The Next Whisky Bar

Mike Behrisch[1], Miki Hermann[2( )], Stefan Mengel[3], and Gernot Salzer[1]

[1] Technische Universität Wien, Vienna, Austria
{behrisch,salzer}@logic.at
[2] LIX (UMR CNRS 7161), École Polytechnique, Palaiseau, France
hermann@lix.polytechnique.fr
[3] CRIL (UMR CNRS 8188), Université d'Artois, Lens, France
mengel@cril.fr

Abstract. We determine the complexity of an optimization problem related to information theory. Taking a conjunctive propositional formula over some finite set of Boolean relations as input, we seek a satisfying assignment of the formula having minimal Hamming distance to a given assignment that is not required to be a model (NearestSolution, NSol). We obtain a complete classification with respect to the relations admitted in the formula. For two classes of constraint languages we present polynomial time algorithms; otherwise, we prove hardness or completeness concerning the classes APX, poly-APX, NPO, or equivalence to well-known hard optimization problems.

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 the following 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 problem asks to find a satisfying assignment with minimal Hamming distance to a given assignment (NearestSolution, NSol). Note that we do not assume the given assignment to satisfy the formula nor the solution to be different from it as was done in [4], where NearestOtherSolution (NOSol) was studied. This would change the complexity classification (e.g. for bijunctive constraints), and proof techniques would become considerably harder due to inapplicability of clone theory.

The title refers to the *Alabama Song* by Bertolt Brecht (lyrics), Kurt Weill (music), and Elisabeth Hauptmann (English translation). Among the numerous cover versions, the one by Jim Morrison and the Doors became particularly popular in the 1970s.

M. Behrisch, and G. Salzer — Supported by Austrian Science Fund (FWF) grant I836-N23.

M. Hermann — Supported by ANR-11-ISO2-003-01 Blanc International grant ALCOCLAN.

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

A.S. Kulikov and G.J. Woeginger (Eds.): CSR 2016, LNCS 9691, pp. 41–56, 2016.
DOI: 10.1007/978-3-319-34171-2_4

This problem appears in several guises throughout literature. E.g., a common problem in AI 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 [3] 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 $\mathsf{NSol}^{\mathrm{d}}_{\mathrm{pp}}$ later on). While [3] 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.

As is common, these restrictions are given by the set of atomic constraints allowed to appear in the instances of the problem. We give a complete classification of the complexity of approximation with respect to this parameterization, applying methods from clone theory. Despite being classical, for NSol this step requires considerably more non-trivial work than for e.g. satisfiability problems. It turns out that our problem can either be solved in polynomial time, or it is complete for a well-known optimization class, or else it is equivalent to a well-known hard optimization problem.

Our study can be understood as a continuation of the minimization problems investigated by Khanna et al. in [10], especially that of MinOnes. The MinOnes optimization problem asks for a solution of a constraint satisfaction problem with minimal Hamming weight, i.e., minimal Hamming distance to the 0-vector. Our work generalizes this by allowing the given vector to be arbitrary.

Moreover, our work can also be seen as a generalization of questions in coding theory. 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.

2 Preliminaries

An n-ary *Boolean relation* R is a subset of $\{0,1\}^n$; its elements $(b_1,\ldots,b_n)$ 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 $R(\boldsymbol{x})$, where R is an n-ary relation and $\boldsymbol{x}$ 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 $R_1(\boldsymbol{x_1}) \wedge \cdots \wedge R_k(\boldsymbol{x_k})$, where the R_i are relations from Γ and the $\boldsymbol{x_i}$ are variable tuples of suitable arity.

An *assignment* is a mapping $m\colon V \to \{0,1\}$ assigning a Boolean value $m(x)$ to each variable $x \in V$. If we arrange the variables in some arbitrary but fixed order, say as a vector $(x_1,\ldots,x_n)$, then the assignments can be identified with vectors from $\{0,1\}^n$. The i-th component of a vector m is denoted by $m[i]$ and

Table 1. List of Boolean functions and relations

$x \oplus y = x + y \pmod 2$	$\mathrm{or}^k = \{0,1\}^k \smallsetminus \{0 \cdots 0\}$
$x \equiv y = x + y + 1 \pmod 2$	$\mathrm{nand}^k = \{0,1\}^k \smallsetminus \{1 \cdots 1\}$
$\mathrm{dup}^3 = \{0,1\}^3 \smallsetminus \{010, 101\}$	$\mathrm{even}^4 = \{(a_1, a_2, a_3, a_4) \in \{0,1\}^4 \mid \sum_{i=1}^4 a_i \text{ is even}\}$
$\mathrm{nae}^3 = \{0,1\}^3 \smallsetminus \{000, 111\}$	

corresponds to the value of the i-th variable, i.e., $m[i] = m(x_i)$. The *Hamming weight* $\mathrm{hw}(m) = |\{i \mid m[i] = 1\}|$ of m is the number of 1s in the vector m. The *Hamming distance* $\mathrm{hd}(m, m') = |\{i \mid m[i] \neq m'[i]\}|$ of m and m' is the number of coordinates on which the vectors disagree. The complement $\overline{m}$ of a vector m is its pointwise complement, $\overline{m}[i] = 1 - m[i]$.

An assignment m satisfies a constraint $R(x_1, \ldots, x_n)$ if $(m(x_1), \ldots, m(x_n)) \in R$ holds. It satisfies the formula φ if it satisfies all of 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. Note that $[\varphi]$ represents a Boolean relation. In sets of relations represented this way we usually omit the brackets. A *literal* is a variable v, or its negation $\neg v$. Assignments m are extended to literals by defining $m(\neg v) = 1 - m(v)$ (Table 1).

Throughout the text we refer to different types of Boolean constraint relations following Schaefer's terminology [11] (see also the monograph [8] and the survey [6]). A Boolean relation R is (1) 1-*valid* if $1 \cdots 1 \in R$ and it is 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) having at most one unnegated (negated) variable in each clause, (3) *monotone* if it is both Horn and dual Horn, (4) *bijunctive* if it can be represented by a CNF having at most two variables in each 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 $\overline{m} \in R$. A set Γ of Boolean relations is called 0-valid (1-valid, Horn, dual Horn, monotone, affine, bijunctive, complementive) if *every* relation in Γ is 0-valid (1-valid, Horn, dual Horn, monotone, affine, bijunctive, complementive). See also Table 3.

