Relating the Time Complexity of Optimization Problems in Light of the Exponential-Time Hypothesis*

Peter Jonsson[1], Victor Lagerkvist[1], Johannes Schmidt[1] and Hannes Uppman[1]

Department of Computer and Information Science, Linköping University, Sweden
{peter.jonsson, victor.lagerkvist, johannes.schmidt,
hannes.uppman}@liu.se

Abstract. Obtaining lower bounds for NP-hard problems has for a long time been an active area of research. Recent algebraic techniques introduced by Jonsson et al. (SODA 2013) show that the time complexity of the parameterized $\mathrm{SAT}(\cdot)$ problem correlates to the lattice of strong partial clones. With this ordering they isolated a relation R such that $\mathrm{SAT}(R)$ can be solved at least as fast as any other NP-hard $\mathrm{SAT}(\cdot)$ problem. In this paper we extend this method and show that such languages also exist for the *max ones problem* ($\mathrm{MAX\text{-}ONES}(\Gamma)$) and the *Boolean valued constraint satisfaction problem* over finite-valued constraint languages ($\mathrm{VCSP}(\Delta)$). With the help of these languages we relate MAX-ONES and VCSP to the exponential time hypothesis in several different ways.

1 Introduction

A superficial analysis of the NP-complete problems may lead one to think that they are a highly uniform class of problems: in fact, under polynomial-time reductions, the NP-complete problems may be viewed as a *single* problem. However, there are many indications (both from practical and theoretical viewpoints) that the NP-complete problems are a diverse set of problems with highly varying properties, and this becomes visible as soon as one starts using more refined methods. This has inspired a strong line of research on the "inner structure" of the set of NP-complete problem. Examples include the intensive search for faster algorithms for NP-complete problems [23] and the highly influential work on the *exponential time hypothesis* (ETH) and its variants [14]. Such research might not directly resolve whether P is equal to NP or not, but rather attempts to explain the seemingly large difference in complexity between NP-hard problems and what makes one problem harder than another. Unfortunately there is still a lack of general methods for studying and comparing the complexity of NP-complete problems with more restricted notions of reducibility. Jonsson et al. [10] presented a framework based on *clone theory*, applicable to problems that can be viewed as "assigning values to variables", such as constraint satisfaction problems, the vertex cover problem, and integer programming problems. To analyze and relate the complexity of these problems

* This is an extended version of *Relating the Time Complexity of Optimization Problems in Light of the Exponential-Time Hypothesis*, appearing in Proceedings of the 39th International Symposium on Mathematical Foundations of Computer Science MFCS 2014 Budapest, August 25-29, 2014

in greater detail we utilize polynomial-time reductions which increase the number of variables by a constant factor (*linear variable reductions* or *LV-reductions*) and reductions which increases the amount of variables by a constant (*constant variable reductions* or *CV-reductions*). Note the following: (1) if a problem A is solvable in $O(c^n)$ time (where n denotes the number of variables) for all $c > 1$ and if problem B is LV-reducible to A then B is also solvable in $O(c^n)$ time for all $c > 1$ and (2) if A is solvable in time $O(c^n)$ and if B is CV-reducible to A then B is also solvable in time $O(c^n)$. Thus LV-reductions preserve subexponential complexity while CV-reductions preserve exact complexity. Jonsson et al. [10] exclusively studied the Boolean satisfiability $\mathrm{SAT}(\cdot)$ problem and identified an NP-hard $\mathrm{SAT}(\{R\})$ problem CV-reducible to all other NP-hard $\mathrm{SAT}(\cdot)$ problems. Hence $\mathrm{SAT}(\{R\})$ is, in a sense, the *easiest* NP-complete $\mathrm{SAT}(\cdot)$ problem since if $\mathrm{SAT}(\Gamma)$ can be solved in $O(c^n)$ time, then this holds for $\mathrm{SAT}(\{R\})$, too. With the aid of this result, they analyzed the consequences of subexponentially solvable $\mathrm{SAT}(\cdot)$ problems by utilizing the interplay between CV- and LV-reductions. As a by-product, Santhanam and Srinivasan's [17] negative result on sparsification of infinite constraint languages was shown not to hold for finite languages.

We believe that the existence and construction of such easiest languages forms an important puzzle piece in the quest of relating the complexity of NP-hard problems with each other, since it effectively gives a lower bound on the time complexity of a given problem with respect to constraint language restrictions. As a logical continuation on the work on $\mathrm{SAT}(\cdot)$ we pursue the study of CV- and LV-reducibility in the context of Boolean optimization problems. In particular we investigate the complexity of MAX-ONES$(\cdot)$ and $\mathrm{VCSP}(\cdot)$ and introduce and extend several non-trivial methods for this purpose. The results confirms that methods based on universal algebra are indeed useful when studying broader classes of NP-complete problems. The MAX-ONES$(\cdot)$ problem [11] is a variant of $\mathrm{SAT}(\cdot)$ where the goal is to find a satisfying assignment which maximizes the number of variables assigned the value 1. This problem is closely related to the 0/1 LINEAR PROGRAMMING problem. The $\mathrm{VCSP}(\cdot)$ problem is a function minimization problem that generalizes the MAX-CSP and MIN-CSP problems [11]. We treat both the unweighted and weighted versions of these problems and use the prefix U to denote the unweighted problem and W to denote the weighted version. These problems are well-studied with respect to separating tractable cases from NP-hard cases [11,22] but much less is known when considering the weaker schemes of LV-reductions and CV-reductions. We begin (in Section 3.1) by identifying the easiest language for W-MAX-ONES$(\cdot)$. The proofs make heavy use of the *algebraic method* for constraint satisfaction problems [7,8] and the *weak base method* [20]. The algebraic method was introduced for studying the computational complexity of constraint satsifaction problems up to polynomial-time reductions while the weak base method [19] was shown by Jonsson et al. [10] to be useful for studying CV-reductions. To prove the main result we however need even more powerful reduction techniques based on *weighted primitive positive implementations* [9,21]. For $\mathrm{VCSP}(\cdot)$ the situation differs even more since the algebraic techniques developed for $\mathrm{CSP}(\cdot)$ are not applicable — instead we use *multimorphisms* [2] when considering the complexity of $\mathrm{VCSP}(\cdot)$. We prove (in Section 3.2) that the binary function $f_{\neq}$ which returns 0 if its two arguments are different and 1 otherwise, results in the easiest NP-hard $\mathrm{VCSP}(\cdot)$ problem. This problem is very familiar since it is the MAX CUT problem slightly disguised. The complexity landscape surrounding these problems is outlined in Section 3.3.

With the aid of the languages identified in Section 3, we continue (in Section 4) by relating MAX-ONES and VCSP with LV-reductions and connect them with the ETH. Our results imply that (1) if the ETH is true then no NP-complete U-MAX-ONES(Γ), W-MAX-ONES(Γ), or VCSP(Δ) is solvable in subexponential time and (2) that if the ETH is false then U-MAX-ONES(Γ) and U-VCSP$_d(\Delta)$ are solvable in subexponential time for every choice of Γ and Δ and $d \geq 0$. Here U-VCSP$_d(\Delta)$ is the U-VCSP(Δ) problem restricted to instances where the sum to minimize contains at most dn terms. Thus, to disprove the ETH, our result implies that it is sufficient to find a single language Γ or a set of cost functions Δ such that U-MAX-ONES(Γ), W-MAX-ONES(Γ) or VCSP(Δ) is NP-hard and solvable in subexponential time.

2 Preliminaries

Let Γ denote a finite set of finitary relations over $\mathbb{B} = \{0,1\}$. We call Γ a *constraint language*. Given $R \subseteq \mathbb{B}^k$ we let $\mathrm{ar}(R) = k$ denote its arity, and similarly for functions. When $\Gamma = \{R\}$ we typically omit the set notation and treat R as a constraint language.

2.1 Problem Definitions

The *constraint satisfaction problem* over Γ (CSP(Γ)) is defined as follows.

INSTANCE: A set V of variables and a set C of constraint applications $R(v_1,\ldots,v_k)$ where $R \in \Gamma$, $k = \mathrm{ar}(R)$, and $v_1,\ldots,v_k \in V$.
QUESTION: Is there a function $f : V \to \mathbb{B}$ such that $(f(v_1),\ldots,f(v_k)) \in R$ for each $R(v_1,\ldots,v_k)$ in C?

For the Boolean domain this problem is typically denoted as SAT(Γ). By SAT(Γ)-B we mean the SAT(Γ) problem restricted to instances where each variable can occur in at most B constraints. This restricted problem is occasionally useful since each instance contains at most Bn constraints. The *weigthed maximum ones problem* over Γ (W-MAX-ONES(Γ)) is an optimization version of SAT(Γ) where we for an instance on variables $\{x_1,\ldots,x_n\}$ and weights $w_i \in \mathbb{Q}_{\geq 0}$ want to find a solution h for which $\sum_{i=1}^{n} w_i h(x_i)$ is maximal. The *unweigthed maximum ones problem* (U-MAX-ONES(Γ)) is the W-MAX-ONES(Γ) problem where all weights have the value 1. A *finite-valued cost function* on $\mathbb{B}$ is a function $f : \mathbb{B}^k \to \mathbb{Q}_{\geq 0}$. The *valued constraint satisfaction problem* over a finite set of finite-valued cost functions Δ (VCSP(Δ)) is defined as follows.

INSTANCE: A set $V = \{x_1,\ldots,x_n\}$ of variables and the objective function $f_I(x_1,\ldots,x_n) = \sum_{i=1}^{q} w_i f_i(\mathbf{x}^i)$ where, for every $1 \leq i \leq q$, $f_i \in \Delta, \mathbf{x}^i \in V^{\mathrm{ar}(f_i)}$, and $w_i \in \mathbb{Q}_{\geq 0}$ is a weight.
GOAL: Find a function $h : V \to \mathbb{B}$ such that $f_I(h(x_1),\ldots,h(x_n))$ is minimal.

When the set of cost functions is singleton VCSP($\{f\}$) is written as VCSP(f). We let U-VCSP be the VCSP problem without weights and U-VCSP$_d$ (for $d \geq 0$) denote the U-VCSP problem restricted to instances containing at most $d\,|\mathrm{Var}(I)|$ constraints. Many optimization problems can be viewed as VCSP(Δ) problems for suitable Δ: well-known examples are the MAX-CSP(Γ) and MIN-CSP(Γ) problems where the number of satisfied constraints in a CSP instance are maximized or minimized. For each Γ, there obviously exists sets of cost functions $\Delta_{\min}, \Delta_{\max}$ such that MIN-CSP(Γ) is polynomial-time equivalent to VCSP($\Delta_{\min}$) and MAX-CSP(Γ) is polynomial-time equivalent to VCSP($\Delta_{\max}$). We have defined the problems U-VCSP, VCSP, U-MAX-ONES and W-MAX-ONES as optimization problems, but to obtain a more uniform treatment we often view them as decision problems, i.e. given k we ask if there is a solution with objective value k or better.

2.2 Size-Preserving Reductions and Subexponential Time

If A is a computational problem we let $I(A)$ be the set of problem instances and $\|I\|$ be the size of any $I \in I(A)$, i.e. the number of bits required to represent I. Many problems can in a natural way be viewed as problems of assigning values from a fixed finite set to a collection of variables. This is certainly the case for SAT($\cdot$), MAX-ONES($\cdot$) and VCSP($\cdot$) but it is also the case for various graph problems such as MAX-CUT and MAX INDEPENDENT SET. We call problems of this kind *variable problems* and let $\mathrm{Var}(I)$ denote the set of variables of an instance I.

