}(\varphi', m')$, then, as $s'(x_i^1) = s'(x_i^j)$ holds for all $j \in J$ and all $1 \leq i \leq l$, we obtain a solution to the MinDistance problem with Hamming weight w such that $t \cdot w \leq \text{hd}(s', m')$. Also, any optimal solution to the MinDistance-instance can be extended to a not-necessarily optimal solution s'' of (φ', m') , for which one can bound the distance to m' as follows: $\text{OPT}(\varphi', m') \leq \text{hd}(s'', m') \leq t \cdot \text{OPT}(A) + p$. Combining these inequalities, we can infer $t \cdot w \leq r \cdot t \cdot \text{OPT}(A) + r \cdot p$, or $w \leq \text{OPT}(A) \cdot (r + r/\text{OPT}(A) \cdot p/t)$. We noted above that p is linearly bounded in the size of the input, thus choosing t quadratic in the size of the input bounds w by $\text{OPT}(A)(r + o(1))$, whence we have an AP-reduction with $\alpha = 1$. \square

Lemma 43 *For constraint languages Γ , where one can decide the existence of and also find a feasible solution of $\text{NOSol}(\Gamma)$ in polynomial time, we have the reduction $\text{NOSol}(\Gamma) \leq_{\text{AP}} \text{NOSol}((\Gamma \setminus \{[x], [\neg x]\}) \cup \{\approx\})$.*

Proof If an instance (φ, m) does not have feasible solutions, then it does not have nearest other solutions either. So we map it to the generic unsolvable instance \perp . Consider now formulas φ over variables x_1, \dots, x_n with models m where some feasible solution $s_0 \neq m$ exists (and has been computed).

We can assume φ to be of the form $\psi(x_1, \dots, x_n) \wedge \bigwedge_{i \in I_1} [x_i] \wedge \bigwedge_{i \in I_0} [\neg x_i]$, where ψ is a $(\Gamma \setminus \{[x], [\neg x]\})$ -formula and $I_1, I_0 \subseteq \{1, \dots, n\}$. We transform φ to $\varphi' := \psi(x_1, \dots, x_n) \wedge \bigwedge_{i \in I_1} x_i \approx y_1 \wedge \bigwedge_{i \in I_0} x_i \approx z_1 \wedge \bigwedge_{i=1}^{1+n^2} (y_i \approx y_1 \wedge z_i \approx z_1)$ and extend models of φ to models of φ' in the natural way. Conversely, if s' is a model of φ' and $s'(y_i) = 1$ and $s'(z_i) = 0$ hold for all $1 \leq i \leq 1 + n^2$, then we can restrict it to a model of φ . Other models of φ' are not optimal and are mapped to s_0 . It is not hard to see that this provides an AP-reduction with $\alpha = 1$. \square

Proposition 44 *For every constraint language Γ satisfying $iL \subseteq \langle \Gamma \rangle \subseteq iL_2$ we have $\text{MinDistance} \equiv_{\text{AP}} \text{NOSol}(\Gamma)$.*

Proof Since we lack compatibility with existential quantification, we shall deal with each co-clone $\mathcal{B} = \langle \Gamma \rangle$ in the interval $\{iL, iL_0, iL_1, iL_2, iL_3\}$ separately. First we perform the reduction from Lemma 42 to $\text{NOSol}(\{R_{\mathcal{B}}, [\neg x]\})$. We need to find a reduction to $\text{NOSol}(\{R_{\mathcal{B}}\})$ as this reduces to $\text{NOSol}(\Gamma)$ by Proposition 15 and Theorem 1.

This is simple in the case of iL_0 and iL_2 since $[\neg x] = \{x \mid R_{iL_0}(x, x, x, x)\} \in \langle\langle R_{iL_0} \rangle\rangle_{\wedge}$ (see Proposition 15) and $[\neg x] = \{x \mid \exists y(R_{iL_2}(x, x, x, y, y, y, x, y))\}$, where the existential quantifier can be handled by an AP-reduction with $\alpha = 1$ which drops the quantifier and extends every model by assigning 1 to all previously existentially quantified variables. Thereby (optimal) distances between models do not change at all.

In the remaining cases, we reduce $\text{NOSol}(\{R_{\mathcal{B}}, [\neg x]\}) \leq_{\text{AP}} \text{NOSol}(\{R_{\mathcal{B}}, [x], [\neg x]\})$ and the latter to $\text{NOSol}(\{R_{\mathcal{B}}, \approx\})$ by Lemma 43, which now has to be reduced to $\text{NOSol}(\{R_{\mathcal{B}}\})$. This is obvious for $\mathcal{B} = iL$ where equality constraints $x \approx y$ can be expressed as $R_{iL}(x, x, x, y) \in \langle\langle R_{iL} \rangle\rangle_{\wedge}$ (cf. Proposition 15). For iL_1 the same can be done using the formula $\exists z(R_{iL_1}(x, y, z, z))$, where the existential quantifier can be removed by the same sort of simple AP-reduction with $\alpha = 1$ as employed for iL_2 . Finally, for iL_3 we want to express equality as $\exists u \exists v(R_{iL_3}(x, x, x, y, u, u, u, v))$. Here, in an AP-reduction, the quantifiers cannot simply be disregarded, as the values of the existentially quantified variables are not constant for all models. They are uniquely determined by the values of x and y for each particular model, though, which allows us to perform a similar blow-up construction as in the proof of Lemma 42.

In more detail, given a $\{R_{iL_3}, \approx\}$ -formula ψ containing variables x_1, \dots, x_l , first note that each atomic R_{iL_3} -constraint $R_{iL_3}(x_1, \dots, x_8)$ can be represented as a linear system of equations, namely $\bigoplus_{i=1}^4 x_i = 0$ and $x_i \oplus x_{i+4} = 1$ for $1 \leq i \leq 4$. Since equalities $x_i \approx x_j$ can be written as $x_i \oplus x_j = 0$, the formula ψ is equivalent to an expression of the form $Ax = b$ where $x = (x_1, \dots, x_l)$. Replacing each equality constraint by the existential formula above and bringing the result into prenex normal form, we get a formula $\exists y_1, \dots, y_p(\varphi(y_1, \dots, y_p, x_1, \dots, x_l))$, which is equivalent to ψ and where φ is a conjunctive $\{R_{iL_3}\}$ -formula. By construction any two models of φ that agree on x_1, \dots, x_l must coincide. Thus, introducing variables x_i^j for $1 \leq i \leq l$ and $j \in J := \{1, \dots, t\}$ and defining φ' in literally the same way as in the proof of Lemma 42, any model s of ψ yields a model s' of φ' by putting $s'(x_i^j) := s(x_i)$ for $1 \leq i \leq l$ and $j \in J$ and extending this with the unique values for y_1, \dots, y_p satisfying $\varphi(y_1, \dots, y_p, x_1, \dots, x_l)$. In this way we obtain a model m' of φ' from a given solution m of ψ . Besides, if s' is any model of φ' , then as in Lemma 42, the vectors $(s'(x_1^1), \dots, s'(x_l^1))$ and $(s'(x_1^1), \dots, s'(x_{i-1}^1), s'(x_i^j), s'(x_{i+1}^1), \dots, s'(x_l^1))$ both satisfy ψ , and thus their difference is in the kernel of A . Since the variable x_i occurs in at least one of the atoms of ψ , the i -th column of A is non-zero, implying that $s'(x_i^j) = s'(x_i^1)$ for $j \in J$ and all $1 \leq i \leq l$. Thus, any model $s' \neq m'$ of φ' gives a model $s \neq m$ of ψ by defining $s(x_i) := s'(x_i^1)$ for all $1 \leq i \leq l$.

The presented construction is an AP-reduction with $\alpha = 1$, which can be proven completely analogously to the last paragraph of the proof of Lemma 42, choosing t quadratic in the size of ψ . \square

6.2.3 MinHornDeletion-Equivalent Cases

As in Proposition 38 the need to use conjunctive closure instead of $\langle \rangle$ causes a case distinction in the proof of the following result, which is the dual variant of [6, Lemma 16]. Correspondingly, Lemma 46 then replaces [6, Lemma 17].