A formula constructed from atoms by conjunction, variable identification, and existential quantification is called a *primitive positive formula* (*pp-formula*). We denote by $\langle \Gamma \rangle$ the set of all relations that can be expressed using relations

Table 2. Some Boolean co-clones with bases

$\mathrm{iM}_2\{x \to y, \neg x, x\}$	$\mathrm{iD}_2\{x \oplus y, x \to y\}$	$\mathrm{iE}_2\{\neg x \vee \neg y \vee z, \neg x, x\}$
$\mathrm{iS}_0^k\{\mathrm{or}^k\}$	$\mathrm{iL}\{\mathrm{even}^4\}$	$\mathrm{iN}\{\mathrm{dup}^3\}$
$\mathrm{iS}_1^k\{\mathrm{nand}^k\}$	$\mathrm{iL}_2\{\mathrm{even}^4, \neg x, x\}$	$\mathrm{iN}_2\{\mathrm{nae}^3\}$
$\mathrm{iS}_{00}^k\{\mathrm{or}^k, x \to y, \neg x, x\}$	$\mathrm{iV}\{x \vee y \vee \neg z\}$	$\mathrm{iI}_0\{\mathrm{even}^4, x \to y, \neg x\}$
$\mathrm{iS}_{10}^k\{\mathrm{nand}^k, \neg x, x, x \to y\}$	$\mathrm{iV}_2\{x \vee y \vee \neg z, \neg x, x\}$	$\mathrm{iI}_1\{\mathrm{even}^4, x \to y, x\}$
$\mathrm{iD}_1\{x \oplus y, x\}$	$\mathrm{iE}\{\neg x \vee \neg y \vee z\}$	

from $\Gamma \cup \{=\}$, conjunction, variable identification (and permutation), cylindrification, and existential quantification. 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 $\langle \Gamma \rangle = \mathcal{B}$. All co-clones, ordered by set inclusion, form a lattice. Together with their respective bases, which were studied in [7], some of them are listed in Table 2. In particular the sets of relations being 0-valid, 1-valid, complementive, Horn, dual Horn, affine, bijunctive, 2affine (both bijunctive and affine) and monotone each form a co-clone denoted by iI_0, iI_1, iN_2, iE_2, iV_2, iL_2, iD_2, iD_1, and iM_2, respectively. See also Table 3.

We assume that the reader has a basic knowledge of approximation algorithms and complexity theory, see e.g. [2,8]. For reductions among decision problems we use polynomial-time many-one reduction denoted by $\leq_{\mathrm{m}}$. Many-one equivalence between decision problems is written as $\equiv_{\mathrm{m}}$. For reductions among optimization problems we employ approximation preserving reductions, also called AP-reductions, represented by $\leq_{\mathrm{AP}}$. AP-equivalence between optimization problems is stated as $\equiv_{\mathrm{AP}}$. Moreover, we shall need the following approximation complexity classes in the hierarchy $\mathrm{PO} \subseteq \mathrm{APX} \subseteq \mathrm{poly\text{-}APX} \subseteq \mathrm{NPO}$.

An optimization problem $\mathcal{P}_1$ AP-*reduces* to another optimization problem $\mathcal{P}_2$ if there are two polynomial-time computable functions f, g, and a constant $\alpha \geq 1$ such that for all $r > 1$ on any input x for $\mathcal{P}_1$ the following holds:

- $f(x)$ is an instance of $\mathcal{P}_2$;
- for any solution y of $f(x)$, $g(x, y)$ is a solution of x;
- whenever y is an r-approximate solution for the instance $f(x)$, then $g(x, y)$ provides a $(1 + (r - 1)\alpha + \mathrm{o}(1))$-approximate solution for x.

If $\mathcal{P}_1$ AP-reduces to $\mathcal{P}_2$ with constant $\alpha \geq 1$ and $\mathcal{P}_2$ has an $f(n)$-approximation algorithm, then there is an $\alpha f(n)$-approximation algorithm for $\mathcal{P}_1$.

We will relate our problems to well-known optimization problems. To this end we make the following convention: For optimization problems $\mathcal{P}$ and $\mathcal{Q}$ we say that $\mathcal{Q}$ is $\mathcal{P}$-*hard* if $\mathcal{P} \leq_{\mathrm{AP}} \mathcal{Q}$, i.e. if $\mathcal{P}$ reduces to it. Moreover, $\mathcal{Q}$ is called $\mathcal{P}$-*complete* if $\mathcal{P} \equiv_{\mathrm{AP}} \mathcal{Q}$.

To prove our results, we refer to the following optimization problems defined and analyzed in [10]. Like our problems they are parameterized by a constraint language Γ.

Problem MinOnes(Γ). Given a conjunctive formula φ over relations from Γ, any assignment m satisfying φ is a feasible solution. The goal is to minimize the Hamming weight $\mathrm{hw}(m)$.

Problem WeightedMinOnes(Γ). Given a conjunctive formula φ over relations in Γ and a weight function $w \colon V \to \mathbb{N}$ on the variables V of φ, solutions are again all assignments m satisfying φ. The objective is to minimize the value $\sum_{x \in V : m(x)=1} w(x)$.

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. Given a matrix $A \in \mathbb{Z}_2^{k \times l}$ and $m \in \mathbb{Z}_2^l$, any vector $x \in \mathbb{Z}_2^k$ is a solution. The objective is to minimize the Hamming distance $\mathrm{hd}(xA, m)$.

Problem MinHornDeletion. For a given conjunctive formula φ over relations from the set $\{[\neg x \vee \neg y \vee z], [x], [\neg x]\}$, an assignment m satisfying φ is sought. The objective is given by the minimum number of unsatisfied conjuncts of φ.

NearestCodeword and MinHornDeletion are known to be NP-hard to approximate within a factor $2^{\Omega(\log^{1-\varepsilon}(n))}$ for every $\varepsilon > 0$ [1,10]. Thus if a problem $\mathcal{P}$ is equivalent to any of these problems, it follows that $\mathcal{P} \notin \mathrm{APX}$ unless P = NP.

We also use the classic satisfiability problem $\mathsf{SAT}(\Gamma)$, given a conjunctive formula φ over relations from Γ, asking if φ is satisfiable. Schaefer presented in [11] a complete classification of complexity for SAT. His dichotomy theorem proves that $\mathsf{SAT}(\Gamma)$ is polynomial-time decidable if Γ is 0-valid ($\Gamma \subseteq \mathrm{iI}_0$), 1-valid ($\Gamma \subseteq \mathrm{iI}_1$), Horn ($\Gamma \subseteq \mathrm{iE}_2$), dual Horn ($\Gamma \subseteq \mathrm{iV}_2$), bijunctive ($\Gamma \subseteq \mathrm{iD}_2$), or affine ($\Gamma \subseteq \mathrm{iL}_2$); otherwise it is NP-complete.

3 Results

This section presents the formal definition of the considered problem, parameterized by a constraint language Γ, and our main result; the proofs follow in subsequent sections.

Problem NearestSolution(Γ), NSol(Γ)
Input: A conjunctive formula φ over relations from Γ and an assignment m of 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 $\mathrm{hd}(m, m')$.