Definition 1. *Let A_1 and A_2 be variable problems in NP. The function f from $I(A_1)$ to $I(A_2)$ is a* many-one linear variable reduction *(LV-reduction) with parameter $C \geq 0$ if: (1) I is a yes-instance of A_1 if and only if $f(I)$ is a yes-instance of A_2, (2) $|\mathrm{Var}(f(I))| = C \cdot |\mathrm{Var}(I)| + O(1)$, and (3) $f(I)$ can be computed in time $O(\mathrm{poly}(\|I\|))$.*

LV-reductions can be seen as a restricted form of SERF-reductions [6]. The term CV-reduction is used to denote LV-reductions with parameter 1, and we write $A_1 \leq^{\mathrm{CV}} A_2$ to denote that the problem A_1 has an CV-reduction to A_2. If A_1 and A_2 are two NP-hard problems we say that A_1 is *at least as easy* as (or *not harder than*) A_2 if A_1 is solvable in $O(c^{|\mathrm{Var}(I)|})$ time whenever A_1 is solvable in $O(c^{|\mathrm{Var}(I)|})$ time. By definition if $A_1 \leq^{\mathrm{CV}} A_2$ then A_1 is not harder than A_2 but the converse is not true in general. A problem solvable in time $O(2^{c|\mathrm{Var}(I)|})$ for all $c > 0$ is a *subexponential problem*, and SE denotes the class of all variable problems solvable in subexponential time. It is straightforward to prove that LV-reductions preserve subexponential complexity in the sense that if A is LV-reducible to B then $A \in \mathrm{SE}$ if $B \in \mathrm{SE}$. Naturally, SE can be defined using other complexity parameters than $|\mathrm{Var}(I)|$ [6].

2.3 Clone Theory

An operation $f : \mathbb{B}^k \to \mathbb{B}$ is a *polymorphism* of a relation R if for every $\mathbf{t}^1, \ldots, \mathbf{t}^k \in R$ it holds that $f(\mathbf{t}^1, \ldots, \mathbf{t}^k) \in R$, where f is applied element-wise. In this case R is *closed*, or *invariant*, under f. For a set of functions F we define $\mathrm{Inv}(\mathsf{F})$ (often abbreviated as IF) to be the set of all relations invariant under all functions in F. Dually $\mathrm{Pol}(\Gamma)$ for a set of relations Γ is defined to be the set of polymorphisms of Γ. Sets of the form $\mathrm{Pol}(\Gamma)$ are known as *clones* and sets of the form $\mathrm{Inv}(\mathsf{F})$ are known as *co-clones*. The reader unfamiliar with these concepts is referred to the textbook by Lau [13]. The relationship between these structures is made explicit in the following *Galois connection* [13].

Theorem 2. *Let Γ, Γ' be sets of relations. Then $\mathrm{Inv}(\mathrm{Pol}(\Gamma')) \subseteq \mathrm{Inv}(\mathrm{Pol}(\Gamma))$ if and only if $\mathrm{Pol}(\Gamma) \subseteq \mathrm{Pol}(\Gamma')$.*

Co-clones can equivalently be described as sets containing all relations R definable through *primitive positive* (p.p.) implementations over a constraint language Γ, i.e. definitions of the form $R(x_1,\ldots,x_n) \equiv \exists y_1,\ldots,y_m \,.\, R_1(\mathbf{x}^1) \wedge \ldots \wedge R_k(\mathbf{x}^k)$, where each $R_i \in \Gamma \cup \{\mathrm{eq}\}$ and each $\mathbf{x}^i$ is a tuple over $x_1,\ldots,x_n,y_1,\ldots,y_m$ and where $\mathrm{eq} = \{(0,0),(1,1)\}$. As a shorthand we let $\langle\Gamma\rangle = \mathrm{Inv}(\mathrm{Pol}(\Gamma))$ for a constraint language Γ, and as can be verified this is the smallest set of relations closed under p.p. definitions over Γ. In this case Γ is said to be a *base* of $\langle\Gamma\rangle$. It is known that if Γ' is finite and $\mathrm{Pol}(\Gamma) \subseteq \mathrm{Pol}(\Gamma')$ then $\mathrm{CSP}(\Gamma')$ is polynomial-time reducible to $\mathrm{CSP}(\Gamma)$ [7]. With this fact and Post's classification of all Boolean clones [15] Schaefer's dichotomy theorem [18] for $\mathrm{SAT}(\cdot)$ follows almost immediately. See Figure 2 and Table 1 in Appendix A.1 for a visualization of this lattice and a list of bases. The complexity of $\textsc{Max-Ones}(\Gamma)$ is also preserved under finite expansions with relations p.p. definable in Γ, and hence follow the standard Galois connection [11]. Note however that $\mathrm{Pol}(\Gamma') \subseteq \mathrm{Pol}(\Gamma)$ does not imply that $\mathrm{CSP}(\Gamma')$ CV-reduces to $\mathrm{CSP}(\Gamma)$ or even that $\mathrm{CSP}(\Gamma')$ LV-reduces to $\mathrm{CSP}(\Gamma)$ since the number of constraints is not necessarily linearly bounded by the number of variables.

To study these restricted classes of reductions we are therefore in need of Galois connections with increased granularity. In Jonsson et al. [10] the $\mathrm{SAT}(\cdot)$ problem is studied with the Galois connection between closure under p.p. definitions without existential quantification and *strong partial clones*. We concentrate on the relational description and present the full definitions of partial polymorphisms and the aforementioned Galois connection in Appendix A.2. If R is an n-ary Boolean relation and Γ a constraint language then R has a *quantifier-free primitive positive* (q.p.p.) implementation in Γ if $R(x_1,\ldots,x_n) \equiv R_1(\mathbf{x}^1) \wedge \ldots \wedge R_k(\mathbf{x}^k)$, where each $R_i \in \Gamma \cup \{\mathrm{eq}\}$ and each $\mathbf{x}^i$ is a tuple over $x_1,\ldots,x_n$. We use $\langle\Gamma\rangle_{\nexists}$ to denote the smallest set of relations closed under q.p.p. definability over Γ. If $\mathsf{IC} = \langle\mathsf{IC}\rangle_{\nexists}$ then IC is a *weak partial co-clone*. In Jonsson et al. [10] it is proven that if $\Gamma' \subseteq \langle\Gamma\rangle_{\nexists}$ and if Γ and Γ' are both finite constraint languages then $\mathrm{CSP}(\Gamma') \leq^{\mathrm{CV}} \mathrm{CSP}(\Gamma)$. It is not hard to extend this result to the $\textsc{Max-Ones}(\cdot)$ problem since it follows the standard Galois connection, and therefore we use this fact without explicit proof. A *weak base* R_w of a co-clone IC is then a base of IC with the property that for any finite base Γ of IC it holds that $R_w \in \langle\Gamma\rangle_{\nexists}$. In particular this means that $\mathrm{SAT}(R_w)$ and $\textsc{Max-Ones}(R_w)$ CV-reduce to $\mathrm{SAT}(\Gamma)$ and $\textsc{Max-Ones}(\Gamma)$ for any base Γ of IC, and R_w can therefore be seen as the easiest language in the co-clone. The formal definition of a weak base is included in Appendix A.2 together with a table of weak bases for all Boolean co-clones with a finite base. These weak bases have the additional property that they can be implemented without the equality relation [12].

2.4 Operations and Relations

An operation f is called *arithmetical* if $f(y,x,x) = f(y,x,y) = f(x,x,y) = y$ for every $x,y \in \mathbb{B}$. The max function is defined as $\max(x,y) = 0$ if $x = y = 0$ and 1 otherwise. We often express a Boolean relation R as a logical formula whose satisfying assignment corresponds to the tuples of R. F and T are the two constant relations $\{(0)\}$ and $\{(1)\}$ while neq denotes inequality, i.e. the relation $\{(0,1),(1,0)\}$. The relation EVEN^n is defined as $\{(x_1,\ldots,x_n) \in \mathbb{B}^n \mid \sum_{i=1}^{n} x_i \text{ is even}\}$. The relation ODD^n is defined dually. The relations OR^n and NAND^n are the relations corresponding to the clauses $(x_1 \vee \ldots \vee x_n)$

and $(\overline{x_1} \vee \ldots \vee \overline{x_n})$. For any n-ary relation and R we let $R_{m\neq}$, $1 \leq m \leq n$, denote the $(n+m)$-ary relation defined as $R_{m\neq}(x_1,\ldots,x_{n+m}) \equiv R(x_1,\ldots,x_n) \wedge \mathrm{neq}(x_1,x_{n+1}) \wedge \ldots \wedge \mathrm{neq}(x_n,x_{n+m})$. We use $R^{1/3}$ for the relation $\{(0,0,1),(0,1,0),(1,0,0)\}$. Variables are typically named $x_1,\ldots,x_n$ or x except when they occur in positions where they are forced to take a particular value, in which case they are named c_0 and c_1 respectively to explicate that they are in essence constants. As convention c_0 and c_1 always occur in the last positions in the arguments to a predicate. We now see that $R_{\mathsf{II}_2}(x_1,\ldots,x_6,c_0,c_1) \equiv R^{1/3}_{3\neq}(x_1,\ldots,x_6) \wedge \mathrm{F}(c_0) \wedge \mathrm{T}(c_1)$ and $R_{\mathsf{IN}_2}(x_1,\ldots,x_8) \equiv \mathrm{EVEN}^4_{4\neq}(x_1,\ldots,x_8) \wedge (x_1x_4 \leftrightarrow x_2x_3)$ from Table 2 in Appendix A.1 are the two relations (where the tuples in the relations are listed as rows)

$$R_{\mathsf{II}_2} = \left\{ \begin{smallmatrix} 0&0&1&1&1&0&0&1 \\ 0&1&0&1&0&1&0&1 \\ 1&0&0&0&1&1&0&1 \end{smallmatrix} \right\} \quad \text{and} \quad R_{\mathsf{IN}_2} = \left\{ \begin{smallmatrix} 0&0&0&0&1&1&1&1 \\ 0&0&1&1&1&1&0&0 \\ 0&1&0&1&1&0&1&0 \\ 1&1&1&1&0&0&0&0 \\ 1&1&0&0&0&0&1&1 \\ 1&0&1&0&0&1&0&1 \end{smallmatrix} \right\}.$$

3 The Easiest NP-Hard MAX-ONES and VCSP Problems

We will now study the complexity of W-MAX-ONES and VCSP with respect to CV-reductions. We remind the reader that constraint languages Γ and sets of cost functions Δ are always finite. We prove that for both these problems there is a single language which is CV-reducible to every other NP-hard language. Out of the infinite number of candidate languages generating different co-clones, the language $\{R_{\mathsf{II}_2}\}$ defines the easiest W-MAX-ONES$(\cdot)$ problem, which is the same language as for SAT$(\cdot)$ [10]. This might be contrary to intuition since one could be led to believe that the co-clones in the lower parts of the co-clone lattice, generated by very simple languages where the corresponding SAT$(\cdot)$ problem is in P, would result in even easier problems.