Lemma 45 *If Γ is exactly dual Horn ($iV \subseteq \langle \Gamma \rangle \subseteq iV_2$) then one of the following relations is in $\langle \Gamma \rangle_\wedge$: $[x \rightarrow y]$, $[x \rightarrow y] \times \{0\}$, $[x \rightarrow y] \times \{1\}$, or $[x \rightarrow y] \times \{01\}$.*

Proof The co-clone $\langle \Gamma \rangle$ is equal to iV , iV_0 , iV_1 , or iV_2 . In the case $\langle \Gamma \rangle = iV$ the relation R_{iV} belongs to $\langle \Gamma \rangle_\wedge$ by Theorem 1; because of $R_{iV}(y, y, y, x) = [x \rightarrow y]$ we have $[x \rightarrow y] \in \langle R_{iV} \rangle_\wedge \subseteq \langle \Gamma \rangle_\wedge$. The case $\langle \Gamma \rangle = iV_1$ leads to $[x \rightarrow y] \times \{1\} \in \langle \Gamma \rangle_\wedge$ in an analogous manner. The cases $\langle \Gamma \rangle = iV_0$ and $\langle \Gamma \rangle = iV_2$ lead to $[x \rightarrow y] \times \{0\} \in \langle \Gamma \rangle_\wedge$ and $[x \rightarrow y] \times \{01\} \in \langle \Gamma \rangle_\wedge$, respectively, by observing that $[S_1(y, y, x)] = [S_0(\neg y, \neg y, \neg x, \neg y)] = [(\neg y \wedge \neg y) \approx (\neg y \wedge \neg x)] = [x \rightarrow y]$. \square

Lemma 46 *If Γ is exactly dual Horn ($iV \subseteq \langle \Gamma \rangle \subseteq iV_2$), then the problem $\text{NOSol}(\Gamma)$ is MinHornDeletion-hard.*

Proof There are four cases to consider, namely $\langle \Gamma \rangle \in \{iV, iV_0, iV_1, iV_2\}$. For simplicity we only present the situation where $\langle \Gamma \rangle = iV_1$; the case $\langle \Gamma \rangle = iV_2$ is very similar, and the other possibilities are even less complicated. At the end we shall give a few hints how to adapt the proof in these cases.

The basic structure of the proof is as follows: we choose a suitable weak base of iV_1 consisting of an irredundant relation R_1 , and identify a relation $H_1 \in \langle \{R_1\} \rangle_\wedge$ which allows us to encode a sufficiently complicated variant of the MinOnes-problem into $\text{NOSol}(\{H_1\})$. Thus by Theorem 1 and Lemma 45 we have $H_1 \in \langle \{R_1\} \rangle_\wedge \subseteq \langle \Gamma \rangle_\wedge$ and $[x \rightarrow y] \times \{1\} \in \langle \Gamma \rangle_\wedge$, wherefore Proposition 15 implies $\text{NOSol}(\Gamma') \leq_{\text{AP}} \text{NOSol}(\Gamma)$ where $\Gamma' = \{H_1, [x \rightarrow y] \times \{1\}\}$. According to [22, Theorem 2.14(4)], MinHornDeletion is equivalent to MinOnes(Δ) for constraint languages Δ being dual Horn, not 0-valid and not implicative hitting set bounded+ with any finite bound, that is, if $\langle \Delta \rangle \in \{iV_1, iV_2\}$. The key point of the construction is to choose R_1 and H_1 in such a way that we can find a relation G_1 satisfying $iV_1 \subseteq \langle \{G_1\} \rangle \subseteq iV_2$ and $((G_1 \times \{1\}) \cup \{0\}) \times \{1\} = H_1$. The latter property will allow us to prove an AP-reduction $\text{MinHornDeletion} \equiv_{\text{AP}} \text{MinOnes}(\{G_1\}) \leq_{\text{AP}} \text{NOSol}(\Gamma')$, completing the chain.

We first check that $R_1 = V_1 \circ \langle \chi_4 \rangle$ satisfies $\langle \{R_1\} \rangle = iV_1$: namely, by construction, this relation is preserved by the disjunction and by the constant operation with value 1, i.e., $\langle R_1 \rangle \subseteq iV_1$. This inclusion cannot be proper, since $0 \notin R_1$ ($\langle R_1 \rangle \not\subseteq iI_0$) and $x \vee (y \wedge z) \notin R_1$ while $x = (e_1 \circ \beta) \vee (e_4 \circ \beta)$, $y = (e_1 \circ \beta) \vee (e_2 \circ \beta)$ and $z = (e_1 \circ \beta) \vee (e_3 \circ \beta)$ belong to $V_1 \circ \langle \chi_4 \rangle$ (cf. before Theorem 2 for the notation), i.e.

the generating function $(x, y, z) \mapsto x \vee (y \wedge z)$ of the clone S_{00} [13, Figure 2, p. 8] fails to be a polymorphism of R_1 . For later we note that when β is chosen such that the coordinates of χ_4 are ordered lexicographically (and we are going to assume this from now on), then this failure can already be observed within the first seven coordinates of R_1 . Now according to Theorem 2, the sedenary relation $R_1 := \vee_1 \circ \langle \chi_4 \rangle$ is a weak base relation for $i\vee_1$ without duplicate coordinates, and a brief moment of inspection shows that none of them is fictitious either. Therefore, R_1 is an irredundant weak base relation for $i\vee_1$. We define H_1 to be $\{(x_0, \dots, x_8) \mid (x_0, \dots, x_7, x_8, \dots, x_8) \in R_1\}$, then clearly $H_1 \in \{\{R_1\}\}_\wedge$. Now we put $G_1 := G'_1 \setminus \{\mathbf{0}\}$ where $G'_1 := \{(x_0, \dots, x_6) \mid (x_0, \dots, x_8) \in H_1\}$, and one quickly verifies that $((G_1 \times \{1\}) \cup \{\mathbf{0}\}) \times \{1\} = H_1$. Since $G'_1 \in \langle H_1 \rangle \subseteq \langle R_1 \rangle = i\vee_1$ and removing the bottom-element $\mathbf{0}$ of a non-trivial join-semilattice with top-element still yields a join-semilattice with top-element, we have $G_1 \in i\vee_1$. With the analogous counterexample as for the relation R_1 above, we can show that $(x, y, z) \mapsto x \vee (y \wedge z)$ is not a polymorphism of G_1 (because the non-membership is witnessed among the first seven coordinates). Thus, $\{\{G_1\}\} = i\vee_1$; in particular G_1 , and any relation conjunctively definable from it, is not 0-valid.

For the reduction let now $\varphi(\mathbf{x}) = G_1(\mathbf{x}_1) \wedge \dots \wedge G_1(\mathbf{x}_k)$ be an instance of $\text{MinOnes}(\{\{G_1\}\})$. We construct a corresponding Γ' -formula φ' as follows.

$$\begin{aligned} \varphi''(\mathbf{x}, y, z) &= H_1(\mathbf{x}_1, y, z) \wedge \dots \wedge H_1(\mathbf{x}_k, y, z) \\ \varphi'''(\mathbf{x}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(\ell)}, z) &= \bigwedge_{i=1}^{\ell} \left((x_i \xrightarrow{z=1} x_i^{(2)}) \wedge \bigwedge_{j=2}^{\ell-1} (x_i^{(j)} \xrightarrow{z=1} x_i^{(j+1)}) \wedge (x_i^\ell \xrightarrow{z=1} x_i) \right) \\ \varphi'(\mathbf{x}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(\ell)}, y, z) &= \varphi''(\mathbf{x}, y, z) \wedge \varphi'''(\mathbf{x}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(\ell)}, z) \end{aligned}$$

where $\ell = |\mathbf{x}|$ is the number of variables of φ , y and z are new global variables, and where we have written $(u \xrightarrow{w=1} v)$ to denote $([x \rightarrow y] \times \{1\})(u, v, w)$. Let m_0 be the assignment to the $\ell^2 + 2$ variables of φ' given by $m_0(z) = 1$ and $m_0(x) = 0$ elsewhere. It is clear that (φ', m_0) is an instance of $\text{NOSol}(\Gamma')$, since m_0 satisfies φ' . The formula φ''' only multiplies each variable x from φ ℓ -times and forces $x \approx x^{(2)} \approx \dots \approx x^{(\ell)}$, which is just a technicality for establishing an AP-reduction. The main idea of this proof is the correspondence between the solutions of φ and φ'' .