Theorem 1 (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* ($\Gamma \subseteq \mathrm{iD}_1$) *or*
- *(b) monotone* ($\Gamma \subseteq \mathrm{iM}_2$)*;*

(ii) APX-*complete if*
- *(a) $\langle\Gamma\rangle$ contains the relation $[x \vee y]$ and $\Gamma \subseteq \langle x_1 \vee \cdots \vee x_k, x \to y, \neg x, x\rangle$* ($\mathrm{iS}_0^2 \subseteq \langle\Gamma\rangle \subseteq \mathrm{iS}_{00}^k$) *for some $k \in \mathbb{N}$, $k \geq 2$, or*
- *(b) Γ is bijunctive and $\langle\Gamma\rangle$ contains the relation $[x \vee y]$* ($\mathrm{iS}_0^2 \subseteq \langle\Gamma\rangle \subseteq \mathrm{iD}_2$)*, or*
- *(c) $\langle\Gamma\rangle$ contains the relation $[\neg x \vee \neg y]$ and $\Gamma \subseteq \langle \neg x_1 \vee \cdots \vee \neg x_k, x \to y, \neg x, x\rangle$* ($\mathrm{iS}_1^2 \subseteq \langle\Gamma\rangle \subseteq \mathrm{iS}_{10}^k$) *for some $k \in \mathbb{N}$, $k \geq 2$, or*
- *(d) Γ is bijunctive and $\langle\Gamma\rangle$ contains the relation $[\neg x \vee \neg y]$* ($\mathrm{iS}_1^2 \subseteq \langle\Gamma\rangle \subseteq \mathrm{iD}_2$)*;*

(iii) NearestCodeword-*complete if Γ is exactly affine* ($\mathrm{iL} \subseteq \langle\Gamma\rangle \subseteq \mathrm{iL}_2$)*;*

(iv) MinHornDeletion-*complete if Γ is*
- *(a) exactly Horn* ($\mathrm{iE} \subseteq \langle\Gamma\rangle \subseteq \mathrm{iE}_2$) *or*
- *(b) exactly dual Horn* ($\mathrm{iV} \subseteq \langle\Gamma\rangle \subseteq \mathrm{iV}_2$)*;*

(v) poly-APX-complete if Γ does not contain an affine relation and it is
- *(a) either* 0-*valid* ($\mathrm{iN} \subseteq \langle\Gamma\rangle \subseteq \mathrm{iI}_0$) *or*

(b) 1-*valid* (iN $\subseteq \langle\Gamma\rangle \subseteq$ iI$_1$)*; and*
(vi) NPO-*complete otherwise* (iN$_2 \subseteq \langle\Gamma\rangle$).

The considered optimization problem can be transformed into a decision problem in the usual way. We add a bound $k \in \mathbb{N}$ to the input and ask if the Hamming distance satisfies the inequality $\mathrm{hd}(m, m') \leq k$. This way we obtain the corresponding decision problem $\mathsf{NSol}^{\mathrm{d}}$. Its complexity follows immediately from the theorem above. All cases in PO become polynomial-time decidable, whereas the other cases, which are APX-hard, become NP-complete. This way we obtain a dichotomy theorem classifying the decision problem as polynomial or NP-complete for all finite sets Γ of Boolean relations.

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 as we continue.

There are two natural versions of $\mathsf{NSol}(\Gamma)$. In one version the formula φ is quantifier free while in the other one we do allow existential quantification. We call the former version $\mathsf{NSol}(\Gamma)$ and the latter $\mathsf{NSol}_{\mathrm{pp}}(\Gamma)$. Fortunately, we will now see that both versions are equivalent.

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

Lemma 2. *For finite sets Γ we have the equivalences* $\mathsf{NSol}^{\mathrm{d}}(\Gamma) \equiv_{\mathrm{m}} \mathsf{NSol}^{\mathrm{d}}_{\mathrm{pp}}(\Gamma)$ *and* $\mathsf{NSol}(\Gamma) \equiv_{\mathrm{AP}} \mathsf{NSol}_{\mathrm{pp}}(\Gamma)$.

Proof. The reduction from left to right is trivial in both cases. For the other direction, consider first an instance with formula φ, assignment m, and bound k for $\mathsf{NSol}^{\mathrm{d}}_{\mathrm{pp}}(\Gamma)$. Let $x_1, \ldots, x_n$ be the free variables of φ and let $y_1, \ldots, y_\ell$ be the existentially quantified variables, which can be assumed to be disjoint. For each variable z we define a set $B(z)$ as follows:

$$B(z) = \begin{cases} \{x_i^j \mid j \in \{1, \ldots, (n+\ell+1)^2\}\} & \text{if } z = x_i \text{ for some } i \in \{1, \ldots, n\}, \\ \{y_i\} & \text{if } z = y_i \text{ for some } i \in \{1, \ldots, \ell\}. \end{cases}$$

We construct a quantifier-free formula φ' over the variables $\bigcup_{i=1}^{n} B(x_i) \cup \bigcup_{i=1}^{\ell} B(y_i)$ that contains for every atom $R(z_1, \ldots, z_s)$ from φ the atom $R(z_1', \ldots, z_s')$ for every combination $(z_1', \ldots, z_s')$ from $B(z_1) \times \cdots \times B(z_s)$. 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 $\mathrm{hd}(m, m') \leq k$ if and only if φ' has a solution m'' with $\mathrm{hd}(B(m), m'') \leq k(n+\ell+1)^2 + \ell$. 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 $m''(y_i) := m'_{\mathrm{e}}(y_i)$ for all i and j. Then m'' is clearly a satisfying assignment of φ'. Moreover, m'' and $B(m)$ differ in at most $k(n+\ell+1)^2$ variables among the x_i^j. Since there exist only ℓ other variables y_i, we get $\mathrm{hd}(m'', B(m)) \leq k(n+\ell+1)^2 + \ell$ as desired.

Now suppose m'' satisfies φ' with $\mathrm{hd}(B(m), m'') \leq k(n+\ell+1)^2 + \ell$. We may assume for each i that $m''(x_i^1) = \cdots = m''(x_i^{(n+\ell+1)^2})$. Indeed, if this is not the case, then setting all x_i^j to $B(m)(x_i^j) = m(x_i)$ will give us a satisfying assignment closer to $B(m)$. After at most n iterations we get some m'' as desired. Now define an assignment m' to 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 $(n+\ell+1)^2\,\mathrm{hd}(m, m') \leq \mathrm{hd}(B(m), m'') \leq k(n+\ell+1)^2 + \ell$. Therefore, we have the inequality $\mathrm{hd}(m, m') \leq k + \ell/(n+\ell+1)^2$ and hence $\mathrm{hd}(m, m') \leq k$. This completes the many-one reduction.