3.1 The MAX-ONES Problem

Here we use a slight reformulation of Khanna et al.'s [11] complexity classification of the MAX-ONES problem expressed in terms of polymorphisms.

Theorem 3 ([11]). *Let Γ be a finite Boolean constraint language.* MAX-ONES(Γ) *is in P if and only if Γ is* 1*-closed,* max*-closed, or closed under an arithmetical operation.*

The theorem holds for both the weighted and the unweighted version of the problem and showcases the strength of the algebraic method since it not only eliminates all constraint languages resulting in polynomial-time solvable problems, but also tells us exactly which cases remain, and which properties they satisfy.

Theorem 4. U-MAX-ONES$(R) \leq^{\mathrm{CV}}$ U-MAX-ONES(Γ) *for some* $R \in \{R_{\mathsf{IS}_1^2}, R_{\mathsf{II}_2}, R_{\mathsf{IN}_2}, R_{\mathsf{IL}_0}, R_{\mathsf{IL}_2}, R_{\mathsf{IL}_3}, R_{\mathsf{ID}_2}\}$ *whenever* U-MAX-ONES(Γ) *is NP-hard.*

Proof. By Theorem 3 in combination with Table 1 and Figure 2 in Appendix A.1 it follows that U-MAX-ONES(Γ) is NP-hard if and only if $\langle\Gamma\rangle \supseteq \mathsf{IS}_1^2$ or if $\langle\Gamma\rangle \in$

$\{\mathsf{IL}_0, \mathsf{IL}_3, \mathsf{IL}_2, \mathsf{IN}_2\}$. In principle we then for every co-clone have to decide which language is CV-reducible to every other base of the co-clone, but since a weak base always have this property, we can eliminate a lot of tedious work and directly consult the precomputed relations in Table 2. From this we first see that $\langle R_{\mathsf{IS}_1^2} \rangle_{\nexists} \subset \langle R_{\mathsf{IS}_1^n} \rangle_{\nexists}$, $\langle R_{\mathsf{IS}_{12}^2} \rangle_{\nexists} \subset \langle R_{\mathsf{IS}_{12}^n} \rangle_{\nexists}$, $\langle R_{\mathsf{IS}_{11}^2} \rangle_{\nexists} \subset \langle R_{\mathsf{IS}_{11}^n} \rangle_{\nexists}$ and $\langle R_{\mathsf{IS}_{10}^2} \rangle_{\nexists} \subset \langle R_{\mathsf{IS}_{10}^n} \rangle_{\nexists}$ for every $n \geq 3$. Hence in the four infinite chains IS_1^n, IS_{12}^n, IS_{11}^n, IS_{10}^n we only have to consider the bottommost co-clones IS_1^2, IS_{12}^2, IS_{11}^2, IS_{10}^2. Observe that if R and R' satisfies $R(x_1, \ldots, x_k) \Rightarrow \exists y_0, y_1 . R'(x_1, \ldots, x_k, y_0, y_1) \wedge F(y_0) \wedge T(y_1)$ and $R'(x_1, \ldots, x_k, y_0, y_1) \Rightarrow R(x_1, \ldots, x_k) \wedge F(y_0)$, and it moreover holds that $R'(x_1, \ldots, x_k, y_0, y_1) \in \langle \Gamma \rangle_{\nexists}$, then U-MAX-ONES$(R) \leq^{\mathrm{CV}}$ U-MAX-ONES(Γ), since we can use y_0 and y_1 as global variables and because an optimal solution to the instance we construct will always map y_1 to 1 if the original instance is satisfiable. For $R_{\mathsf{IS}_1^2}(x_1, x_2, c_0)$ we can q.p.p. define predicates $R'_{\mathsf{IS}_1^2}(x_1, x_2, c_0, y_0, y_1)$ with $R_{\mathsf{IS}_{12}^2}, R_{\mathsf{IS}_{11}^2}, R_{\mathsf{IS}_{10}^2}, R_{\mathsf{IE}_2}, R_{\mathsf{IE}_0}$ satisfying these properties as follows:

- $R'_{\mathsf{IS}_1^2}(x_1, x_2, c_0, y_0, y_1) \equiv R_{\mathsf{IS}_{12}^2}(x_1, x_2, c_0, y_1) \wedge R_{\mathsf{IS}_{12}^2}(x_1, x_2, y_0, y_1)$,
- $R'_{\mathsf{IS}_1^2}(x_1, x_2, c_0, y_0, y_1) \equiv R_{\mathsf{IS}_{11}^2}(x_1, x_2, c_0, c_0) \wedge R_{\mathsf{IS}_{11}^2}(x_1, x_2, y_0, y_0)$,
- $R'_{\mathsf{IS}_1^2}(x_1, x_2, c_0, y_0, y_1) \equiv R_{\mathsf{IS}_{10}^2}(x_1, x_2, c_0, c_0, y_1) \wedge R_{\mathsf{IS}_{10}^2}(x_1, x_2, c_0, y_0, y_1)$,
- $R'_{\mathsf{IS}_1^2}(x_1, x_2, c_0, y_0, y_1) \equiv R_{\mathsf{IE}_2}(c_0, x_1, x_2, c_0, y_1) \wedge R_{\mathsf{IE}_2}(c_0, x_1, x_2, y_0, y_1)$,
- $R'_{\mathsf{IS}_1^2}(x_1, x_2, c_0, y_0, y_1) \equiv R_{\mathsf{IE}_0}(c_0, x_1, x_2, y_1, c_0) \wedge R_{\mathsf{IE}_0}(y_0, x_1, x_2, y_1, y_0)$,

and similarly a relation R'_{II_2} using R_{II_0} as follows $R'_{\mathsf{II}_2}(x_1, x_2, x_3, x_4, x_5, x_6, c_0, c_1, y_0, y_1) \equiv R_{\mathsf{II}_0}(x_1, x_2, x_3, c_0) \wedge R_{\mathsf{II}_0}(c_0, c_1, y_1, y_0) \wedge R_{\mathsf{II}_0}(x_1, x_4, y_1, y_0) \wedge R_{\mathsf{II}_0}(x_2, x_5, y_1, y_0) \wedge R_{\mathsf{II}_0}(x_3, x_6, y_1, y_0)$. By Figure 2 in Appendix A.1 we then see that the only remaining cases for Γ when $\langle \Gamma \rangle \supset \mathsf{IS}_1^2$ is when $\langle \Gamma \rangle = \mathsf{II}_2$ or when $\langle \Gamma \rangle = \mathsf{ID}_2$. This concludes the proof. □

Using q.p.p. implementations to further decrease the set of relations in Theorem 4 appears difficult and we therefore make use of more powerful implementations. Let Optsol(I) be the set of all optimal solutions of a W-MAX-ONES(Γ) instance I. A relation R has a *weighted p.p. definition* (w.p.p. definition) [9,21] in Γ if there exists an instance I of W-MAX-ONES(Γ) on variables V such that $R = \{(\phi(v_1), \ldots, \phi(v_m)) \mid \phi \in \mathrm{Optsol}(I)\}$ for some $v_1, \ldots, v_m \in V$. The set of all relations w.p.p. definable in Γ is denoted $\langle \Gamma \rangle_w$ and we furthermore have that if $\Gamma' \subseteq \langle \Gamma \rangle_w$ is a finite then W-MAX-ONES(Γ') is polynomial-time reducible to W-MAX-ONES(Γ) [9,21]. If there is a W-MAX-ONES(Γ) instance I on V such that $R = \{(\phi(v_1), \ldots, \phi(v_m)) \mid \phi \in \mathrm{Optsol}(I)\}$ for $v_1, \ldots, v_m \in V$ satisfying $\{v_1, \ldots, v_m\} = V$, then we say that R is q.w.p.p. definable in Γ. We use $\langle \Gamma \rangle_{\nexists, w}$ for set of all relations q.w.p.p. definable in Γ. It is not hard to check that if $\Gamma' \subseteq \langle \Gamma \rangle_{\nexists, w}$, then every instance is mapped to an instance of equally many variables — hence W-MAX-ONES(Γ') is CV-reducible to W-MAX-ONES(Γ) whenever Γ' is finite.

Theorem 5. *Let Γ be a constraint language such that* W-MAX-ONES(Γ) *is NP-hard. Then it holds that* W-MAX-ONES$(R_{\mathsf{II}_2}) \leq^{\mathrm{CV}}$ W-MAX-ONES(Γ).

Proof. We utilize q.w.p.p. definitions and note that the following holds.

$R_{\mathsf{II}_2} = \arg\max_{\mathbf{x}\in\mathbb{B}^8:(\mathbf{x}_7,\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_6,\mathbf{x}_8,\mathbf{x}_4,\mathbf{x}_5,\mathbf{x}_3)\in R_{\mathsf{IN}_2}} \mathbf{x}_8$,
$R_{\mathsf{II}_2} = \arg\max_{\mathbf{x}\in\mathbb{B}^8:(\mathbf{x}_5,\mathbf{x}_4,\mathbf{x}_2,\mathbf{x}_1,\mathbf{x}_7,\mathbf{x}_8),(\mathbf{x}_6,\mathbf{x}_4,\mathbf{x}_3,\mathbf{x}_1,\mathbf{x}_7,\mathbf{x}_8),(\mathbf{x}_6,\mathbf{x}_5,\mathbf{x}_3,\mathbf{x}_4,\mathbf{x}_7,\mathbf{x}_8)\in R_{\mathsf{ID}_2}} (\mathbf{x}_1+\mathbf{x}_2+\mathbf{x}_3)$,
$R_{\mathsf{II}_2} = \arg\max_{\mathbf{x}\in\mathbb{B}^8:(\mathbf{x}_4,\mathbf{x}_5,\mathbf{x}_6,\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\mathbf{x}_7,\mathbf{x}_8)\in R_{\mathsf{IL}_2}} (\mathbf{x}_4+\mathbf{x}_5+\mathbf{x}_6)$,
$R_{\mathsf{IL}_2} = \arg\max_{\mathbf{x}\in\mathbb{B}^8:(\mathbf{x}_7,\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\mathbf{x}_8,\mathbf{x}_4,\mathbf{x}_5,\mathbf{x}_6)\in R_{\mathsf{IL}_3}} \mathbf{x}_8$,
$R_{\mathsf{IL}_2} = \arg\max_{\mathbf{x}\in\mathbb{B}^8:(\mathbf{x}_4,\mathbf{x}_5,\mathbf{x}_6,\mathbf{x}_7),(\mathbf{x}_8,\mathbf{x}_1,\mathbf{x}_4,\mathbf{x}_7),(\mathbf{x}_8,\mathbf{x}_2,\mathbf{x}_5,\mathbf{x}_7),(\mathbf{x}_8,\mathbf{x}_3,\mathbf{x}_6,\mathbf{x}_7)\in R_{\mathsf{IL}_0}} \mathbf{x}_8$,
$R_{\mathsf{II}_2} = \arg\max_{\mathbf{x}\in\mathbb{B}^8:(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_7),(\mathbf{x}_1,\mathbf{x}_3,\mathbf{x}_7),(\mathbf{x}_2,\mathbf{x}_3,\mathbf{x}_7),(\mathbf{x}_1,\mathbf{x}_4,\mathbf{x}_7),(\mathbf{x}_2,\mathbf{x}_5,\mathbf{x}_7),(\mathbf{x}_3,\mathbf{x}_6,\mathbf{x}_7)\in R_{\mathsf{IS}_1^2}} (\mathbf{x}_1+\cdots+\mathbf{x}_8)$.