For each solution s of $\varphi(\mathbf{x})$ there exists a solution s' of $\varphi''(\mathbf{x}, y)$ with $s'(y) = 1$ (and $s'(z) = 1$). Each solution s' of φ'' has always $s'(z) = 1$ and either $s'(y) = 0$ or $s'(y) = 1$. Because every variable from \mathbf{x} is part of one of the \mathbf{x}_i , the assignment m_0 restricted to (\mathbf{x}, y, z) is the only solution s' of φ'' satisfying $s'(y) = 0$. If otherwise $s'(y)$ equals 1, then s' restricted to the variables \mathbf{x} satisfies $\varphi(\mathbf{x})$, following the correspondence between the relations G_1 and H_1 .

For $r \in [1, \infty)$ let s' be an r -approximate solution of the $\text{NOSol}(\Gamma')$ -instance (φ', m_0) . Let $s := s' \upharpoonright_{\mathbf{x}}$ be the restriction of s' to the variables of φ . Since $s' \neq m_0$, by what we showed before, $s'(y) = 1$ and s is a solution of $\varphi(\mathbf{x})$. We have $\text{OPT}(\varphi', m_0) \geq 2$ and $\text{OPT}(\varphi) \geq 1$, since solutions of the $\text{NOSol}(\Gamma')$ -instance (φ', m_0) must be different from m_0 , whereby y is forced to have value 1, and $[\varphi] \in \{\{G_1\}\}_\wedge$ is not 0-valid. Moreover, $\text{hw}(s) = \text{hd}(\mathbf{0}, s)$, $\text{hd}(s', m_0) = \ell \text{hw}(s) + 1$,

$\text{OPT}(\varphi', m_0) = \ell\text{OPT}(\varphi) + 1$, and $\text{hd}(s', m_0) \leq r\text{OPT}(\varphi', m_0)$. From this and $\text{OPT}(\varphi) \geq 1$ it follows that

$$\begin{aligned} \ell\text{hw}(s) < \ell\text{hw}(s) + 1 &= \text{hd}(s', m_0) \leq r\text{OPT}(\varphi', m_0) = r\ell\text{OPT}(\varphi) + r \\ &\leq r\ell\text{OPT}(\varphi) + r\text{OPT}(\varphi) \\ &\leq r\ell\text{OPT}(\varphi) + r\text{OPT}(\varphi) + (r - 1)\ell\text{OPT}(\varphi) \\ &= (2r - 1 + r/\ell)\ell\text{OPT}(\varphi) \\ &= (1 + 2(r - 1) + r/\ell)\ell\text{OPT}(\varphi). \end{aligned}$$

Hence s is an $(1 + \alpha(r - 1) + o(1))$ -approximate solution of the instance φ of the problem $\text{MinOnes}(\{G_1\})$ where $\alpha = 2$.

In the case when $\langle \Gamma \rangle = \text{iV}_2$, the proof goes through with minor changes: $R_2 = \text{V}_2 \circ \langle \chi_4 \rangle = R_1 \setminus \{\mathbf{1}\}$, so we define H_2 and G_2 like H_1 and G_1 just using R_2 and H_2 in place of R_1 and H_1 . Then we have $H_2 = H_1 \setminus \{\mathbf{1}\}$, $G_2 = G_1 \setminus \{\mathbf{1}\}$ and $\langle \{G_2\} \rangle = \text{iV}_2$. Moreover, for the reduction we shall need an additional global variable w for φ''' (and φ') since the encoding of the implication from Lemma 45 requires it (and forces it to zero in every model).

For $\langle \Gamma \rangle = \text{iV}_0$ we can use $R_0 = \text{V}_0 \circ \langle \chi_4 \rangle = R_2 \cup \{\mathbf{0}\}$; then, letting $H_0 = \{(x_0, \dots, x_7) \mid (x_0, \dots, x_7, x_7, \dots, x_7) \in R_0\} \in \langle \{R_0\} \rangle_\wedge$, we have $H_0 = (G_2 \times \{\mathbf{1}\}) \cup \{\mathbf{0}\}$. On a side note, we observe that $H_0 = \text{V}_0 \circ \langle \chi_3 \rangle$, which we can use alternatively without detouring via R_0 . Given the relationship between G_2 and H_0 , we do not need the global variable z in the definition of φ'' , but we need to have it in the definition of φ''' , where the relation given by Lemma 45 necessitates atoms of the form $(u \xrightarrow{z=0} v)$ forcing z to zero in every model.

The case where $\langle \Gamma \rangle = \text{iV}$ is similar to the previous: we can use the irredundant weak base relation $H = \text{V} \circ \langle \chi_3 \rangle = H_0 \cup \{\mathbf{1}\} = (G_1 \times \{\mathbf{1}\}) \cup \{\mathbf{0}\}$. Except for y in the definition of φ'' no additional global variables are needed in the definition of φ' , because $[u \rightarrow v]$ atoms are directly available for φ''' . □

Corollary 47 *If Γ is exactly Horn ($\text{iE} \subseteq \langle \Gamma \rangle \subseteq \text{iE}_2$) or exactly dual-Horn ($\text{iV} \subseteq \langle \Gamma \rangle \subseteq \text{iV}_2$) then $\text{NOSol}(\Gamma)$ is MinHornDeletion-complete under AP-Turing-reductions.*

Proof Hardness follows from Lemma 46 and duality. Moreover, $\text{NOSol}(\Gamma)$ can be AP-Turing-reduced to $\text{NSol}(\Gamma \cup \{[x], [\neg x]\})$ as follows: Given a Γ -formula φ and a model m , we construct for every variable x of φ a formula $\varphi_x = \varphi \wedge (x \approx \bar{m}(x))$. Then for every x where $[\varphi_x] \neq \emptyset$ we run an oracle algorithm for $\text{NSol}(\Gamma \cup \{[x], [\neg x]\})$ on (φ_x, m) and output one result of these oracle calls that is closest to m .

We claim that this algorithm provides indeed an AP-Turing reduction. To see this observe first that the instance (φ, m) has feasible solutions if and only if this holds for (φ_x, m) and at least one variable x . Moreover, we have $\text{OPT}(\varphi, m) =$

$\min_{x, [\varphi_x] \neq \emptyset}(\text{OPT}(\varphi_x, m))$. Let $A(\varphi, m)$ be the answer of the algorithm on (φ, m) and let $B(\varphi_x, m)$ be the answers to the oracle calls. Consider a variable x^* such that $\text{OPT}(\varphi, m) = \min_{x, [\varphi_x] \neq \emptyset}(\text{OPT}(\varphi_x, m)) = \text{OPT}(\varphi_{x^*}, m)$, and assume that $B(\varphi_{x^*}, m)$ is an r -approximate solution of (φ_{x^*}, m) . Then we get

$$\frac{\text{hd}(m, A(\varphi, m))}{\text{OPT}(\varphi, m)} = \frac{\min_{y, [\varphi_y] \neq \emptyset}(\text{hd}(m, B(\varphi_y, m)))}{\text{OPT}(\varphi_{x^*}, m)} \leq \frac{\text{hd}(m, B(\varphi_{x^*}, m))}{\text{OPT}(\varphi_{x^*}, m)} \leq r.$$

Thus the algorithm is indeed an AP-Turing-reduction from $\text{NOSol}(\Gamma)$ to $\text{NSol}(\Gamma \cup \{[x], [\neg x]\})$. Note that for $\Gamma \subseteq \text{iV}_2$ the problem $\text{NSol}(\Gamma \cup \{[x], [\neg x]\})$ reduces to MinHornDeletion according to Propositions 29 and 27. Duality completes the proof. \square

7 Finding the Minimal Distance Between Solutions

In this section we study the optimization problem $\text{MinSolutionDistance}$. We first consider the polynomial-time cases and then the cases of higher complexity.

7.1 Polynomial-Time Cases

We show that for bijunctive constraints the problem $\text{MinSolutionDistance}$ can be solved in polynomial time. After stating the result we present an algorithm and analyze its complexity and correctness.