We claim that the above construction is an AP-reduction, too. To this end, let m'' be an r-approximation for φ' and $B(m)$, i.e., $\mathrm{hd}(B(m), m'') \leq r \cdot \mathrm{OPT}(\varphi', B(m))$. Construct m' as before, so $(n+\ell+1)^2\,\mathrm{hd}(m, m') \leq \mathrm{hd}(B(m), m'') \leq r \cdot \mathrm{OPT}(\varphi', B(m))$. Since $\mathrm{OPT}(\varphi', B(m))$ is at most $(n+\ell+1)^2$ $\mathrm{OPT}(\varphi, m) + \ell$ as before, we get $(n+\ell+1)^2\,\mathrm{hd}(m, m') \leq r((n+\ell+1)^2$ $\mathrm{OPT}(\varphi, m) + \ell)$. This implies the inequality $\mathrm{hd}(m, m') \leq r \cdot \mathrm{OPT}(\varphi, m) + r \cdot \ell/(n+\ell+1)^2 = (r + \mathrm{o}(1)) \cdot \mathrm{OPT}(\varphi, m)$ and shows that the construction is an AP-reduction with $\alpha = 1$. □

Remark 3. Note that in the reduction from $\mathsf{NSol}^{\mathrm{d}}_{\mathrm{pp}}(\Gamma)$ to $\mathsf{NSol}^{\mathrm{d}}(\Gamma)$ 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.

We can also show that introducing explicit equality constraints does not change the complexity of our problem.

Lemma 4. *For constraint languages Γ we have $\mathsf{NSol}^{\mathrm{d}}(\Gamma) \equiv_{\mathrm{m}} \mathsf{NSol}^{\mathrm{d}}(\Gamma \cup \{=\})$ and $\mathsf{NSol}(\Gamma) \equiv_{\mathrm{AP}} \mathsf{NSol}(\Gamma \cup \{=\})$.*

Although a proof of this statement can be established by similar methods as those used in Lemma 2, it is a technically rather involved case distinction whose length exceeds the scope of this presentation. The proof is therefore omitted.

Lemmas 2 and 4 are very convenient, because they allow us to freely switch between formulas with quantifiers and equality and those without. This allows us to give all upper bounds in the setting without quantifiers and equality while freely using them in all hardness reductions. In particular it follows that 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 $\mathsf{NSol}(\Gamma)$ for every finite set of Boolean relations Γ.

Corollary 5. *For constraint languages Γ, Γ' such that $\Gamma' \subseteq \langle \Gamma \rangle$, we have the reductions $\mathsf{NSol}^{\mathrm{d}}(\Gamma') \leq_{\mathrm{m}} \mathsf{NSol}^{\mathrm{d}}(\Gamma)$ and $\mathsf{NSol}(\Gamma') \leq_{\mathrm{AP}} \mathsf{NSol}(\Gamma)$. Thus, if*

$\langle\Gamma'\rangle = \langle\Gamma\rangle$ is satisfied, then the equivalences $\mathsf{NSol}^{\mathrm{d}}(\Gamma) \equiv_{\mathrm{m}} \mathsf{NSol}^{\mathrm{d}}(\Gamma')$ *and* $\mathsf{NSol}(\Gamma) \equiv_{\mathrm{AP}} \mathsf{NSol}(\Gamma')$ *hold.*

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

Lemma 6. *We have the equivalence* $\mathsf{NSol}(\Gamma) \equiv_{\mathrm{AP}} \mathsf{NSol}(\Gamma \cup \{[x], [\neg x]\})$ *for any constraint language Γ where the problem of finding feasible solutions of* $\mathsf{NSol}(\Gamma)$ *is polynomial-time decidable.*

Proof. The direction from left to right is obvious. For the other direction, we show an AP-reduction from $\mathsf{NSol}(\Gamma \cup \{[x], [\neg x]\})$ to $\mathsf{NSol}(\Gamma \cup \{[x \equiv y]\})$. Since $[x \equiv y]$ is by definition in every co-clone and thus in $\langle\Gamma\rangle$, the result follows from Corollary 5.

The idea of the construction is to introduce two sets of variables $y_1, \ldots, y_{n^2}$ and $z_1, \ldots, z_{n^2}$ such that in any feasible solution all y_i and all z_i take the same value. Then 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 unary relations x and $\neg x$ by $x \equiv y_1$ and $x \equiv z_1$, respectively. We now describe the reduction formally.

Let the formula φ and the assignment m be a $\Gamma \cup \{[x], [\neg x]\}$-formula over the variables $x_1, \ldots, x_n$ with a feasible solution. We construct a $\Gamma \cup \{[x \equiv y]\}$-formula φ' over the variables $x_1, \ldots x_n$, $y_1, \ldots, y_{n^2}$, $z_1, \ldots, z_{n^2}$ and an assignment m'. We get φ' from φ by substituting every occurrence of a constraint $[x_i]$ for some variable x_i by $x_i \equiv y_1$ and substituting every occurrence $[\neg x_i]$ for every variable x_i by $x_i \equiv z_1$. Finally, add $y_i \equiv y_j$ for all $i, j \in \{1, \ldots, n^2\}$ and $z_i \equiv z_j$ for all $i, j \in \{1, \ldots, n^2\}$. Let m' be the assignment of the variables of φ' given by $m'(x_i) = m(x_i)$ for each $i \in \{1, \ldots, n\}$, and $m'(y_i) = 1$ and $m'(z_i) = 0$ for all $i \in \{1, \ldots, n^2\}$. 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 $i \in \{1, \ldots, n^2\}$ or $m''(z_i) = 1$ for all i in $\{1, \ldots, n^2\}$, we define $g(\varphi, m, m'')$ to be any satisfying assignment of φ.
3. Otherwise, $m''(y_i) = 1$ for all $i \in \{1, \ldots, n^2\}$ and $m''(z_i) = 0$, we define $g(\varphi, m, m'')$ to be the restriction of m'' onto $x_1, \ldots, x_n$.

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 by assumption, so $g(\varphi, m, m'')$ is well-defined and feasible by construction. Observe that m' and m'' disagree on all y_i or on all z_i, so $\mathrm{hd}(m', m'') \geq n^2$ holds. Moreover, since φ has a feasible solution, it follows that $\mathrm{OPT}(\varphi', m') \leq n$. Since m'' is an r-approximate solution, we have that $r \geq \mathrm{hd}(m', m'')/\mathrm{OPT}(\varphi', m') \geq n$. Consequently, the distance $\mathrm{hd}(m, g(\varphi, m, m''))$ is bounded above by $n \leq r \leq r \cdot \mathrm{OPT}(\varphi, m)$, where the last inequality holds because φ is not satisfied by m and thus the distance of the optimal solution from m is at least 1.