Hence, $R_{\mathsf{II}_2} \in \langle R\rangle_{\nexists,w}$ for every $R \in \{R_{\mathsf{IS}_1^2}, R_{\mathsf{IN}_2}, R_{\mathsf{IL}_0}, R_{\mathsf{IL}_2}, R_{\mathsf{IL}_3}, R_{\mathsf{ID}_2}\}$ which by Theorem 4 completes the proof. □

3.2 The VCSP Problem

Since VCSP does not adhere to the standard Galois connection in Theorem 2, the weak base method is not applicable and alternative methods are required. For this purpose we use *multimorphisms* from Cohen et al. [2]. Let Δ be a set of cost functions on $\mathbb{B}$, let p be a unary operation on $\mathbb{B}$, and let f,g be binary operations on $\mathbb{B}$. We say that Δ admits the binary *multimorphism* (f,g) if it holds that $\nu(f(x,y)) + \nu(g(x,y)) \leq \nu(x) + \nu(y)$ for every $\nu \in \Delta$ and $x,y \in \mathbb{B}^{\mathrm{ar}(\nu)}$. Similarly Δ admits the unary *multimorphism* (p) if it holds that $\nu(p(x)) \leq \nu(x)$ for every $\nu \in \Delta$ and $x \in \mathbb{B}^{\mathrm{ar}(\nu)}$. Recall that the function $f_{\neq}$ equals $\{(0,0)\mapsto 1,(0,1)\mapsto 0,(1,0)\mapsto 0,(1,1)\mapsto 1\}$ and that the minimisation problem $\mathrm{VCSP}(f_{\neq})$ and the maximisation problem MAX CUT are trivially CV-reducible to each other. We will make use of (a variant of) the concept of *expressibility* [2]. We say that a cost function g is *$\nexists$-expressible* in Δ if $g(x_1,\ldots,x_n) = \sum_i w_i f_i(\mathbf{s}^i) + w$ for some tuples $\mathbf{s}^i$ over $\{x_1,\ldots,x_n\}$, weights $w_i \in \mathbb{Q}_{\geq 0}$, $w \in \mathbb{Q}$ and $f_i \in \Delta$. It is not hard to see that if every function in a finite set Δ' is $\nexists$-expressible in Δ, then $\mathrm{VCSP}(\Delta') \leq^{\mathrm{CV}} \mathrm{VCSP}(\Delta)$. Note that if the constants 0 and 1 are expressible in Δ then we may allow tuples $\mathbf{s}^i$ over $\{x_1,\ldots,x_n,0,1\}$, and still obtain a CV-reduction.

Theorem 6. *Let Δ be a set of finite-valued cost functions on $\mathbb{B}$. If the problem $\mathrm{VCSP}(\Delta)$ is NP-hard, then $\mathrm{VCSP}(f_{\neq}) \leq^{\mathrm{CV}} \mathrm{VCSP}(\Delta)$.*

Proof. Since $\mathrm{VCSP}(\Delta)$ is NP-hard (and since we assume $\mathrm{P} \neq \mathrm{NP}$) we know that Δ does not admit the unary (0)-multimorphism or the unary (1)-multimorphism [2]. Therefore there are $g,h \in \Delta$ and $\mathbf{u} \in \mathbb{B}^{\mathrm{ar}(g)}$, $\mathbf{v} \in \mathbb{B}^{\mathrm{ar}(h)}$ such that $g(\mathbf{0}) > g(\mathbf{u})$ and $h(\mathbf{1}) > h(\mathbf{v})$. Let $\mathbf{w} \in \arg\min_{\mathbf{x}\in\mathbb{B}^b}(g(\mathbf{x}_1,\ldots,\mathbf{x}_a) + h(\mathbf{x}_{a+1},\ldots,\mathbf{x}_b))$ and then define $o(x,y) = g(z_1,\ldots,z_a) + h(z_{a+1},\ldots,z_b)$ where $z_i = x$ if $\mathbf{w}_i = 0$ and $z_i = y$ otherwise. Clearly $(0,1) \in \arg\min_{\mathbf{x}\in\mathbb{B}^2} o(\mathbf{x})$, $o(0,1) < o(0,0)$, and $o(0,1) < o(1,1)$. We will show that we always can force two fresh variables v_0 and v_1 to 0 and 1, respectively. If $o(0,0) \neq o(1,1)$, then assume without loss of generality that $o(0,0) < o(1,1)$. In this case we force v_0 to 0 with the (sufficiently weighted) term $o(v_0,v_0)$. Define $g'(x) = g(z_1,\ldots,z_{\mathrm{ar}(g)})$ where $z_i = x$ if $u_i = 1$ and $z_i = v_0$ otherwise. Note that $g'(1) < g'(0)$ which means that we can force v_1 to 1. Otherwise $o(0,0) = o(1,1)$. If $o(0,1) = o(1,0)$, then $f_{\neq} = \alpha_1 o + \alpha_2$, otherwise assume without loss of generality that $o(0,1) < o(1,0)$. In this case v_0, v_1 can be forced to $0,1$ with the help of the (sufficiently weighted) term $o(v_0,v_1)$.

We also know that Δ does not admit the $(\min,\max)$-multimorphism [2] since $\mathrm{VCSP}(\Delta)$ is NP-hard by assumption. Hence, there exists a k-ary function $f \in \Delta$ and $\mathbf{s},\mathbf{t} \in \mathbb{B}^k$ such that $f(\min(\mathbf{s},\mathbf{t})) + f(\max(\mathbf{s},\mathbf{t})) > f(\mathbf{s}) + f(\mathbf{t})$. Let $f_1(x) = \alpha_1 o(v_0,x) + \alpha_2$ for some $\alpha_1 \in \mathbb{Q}_{\geq 0}$ and $\alpha_2 \in \mathbb{Q}$ such that $f_1(1) = 0$ and $f_1(0) = 1$. Let also $g(x,y) = f(z_1,\ldots,z_k)$ where $z_i = v_1$ if $\min(\mathbf{s}_i,\mathbf{t}_i) = 1$, $z_i = v_0$ if $\max(\mathbf{s}_i,\mathbf{t}_i) = 0$, $z_i = x$ if $s_i > t_i$ and $z_i = y$ otherwise. Note that $g(0,0) = f(\min(\mathbf{s},\mathbf{t}))$, $g(1,1) = f(\max(\mathbf{s},\mathbf{t}))$, $g(1,0) = f(\mathbf{s})$ and $g(0,1) = f(\mathbf{t})$. Set $h(x,y) = g(x,y) + g(y,x)$. Now $h(0,1) = h(1,0) < \frac{1}{2}(h(0,0) + h(1,1))$. If $h(0,0) = h(1,1)$, then $f_{\neq} = \alpha_1 h + \alpha_2$ for some $\alpha_1 \in \mathbb{Q}_{\geq 0}$ and $\alpha_2 \in \mathbb{Q}$. Hence, we can without loss of generality assume that $h(1,1) - h(0,0) = 2$. Note now that $h'(x,y) = f_1(x) + f_1(y) + h(x,y)$ satisfies $h'(0,0) = h'(1,1) = \frac{1}{2}(h(0,0) + h(1,1) + 2)$ and $h'(0,1) = h'(1,0) = \frac{1}{2}(2 + h(0,1) + h(1,0))$. Hence, $h'(0,0) = h'(1,1) > h'(0,1) = h'(1,0)$. So $f_{\neq} = \alpha_1 h' + \alpha_2$ for some $\alpha_1 \in \mathbb{Q}_{\geq 0}$ and $\alpha_2 \in \mathbb{Q}$. $\square$

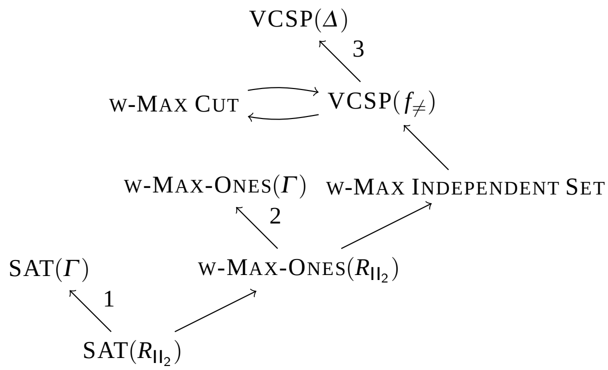
3.3 The Broader Picture

Theorems 5 and 6 does not describe the relative complexity between the $\mathrm{SAT}(\cdot)$, MAX-ONES$(\cdot)$ and $\mathrm{VCSP}(\cdot)$ problems. However we readily see (1) that $\mathrm{SAT}(R_{\mathrm{II}_2}) \leq^{\mathrm{CV}}$ W-MAX-ONES(R_{II_2}), and (2) that W-MAX-ONES$(R_{\mathrm{II}_2}) \leq^{\mathrm{CV}}$ W-MAX INDEPENDENT SET since W-MAX INDEPENDENT SET can be expressed by W-MAX-ONES(NAND^2). The problem W-MAX-ONES(NAND^2) is in turn expressible by MAX-CSP$(\{\mathrm{NAND}^2, \mathrm{T}, \mathrm{F}\})$. To show that W-MAX INDEPENDENT SET $\leq^{\mathrm{CV}}$ $\mathrm{VCSP}(f_{\neq})$ it is in fact, since MAX-CSP(neq) and $\mathrm{VCSP}(f_{\neq})$ is the same problem, sufficient to show that MAX-CSP$(\{\mathrm{NAND}^2,\mathrm{T},\mathrm{F}\}) \leq^{\mathrm{CV}}$ MAX-CSP(neq). We do this as follows. Let v_0 and v_1 be two global variables. We force v_0 and v_1 to be mapped to different values by assigning a sufficiently high weight to the constraint $\mathrm{neq}(v_0,v_1)$. It then follows that $\mathrm{T}(x) = \mathrm{neq}(x,v_0)$, $\mathrm{F}(x) = \mathrm{neq}(x,v_1)$ and $\mathrm{NAND}^2(x,y) = \frac{1}{2}(\mathrm{neq}(x,y) + \mathrm{F}(x) + \mathrm{F}(y))$ and we are done. It follows from this proof that MAX-CSP$(\{\mathrm{NAND}^2,\mathrm{T},\mathrm{F}\})$ and $\mathrm{VCSP}(f_{\neq})$ are mutually CV-interreducible. Since MAX-CSP$(\{\mathrm{NAND}^2,\mathrm{T},\mathrm{F}\})$ can also be formulated as a VCSP it follows that $\mathrm{VCSP}(\cdot)$ does not have a unique easiest set of cost functions. The complexity results are summarized in Figure 1. Some trivial inclusions are omitted in the figure: for example it holds that $\mathrm{SAT}(\Gamma) \leq^{\mathrm{CV}}$ W-MAX-ONES(Γ) for all Γ.

