The complexity of Boolean surjective general-valued CSPs

PETER FULLA, University of Oxford, UK
HANNES UPPMAN, Linköping University, Sweden
STANISLAV ŽIVNÝ, University of Oxford, UK

Valued constraint satisfaction problems (VCSPs) are discrete optimisation problems with a $(\mathbb{Q} \cup \{\infty\})$-valued objective function given as a sum of fixed-arity functions. In Boolean surjective VCSPs, variables take on labels from $D = \{0, 1\}$ and an optimal assignment is required to use both labels from D. Examples include the classical *global Min-Cut* problem in graphs and the *Minimum Distance* problem studied in coding theory.

We establish a dichotomy theorem and thus give a complete complexity classification of Boolean surjective VCSPs with respect to exact solvability. Our work generalises the dichotomy for $\{0, \infty\}$-valued constraint languages (corresponding to surjective decision CSPs) obtained by Creignou and Hébrard. For the maximisation problem of $\mathbb{Q}_{\geq 0}$-valued surjective VCSPs, we also establish a dichotomy theorem with respect to approximability.

Unlike in the case of Boolean surjective (decision) CSPs, there appears a novel tractable class of languages that is trivial in the non-surjective setting. This newly discovered tractable class has an interesting mathematical structure related to downsets and upsets. Our main contribution is identifying this class and proving that it lies on the borderline of tractability. A crucial part of our proof is a polynomial-time algorithm for enumerating all near-optimal solutions to a generalised Min-Cut problem, which might be of independent interest.

CCS Concepts: • **Theory of computation → Problems, reductions and completeness**;

Additional Key Words and Phrases: constraint satisfaction problems, surjective CSP, valued CSP, min-cut, polymorphisms, multimorphisms

ACM Reference Format:
Peter Fulla, Hannes Uppman, and Stanislav Živný. 2018. The complexity of Boolean surjective general-valued CSPs. *ACM Trans. Comput. Theory* 1, 1, Article 1 (January 2018), 31 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

The framework of valued constraint satisfaction problems (VCSPs) captures many fundamental discrete optimisation problems. A VCSP *instance* $I = (V, D, \phi_I)$ is given by a finite set of variables $V = \{x_1, \ldots, x_n\}$, a finite set of labels D called the *domain*, and an objective function $\phi_I : D^n \to \overline{\mathbb{Q}}$, where $\overline{\mathbb{Q}} = \mathbb{Q} \cup \{\infty\}$ denotes the set of extended rationals. The objective function is expressed by a weighted sum of *valued constraints*

$$\phi_I(x_1, \ldots, x_n) = \sum_{i=1}^{q} w_i \cdot \gamma_i(\mathbf{x}_i)\,, \tag{1}$$

where $\gamma_i : D^{\mathrm{ar}(\gamma_i)} \to \overline{\mathbb{Q}}$ is a *weighted relation* of *arity* $\mathrm{ar}(\gamma_i) \in \mathbb{Z}_{\geq 1}$, $w_i \in \mathbb{Q}_{\geq 0}$ is the *weight* and $\mathbf{x}_i \in V^{\mathrm{ar}(\gamma_i)}$ the *scope* of the ith valued constraint. (Note that zero weights are allowed; we

Authors' addresses: Peter Fulla, University of Oxford, Oxford, UK, peter.fulla@cs.ox.ac.uk; Hannes Uppman, Linköping University, Linköping, Sweden; Stanislav Živný, University of Oxford, Oxford, UK, standa.zivny@cs.ox.ac.uk.

1942-3454/2018/1-ART1 $15.00
https://doi.org/0000001.0000001

define $0 \cdot \infty = \infty$.) The value of an assignment of domain labels to variables $s : V \to D$ equals $\phi_I(s) = \phi_I(s(x_1), \ldots, s(x_n))$. An assignment s is *feasible* if $\phi_I(s) < \infty$, and it is *optimal* if it is feasible and $\phi_I(s) \leq \phi_I(s')$ for all assignments s'. Given an instance I, the goal is to find an optimal assignment, i.e. one that minimises ϕ_I. A *valued constraint language* (or just a language) Γ is set of weighted relations over a domain D. We denote by VCSP(Γ) the class of all VCSP instances that use only weighted relations from a language Γ in their objective function. VCSPs are also called general-valued CSPs [31] to emphasise the fact that (decision) CSPs are a special case of VCSPs in which weighted relations only assign values 0 and ∞. (However, $\mathbb{Q}$-valued VCSPs [42] do *not* include CSPs as a special case.)