Case 3: The variables x_i for which $[x_i]$ is a constraint all have $g(\varphi, m, m'')(x_i) = 1$ 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, $\mathrm{OPT}(\varphi', m') \leq n$, so the optimal solution to (φ', m') must set all variables y_i to 1 and all z_i to 0. It follows that $\mathrm{OPT}(\varphi, m) = \mathrm{OPT}(\varphi', m')$. Thus we get

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

which completes the proof. □

Given a relation $R \subseteq \{0,1\}^n$, its *dual* relation is $\mathrm{dual}(R) = \{\overline{m} \mid m \in R\}$, i.e., the relation containing the complements of vectors from R. Duality naturally extends to sets of relations and co-clones. We define $\mathrm{dual}(\Gamma) = \{\mathrm{dual}(R) \mid R \in \Gamma\}$ as the set of dual relations to Γ. 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 lemma shows that it is sufficient to consider only one half of Post's lattice of co-clones.

Lemma 7. *For any set Γ of Boolean relations we have* $\mathsf{NSol}^{\mathrm{d}}(\Gamma) \equiv_{\mathrm{m}} \mathsf{NSol}^{\mathrm{d}}(\mathrm{dual}(\Gamma))$ *and* $\mathsf{NSol}(\Gamma) \equiv_{\mathrm{AP}} \mathsf{NSol}(\mathrm{dual}(\Gamma))$.

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

5 Finding the Nearest Solution

This section contains the proof of Theorem 1. We first consider the polynomial-time cases followed by the cases of higher complexity.

5.1 Polynomial-Time Cases

Proposition 8. *If a constraint language Γ is both bijunctive and affine ($\Gamma \subseteq \mathrm{iD}_1$), then $\mathsf{NSol}(\Gamma)$ can be solved in polynomial time.*

Proof. Since $\Gamma \subseteq \mathrm{iD}_1 = \langle \Gamma' \rangle$ with $\Gamma' := \{[x \oplus y], [x]\}$, we have the reduction $\mathsf{NSol}(\Gamma) \leq_{\mathrm{AP}} \mathsf{NSol}(\Gamma')$ by Corollary 5. Every Γ'-formula φ is equivalent to a linear system of equations over the Boolean ring $\mathbb{Z}_2$ of the 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. 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. 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 $\min\{\mathrm{hd}(m, m_1), \mathrm{hd}(m, m_2)\}$. □

Proposition 9. *If a constraint language Γ is monotone ($\Gamma \subseteq \mathrm{iM}_2$), then* $\mathsf{NSol}(\Gamma)$ *can be solved in polynomial time.*

Proof. We have $\mathrm{iM}_2 = \langle \Gamma' \rangle$ where $\Gamma' := \{[x \to y], [\neg x], [x]\}$. Thus Corollary 5 and $\Gamma \subseteq \langle \Gamma' \rangle$ imply $\mathsf{NSol}(\Gamma) \leq_{\mathrm{AP}} \mathsf{NSol}(\Gamma')$. The relations $[\neg x]$ and $[x]$ determine the unique value of the considered variable, therefore we can eliminate the unit clauses built from the two latter relations and propagate. We consider formulas φ built only from the relation $[x \to y]$, i.e., formulas containing only binary implicative clauses of the type $x \to y$.

Let V the set of variables of the formula φ. According to the value assigned to the variables by the vector m, we can divide V into two disjoint subsets V_0 and V_1, such that $V_i = \{x \in V \mid m(x) = i\}$. We transform the formula φ to an integer programming problem P. First, for each clause $x \to y$ from φ we add to P the relation $y \geq x$. For each variable $x \in V$ we add the constraints $x \geq 0$ and $x \leq 1$, with $x \in \{0, 1\}$. Finally, we construct the linear function f_φ by defining

$$f_\varphi(m') = \sum_{x_i \in V_0} m'(x_i) + \sum_{x_j \in V_1} (1 - m'(x_j))$$

for assignments m' of φ. Obviously, $f_\varphi(m')$ counts the number of variables changing their parity between m and m', i.e., $f_\varphi(m') = \mathrm{hd}(m, m')$. As P is totally unimodular, the minimum of f_φ can be computed in polynomial time (see e.g. [12]). □

5.2 Hard Cases

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

Lemma 10. *Let Γ be a finite set of Boolean relations. If $\mathrm{iN}_2 \subseteq \langle \Gamma \rangle$, then* $\mathsf{NSol}(\Gamma)$ *is* NPO*-complete; otherwise,* $\mathsf{NSol}(\Gamma) \in$ *poly-APX.*

Proof. If $\mathrm{iN}_2 \subseteq \langle \Gamma \rangle$ holds, finding a solution for $\mathsf{NSol}(\Gamma)$ is NP-hard by Schaefer's theorem [11], hence $\mathsf{NSol}(\Gamma)$ is NPO-complete.

We give an n-approximation algorithm for the other case. Let a formula φ and a model m be an instance of $\mathsf{NSol}(\Gamma)$. If m is a solution of φ, return m. Otherwise, compute an arbitrary solution m' of φ, which is possible by Schaefer's theorem, and return m'.

The approximation ratio of this algorithm is n. Indeed, if m satisfies φ, this is obviously true, because we return the exact solution. Otherwise, we have $\mathrm{OPT}(\varphi, m) \geq 1$ and so, trivially, $\mathrm{hd}(m, m') \leq n$ whence the claim follows. □

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 11. *$\mathsf{NSol}(\Gamma)$ is APX-hard for every constraint language Γ satisfying the inclusion $\mathrm{iS}_0^2 \subseteq \langle \Gamma \rangle$ or $\mathrm{iS}_1^2 \subseteq \langle \Gamma \rangle$.*

Proof. We have $\mathrm{iS}_0^2 = \langle \{[x \vee y]\} \rangle$, whereas $\mathrm{iS}_1^2 = \langle \{[\neg x \vee \neg y]\} \rangle$. So we discuss the former case, the latter one being symmetric and provable from the first one by Corollary 5.