4 Subexponential Time and the Exponential-Time Hypothesis

The exponential-time hypothesis states that 3-SAT $\notin$ SE [5]. We remind the reader that the ETH can be based on different size parameters (such as the number of variables or the number of clauses) and that these different definitions often coincide [6]. In this section we investigate the consequences of the ETH for the U-MAX-ONES and U-VCSP problems. A direct consequence of Section 3 is that if there exists any finite constraint language Γ or set of cost functions Δ such that W-MAX-ONES(Γ) or $\mathrm{VCSP}(\Delta)$ is NP-hard and in SE, then $\mathrm{SAT}(R_{\mathrm{II}_2})$ is in SE which implies that the ETH is false [10]. The other direction is interesting too since it highlights the likelihood of subexponential time algorithms for the problems, relative to the ETH.

Lemma 7. *If* U-MAX-ONES(Γ) *is in* SE *for some finite constraint languages* Γ *such that* U-MAX-ONES(Γ) *is NP-hard, then the ETH is false.*



1. Holds for every Γ such that SAT(Γ) is NP-hard.
2. Holds for every Γ such that W-MAX-ONES(Γ) is NP-hard.
3. Holds for every finite-valued Δ such that VCSP(Δ) is NP-hard.

Fig. 1. The complexity landscape of some Boolean optimization and satisfiability problems. A directed arrow from one node A to B means that $A \leq^{\mathrm{CV}} B$.

Proof. From Jonsson et al. [10] it follows that 3-SAT is in SE if and only if $\mathrm{SAT}(R_{\mathsf{II}_2})$-2 is in SE. Combining this with Theorem 4 we only have to prove that $\mathrm{SAT}(R_{\mathsf{II}_2})$-2 LV-reduces to U-MAX-ONES(R) for $R \in \{R_{\mathsf{IS}_1^2}, R_{\mathsf{IN}_2}, R_{\mathsf{IL}_0}, R_{\mathsf{IL}_2}, R_{\mathsf{IL}_3}, R_{\mathsf{ID}_2}\}$. We provide an illustrative reduction from $\mathrm{SAT}(R_{\mathsf{II}_2})$-2 to U-MAX-ONES$(R_{\mathsf{IS}_1^2})$; the remaining reductions are presented in Lemmas 11–15 in Appendix A.3. Since $R_{\mathsf{IS}_1^2}$ is the NAND relation with one additional constant column, the U-MAX-ONES$(R_{\mathsf{IS}_1^2})$ problem is basically the maximum independent set problem or, equivalently, the maximum clique problem in the complement graph. Given an instance I of $\mathrm{CSP}(R_{\mathsf{II}_2})$-2 we create for every constraint 3 vertices, one corresponding to each feasible assignment of values to the variables occurring in the constraint. We add edges between all pairs of vertices that are not inconsistent and that do not correspond to the same constraint. The instance I is satisfied if and only if there is a clique of size m where m is the number of constraints in I. Since $m \leq 2n$ this implies that the number of vertices is $\leq 2n$. $\square$

Theorem 8. *The following statements are equivalent.*

1. *The exponential-time hypothesis is false.*
2. U-MAX-ONES$(\Gamma) \in \mathrm{SE}$ *for every finite* Γ.
3. U-MAX-ONES$(\Gamma) \in \mathrm{SE}$ *for some finite* Γ *such that* U-MAX-ONES(Γ) *is NP-hard.*
4. U-VCSP$(\Delta)_d \in \mathrm{SE}$ *for every finite set of finite-valued cost functions* Δ *and* $d \geq 0$.

Proof. The implication $1 \Rightarrow 2$ follows from Lemma 16 in Appendix A.3, $2 \Rightarrow 3$ is trivial, and $3 \Rightarrow 1$ follows by Lemma 7. The implication $2 \Rightarrow 4$ follows from Lemma 17 in Appendix A.3. We finish the proof by showing $4 \Rightarrow 1$. Let $I = (V, C)$ be an instance of $\mathrm{SAT}(R_{\mathsf{II}_2})$-2. Note that I contains at most $2|V|$ constraints. Let f be the function defined by $f(\mathbf{x}) = 0$ if $\mathbf{x} \in R_{\mathsf{II}_2}$ and $f(\mathbf{x}) = 1$ otherwise. Create an instance of U-VCSP$_2(f)$ by, for every constraint $C_i = R_{\mathsf{II}_2}(x_1, \ldots, x_8) \in C$, adding to the cost function the term $f(x_1, \ldots, x_8)$. This instance has a solution with objective value 0 if and only if I is satisfiable. Hence, $\mathrm{SAT}(R_{\mathsf{II}_2})$-2 $\in \mathrm{SE}$ which contradicts the ETH [10]. $\square$

5 Future Research

Other problems. The weak base method naturally lends itself to other problems parameterized by constraint languages. In general, one has to consider all co-clones where the problem is NP-hard, take the weak bases for these co-clones and find out which of these are CV-reducible to the other cases. The last step is typically the most challenging — this was demonstrated by the U-MAX-ONES problems where we had to introduce q.w.p.p. implementations. An example of an interesting problem where this strategy works is the *non-trivial* SAT problem ($\mathrm{SAT}^*(\Gamma)$), i.e. the problem of deciding whether a given instance has a solution in which not all variables are mapped to the same value. This problem is NP-hard in exactly six cases [3] and by following the aforementioned procedure one can prove that the relation R_{II_2} results in the easiest NP-hard $\mathrm{SAT}^*(\Gamma)$ problem. Since $\mathrm{SAT}^*(R_{\mathrm{II}_2})$ is in fact the same problem as $\mathrm{SAT}(R_{\mathrm{II}_2})$ this shows that restricting solutions to non-trivial solutions does not make the satisfiability problem easier. This result can also be extended to the co-NP-hard *implication problem* [3] and we believe that similar methods can also be applied to give new insights into the complexity of e.g. *enumeration*, which also follows the same complexity classification [3]. Such results would naturally give us insights into the structure of NP but also into the applicability of clone-based methods.

Weighted versus unweighted problems. Theorem 8 only applies to unweighted problems and lifting these results to the weighted case does not appear straightforward. We believe that some of these obstacles could be overcome with generalized sparsification techniques. We provide an example by proving that if any NP-hard $\mathrm{W\text{-}MAX\text{-}ONES}(\Gamma)$ problem is in SE, then MAX-CUT can be approximated within a multiplicative error of $(1 \pm \varepsilon)$ (for any $\varepsilon > 0$) in subexponential time. Assume that $\mathrm{W\text{-}MAX\text{-}ONES}(\Gamma)$ is NP-hard and a member of SE, and arbitrarily choose $\varepsilon > 0$. Let $\mathrm{MAX\text{-}CUT}_c$ be the MAX-CUT problem restricted to graphs $G = (V,E)$ where $|E| \leq c \cdot |V|$. We first prove that $\mathrm{MAX\text{-}CUT}_c$ is in SE for arbitrary $c \geq 0$. By Theorem 5, we infer that $\mathrm{W\text{-}MAX\text{-}ONES}(R_{\mathrm{II}_2})$ is in SE. Given an instance (V,E) of $\mathrm{MAX\text{-}CUT}_c$, one can introduce one fresh variable x_v for each $v \in V$ and one fresh variable x_e for each edge $e \in E$. For each edge $e = (v,w)$, we then constrain the variables x_v, x_w and x_e as $R(x_v, x_w, x_e)$ where $R = \{(0,0,0),(0,1,1),(1,0,1),(1,1,0)\} \in \langle R_{\mathrm{II}_2} \rangle$. It can then be verified that, for an optimal solution h, that the maximum value of $\sum_{e \in E} w_e h(x_e)$ (where w_e is the weight associated with the edge e) equals the weight of a maximum cut in (V,E). This is an LV-reduction since $|E| = c \cdot |V|$. Now consider an instance (V,E) of the unrestricted MAX-CUT problem. By Batson et al. [1], we can (in polynomial time) compute a *cut sparsifier* (V',E') with only $D_\varepsilon \cdot n/\varepsilon^2$ edges (where D_ε is a constant depending only on ε), which approximately preserves the value of the maximum cut of (V,E) to within a multiplicative error of $(1 \pm \varepsilon)$. By using the LV-reduction above from $\mathrm{MAX\text{-}CUT}_{D_\varepsilon/\varepsilon^2}$ to $\mathrm{W\text{-}MAX\text{-}ONES}(\Gamma)$, it follows that we can approximate the maximum cut of (V,E) within $(1 \pm \varepsilon)$ in subexponential time.

References

1. J. Batson, D. A. Spielman, and N. Srivastava. Twice-ramanujan sparsifiers. *SIAM Journal on Computing*, 41(6):1704 – 1721, 2012.

2. D. A. Cohen, M. C. Cooper, P. G. Jeavons, and A. A. Krokhin. The complexity of soft constraint satisfaction. *Artificial Intelligence*, 170(11):983–1016, 2006.
3. N. Creignou and J. J. Hébrard. On generating all solutions of generalized satisfiability problems. *Informatique Théorique et Applications*, 31(6):499–511, 1997.
4. T. Feder and M. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.
5. R. Impagliazzo and R. Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367 – 375, 2001.
6. R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512 – 530, 2001.
7. P. Jeavons. On the algebraic structure of combinatorial problems. *Theoretical Computer Science*, 200:185–204, 1998.
8. P. Jeavons, D. Cohen, and M. Gyssens. Closure properties of constraints. *Journal of the ACM*, 44(4):527–548, 1997.
9. P. Jonsson, F. Kuivinen, and J. Thapper. Min CSP on four elements: Moving beyond submodularity. In *Proceedings of the Principles and Practice of Constraint Programming*, CP 2011, pages 438–453. Springer Berlin Heidelberg, 2011.
10. P. Jonsson, V. Lagerkvist, G. Nordh, and B. Zanuttini. Complexity of SAT problems, clone theory and the exponential time hypothesis. In Sanjeev Khanna, editor, *SODA 2013*, pages 1264–1277. SIAM, 2013.
11. S. Khanna, M. Sudan, L. Trevisan, and D. Williamson. The approximability of constraint satisfaction problems. *SIAM Journal on Computing*, 30(6):1863–1920, 2000.
12. V. Lagerkvist. Weak bases of Boolean co-clones. *Information Processing Letters*, 114(9):462–468, 2014.
13. D. Lau. *Function Algebras on Finite Sets: Basic Course on Many-Valued Logic and Clone Theory (Springer Monographs in Mathematics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
14. D. Lokshtanov, D. Marx, and S. Saurabh. Lower bounds based on the exponential time hypothesis. *Bulletin of the EATCS*, 105:41–72, 2011.
15. E. Post. The two-valued iterative systems of mathematical logic. *Annals of Mathematical Studies*, 5:1–122, 1941.
16. B. A. Romov. The algebras of partial functions and their invariants. *Cybernetics*, 17(2):157–167, 1981.
17. R. Santhanam and S. Srinivasan. On the limits of sparsification. In *Proceeding of the 39th International Colloquium on Automata, Languages, and Programming (ICALP-2012)*, pages 774–785, 2012.
18. T. J. Schaefer. The complexity of satisfiability problems. In *Proceedings 10th Symposium on Theory of Computing*, STOC 1978, pages 216–226. ACM Press, 1978.
19. H. Schnoor and I. Schnoor. Partial polymorphisms and constraint satisfaction problems. In N. Creignou, P. G. Kolaitis, and H. Vollmer, editors, *Complexity of Constraints*, volume 5250 of *Lecture Notes in Computer Science*, pages 229–254. Springer Berlin Heidelberg, 2008.
20. I. Schnoor. *The weak base method for constraint satisfaction*. PhD thesis, Gottfried Wilhelm Leibniz Universität, Hannover, Germany, 2008.
21. J. Thapper. *Aspects of a Constraint Optimisation Problem*. PhD thesis, Linköping University, The Institute of Technology, 2010.
22. J. Thapper and S. Živný. The complexity of finite-valued CSPs. In *Proceedings of the 45th Annual ACM Symposium on Symposium on Theory of Computing*, STOC 2013, pages 695–704, New York, NY, USA, 2013. ACM.