Proposition 48 *If Γ is a bijunctive constraint language ($\Gamma \subseteq \text{iD}_2$) then the problem $\text{MSD}(\Gamma)$ is in PO.*

By Proposition 32, an algorithm for bijunctive constraint languages Γ can be restricted to at most binary clauses. Alternatively, one can use the plain base

$$\{[x], [\neg x], [x \vee y], [\neg x \vee y], [\neg x \vee \neg y]\}$$

of iD_2 exhibited in [12] to see that every relation in Γ can be written as a conjunction of disjunctions of two not necessarily distinct literals. We shall treat these disjunctions as one- or two-element sets of literals when extending the algorithm of Aspvall, Plass, and Tarjan [2] to compute the minimum distance between distinct models of a bijunctive constraint formula.

Algorithm BIJUNCTIVE MSD

Input: An iD_2 -formula φ given as a collection of one- or two-element sets of literals (bijunctive clauses).

Output: “ ≤ 1 model” or the minimal Hamming distance of any two distinct models of φ .

Method:

Let \mathcal{V} be the set of variables occurring in φ .

Let $\mathcal{L} := \{v, \neg v \mid v \in \mathcal{V}\}$ be the set of literals.

Let \bar{u} denote the literal complementary to $u \in \mathcal{L}$.

Construct the relation $R := \{(\bar{u}, v), (\bar{v}, u) \mid \{u, v\} \in \varphi \wedge u \neq v\} \cup \{(\bar{u}, u) \mid \{u\} \in \varphi\}$.
Let \leq be the reflexive and transitive closure of R , i.e. the least preorder on \mathcal{L} extending R .

Construct the sets

$$\mathcal{V}_0 := \{v \in \mathcal{V} \mid v \leq x \leq \neg x \quad \text{or} \quad \neg x \leq x \leq \neg v \quad \text{for some } x \in \mathcal{V}\}$$

$$\mathcal{V}_1 := \{v \in \mathcal{V} \mid x \leq \neg x \leq v \quad \text{or} \quad \neg v \leq \neg x \leq x \quad \text{for some } x \in \mathcal{V}\}$$

If $\mathcal{V}_0 \cap \mathcal{V}_1 \neq \emptyset$ or $\mathcal{V}_0 \cup \mathcal{V}_1 = \mathcal{V}$ holds, then return “ ≤ 1 model”.

Let $\mathcal{L}' := \mathcal{L} \setminus \{v, \neg v \mid v \in \mathcal{V}_0 \cup \mathcal{V}_1\}$.

Let $\sim := \{(u, v) \in \mathcal{L}' \times \mathcal{L}' \mid u \leq v \wedge v \leq u\}$.

Return $\min\{|L| \mid L \in \mathcal{L}'/\sim\}$ as minimal Hamming distance.

End of algorithm

Complexity The size of \mathcal{L} is linear in the number of variables, the reflexive closure can be computed in time linear in $|\mathcal{L}|$, the transitive closure in time cubic in $|\mathcal{L}|$, see [32]. The equivalence relation \sim is the intersection of \leq restricted to \mathcal{L}' and its inverse (quadratic in $|\mathcal{L}'|$); from it we can obtain the partition \mathcal{L}'/\sim in linear time in $|\mathcal{L}'| \leq |\mathcal{L}|$, including the cardinalities of the equivalence classes and their minimization. Similarly, the remaining sets from the proof (\mathcal{V}_0 , \mathcal{V}_1 , their intersection and union, and thus also \mathcal{L}') can be computed with polynomial time complexity.

Correctness The pairs in R arise from interpreting the atomic constraints in φ as implications. By transitivity of implication, the inequality $u \leq v$ for literals u, v means that every model m of φ satisfies the implication $u \rightarrow v$ or, equivalently, $m(u) \leq m(v)$. In particular, $x \leq \neg x$ implies $m(x) = 0$ and $\neg x \leq x$ implies $m(x) = 1$. Therefore \mathcal{V}_0 can be seen to be the set of variables that have to be false in every model of φ , and \mathcal{V}_1 the set of variables true in every model.

If $\mathcal{V}_0 \cap \mathcal{V}_1 \neq \emptyset$ holds then the formula φ is inconsistent and has no solution. If $\mathcal{V}_0 \cup \mathcal{V}_1 = \mathcal{V}$ holds, then every variable has a unique fixed value, hence φ has only one solution. Otherwise the formula is consistent and not all variables are fixed, hence there are at least two models.

To determine the minimal number of variables, whose values can be flipped between any two models of φ , it suffices to consider the literals without fixed value,

\mathcal{L}' . If we have $u \leq v$ and $v \leq u$, the literals are equivalent, $u \sim v$, and must have the same value in every model. This means that any two distinct models have to differ on all literals of at least one equivalence class in \mathcal{L}'/\sim . Therefore, the return value of the algorithm is a lower bound for the minimal distance.

To prove that the return value can indeed be attained, we exhibit two models $m_0 \neq m_1$ of φ having the least cardinality of any equivalence class in \mathcal{L}'/\sim as their Hamming distance. Let $L \in \mathcal{L}'/\sim$ be a class of minimum cardinality. Define $m_0(u) := 0$ and $m_1(u) := 1$ for all literals $u \in L$. We extend this by setting $m_0(w) := m_1(w) := 0$ for all $w \in \mathcal{L}$ such that $w \leq u$ for some $u \in L$, and by $m_0(w) := m_1(w) := 1$ for all $w \in \mathcal{L}$ such that $u \leq w$ for some $u \in L$. For variables $v \in \mathcal{V}$ satisfying $v \leq \neg v$ or $\neg v \leq v$ we have $v \in \mathcal{V}_0 \cup \mathcal{V}_1$, and thus $v \notin \mathcal{L}'$; in other words, for $[v]_{\sim} \in \mathcal{L}'/\sim$ the classes $[v]_{\sim}$ and $[\neg v]_{\sim}$ are incomparable. Thus, so far, we have not defined m_0 and m_1 on a variable $v \in \mathcal{V}$ and on its negation $\neg v$ at the same time. Of course, fixing a value for a negative literal $\neg v$ implicitly means that we bind the assignment for $v \in \mathcal{V}$ to the opposite value.

It remains to fix the value of literals in \mathcal{L}' that are neither related to the literals in L nor have fixed values in all models. Suppose $(\bar{u}, v) \in R$ is a constraint such that the value of at least one literal has not yet been defined. There are three cases: either both literals have not yet received a value, or \bar{u} is undefined and v has been assigned the value 1 (either as a fixed value in all models or because of being greater than a literal in L or because of being lesser than a complement of a literal in L), or v is undefined and \bar{u} has been assigned the value 0 (either as a fixed value in all models or because of being smaller than a literal in L or greater than a complement of a literal in L). All three cases can be handled by defining both models, m_0 and m_1 , on the remaining variables identically: starting with a minimal literal u , where m_0 and m_1 are not yet defined, we assign $m_0(u) := m_1(\bar{u}) := 0$ and $m_1(u) := m_0(\bar{u}) := 1$.

This way none of the constraints is violated, and m_0 and m_1 are distinct only on variables corresponding to literals in L . Iterate this procedure until all variables (and their complements) have been assigned values. If $\mathcal{L}'' \subseteq \mathcal{L}'$ denotes the literals remaining after propagating the values of m_0 and m_1 on L , then the presented method can be implemented by partitioning \mathcal{L}'' into two classes L_0 and L_1 such that $L_0 \cap \{u, \bar{u}\}$ is a singleton for every $u \in \mathcal{L}''$ and each weakly connected component of the quasiordered set (\mathcal{L}'', \leq) is either a subset of L_0 or L_1 . Then set m_0 and m_1 to k on the literals belonging to L_k for $k \in \{0, 1\}$.

By construction, m_0 differs from m_1 only in the variables corresponding to the literals in L , so their Hamming distance is $|L|$ as desired. Moreover, both assignments respect the order constraints in (\mathcal{L}, \leq) . As these faithfully reflect all original atomic constraints, m_0 and m_1 are indeed models of φ .

Proposition 49 *If Γ is a Horn ($\Gamma \subseteq \text{iE}_2$) or a dual Horn ($\Gamma \subseteq \text{iV}_2$) constraint language then $\text{MSD}(\Gamma)$ is in PO.*

We only discuss the Horn case ($\Gamma \subseteq \text{iE}_2$), dual Horn ($\Gamma \subseteq \text{iV}_2$) being symmetric. The following algorithm improves the description given in [6] by correctly treating two marginal cases where the output is evident.