We encode $\mathsf{VertexCover}$ into $\mathsf{NSol}(\{[x \vee y]\}) \leq_{\mathrm{AP}} \mathsf{NSol}(\Gamma)$ (see Corollary 5). 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. Taking $m = \mathbf{0}$, then $\mathrm{hd}(\mathbf{0}, m')$ 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 $\mathsf{VertexCover}$ is APX-complete (see e.g. [2]), the result follows. □

Proposition 12. *We have $\mathsf{NSol}(\Gamma) \in \mathrm{APX}$ for constraint languages $\Gamma \subseteq \mathrm{iD}_2$.*

Proof. As $\{x \oplus y, x \rightarrow z\}$ is a basis of iD_2, it suffices to show that $\mathsf{NSol}(\{x \oplus y, x \rightarrow y\})$ is in APX by Corollary 5. Let (φ, m) be an input of this problem. Feasibility for φ can be written as an integer program as follows: Every constraint $x_i \oplus x_j$ induces a linear equation $x_i + x_j = 1$. Every constraint $x_i \rightarrow x_j$ can be written as $x_i \leq x_j$. If we restrict all variables to $\{0, 1\}$ by the appropriate inequalities, it is clear that any assignment m' satisfies φ if it satisfies the linear system with inequality side conditions. We complete the construction of the linear program by adding the objective function $c(m') := \sum_{i:m(x_i)=0} m'(x_i) + \sum_{i:m(x_i)=1} (1 - m'(x_i))$. Clearly, for every m' we have $c(m') = \mathrm{hd}(m, m')$. The 2-approximation algorithm from [9] for integer linear programs, in which in every inequality at most two variables appear, completes the proof. □

Proposition 13. *We have $\mathsf{NSol}(\Gamma) \in \mathrm{APX}$ for constraint languages $\Gamma \subseteq \mathrm{iS}_{00}^{\ell}$ with $\ell \geq 2$.*

Proof. Due to $\{x_1 \vee \cdots \vee x_\ell, x \rightarrow y, \neg x, x\}$ being a basis of iS_{00}^{ℓ} and Corollary 5, it suffices to show $\mathsf{NSol}(\{x_1 \vee \cdots \vee x_\ell, x \rightarrow y, \neg x, x\}) \in \mathrm{APX}$. Let formula φ and assignment m be an instance of that problem. We will use an approach similar to that for the corresponding case in [10], again writing φ as an integer program. Every constraint $x_{i_1} \vee \cdots \vee x_{i_\ell}$ is translated to an inequality $x_{i_1} + \cdots + x_{i_\ell} \geq 1$. Every constraint $x_i \rightarrow x_j$ is written as $x_i \leq x_j$. Each $\neg x_i$ is turned into $x_i = 0$, every constraint x_i yields $x_i = 1$. Add $x_i \geq 0$ and

$x_i \leq 1$ for each variable x_i. Again, it is easy to check that feasible Boolean solutions of φ and the linear system coincide. Defining again the objective function $c(m') = \sum_{i:m(x_i)=0} m'(x_i) + \sum_{i:m(x_i)=1}(1 - m'(x_i))$, we have $\mathrm{hd}(m, m') = c(m')$ for every m'. Therefore it suffices to approximate the optimal solution for the 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 $m'(x_i) = 0$ if $m''(x_i) < 1/\ell$ and $m'(x_i) = 1$ if $m''(x_i) \geq 1/\ell$. As $\ell \geq 2$, we get $\mathrm{hd}(m, m') = c(m') \leq \ell c(m'') \leq \ell \cdot \mathrm{OPT}(\varphi, m)$. It is easy to check that m' is a feasible solution, which completes the proof. □

Lemma 14. *We have* $\mathsf{MinOnes}(\Gamma) \leq_{\mathrm{AP}} \mathsf{NSol}(\Gamma)$ *for any constraint language* Γ.

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

Proposition 15 (Khanna et al. [10, Theorem 2.14]**).** *The problem* $\mathsf{MinOnes}(\Gamma)$ *is* $\mathsf{NearestCodeword}$*-complete for constraint languages* Γ *satisfying* $\langle\Gamma\rangle = \mathrm{iL}_2$.

Corollary 16. *For a constraint language* Γ *satisfying* $\mathrm{iL} \subseteq \langle\Gamma\rangle$, *the problem* $\mathsf{NSol}(\Gamma)$ *is* $\mathsf{NearestCodeword}$*-hard.*

Proof. Let $\Gamma' := \{\mathrm{even}^4, [x], [\neg x]\}$. Since $\langle\Gamma'\rangle = \mathrm{iL}_2$, $\mathsf{NearestCodeword}$ is equivalent to $\mathsf{MinOnes}(\Gamma')$, which reduces to $\mathsf{NSol}(\Gamma')$ by Lemma 14. We have now the AP-equivalence $\mathsf{NSol}(\Gamma') \equiv_{\mathrm{AP}} \mathsf{NSol}(\{\mathrm{even}^4\})$ by appealing to Lemma 6 and the reduction $\mathsf{NSol}(\{\mathrm{even}^4\}) \leq_{\mathrm{AP}} \mathsf{NSol}(\Gamma)$ due to $\mathrm{even}^4 \in \mathrm{iL} \subseteq \langle\Gamma\rangle$ and Corollary 5. □

Proposition 17. *We have* $\mathsf{NSol}(\Gamma) \leq_{\mathrm{AP}} \mathsf{MinOnes}(\{\mathrm{even}^4, [\neg x], [x]\})$ *for any constraint language* $\Gamma \subseteq \mathrm{iL}_2$.

Proof. The set $\Gamma' := \{\mathrm{even}^4, [\neg x], [x]\}$ is a basis of iL_2, therefore by Corollary 5 it is sufficient to show $\mathsf{NSol}(\Gamma') \leq_{\mathrm{AP}} \mathsf{MinOnes}(\Gamma')$.

We proceed by reducing $\mathsf{NSol}(\Gamma')$ to a subproblem of $\mathsf{NSol}_{\mathrm{pp}}(\Gamma')$, where only instances $(\varphi, \mathbf{0})$ are considered. Then, using Lemma 2 and Remark 3, this reduces to a subproblem of $\mathsf{NSol}(\Gamma')$ with the same restriction on the assignments, which is exactly $\mathsf{MinOnes}(\Gamma')$. Note that $[x \oplus y]$ is equal to $[\exists z \exists z' (\mathrm{even}^4(x, y, z, z') \wedge \neg z \wedge z']$ so we can freely use $[x \oplus y]$ in any Γ'-formula. Let formula φ and assignment m be an instance of $\mathsf{NSol}(\Gamma')$. 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 $m(x)$. Moreover, we have that $\mathrm{hd}(m, s) = \mathrm{hd}(\mathbf{0}, I(s))$. This yields a trivial AP-reduction with $\alpha = 1$. □

Proposition 18 (Khanna et al. [10]). *The problems* $\mathsf{MinOnes}(\{x \lor y \lor \neg z, x, \neg x\})$ *and* $\mathsf{WeightedMinOnes}(\{x \lor y \lor \neg z, x \lor y\})$ *are* $\mathsf{MinHornDeletion}$-*complete.*

Lemma 19. $\mathsf{NSol}(\{x \lor y \lor \neg z\}) \leq_{\mathrm{AP}} \mathsf{WeightedMinOnes}(\{x \lor y \lor \neg z, x \lor y\})$.