23. G. Woeginger. Exact algorithms for NP-hard problems: a survey. In M. Juenger, G. Reinelt, and G. Rinaldi, editors, *Combinatorial Optimization – Eureka! You Shrink!*, pages 185–207, 2000.

A Appendix

A.1 Bases of Boolean Clones and the Clone Lattice

In Table 1 we present a full table of bases for all Boolean clones. These were first introduced by Post [15] and the lattice is hence known as *Post's lattice*. It is visualized in Figure 2.

Table 1. List of all Boolean clones with definitions and bases, where $\mathrm{id}(x) = x$ and $h_n(x_1,\dots,x_{n+1}) = \bigvee_{i=1}^{n+1} x_1 \cdots x_{i-1}x_{i+1} \cdots x_{n+1}$, $\mathrm{dual}(f)(a_1,\dots,a_n) = 1 - f(\overline{a_1},\dots,\overline{a_n})$.

Clone	Definition	Base
BF	All Boolean functions	$\{x \wedge y, \neg x\}$
R_0	$\{f \mid f$ is 0-reproducing$\}$	$\{x \wedge y, x \oplus y\}$
R_1	$\{f \mid f$ is 1-reproducing$\}$	$\{x \vee y, x \oplus y \oplus 1\}$
R_2	$\mathsf{R}_0 \cap \mathsf{R}_1$	$\{x \vee y, x \wedge (y \oplus z \oplus 1)\}$
M	$\{f \mid f$ is monotonic$\}$	$\{x \vee y, x \wedge y, 0, 1\}$
M_1	$\mathsf{M} \cap \mathsf{R}_1$	$\{x \vee y, x \wedge y, 1\}$
M_0	$\mathsf{M} \cap \mathsf{R}_0$	$\{x \vee y, x \wedge y, 0\}$
M_2	$\mathsf{M} \cap \mathsf{R}_2$	$\{x \vee y, x \wedge y\}$
S_0^n	$\{f \mid f$ is 0-separating of degree $n\}$	$\{x \rightarrow y, \mathrm{dual}(h_n)\}$
S_0	$\{f \mid f$ is 0-separating$\}$	$\{x \rightarrow y\}$
S_1^n	$\{f \mid f$ is 1-separating of degree $n\}$	$\{x \wedge \neg y, h_n\}$
S_1	$\{f \mid f$ is 1-separating$\}$	$\{x \wedge \neg y\}$
S_{02}^n	$\mathsf{S}_0^n \cap \mathsf{R}_2$	$\{x \vee (y \wedge \neg z), \mathrm{dual}(h_n)\}$
S_{02}	$\mathsf{S}_0 \cap \mathsf{R}_2$	$\{x \vee (y \wedge \neg z)\}$
S_{01}^n	$\mathsf{S}_0^n \cap \mathsf{M}$	$\{\mathrm{dual}(h_n), 1\}$
S_{01}	$\mathsf{S}_0 \cap \mathsf{M}$	$\{x \vee (y \wedge z), 1\}$
S_{00}^n	$\mathsf{S}_0^n \cap \mathsf{R}_2 \cap \mathsf{M}$	$\{x \vee (y \wedge z), \mathrm{dual}(h_n)\}$
S_{00}	$\mathsf{S}_0 \cap \mathsf{R}_2 \cap \mathsf{M}$	$\{x \vee (y \wedge z)\}$
S_{12}^n	$\mathsf{S}_1^n \cap \mathsf{R}_2$	$\{x \wedge (y \vee \neg z), h_n\}$
S_{12}	$\mathsf{S}_1 \cap \mathsf{R}_2$	$\{x \wedge (y \vee \neg z)\}$
S_{11}^n	$\mathsf{S}_1^n \cap \mathsf{M}$	$\{h_n, 0\}$
S_{11}	$\mathsf{S}_1 \cap \mathsf{M}$	$\{x \wedge (y \vee z), 0\}$
S_{10}^n	$\mathsf{S}_1^n \cap \mathsf{R}_2 \cap \mathsf{M}$	$\{x \wedge (y \vee z), h_n\}$
S_{10}	$\mathsf{S}_1 \cap \mathsf{R}_2 \cap \mathsf{M}$	$\{x \wedge (y \vee z)\}$
D	$\{f \mid f$ is self-dual$\}$	$\{(x \wedge \neg y) \vee (x \wedge \neg z) \vee (\neg y \wedge \neg z)\}$
D_1	$\mathsf{D} \cap \mathsf{R}_2$	$\{(x \wedge y) \vee (x \wedge \neg z) \vee (y \wedge \neg z)\}$
D_2	$\mathsf{D} \cap \mathsf{M}$	$\{h_2\}$
L	$\{f \mid f$ is affine$\}$	$\{x \oplus y, 1\}$
L_0	$\mathsf{L} \cap \mathsf{R}_0$	$\{x \oplus y\}$
L_1	$\mathsf{L} \cap \mathsf{R}_1$	$\{x \oplus y \oplus 1\}$
L_2	$\mathsf{L} \cap \mathsf{R}_2$	$\{x \oplus y \oplus z\}$
L_3	$\mathsf{L} \cap \mathsf{D}$	$\{x \oplus y \oplus z \oplus 1\}$
V	$\{f \mid f$ is a disjunction or constants$\}$	$\{x \vee y, 0, 1\}$
V_0	$\mathsf{V} \cap \mathsf{R}_0$	$\{x \vee y, 0\}$
V_1	$\mathsf{V} \cap \mathsf{R}_1$	$\{x \vee y, 1\}$
V_2	$\mathsf{V} \cap \mathsf{R}_2$	$\{x \vee y\}$
E	$\{f \mid f$ is a conjunction or constants$\}$	$\{x \wedge y, 0, 1\}$
E_0	$\mathsf{E} \cap \mathsf{R}_0$	$\{x \wedge y, 0\}$
E_1	$\mathsf{E} \cap \mathsf{R}_1$	$\{x \wedge y, 1\}$
E_2	$\mathsf{E} \cap \mathsf{R}_2$	$\{x \wedge y\}$
N	$\{f \mid f$ depends on at most one variable$\}$	$\{\neg x, 0, 1\}$
N_2	$\mathsf{N} \cap \mathsf{R}_2$	$\{\neg x\}$
I	$\{f \mid f$ is a projection or a constant$\}$	$\{\mathrm{id}, 0, 1\}$
I_0	$\mathsf{I} \cap \mathsf{R}_0$	$\{\mathrm{id}, 0\}$
I_1	$\mathsf{I} \cap \mathsf{R}_1$	$\{\mathrm{id}, 1\}$
I_2	$\mathsf{I} \cap \mathsf{R}_2$	$\{\mathrm{id}\}$

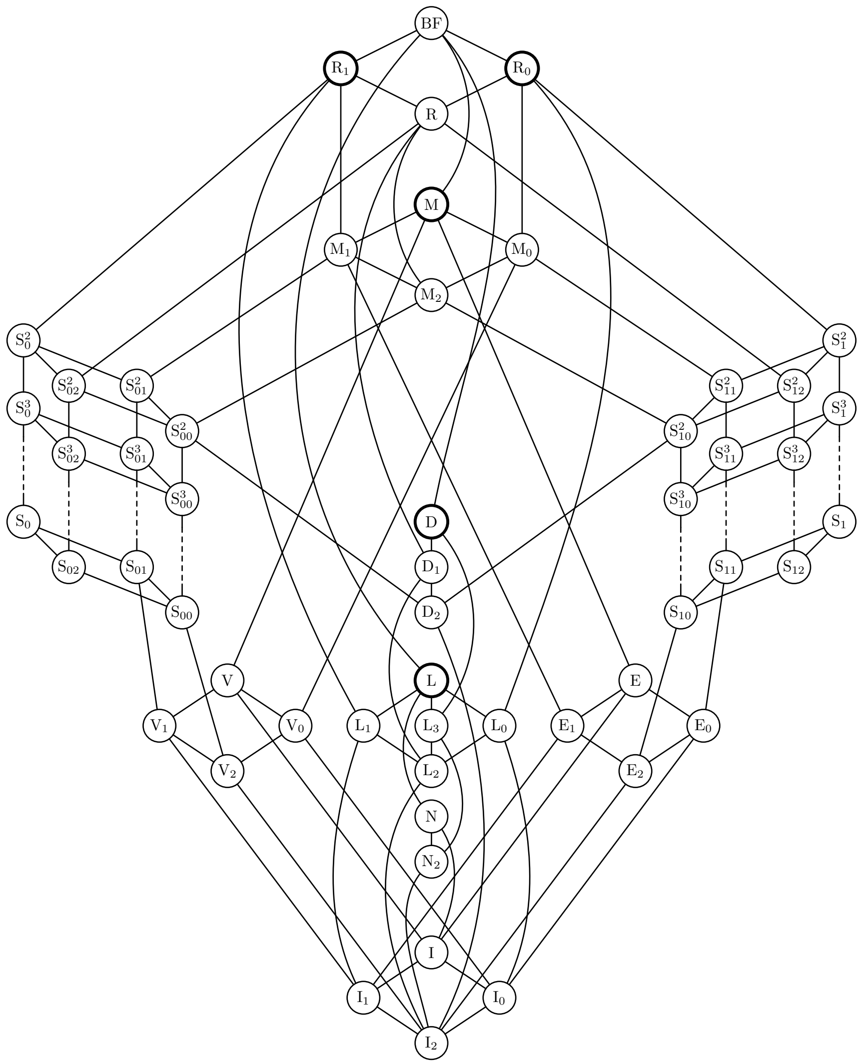


Fig. 2. The lattice of Boolean clones.

A.2 Weak Bases

We extend the definition of a polymorphism and say that a partial function f is a *partial polymorphism* to a relation R if R is closed under f for every sequence of tuples for which f is defined. A set of partial functions F is said to be a *strong partial clone* if it contains all (total and partial) projection functions and is closed under composition of functions. By $\mathrm{pPol}(\Gamma)$ we denote the set of partial polymorphisms to the set of relations Γ. Obviously sets of the form $\mathrm{pPol}(\Gamma)$ always form strong partial clones and again we have a Galois connection between clones and co-clones.

Theorem 9. *[16] Let Γ and Γ' be two sets of relations. Then $\langle\Gamma\rangle_{\nexists} \subseteq \langle\Gamma'\rangle_{\nexists}$ if and only if* $\mathrm{pPol}(\Gamma') \subseteq \mathrm{pPol}(\Gamma)$.