Algorithm HORN MSD

Input: A Horn formula φ given as a set of Horn clauses (cf. the plain base of iE_2 given in [12]).

Output: “ ≤ 1 model” or the minimal Hamming distance of any two distinct models of φ .

Method:

For each variable x in φ , add the clause $(\neg x \vee x)$.

Let $\mathcal{U} := \emptyset$.

Apply the following rules to φ until no more clauses and literals can be removed and no new clauses can be added.

Unit resolution and unit subsumption: Let \bar{u} denote the complement of a literal u . If the clause set contains a unit clause u , remove all clauses containing the literal u and remove all literals \bar{u} from the remaining clauses. Add u to the set \mathcal{U} .

Hyper-resolution with binary implications: Resolve all negative literals of a clause simultaneously with binary implications possessing identical premises.

$$\frac{(\neg x \vee y_1) \cdots (\neg x \vee y_k) \quad (\neg y_1 \vee \cdots \vee \neg y_k \vee z)}{(\neg x \vee z)}$$

$$\frac{(\neg x \vee y_1) \cdots (\neg x \vee y_k) \quad (\neg y_1 \vee \cdots \vee \neg y_k)}{(\neg x)}$$

Let \mathcal{D} be the set of clauses after applying the two rules exhaustively.

If \mathcal{D} contains the empty clause, return “ ≤ 1 model”.

If \mathcal{U} contains a literal for every variable in φ , return “ ≤ 1 model”.

If φ contains a variable that appears neither in \mathcal{D} nor in \mathcal{U} , return 1 as the minimal Hamming distance.

Otherwise, let \mathcal{V} be the set of variables occurring in \mathcal{D} , and let $\sim \subseteq \mathcal{V}^2$ be the relation defined by $x \sim y$ if $\{\neg x \vee y, \neg y \vee x\} \subseteq \mathcal{D}$. Note that \sim is an equivalence, since the tautological clauses ensure reflexivity and resolution of implications computes their transitive closure. We say that a variable z depends on variables y_1, \dots, y_k , if \mathcal{D} contains the clauses $\neg y_1 \vee \cdots \vee \neg y_k \vee z, \neg z \vee y_1, \dots, \neg z \vee y_k$ and $z \not\sim y_i$ holds for all $i = 1, \dots, k$.

Return $\min\{|X| \mid X \in \mathcal{V}/\sim, X \text{ does not contain dependent variables}\}$ as the minimal Hamming distance.

End of algorithm

Complexity The run-time of the algorithm is polynomial in the number of clauses in φ : Unit resolution/subsumption can be applied at most once for each variable, and hyper-resolution has to be applied at most once for each variable x and each clause $\neg y_1 \vee \cdots \vee \neg y_k \vee z$ and $\neg y_1 \vee \cdots \vee \neg y_k$.

Correctness Adding resolvents and removing subsumed clauses maintains logical equivalence, therefore $\mathcal{D} \cup \mathcal{U}$ is logically equivalent to φ , i.e., both clause sets have the same models. We note that the sets of variables of \mathcal{U} and of \mathcal{D} are disjoint. The unit clauses in \mathcal{U} are always (uniquely) satisfiable, thus \mathcal{D} and φ are equisatisfiable.

Therefore, if \mathcal{D} contains the empty clause, φ is also unsatisfiable; otherwise \mathcal{D} is satisfiable, e.g., by assigning 0 to every $x \in \mathcal{V}$. In this case, if \mathcal{U} contains a literal for every variable of φ , the unit clauses in \mathcal{U} define a unique model of φ .

Otherwise φ has at least two models $m_1 \neq m_2$. In the simplest case some variable x in φ has been left unconstrained by \mathcal{D} and \mathcal{U} ; in this case we can pick any model of \mathcal{D} and \mathcal{U} and extend it to two different models of φ with Hamming distance 1 by setting $m_1(x) = 0$ and $m_2(x) = 1$ and setting $m_1(y) = m_2(y) = 0$ for any other variable y outside \mathcal{D} and \mathcal{U} . For the remaining situations it is sufficient to consider the models of \mathcal{D} only, as each model m of \mathcal{D} uniquely extends to a model of φ by defining $m(x) = 1$ for $(x) \in \mathcal{U}$ and $m(x) = 0$ for $(\neg x) \in \mathcal{U}$; hence the minimal Hamming distances of the models of φ and \mathcal{D} will be the same.

We are thus looking for models m_1, m_2 of \mathcal{D} such that the size of the difference set $\Delta(m_1, m_2) = \{x \mid m_1(x) \neq m_2(x)\}$ is minimal. In fact, since the models of Horn formulas are closed under minimum, we may assume $m_1 < m_2$, i.e., we have $m_1(x) = 0$ and $m_2(x) = 1$ for all variables $x \in \Delta(m_1, m_2)$. Indeed, given two models m_2 and m'_2 of \mathcal{D} where neither $m_2 \leq m'_2$ nor $m'_2 \leq m_2$, $m_1 = m_2 \wedge m'_2$ is also a model, and it is distinct from m_2 . Since $\text{hd}(m_1, m_2) \leq \text{hd}(m_2, m'_2)$, the minimal Hamming distance will occur between models m_1 and m_2 satisfying $m_1 < m_2$.

Note the following facts regarding the equivalence relation \sim and dependent variables.

- If $x \sim y$ then the two variables must have the same value in every model of \mathcal{D} in order to satisfy the implications $\neg x \vee y$ and $\neg y \vee x$. This means that for all models m of \mathcal{D} and all $X \in \mathcal{V}/\sim$, we have either $m(x) = 0$ for all $x \in X$ or $m(x) = 1$ for all $x \in X$.
- The dependence of variables is acyclic: If, for some $l \geq 2$, for every $1 \leq i < l$ we have that z_i depends on variables including one, say y_i , which is equivalent to z_{i+1} , and $z_l = z_1$, then there is a cycle of binary implications between the variables and thus $z_i \sim y_i \sim z_j$ for all i, j , contradicting the definition of dependence.
- If a variable z depending on y_1, \dots, y_k belongs to a difference set $\Delta(m_1, m_2)$, then at least one of the y_j s also has to belong to $\Delta(m_1, m_2)$: $m_2(z) = 1$ implies $m_2(y_j) = 1$ for all $j = 1, \dots, k$ (because of the clauses $\neg z \vee y_i$), and $m_1(z) = 0$ implies $m_1(y_i) = 0$ for at least one i (because of the clause $\neg y_1 \vee \dots \vee \neg y_k \vee z$). Therefore $\Delta(m_1, m_2)$ is the union of at least two sets in \mathcal{V}/\sim , namely the equivalence class of z and the one of y_i .
- If some $z_1 \in \Delta(m_1, m_2)$ is equivalent to a variable z'_1 that depends on some other variables, then we have a variable z_2 among them, which also belongs to $\Delta(m_1, m_2)$. If the equivalence class of z_2 still contains a variable z'_2 depending on other variables, we can iterate this procedure. In this way we obtain a sequence $z_1 \sim z'_1, z_2 \sim z'_2, z_3 \sim z'_3, \dots$ where z'_i depends on variables including z_{i+1} , which is equivalent to z'_{i+1} . Because there are only finitely many variables and because of acyclicity, after a linear number of steps we must reach a variable $z_n \in \Delta(m_1, m_2)$ such that its equivalence class (being a subset of the difference set) does not contain any dependent variables.

Hence the difference between any two models cannot be smaller than the cardinality of the smallest set in \mathcal{V}/\sim without dependent variables. It remains to show that we can indeed find two such models.

Let X be a set in \mathcal{V}/\sim which has minimal cardinality among the sets without dependent variables, and let m_0, m_1 be interpretations defined as follows: (1) $m_0(y) = 0$ and $m_1(y) = 1$ if $y \in X$; (2) $m_0(y) = 1$ and $m_1(y) = 1$ if $y \notin X$ and $(\neg x \vee y) \in \mathcal{D}$ for some $x \in X$; (3) $m_0(y) = 0$ and $m_1(y) = 0$ otherwise. We have to show that m_0 and m_1 satisfy all clauses in \mathcal{D} . Let m be any of these models. \mathcal{D} contains two types of clauses.