Proof. Let formula φ and assignment m be an instance of $\mathsf{NSol}(x \lor y \lor \neg z)$ over the variables $x_1, \ldots, x_n$. If m satisfies φ then the reduction is trivial. We assume in the remainder of the proof that $\mathrm{OPT}(\varphi, m) > 0$. Let $T(m)$ be the set of variables x_i with $m(x_i) = 1$. We construct a $\{x \lor y \lor \neg z, x \lor y\}$-formula from φ by adding for each $x_i \in T(m)$ the constraint $x_i \lor x_i'$ where x_i' is a new variable. We set the weights of the variables of φ' as follows. For $x_i \in T(m)$ 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 $\mathrm{OPT}(\varphi')$. Note that whenever for $x_i \in T(m)$ 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 $x_i \in T(m)$ 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

$$n_0 = |\{i \mid x_i \in T(m), m'(x_i) = 0\}| = |\{i \mid x_i \in T(m), m'(x_i) \neq m(x_i)\}|$$
$$n_1 = |\{i \mid x_i \notin T(m), m'(x_i) = 1\}| = |\{i \mid x_i \notin T(m), m'(x_i) \neq m(x_i)\}|,$$

which means that $w(m')$ equals $\mathrm{hd}(m, m'')$. Analogously, the optima in both problems correspond, that is we have $\mathrm{OPT}(\varphi') = \mathrm{OPT}(\varphi, m)$. From this we deduce the final inequality $\mathrm{hd}(m, m'')/\mathrm{OPT}(\varphi, m) = w(m')/\mathrm{OPT}(\varphi') \leq r$. □

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

$\Gamma \subseteq \mathrm{iI}_0 \Leftrightarrow \Gamma$ is 0-valid	$\Gamma \subseteq \mathrm{iI}_1 \Leftrightarrow \Gamma$ is 1-valid
$\Gamma \subseteq \mathrm{iE}_2 \Leftrightarrow \Gamma$ is Horn	$\Gamma \subseteq \mathrm{iV}_2 \Leftrightarrow \Gamma$ is dual Horn
$\Gamma \subseteq \mathrm{iM}_2 \Leftrightarrow \Gamma$ is monotone	$\Gamma \subseteq \mathrm{iD}_2 \Leftrightarrow \Gamma$ is bijunctive
$\Gamma \subseteq \mathrm{iL}_2 \Leftrightarrow \Gamma$ is affine	$\Gamma \subseteq \mathrm{iD}_1 \Leftrightarrow \Gamma$ is 2affine
$\Gamma \subseteq \mathrm{iN}_2 \Leftrightarrow \Gamma$ is complementive	$\Gamma \subseteq \mathrm{iI} \Leftrightarrow \Gamma$ is both 0- and 1-valid

Proposition 20. *For every dual Horn constraint language* $\Gamma \subseteq \mathrm{iV}_2$ *we have the reduction* $\mathsf{NSol}(\Gamma) \leq_{\mathrm{AP}} \mathsf{WeightedMinOnes}(\{x \lor y \lor \neg z, x \lor y\})$.

Proof. Since $\{x \lor y \lor \neg z, x, \neg x\}$ is a basis of iV_2, by Corollary 5 it suffices to prove the reduction $\mathsf{NSol}(\{x \lor y \lor \neg z, x, \neg x\}) \leq_{\mathrm{AP}} \mathsf{WeightedMinOnes}(\{x \lor y \lor \neg z, x \lor y\})$. To this end, first reduce $\mathsf{NSol}(\{x \lor y \lor \neg z, x, \neg x\})$ to $\mathsf{NSol}(x \lor y \lor \neg z)$ by Lemma 6 and then use Lemma 19. □

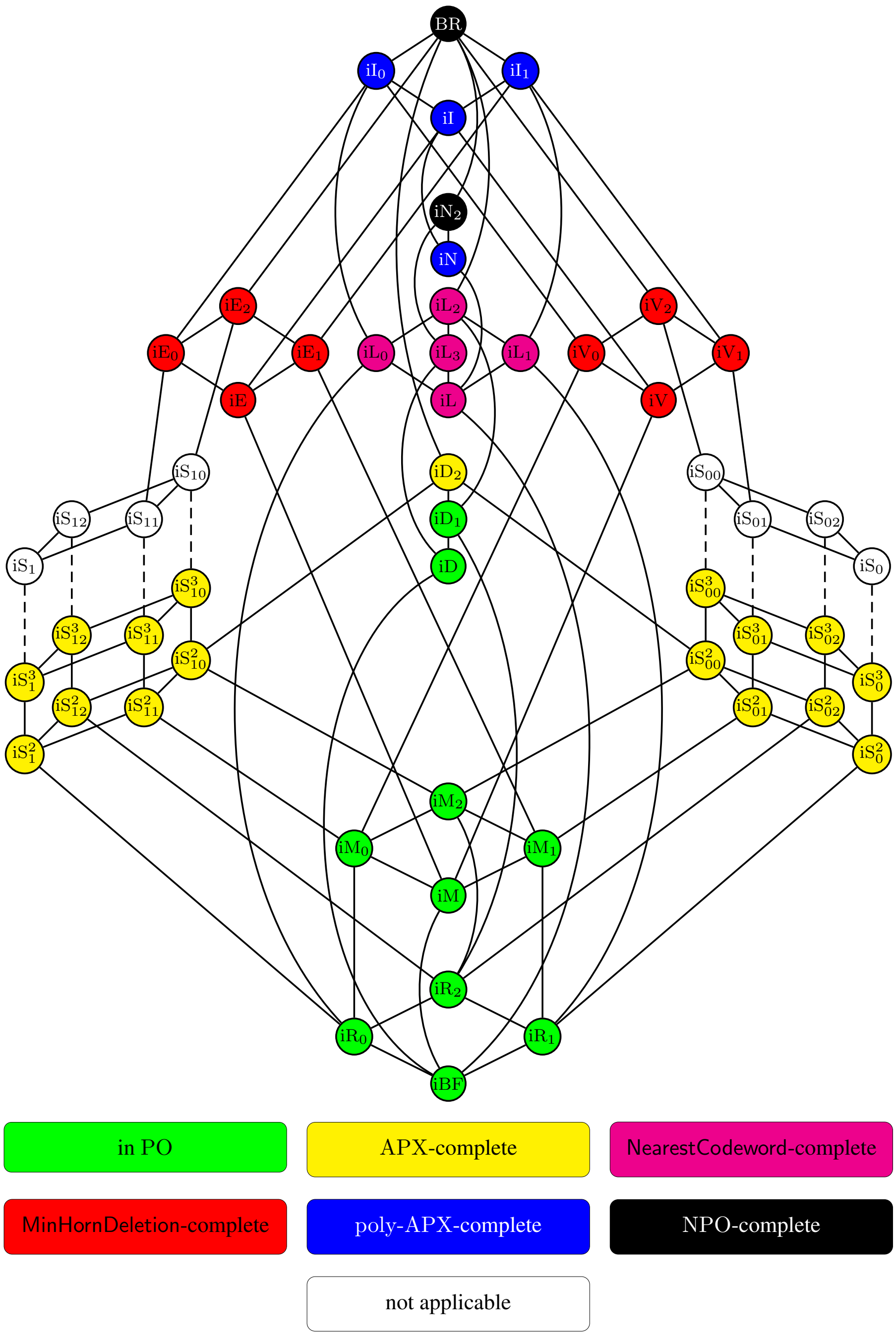

Fig. 1. Lattice of coclones with complexity classification for NSol

Proposition 21. *$\mathsf{NSol}(\Gamma)$ is $\mathsf{MinHornDeletion}$-hard for finite Γ with* $\mathrm{iV}_2 \subseteq \langle\Gamma\rangle$.