We define the *weak base* of a co-clone IC to be the base of the smallest member of the interval $\mathscr{I}(\mathsf{IC}) = \{\mathsf{ID} \mid \mathsf{ID} = \langle\mathsf{ID}\rangle_{\nexists} \text{ and } \langle\mathsf{ID}\rangle = \mathsf{IC}\}$. Weak bases were first introduced in Schnoor and Schnoor [19,20] but their construction resulted in relations that were in many cases exponentially larger than the plain bases with respect to arity. Weak bases fulfilling additional minimality conditions was given in Lagerkvist [12] using relational descriptions. By construction the weak base of a co-clone is always a single relation.

Theorem 10 ([19]). *Let R_w be the weak base of some co-clone IC. Then for any finite base Γ of IC it holds that $R_w \in \langle\Gamma\rangle_{\nexists}$.*

See Table 2 for a complete list of weak bases.

A.3 Additional Proofs for Section 4

Lemma 11. $\mathrm{SAT}(R_{\mathsf{II}_2})$*-2 LV-reduces to* U-MAX-ONES(R_{IL_2}).

Proof. We reduce an instance I of $\mathrm{SAT}(R_{\mathsf{II}_2})$-2 on n variables constraints to an instance of U-MAX-ONES(R_{IL_2}) containing at most $2+8n$ variables. Let v_0, v_1 be two fresh global variables constrained as $R_{\mathsf{IL}_2}(v_0,v_0,v_0,v_1,v_1,v_1,v_0,v_1)$. Note that this forces v_0 to 0 and v_1 to 1 in any satisfying assignment. Now, for every variable x in the SAT-instance we create an additional variable x' which we constrain as $R_{\mathsf{IL}_2}(x',x,v_1,x,x',v_0,v_0,v_1)$. This correctly implements $\mathrm{neq}(x,x')$. For the i-th constraint, $R_{\mathsf{II}_2}(x_1,\dots,x_6,c_0,c_1)$, in I we create three variables z_i^1, z_i^2, z_i^3 and constrain them as $R_{\mathsf{IL}_2}(z_i^1,z_i^2,z_i^3,x_1,x_2,x_3,c_0,c_1)$, we also add the constraint $R_{\mathsf{IL}_2}(x_4,x_5,x_6,x_1,x_2,x_3,c_0,c_1)$. Since every variable in the SAT-instance I can occur in at most two constraints we have that $m \leq 2n$. Hence the resulting U-MAX-ONES instance contains at most $2+2n+3\cdot 2n = 2+8n$ variables. Since x and x', and v_0 and v_1, must take different values it holds that the measure of a solution of this new instance is exactly the number of variables z_i^j that are mapped to 1. Hence, for an optimal solution the objective value is $\geq 2m$ if and only if I is satisfiable. □

Lemma 12. U-MAX-ONES(R_{IL_2}) *LV-reduces to* U-MAX-ONES(R_{IL_0}).

Table 2. Weak bases for all Boolean co-clones with a finite base

Co-clone	Weak base
IBF	$\mathrm{Eq}(x_1,x_2)$
IR_0	$\mathrm{F}(c_0)$
IR_1	$\mathrm{T}(c_1)$
IR_2	$\mathrm{F}(c_0) \wedge \mathrm{T}(c_1)$
IM	$(x_1 \rightarrow x_2)$
IM_0	$(x_1 \rightarrow x_2) \wedge \mathrm{F}(c_0)$
IM_1	$(x_1 \rightarrow x_2) \wedge \mathrm{T}(c_1)$
IM_2	$(x_1 \rightarrow x_2) \wedge \mathrm{F}(c_0) \wedge \mathrm{T}(c_1)$
$\mathsf{IS}_0^{\mathsf{n}}, n \geq 2$	$\mathrm{OR}^n(x_1,\ldots,x_n) \wedge \mathrm{T}(c_1)$
$\mathsf{IS}_{02}^{\mathsf{n}}, n \geq 2$	$\mathrm{OR}^n(x_1,\ldots,x_n) \wedge \mathrm{F}(c_0) \wedge \mathrm{T}(c_1)$
$\mathsf{IS}_{01}^{\mathsf{n}}, n \geq 2$	$\mathrm{OR}^n(x_1,\ldots,x_n) \wedge (x \rightarrow x_1 \cdots x_n) \wedge \mathrm{T}(c_1)$
$\mathsf{IS}_{00}^{\mathsf{n}}, n \geq 2$	$\mathrm{OR}^n(x_1,\ldots,x_n) \wedge (x \rightarrow x_1 \cdots x_n) \wedge \mathrm{F}(c_0) \wedge \mathrm{T}(c_1)$
$\mathsf{IS}_1^{\mathsf{n}}, n \geq 2$	$\mathrm{NAND}^n(x_1,\ldots,x_n) \wedge \mathrm{F}(c_0)$
$\mathsf{IS}_{12}^{\mathsf{n}}, n \geq 2$	$\mathrm{NAND}^n(x_1,\ldots,x_n) \wedge \mathrm{F}(c_0) \wedge \mathrm{T}(c_1)$
$\mathsf{IS}_{11}^{\mathsf{n}}, n \geq 2$	$\mathrm{NAND}^n(x_1,\ldots,x_n) \wedge (x \rightarrow x_1 \cdots x_n) \wedge \mathrm{F}(c_0)$
$\mathsf{IS}_{10}^{\mathsf{n}}, n \geq 2$	$\mathrm{NAND}^n(x_1,\ldots,x_n) \wedge (x \rightarrow x_1 \cdots x_n) \wedge \mathrm{F}(c_0) \wedge \mathrm{T}(c_1)$
ID	$(x_1 \neq x_2)$
ID_1	$(x_1 \neq x_2) \wedge \mathrm{F}(c_0) \wedge \mathrm{T}(c_1)$
ID_2	$\mathrm{OR}^2_{2\neq}(x_1,x_2,x_3,x_4) \wedge \mathrm{F}(c_0) \wedge \mathrm{T}(c_1)$
IL	$\mathrm{EVEN}^4(x_1,x_2,x_3,x_4)$
IL_0	$\mathrm{EVEN}^3(x_1,x_2,x_3) \wedge \mathrm{F}(c_0)$
IL_1	$\mathrm{ODD}^3(x_1,x_2,x_3) \wedge \mathrm{T}(c_1)$
IL_2	$\mathrm{EVEN}^3_{3\neq}(x_1,\ldots,x_6) \wedge \mathrm{F}(c_0) \wedge \mathrm{T}(c_1)$
IL_3	$\mathrm{EVEN}^4_{4\neq}(x_1,\ldots,x_8)$
IV	$(\overline{x_1} \leftrightarrow \overline{x_2}\,\overline{x_3}) \wedge (\overline{x_2} \vee \overline{x_3} \rightarrow \overline{x_4})$
IV_0	$(\overline{x_1} \leftrightarrow \overline{x_2}\,\overline{x_3}) \wedge \mathrm{F}(c_0)$
IV_1	$(\overline{x_1} \leftrightarrow \overline{x_2}\,\overline{x_3}) \wedge (\overline{x_2} \vee \overline{x_3} \rightarrow \overline{x_4}) \wedge \mathrm{T}(c_1)$
IV_2	$(\overline{x_1} \leftrightarrow \overline{x_2}\,\overline{x_3}) \wedge \mathrm{F}(c_0) \wedge \mathrm{T}(c_1)$
IE	$(x_1 \leftrightarrow x_2x_3) \wedge (x_2 \vee x_3 \rightarrow x_4)$
IE_0	$(x_1 \leftrightarrow x_2x_3) \wedge (x_2 \vee x_3 \rightarrow x_4) \wedge \mathrm{F}(c_0)$
IE_1	$(x_1 \leftrightarrow x_2x_3) \wedge \mathrm{T}(c_1)$
IE_2	$(x_1 \leftrightarrow x_2x_3) \wedge \mathrm{F}(c_0) \wedge \mathrm{T}(c_1)$
IN	$\mathrm{EVEN}^4(x_1,x_2,x_3,x_4) \wedge x_1x_4 \leftrightarrow x_2x_3$
IN_2	$\mathrm{EVEN}^4_{4\neq}(x_1,\ldots,x_8) \wedge x_1x_4 \leftrightarrow x_2x_3$
II	$(x_1 \leftrightarrow x_2x_3) \wedge (\overline{x_4} \leftrightarrow \overline{x_2}\,\overline{x_3})$
II_0	$(\overline{x_1} \vee \overline{x_2}) \wedge (\overline{x_1}\,\overline{x_2} \leftrightarrow \overline{x_3}) \wedge \mathrm{F}(c_0)$
II_1	$(x_1 \vee x_2) \wedge (x_1x_2 \leftrightarrow x_3) \wedge \mathrm{T}(c_1)$
II_2	$R^{1/3}_{3\neq}(x_1,\ldots,x_6) \wedge \mathrm{F}(c_0) \wedge \mathrm{T}(c_1)$

Proof. We reduce an instance I of U-MAX-ONES(R_{IL_2}) on n variables to an instance of U-MAX-ONES(R_{IL_0}) on $2+2n$ variables. Let $v_0, v_1, y_1, \ldots, y_n$ be fresh variables and constrain them as $R_{\mathsf{IL}_0}(v_0,v_0,v_0,v_0) \wedge R_{\mathsf{IL}_0}(v_1,v_0,y_1,v_0) \wedge \ldots \wedge R_{\mathsf{IL}_0}(v_1,v_0,y_n,v_0)$. Note that this forces v_0 to 0, and that if v_1 is mapped to 0, then so are the variables $y_1,\ldots,y_n$. If v_1 is mapped to 1 on the other hand, then $y_1,\ldots,y_n$ can be mapped to 1. For every constraint $R_{\mathsf{IL}_2}(x_1,x_2,x_3,x_4,x_5,x_6,c_0,c_1)$ we create the constraints $R_{\mathsf{IL}_0}(x_1,x_2,x_3,v_0) \wedge R_{\mathsf{IL}_0}(v_1,x_1,x_4,v_0) \wedge R_{\mathsf{IL}_0}(v_1,x_2,x_5,v_0) \wedge R_{\mathsf{IL}_0}(v_1,x_3,x_6,v_0) \wedge R_{\mathsf{IL}_0}(v_1,c_0,c_1,v_0)$. The resulting U-MAX-ONES(R_{IL_0}) instance has $2+2n$ variables and has a solution with measure $n+1+k$ if and only if I has a solution with measure k. $\square$

Lemma 13. U-MAX-ONES(R_{II_2}) *LV-reduces to* U-MAX-ONES(R_{IN_2}).

Proof. We reduce an instance I of U-MAX-ONES(R_{II_2}) over n variables to an instance of U-MAX-ONES(R_{IN_2}) over $2+3n$ variables. Create two fresh variables v_0, v_1 and constrain them as $R_{\mathsf{IN}_2}(v_0,v_0,v_0,v_0,v_1,v_1,v_1,v_1)$ in order to force v_0 and v_1 to be mapped to different values. We then create the $2n$ variables $y_1,\ldots,y_{2n}$ and constrain them as $\bigwedge_{i=1}^{2n} R_{\mathsf{IN}_2}(v_0,v_0,v_0,v_0,y_i,y_i,y_i,y_i)$. This forces all of the variables y_i to be mapped to the same value as v_1. We can now express $R_{\mathsf{II}_2}(x_1,x_2,x_3,x_4,x_5,x_6,c_0,c_1)$ using the implementation $R_{\mathsf{IN}_2}(v_0,x_1,x_2,x_6,v_1,x_4,x_5,x_3) \wedge R_{\mathsf{IN}_2}(v_0,c_0,c_0,v_0,v_1,c_1,c_1,v_1)$. Note that in any optimal solution of the new instance v_1 will be mapped to 1 which means that the implementation of R_{II_2} given above will be correct. The resulting instance has a solution with measure $1+2n+k$ if and only if I has a solution with measure k. $\square$

Lemma 14. U-MAX-ONES($R_{\mathsf{IS}_1^2}$) *LV-reduces to* U-MAX-ONES(R_{ID_2}).