Type 1: Horn clauses with a positive literal $\neg y_1 \vee \dots \vee \neg y_k \vee z$. If $m(y_i) = 0$ for any i , we are done. So suppose $m(y_i) = 1$ for all $i = 1, \dots, k$; we have to show $m(z) = 1$. The condition $m(y_i) = 1$ means that either $y_i \in X$ (for $m = m_1$) or that there is a clause $(\neg x_i \vee y_i) \in \mathcal{D}$ for some $x_i \in X$. We distinguish the two cases $z \in X$ and $z \notin X$.

Let $z \in X$. If $z \sim y_i$ for any i , we are done for we have $m(z) = m(y_i) = 1$. So suppose $z \not\sim y_i$ for all i . As the elements in X , in particular z and the x_i s, are equivalent and the binary clauses are closed under resolution, \mathcal{D} contains the clause $\neg z \vee y_i$ for all i . But this would mean that z is a variable depending on the y_i s, contradicting the assumption $z \in X$.

Let $z \notin X$, and let $x \in X$. As the elements in X are equivalent and the binary clauses are closed under resolution, \mathcal{D} contains $\neg x \vee y_i$ for all i . Closure under hyper-resolution with the clause $\neg y_1 \vee \dots \vee \neg y_k \vee z$ means that \mathcal{D} also contains $\neg x \vee z$, whence $m(z) = 1$.

Type 2: Horn clauses with only negative literals $\neg y_1 \vee \dots \vee \neg y_k$. If $m(y_i) = 0$ for any i , we are done. It remains to show that the assumption $m(y_i) = 1$ for all $i = 1, \dots, k$ leads to a contradiction. The condition $m(y_i) = 1$ means that either $y_i \in X$ (for $m = m_1$) or that there is a clause $(\neg x_i \vee y_i) \in \mathcal{D}$ for some $x_i \in X$. Let x be some particular element of X . Since the elements in X are equivalent and the binary clauses are closed under resolution, \mathcal{D} contains the clause $\neg x \vee y_i$ for all i . But then a hyper-resolution step with the clause $\neg y_1 \vee \dots \vee \neg y_k$ would yield the unit clause $\neg x$, which by construction does not occur in \mathcal{D} . Therefore at least one y_i is neither in X nor part of a clause $\neg x \vee y_i$ with $x \in X$, i.e., $m(y_i) = 0$.

7.2 Hard Cases

7.2.1 Two Solution Satisfiability

In this section we study the feasibility problem of $\text{MSD}(\Gamma)$ which is, given a Γ -formula φ , to decide if φ has two distinct solutions.

Problem: TwoSolutionSAT(Γ)

Input: A conjunctive formula φ over the relations from the constraint language Γ .

Question: Are there two satisfying assignments $m \neq m'$ of φ ?

A priori it is not clear that the tractability of TwoSolutionSAT is fully characterized by co-clones. The problem is that the implementation of relations of some language Γ by another language Γ' might not be parsimonious, that is, in the implementation one solution to a constraint might be blown up into several ones in the implementation. Fortunately we can still determine the tractability frontier for TwoSolutionSAT by combining the corresponding results for SAT and AnotherSAT.

Lemma 50 *Let Γ be a constraint language for which $\text{SAT}(\Gamma)$ is NP-hard. Then the problem $\text{TwoSolutionSAT}(\Gamma)$ is NP-hard.*

Proof Since $\text{SAT}(\Gamma)$ is NP-hard, there must be a relation R in Γ having more than one tuple, because every relation containing only one tuple is at the same time Horn, dual Horn, bijunctive, and affine. Given an instance φ for $\text{SAT}(\Gamma)$, construct φ' as $\varphi \wedge R(y_1, \dots, y_\ell)$ where ℓ is the arity of R and y_1, \dots, y_ℓ are new variables not appearing in φ . Obviously, φ has a solution if and only if φ' has at least two solutions. Hence, we have proved $\text{SAT}(\Gamma) \leq_m \text{TwoSolutionSAT}(\Gamma)$. \square

Lemma 51 *Let Γ be a constraint language for which $\text{AnotherSAT}(\Gamma)$ is NP-hard. Then the problem $\text{TwoSolutionSAT}(\Gamma)$ is NP-hard.*

Proof Let a Γ -formula φ and a satisfying assignment m be an instance of the problem $\text{AnotherSAT}(\Gamma)$. Then φ has a solution other than m if and only if it has two distinct solutions. \square

Lemma 52 *Let Γ be a constraint language for which both problems $\text{SAT}(\Gamma)$ and $\text{AnotherSAT}(\Gamma)$ are in P. Then TwoSolutionSAT is also in P.*

Proof Let φ be an instance of $\text{TwoSolutionSAT}(\Gamma)$. All polynomial-time decidable cases of $\text{SAT}(\Gamma)$ are constructive, i.e., whenever that problem is polynomial-time decidable, there exists a polynomial-time algorithm computing a satisfying assignment provided it exists. If φ is not satisfiable, we reject the instance. Otherwise, we can compute in polynomial time a satisfying assignment m of φ . Now use the algorithm for $\text{AnotherSAT}(\Gamma)$ on the instance (φ, m) to decide if there is a second solution to φ . \square

Corollary 53 *For any constraint language Γ , the problem $\text{TwoSolutionSAT}(\Gamma)$ is in P if both $\text{SAT}(\Gamma)$ and $\text{AnotherSAT}(\Gamma)$ are in P. Otherwise, $\text{TwoSolutionSAT}(\Gamma)$ is NP-hard.*

Proposition 54 *Let Γ be a constraint language for which $\text{TwoSolutionSAT}(\Gamma)$ is in P. Then there is a polynomial-time n -approximation algorithm for $\text{MSD}(\Gamma)$, where n is the number of variables of the Γ -formula on input.*

Proof Since $\text{TwoSolutionSAT}(\Gamma)$ is in P, both $\text{SAT}(\Gamma)$ and $\text{AnotherSAT}(\Gamma)$ must be in P by Corollary 53. Since $\text{SAT}(\Gamma)$ is in P, we can compute a model m of the input φ in polynomial time if it exists. Now we check the $\text{AnotherSAT}(\Gamma)$ -instance (φ, m) .

If it has a solution $m' \neq m$, it is also polynomial time computable, and we return (m, m') . If we fail somewhere in this process, then the $\text{MSD}(\Gamma)$ -instance φ does not have feasible solutions; otherwise, $\text{hd}(m, m') \leq n \leq n \cdot \text{OPT}(\varphi)$. \square

7.2.2 MinDistance-Equivalent Cases

In this section we show that, as for the `NearestOtherSolution` problem, the affine cases of MSD are `MinDistance`-complete.

Proposition 55 *$\text{MSD}(\Gamma)$ is `MinDistance`-complete if the constraint language Γ satisfies the inclusions $\text{iL} \subseteq \langle \Gamma \rangle \subseteq \text{iL}_2$.*

Proof We prove $\text{MSD}(\Gamma) \equiv_{\text{AP}} \text{NearestOtherSolution}(\Gamma)$, which is `MinDistance`-complete for each constraint language Γ satisfying the inclusions $\text{iL} \subseteq \langle \Gamma \rangle \subseteq \text{iL}_2$, according to Proposition 44. As the inclusion $\Gamma \subseteq \text{iL}_2 = \langle \{\text{even}^4, [x], [\neg x]\} \rangle$ holds, any Γ -formula ψ is expressible as $\exists y(A_1x + A_2y \approx c)$. The projection of the affine solution space is again an affine space, so it can be understood as solutions of a system $Ax = b$. If (ψ, m_0) is an instance of `NOSol`(Γ), then ψ is a $\text{MSD}(\Gamma)$ -instance, and a feasible solution $m_1 \neq m_2$ satisfying ψ gives a feasible solution $m_3 := m_0 + (m_2 - m_1)$ for (ψ, m_0) , where $\text{hd}(m_0, m_3) = \text{hd}(m_2, m_1)$. Conversely, a solution $m_3 \neq m_0$ to (ψ, m_0) yields a feasible answer to the MSD -instance ψ . Thus, $\text{OPT}(\psi) = \text{OPT}(\psi, m_0)$ and so $\text{NOSol}(\Gamma) \leq_{\text{AP}} \text{MSD}(\Gamma)$. The other way round, if ψ is an MSD -instance, then attempt to solve the system $Ax = b$ defined by it; if there is no or a unique solution, then the instance does not have feasible solutions. Otherwise, we have at least two distinct models of ψ ; let m_0 be one of these. As above we conclude $\text{OPT}(\psi) = \text{OPT}(\psi, m_0)$, and therefore, $\text{MSD}(\Gamma) \leq_{\text{AP}} \text{NOSol}(\Gamma)$. \square

7.2.3 Tightness Results

We prove that Proposition 54 is essentially tight for some constraint languages. This result builds heavily on the previous results from Section 6.2.1.