Proof. For $\Gamma' := \{x \vee y \vee \neg z, x, \neg x\}$ we have $\mathsf{MinHornDeletion} \equiv_{\mathrm{AP}} \mathsf{MinOnes}(\Gamma')$ by Proposition 18. Now it follows $\mathsf{MinOnes}(\Gamma') \leq_{\mathrm{AP}} \mathsf{NSol}(\Gamma') \leq_{\mathrm{AP}} \mathsf{NSol}(\Gamma)$ using Lemma 14 and Corollary 5 on the assumption $\Gamma' \subseteq \mathrm{iV}_2 \subseteq \langle\Gamma\rangle$. □

Proposition 22. *The problem $\mathsf{NSol}(\Gamma)$ is poly-APX-hard for constraint languages Γ verifying* $\mathrm{iN} \subseteq \langle\Gamma\rangle$.

Proof. The constraint language $\Gamma_1 := \{\mathrm{even}^4, x \to y, x\}$ is a base of iI_1. $\mathsf{MinOnes}(\Gamma_1)$ is poly-APX-hard by Theorem 2.14 of [10] and reduces to $\mathsf{NSol}(\Gamma_1)$ by Lemma 14. Since $(x \to y) = \mathrm{dup}^3(0, x, y) = \exists z(\mathrm{dup}^3(z, x, y) \wedge \neg z)$, we have the reductions $\mathsf{NSol}(\Gamma_1) \leq_{\mathrm{AP}} \mathsf{NSol}(\Gamma_1 \cup \{\neg x, \mathrm{dup}^3\}) \leq_{\mathrm{AP}} \mathsf{NSol}(\{\mathrm{even}^4, \mathrm{dup}^3, x, \neg x\})$ by Corollary 5. Lemma 6 implies $\mathsf{NSol}(\{\mathrm{even}^4, \mathrm{dup}^3, x, \neg x\}) \equiv_{\mathrm{AP}} \mathsf{NSol}(\{\mathrm{even}^4, \mathrm{dup}^3\})$; the latter problem reduces to $\mathsf{NSol}(\Gamma)$ because of $\{\mathrm{even}^4, \mathrm{dup}^3\} \subseteq \mathrm{iN} \subseteq \langle\Gamma\rangle$ and Corollary 5 □

6 Concluding Remarks

Considering the optimization problem NSol is part of a more general research program (cf. [4,5]) studying the approximation complexity of Boolean constraint satisfaction problems in connection with Hamming distance. The studied problems fundamentally differ in the resulting complexity classification as well as in the methods applicable to them (e.g. stability under pp-definitions and applicability of classical Galois theory for Boolean clones vs. the need for minimal weak bases for weak co-clones).

The problem NSol is in PO for constraints, which are both bijunctive and affine, or both Horn and dual Horn (also called monotone). In the interval of constraint languages starting from those encoding vertex cover up to those encoding hitting set for fixed arity hypergraphs or up to bijunctive constraints, NSol becomes APX-complete. This indicates that the solution structure for these types of constraints is more complex, and it becomes even more complicated for Horn or dual Horn constraints. The next complexity stage of the solution structure is characterized by affine constraints. In fact, these represent the error correcting codes used in real-word applications. Even if we know that the given assignment satisfies the constraint – contrary to the real-word situation in the case of nearest neighbor decoding – the optimization problem NSol is surprisingly equivalent to the one of finding the nearest codeword. The penultimate stage of solution structure complexity is given by 0-valid or 1-valid constraint languages, where one finds poly-APX-completeness. This implies that we cannot get a suitable approximation for these problems. It is implicit in NSol to check for the existence of at least one solution. For the last case, when the constraint language is equivalent to NAESAT, this is hard, where membership in iN_2 implies intractability of the SAT problem. Hence, a polynomial-time approximation is not possible at all.

It can be observed that NSol has a similar complexity classification as the problem MinOnes. However, the relations inhabiting these classification cases are different. For instance, the Horn case is in PO for MinOnes, whereas it is MinHornDeletion-complete for NSol. Another diffence w.r.t. MinOnes is that our complexity classification preserves duality, i.e. that NSol(Γ) and NSol(dual(Γ)) always have the same complexity.

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. Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A., Protasi, M.: Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties. Springer, Heidelberg (1999)
3. Bailleux, O., Marquis, P.: Some computational aspects of Distance-SAT. J. Autom. Reasoning **37**(4), 231–260 (2006)
4. Behrisch, M., Hermann, M., Mengel, S., Salzer, G.: Give me another one!. In: Elbassioni, K., Makino, K. (eds.) ISAAC 2015. LNCS, vol. 9472, pp. 664–676. Springer, Heidelberg (2015). doi:10.1007/978-3-662-48971-0_56
5. Behrisch, M., Hermann, M., Mengel, S., Salzer, G.: As close as it gets. In: Kaykobad, M., Petreschi, R. (eds.) WALCOM 2016. LNCS, vol. 9627, pp. 222–235. Springer, Heidelberg (2016). doi:10.1007/978-3-319-30139-6_18
6. Böhler, E., Creignou, N., Reith, S., Vollmer, H.: Playing with Boolean blocks, part II: constraint satisfaction problems. SIGACT News, Complex. Theor. Column 43 **35**(1), 22–35 (2004)
7. Böhler, E., Reith, S., Schnoor, H., Vollmer, H.: Bases for Boolean co-clones. Inf. Process. Lett. **96**(2), 59–66 (2005)
8. Creignou, N., Khanna, S., Sudan, M.: Complexity Classifications of Boolean Constraint Satisfaction Problems. SIAM Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia (PA) (2001)
9. Hochbaum, D.S., Megiddo, N., Naor, 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)
10. Khanna, S., Sudan, M., Trevisan, L., Williamson, D.P.: The approximability of constraint satisfaction problems. SIAM J. Comput. **30**(6), 1863–1920 (2000)
11. Schaefer, T.J.: The complexity of satisfiability problems. In: Proceedings of the 10th Symposium on Theory of Computing (STOC 1978), San Diego, pp. 216–226 (1978)
12. Schrijver, A.: Theory of Linear and Integer Programming. John Wiley & Sons, New York (1986)