Proof. We reduce an instance of U-MAX-ONES($R_{\mathsf{IS}_1^2}$) on n variables to an instance of U-MAX-ONES(R_{ID_2}) on $2+3n$ variables. Create two new variables v_0 and v_1 and constrain them as $R_{\mathsf{ID}_2}(v_1,v_1,v_0,v_0,v_0,v_1)$. Note that this forces v_0 to 0 and v_1 to 1. For every variable x we introduce two extra variables x' and x'' and constrain them as $R_{\mathsf{ID}_2}(x,x',x',x,v_0,v_1) \wedge R_{\mathsf{ID}_2}(x',x'',x'',x',v_0,v_1)$. Note that this implements the constraints $\mathrm{neq}(x,x')$ and $\mathrm{neq}(x',x'')$, and that no matter what x is mapped to exactly one of x' and x'' is mapped to 1. For every constraint $R_{\mathsf{IS}_1^2}(x,y,c_0)$ we then introduce the constraint $R_{\mathsf{ID}_2}(x',y',x,y,c_0,v_1)$. The resulting instance has a solution with measure $1+n+k$ if and only if I has a solution with measure k. $\square$

Lemma 15. U-MAX-ONES(R_{IL_2}) *LV-reduces to* U-MAX-ONES(R_{IL_3}).

Proof. We reduce an instance of U-MAX-ONES(R_{IL_2}) on n variables to an instance of U-MAX-ONES(R_{IL_3}) on $2+3n$ variables. Create two new variables v_0 and v_1 and constrain them as $R_{IL_3}(v_0,v_0,v_0,v_0,v_1,v_1,v_1,v_1)$. Note that this forces v_0 and v_1 to be mapped to different values. We then introduce fresh variables $y_1,\ldots,y_{2n}$ and constrain them as $\bigwedge_{i=1}^{2n} R_{IL_3}(v_0,v_0,v_0,v_0,y_i,y_i,y_i,y_i)$. This will ensure that every variables y_i is mapped to the same value as v_1 and therefore that in every optimal solution v_0 is mapped to 0 and v_1 is mapped to 1. For every constraint $R_{IL_2}(x_1,\ldots,x_6,c_0,c_1)$ we introduce the constraints $R_{IL_3}(c_0,x_1,x_2,x_3,c_1,x_4,x_5,x_6) \wedge R_{IL_3}(c_0,c_0,c_0,c_0,v_1,v_1,v_1,v_1) \wedge R_{IL_3}(v_0,v_0,v_0,v_0,c_1,c_1,c_1,c_1)$. The resulting instance has a solution with measure $1+2n+k$ if and only if I has a solution with measure k. $\square$

Lemma 16. *If the ETH is false, then* u-Max-Ones$(\Gamma) \in$ SE *for every finite Boolean constraint language* Γ.

Proof. Define SNP to be the class of properties expressible by formulas of the type $\exists S_1 \ldots \exists S_n \forall x_1 \ldots \forall x_m . F$ where F is a quantifier-free logical formula, $\exists S_1 \ldots \exists S_n$ are second order existential quantifiers, and $\forall x_1 \ldots \forall x_m$ are first-order universal quantifiers. Monadic SNP (MSNP) is the restriction of SNP where all second-order predicates are required to be unary [4]. The associated search problem tries to identify instantiations of $S_1, \ldots, S_n$ that make the resulting first-order formula true. We will be interested in properties that can be expressed by formulas that additionally contain *size-constrained* existential quantifier. A size-constrained existential quantifier is of the form $\exists S$, $|S| \oplus s$, where $|S|$ is the number of inputs where relation S holds, and $\oplus \in \{=, \leq, \geq\}$. Define size-constrained SNP as the class of properties of relations and numbers that are expressible by formulas $\exists S_1 \ldots \exists S_n \forall x_1 \ldots \forall x_m . F$ where the existential quantifiers are allowed to be size-constrained.

If the ETH is false then 3-SAT is solvable in subexponential time. By Impagliazzo et al. [6] this problem is size-constrained MSNP-complete under size-preserving SERF reductions. Hence we only have to prove that u-Max-Ones$(\cdot)$ is included in size-constrained MSNP for it to be solvable in subexponential time. Impagliazzo et al. [6] shows that k-SAT is in SNP by providing an explicit formula $\exists S . F$ where F is a universal formula and S a unary predicate interpreted such that $x \in S$ if and only if x is true. Let k be the highest arity of any relation in Γ. Since k-SAT can q.p.p. implement any k-ary relation it is therefore sufficient to prove that u-Max-Ones(Γ^k_{SAT}) is in size-constrained MSNP, where Γ^k_{SAT} is the language corresponding to all satisfying assignments of k-SAT. This is easy to do with the formula

$$\exists S, |S| \geq K . F$$

where K is the parameter corresponding to the number of variables that has to be assigned 1. $\square$

Lemma 17. *If* u-Max-Ones$(\Gamma) \in$ SE *for every finite Boolean constraint language* Γ*, then* u-VCSP$_d(\Delta) \in$ SE *for every finite set of Boolean cost functions* Δ *and arbitrary* $d \geq 0$.

Proof. We first show that if every u-Max-Ones$(\Gamma) \in$ SE, then the minimization variant u-Min-Ones$(\Gamma) \in$ SE for all Γ, too. Arbitrarily choose a finite constraint language Γ over $\mathbb{B}$. We present an LV-reduction from u-Min-Ones(Γ) to u-Max-Ones$(\Gamma \cup \{\text{neq}\})$. Let $(\{v_1, \ldots, v_n\}, C)$ be an arbitrary instance of u-Min-Ones(Γ) with optimal value K. Consider the following instance I' of u-Max-Ones$(\Gamma \cup \{\text{neq}\})$:

$$(\{v_1, v'_1, v''_1, \ldots, v_n, v'_n, v''_n\}, C \cup \{\text{neq}(v_1, v'_1), \text{neq}(v_1, v''_1), \ldots, \text{neq}(v_n, v'_n), \text{neq}(v_n, v''_n\}).$$

For each variable $v_i \in \{v_1, \ldots, v_n\}$ that is assigned 0, the corresponding variables v'_i, v''_i are assigned 1, and vice-versa. It follows that the optimal value of I' is $2n - K$. Hence, u-Min-Ones$(\Gamma) \in$ SE since u-Max-Ones$(\Gamma \cup \{\text{neq}\}) \in$ SE.

Now, arbitrarily choose $d \geq 0$ and a finite set of Boolean cost functions Δ. Since Δ is finite, we may without loss of generality assume that each function $f \in \Delta$ has its range in $\{0,1,2,\ldots\}$.

We show that U-VCSP$_d(\Delta) \in$ SE by exhibiting an LV-reduction from U-VCSP$_d(\Delta)$ to U-MIN-ONES(Γ) where Γ is finite and only depends on Δ. Given a tuple $\mathbf{a} = (a_1,\ldots,a_k) \in \mathbb{B}^k$, let $\mathrm{val}(\mathbf{a}) = 1 + \sum_{j:a_j=1} 2^{j-1}$. For each $f \in \Delta$ of arity k, define

$$R_f = \left\{ (x_1,\ldots,x_k,y_1,\ldots,y_{2^k}) \in \mathbb{B}^{k+2^k} \;\middle|\; \begin{array}{l} f(x_1,\ldots,x_k) > 0, \\ \{i : y_i \neq 0\} = \{\mathrm{val}(x_1,\ldots,x_k)\} \end{array} \right\}$$
$$\cup \{(x_1,\ldots,x_k,0,\ldots,0) \in \mathbb{B}^{k+2^k} \mid f(x_1,\ldots,x_k) = 0\},$$

and let $\Gamma = \{\mathrm{eq},\mathrm{neq}\} \cup \{R_f \mid f \in \Delta\}$.

One may interpret R_f as follows: for each $(x_1,\ldots,x_k) \in \mathbb{B}^k$ the relation R_f contains exactly one tuple $(x_1,\ldots,x_k,y_1,\ldots,y_{2^k})$. If $f(x_1,\ldots,x_k) = 0$, then this is the tuple $(x_1,\ldots,x_k,0,\ldots,0)$. If $f(x_1,\ldots,x_k) > 0$, then this is the tuple $(x_1,\ldots,x_k,0,\ldots,1,\ldots,0)$ where the 1 is in position $k + \mathrm{val}(x_1,\ldots,x_k)$. We show below how R_f can be used for "translating" each $\mathbf{x} \in \mathbb{B}^k$ into its corresponding weight as prescribed by f.

Let $(V, \sum_{i=1}^m f_i(\mathbf{x}_i))$ be an arbitrary instance of U-VCSP$_d(\Delta)$ where $V = \{v_1,\ldots,v_n\}$. Let $\mathrm{ar}(f_i)$ denote the arity of function f_i. Assume the instance has an optimal solution with value K. For each term $f_i(v_1,\ldots,v_k)$ in the sum, do the following:

1. introduce 2^k fresh variables $v'_1,\ldots,v'_{2^k}$,
2. introduce k fresh variables $w_1,\ldots,w_k$,
3. for each $\mathbf{a} \in \mathbb{B}^k$ such that $f(\mathbf{a}) > 1$, introduce $n' = f(\mathbf{a})$ fresh variables $u_0,\ldots,u_{n'-1}$,
4. introduce the constraint $R_f(v_1,\ldots,v_k,v'_1,\ldots,v'_{2^k})$,
5. introduce the constraints $\mathrm{neq}(v_1,w_1),\ldots,\mathrm{neq}(v_k,w_k)$, and
6. for each $\mathbf{a} \in \mathbb{B}^k$, let $n' = f(\mathbf{a})$ and do the following if $n' > 1$: let $p = \mathrm{val}(\mathbf{a})$ and introduce the constraints $\mathrm{eq}(v'_p,u_0),\mathrm{eq}(u_0,u_1),\ldots,\mathrm{eq}(u_{n'-2},u_{n'-1})$.

It is not difficult to realize that the resulting instance has optimal value $K + \sum_{i=1}^m \mathrm{ar}(C_i)$ given the interpretation of R_f and the following motivation of step 5: the neq constraints introduced in step 5 ensure that the weight of $(x_1,\ldots,x_k)$ does not influence the weight of the construction and this explains that we need to adjust the optimal value with $\sum_{i=1}^m \mathrm{ar}(C_i)$.

Furthermore, the instance contains at most

$$|V| + |C| \cdot (2s + t \cdot (2^s + 1))$$

variables where $s = \max\{\mathrm{ar}(f) \mid f \in \Delta\}$ and $t = \max\{f(\mathbf{a}) \mid f \in \Delta \text{ and } \mathbf{a} \in \mathbb{B}^{\mathrm{ar}(f)}\}$. By noting that $|C| \leq d|V|$ and that s,t are constants that only depend on Δ, it follows that the reduction is an LV-reduction. $\square$