Proposition 56 *For a constraint language Γ satisfying the inclusions $\text{iN} \subseteq \langle \Gamma \rangle \subseteq \text{iI}$ and any $\varepsilon > 0$ there is no polynomial-time $n^{1-\varepsilon}$ -approximation algorithm for $\text{MSD}(\Gamma)$, unless $\text{P} = \text{NP}$.*

Proof We show that any polynomial time $n^{1-\varepsilon}$ -approximation algorithm for $\text{MSD}(\Gamma)$ would also allow to decide in polynomial time the problem `AnotherSAT`_{nc}(Γ), which is NP-complete by Proposition 38.

The algorithm works as follows. Given an instance (φ, m) for `AnotherSAT`_{nc}(Γ), the algorithm accepts if m is not a constant assignment. Since Γ is 0-valid (and 1-valid), this output is correct. If φ has only one variable, reject because φ has only two models; otherwise, proceed as follows.

For each variable x of φ , we construct a new formula φ'_x as follows. Let k be the smallest integer greater than $1/\varepsilon$. Introduce $n^k - n$ new variables x^i for $i = 1, \dots, n^k - n$. For every $i \in \{1, \dots, n^k - n\}$ and every constraint $R(y_1, \dots, y_\ell)$

in φ , such that $x \in \{y_1, \dots, y_\ell\}$, construct a new constraint $R(z_1^i, \dots, z_\ell^i)$ by $z_j^i = x^i$ if $y_j = x$ and $z_j^i = y_j$ otherwise; add all the newly constructed constraints to φ in order to get φ'_x . Note, that we can extend models s of φ to models s' of φ'_x by setting $s'(x^i) = s(x)$. In particular, this can be done for m , yielding $m' \in [\varphi'_x]$. As $\Gamma \subseteq \text{iI} = \text{iI}_0 \cap \text{iI}_1$, the $\text{MSD}(\Gamma)$ -instance φ'_x has feasible solutions; thus run the $n^{1-\varepsilon}$ -approximation algorithm for $\text{MSD}(\Gamma)$ on φ'_x . If for every x the answer is a pair (m_1, m_2) with $m_2 = \overline{m_1}$, then reject, otherwise accept.

This procedure is a correct polynomial-time algorithm for $\text{AnotherSAT}_{\text{nc}}(\Gamma)$. For polynomial runtime is clear, it remains to show correctness. If φ has only constant models, then the same is true for every φ'_x since φ contains a variable distinct from x . Thus each approximation must result in a pair of complementary constant assignments, and the output is correct. Assume now that there is a model s of φ different from $\mathbf{0}$ and $\mathbf{1}$. Hence, there exists a variable x such that $s(x) = m(x)$ because m is constant. It follows that φ'_x has a model s' fulfilling $\text{OPT}(\varphi'_x) \leq \text{hd}(s', m') < n$, where n is the number of variables of φ . But then the approximation algorithm must find two distinct models $m_1 \neq m_2$ of φ'_x satisfying $\text{hd}(m_1, m_2) < n \cdot (n^k)^{1-\varepsilon} = n^{k(1-\varepsilon)+1}$. Since we stipulated $k > 1/\varepsilon$, it follows that $\text{hd}(m_1, m_2) < n^k$. Consequently, we have $m_2 \neq \overline{m_1}$ and the output of our algorithm is again correct. \square

8 Concluding Remarks

The problems investigated in this paper are quite natural. In the space of bit-vectors we search for a solution of a formula that is closest to a given point, or for a solution next to a given solution, or for two solutions witnessing the smallest Hamming distance between any two solutions. Our results describe the complexity of exploring the solution space for arbitrary families of Boolean relations. Moreover, our problems generalize problems familiar from the literature: MinOnes , NearestCodeword , and DistanceSAT are instances of our NearestSolution , while MinDistance is the same as our problem $\text{MinSolutionDistance}$ when restricting the latter to affine relations.

To prove the results, we first had to extend the notion of AP-reduction. The optimization problems considered in the literature have the property that each instance has at least one feasible solution. This is not the case when looking for nearest solutions regarding a given solution or a prescribed Boolean tuple, as a formula may have just a single solution or no solution at all. Therefore we had to refine the notion of AP-reductions such that it correctly handles instances without feasible solutions.

The complexity of NearestSolution can be classified by the usual approach: We first show that for each constraint language the complexity of the problem does not change when admitting existential quantifiers and equality, and then check all finitely related clones according to Post’s lattice. This approach does not work for the problems $\text{NearestOtherSolution}$ and $\text{MinSolutionDistance}$: It does not seem to be possible to show a priori that the complexity remains unaffected under such language extensions. In principle the complexity of a problem might well differ for two constraint languages Γ_1 and Γ_2 that represent the same clone ($\langle \Gamma_1 \rangle = \langle \Gamma_2 \rangle$) but that differ with

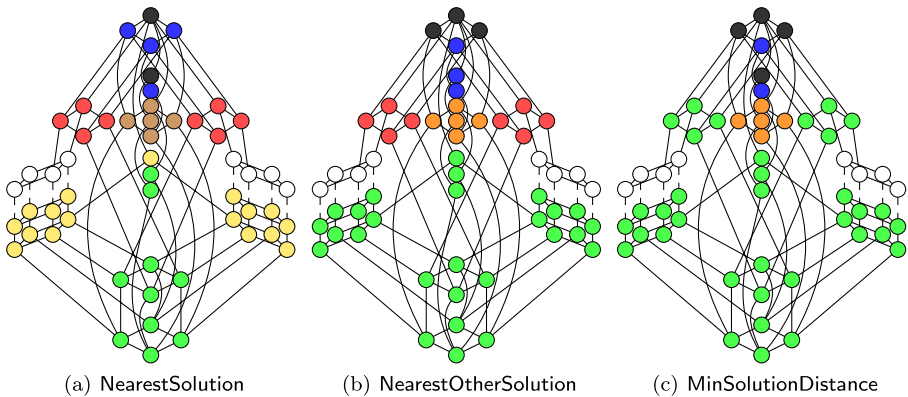


Fig. 4 Comparing the complexities: The hard cases (colored blue and black) are the same, whereas the polynomial cases (green) increase from left to right

respect to partial polymorphisms ($\langle \Gamma_1 \cup \{\approx\} \rangle_{\wedge} \neq \langle \Gamma_2 \cup \{\approx\} \rangle_{\wedge}$). Theorems 4 and 5 finally show that this is not the case, but we learn this only a posteriori. Our method of proof fundamentally relies on irredundant weak bases that seem to be the perfect fit for such a situation: a priori compatibility with existential quantification is not required, but it will follow once the proof succeeds just using weak bases.

Figure 4 compares the complexity classifications of the three problems. Regarding *NearestSolution* and *NearestOtherSolution*, knowing that an assignment is a solution apparently helps in finding a solution nearby. For expressive constraint languages it is NP-complete to decide whether a feasible solution exists at all; for *NearestSolution* this requires the existence of at least one satisfying assignment, while the other two problems need even two. Kann proved in [21] that $\text{MinOnes}(\Gamma)$ is NPOPB-complete for $\langle \Gamma \rangle = \text{BR}$, where NPOPB is the class of NPO problems with a polynomially bounded objective function. This result implies that $\text{NearestSolution}(\Gamma)$ is NPOPB-complete for $\langle \Gamma \rangle = \text{BR}$ as well. It is unclear whether this result also holds for $\langle \Gamma \rangle = \text{iN}_2$. It may be possible to find a suitable constraint language Γ' satisfying $\langle \Gamma' \rangle = \text{BR}$ such that $\text{MinOnes}(\Gamma')$ reduces to $\text{NearestOtherSolution}(\Gamma)$ for $\text{iI}_0 \subseteq \langle \Gamma \rangle$ or $\text{iI}_1 \subseteq \langle \Gamma \rangle$, proving thus that $\text{NOSol}(\Gamma)$ is NPOPB-complete for these cases. Likewise, the NPOPB-hardness of $\text{MSD}(\Gamma)$ for $\text{iN}_2 \subseteq \langle \Gamma \rangle$ or $\text{iI}_0 \subseteq \langle \Gamma \rangle$ or $\text{iI}_1 \subseteq \langle \Gamma \rangle$ remains open for the time being.

Acknowledgments Miki Hermann was supported by ANR-11-ISO2-003-01 Blanc International grant ALCOCLAN. Part of the work was done during his stay at the Wolfgang Pauli Institute (ICP, UMI CNRS 2842) in Vienna, Austria.

Research of Stefan Mengel was done during his post-doctoral stay in LIX at École Polytechnique. Supported by a QUALCOMM grant.

Gernot Salzer was supported by Austrian Science Fund (FWF) grant I836-N23.

Funding Information Open access funding provided by TU Wien (TUW).

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Arora, S., Babai, L., Stern, J., Sweedyk, Z.: The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. Syst. Sci.* **54**(2), 317–331 (1997)
2. Aspvall, B., Plass, M.R., Tarjan, R.E.: A linear-time algorithm for testing the truth of certain quantified Boolean formulas. *Inf. Process. Lett.* **8**(3), 121–123 (1979)
3. Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A., Protasi, M.: *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer, Berlin (1999)
4. Bailleux, O., Marquis, P.: Some computational aspects of Distance-SAT. *J. Autom. Reason.* **37**(4), 231–260 (2006)
5. Baker, K.A., Pixley, A.F.: Polynomial interpolation and the Chinese Remainder Theorem for algebraic systems. *Math. Z.* **143**(2), 165–174 (1975)
6. Behrisch, M., Hermann, M., Mengel, S., Salzer, G.: Give me another one! In: Elbassioni, K., Makino, K. (eds.) *Proceedings 26th International Symposium on Algorithms and Computation (ISAAC 2015)*, Nagoya (Japan), *Lecture Notes in Computer Science*, vol. 9472, pp. 664–676. Springer, Berlin (2015)
7. Behrisch, M., Hermann, M., Mengel, S., Salzer, G.: As close as it gets. In: Kaykobad, M., Petreschi, R. (eds.) *Proceedings 10th International Workshop on Algorithms and Computation (WALCOM 2016)*, Kathmandu (Nepal), *Lecture Notes in Computer Science*, vol. 9627, pp. 222–235. Springer (2016)
8. Behrisch, M., Hermann, M., Mengel, S., Salzer, G.: The next whisky bar. In: Kulikov, A.S., Woeginger, G.J. (eds.) *Proceedings 11th International Computer Science Symposium in Russia (CSR 2016)*, St. Petersburg (Russia), *Lecture Notes in Computer Science*, vol. 9691, pp. 41–56. Springer, Berlin (2016)
9. Böhrer, E., Creignou, N., Reith, S., Vollmer, H.: Playing with Boolean blocks, part II: constraint satisfaction problems. *SIGACT News Complex. Theory Column* **43** **35**(1), 22–35 (2004)
10. Böhrer, E., Reith, S., Schnoor, H., Vollmer, H.: Bases for Boolean co-clones. *Inf. Process. Lett.* **96**(2), 59–66 (2005)
11. Creignou, N., Khanna, S., Sudan, M.: Complexity classifications of Boolean constraint satisfaction problems. In: *SIAM Monographs on Discrete Mathematics and Applications*, vol. 7. SIAM, Philadelphia (2001)
12. Creignou, N., Kolaitis, P.G., Zanuttini, B.: Structure identification of Boolean relations and plain bases for co-clones. *J. Comput. Syst. Sci.* **74**(7), 1103–1115 (2008)
13. Creignou, N., Vollmer, H.: Boolean constraint satisfaction problems: when does post's lattice help? In: Creignou, N., Kolaitis, P.G., Vollmer, H. (eds.) *Complexity of Constraints—An Overview of Current Research Themes [Result of a Dagstuhl Seminar]*, *Lecture Notes in Computer Science*, vol. 5250, pp. 3–37. Springer, Berlin (2008)
14. Crescenzi, P., Rossi, G.: On the Hamming distance of constraint satisfaction problems. *Theor. Comput. Sci.* **288**(1), 85–100 (2002)
15. Dalal, M.: Investigations into a theory of knowledge base revision. In: Shrobe, H.E., Mitchell, T.M., Smith, R.G. (eds.) *Proceedings 7th National Conference on Artificial Intelligence (AAAI-88)*, St. Paul (Minnesota, USA), pp. 475–479. AAAI Press/MIT Press (1988)
16. Dumer, I., Micciancio, D., Sudan, M.: Hardness of approximating the minimum distance of a linear code. *IEEE Trans. Inf. Theory* **49**(1), 22–37 (2003)
17. Hochbaum, D.S., Megiddo, N., Naoar, J., Tamir, A.: Tight bounds and 2-approximation algorithms for integer programs with two variables per inequality. *Math. Program.* **62**(1–3), 69–83 (1993)
18. Janov, Ju.I., Mučnik, A.A.: о существовании k -значных замкнутых классов, не имеющих конечного базиса [Existence of k -valued closed classes without a finite basis]. *Doklady Akademii Nauk SSSR* **127**(1), 44–46 (1959)
19. Jeavons, P., Cohen, D., Gyssens, M.: Closure properties of constraints. *J. Assoc. Comput. Mach.* **44**(4), 527–548 (1997)

20. Juban, L.: Dichotomy theorem for the generalized unique satisfiability problem. In: Ciobanu, G., Păun, G. (eds.) *Proceedings 12th Fundamentals of Computation Theory (FCT'99) Iași (Romania)*, Lecture Notes in Computer Science, vol. 1684, pp. 327–337. Springer, Berlin (1999)
21. Kann, V.: Polynomially bounded minimization problems that are hard to approximate. *Nordic J. Comput.* **1**(3), 317–331 (1994)
22. Khanna, S., Sudan, M., Trevisan, L., Williamson, D.P.: The approximability of constraint satisfaction problems. *SIAM J. Comput.* **30**(6), 1863–1920 (2000)
23. Lagerkvist, V.: Weak bases of Boolean co-clones. *Inf. Process. Lett.* **114**(9), 462–468 (2014)
24. Lau, D.: *Function Algebras on Finite Sets: A Basic Course of Many-Valued Logic and Clone Theory*. Springer Monographs in Mathematics. Springer, Berlin (2006)
25. Pöschel, R., Kalužnin, L.A.: *Funktionen-und Relationenalgebren*. Deutscher Verlag der Wissenschaften, Berlin (1979)
26. Romov, B.A.: The algebras of partial functions and their invariants. *Cybernetics* **17**(2), 157–167 (1981). Translated from *Алгебры частичных функций и их инварианты*. *Kibernetika* **17**(2), 1–11 (1981)
27. Schaefer, T.J.: The complexity of satisfiability problems. In: *Proceedings 10th Symposium on Theory of Computing (STOC'78)*, San Diego (California, USA), pp. 216–226 (1978)
28. Schnoor, H., Schnoor, I.: Partial polymorphisms and constraint satisfaction problems. In: Creignou, N., Kolaitis, P.G., Vollmer, H. (eds.) *Complexity of Constraints—An Overview of Current Research Themes [Result of a Dagstuhl Seminar]*, Lecture Notes in Computer Science, vol. 5250, pp. 229–254. Springer, Berlin (2008)
29. Schnoor, I.: *The Weak Base Method for Constraint Satisfaction*. Dissertation, Gottfried Wilhelm Leibniz Universität Hannover (2008)
30. Schrijver, A.: *Theory of Linear and Integer Programming*. Wiley, New York (1986)
31. Vardy, A.: The intractability of computing the minimum distance of a code. *IEEE Trans. Inf. Theory* **IT-43**(6), 1757–1766 (1997)
32. Warshall, S.: A theorem on Boolean matrices. *J. Assoc. Comput. Mach.* **9**(1), 11–12 (1962)