For an example of a VCSP, consider the (s,t)-Min-Cut problem [40]. Given a digraph $G = (V, E)$ with a source $s \in V$, sink $t \in V$, and edge weights $w : E \to \mathbb{Q}_{>0}$, the goal is to find a set $C \subseteq V$ with $s \in C$ and $t \notin C$ that minimises

$$\sum_{(u,v) \in E,\, u \in C,\, v \notin C} w(u,v)\,. \tag{2}$$

We show how the (s,t)-Min-Cut problem can be expressed as a VCSP over a domain $D = \{0,1\}$ (a domain of size 2 such as this one is called *Boolean*). We define a language $\Gamma_{\text{cut}} = \{\rho_0, \rho_1, \gamma\}$ as follows: For $d \in D$, $\rho_d : D \to \overline{\mathbb{Q}}$ is defined by $\rho_d(x) = 0$ if $x = d$ and $\rho_d(x) = \infty$ if $x \neq d$. Weighted relation $\gamma : D^2 \to \overline{\mathbb{Q}}$ is defined by $\gamma(x,y) = 1$ if $x = 0$ and $y = 1$, and $\gamma(x,y) = 0$ otherwise. Given an (s,t)-Min-Cut instance on a digraph $G = (V,E)$, the problem of finding an optimal (s,t)-Min-Cut in G is equivalent to solving an instance $I = (V, D, \phi_I)$ of VCSP(Γ_{cut}) such that

$$\phi_I(x_1, \ldots, x_n) = \rho_0(s) + \rho_1(t) + \sum_{(u,v) \in E} w(u,v) \cdot \gamma(u,v)\,. \tag{3}$$

It is well known that the (s,t)-Min-Cut problem is solvable in polynomial time. Since every instance I of VCSP(Γ_{cut}) can be reduced to an instance of the (s,t)-Min-Cut problem, VCSP(Γ_{cut}) is solvable in polynomial time.

A language Γ is called *tractable* if, for every finite $\Gamma' \subseteq \Gamma$, VCSP(Γ') is solvable in polynomial time. If there exists a finite $\Gamma' \subseteq \Gamma$ such that VCSP(Γ') is NP-hard, then Γ is called *intractable*.[1] For example, language Γ_{cut} is tractable. It is natural to ask about the complexity of VCSP(Γ) for a fixed language Γ. Cohen et al. [12] obtained a dichotomy classification of *Boolean* languages: They identified eight tractable classes (one of which correspons to submodularity [40] and includes Γ_{cut}) and showed that the remaining languages are intractable. The dichotomy classification from [12] is an extension of Schaefer's celebrated result [39], which gave a dichotomy for Boolean $\{0,\infty\}$-valued constraint languages, and the work of Creignou [13], which established a dichotomy classification for Boolean $\{0,1\}$-valued constraint languages.

The *surjective* variant of VCSPs further requires that assignments of domain labels to variables be surjective (an assignment $s : V \to D$ is surjective if, for every $d \in D$, there exists $x \in V$ such that $s(x) = d$). Thus, the goal is to find an assignment that is optimal among surjective assignments. For Boolean VCSPs with $D = \{0,1\}$, this simply means that the all-zero and all-one assignments are disregarded. We define VCSP(Γ), tractability, and intractability in the surjective setting analogously with regular VCSPs, and refer to them as VCSP$_{\text{s}}$(Γ), s-tractability, and s-intractability.

For an example of a surjective VCSP, consider the (global) Min-Cut problem [40]. Given a graph $G = (V,E)$ and edge weights $w : E \to \mathbb{Q}_{>0}$, the goal is to find a set $C \subseteq V$ with $\emptyset \subsetneq C \subsetneq V$ that

[1] Defining tractability in terms of finite subsets ensures that the tractability of a language is independent of whether the weighted relations are represented explicitly (by tables of values) or implicitly (by oracles).

minimises

$$\sum_{\{u,v\}\in E, |\{u,v\}\cap C|=1} w(u,v)\,. \tag{4}$$

Again, this problem can be expressed over a Boolean domain $D = \{0, 1\}$. We define a weighted relation $\gamma : D^2 \to \overline{\mathbb{Q}}$ by $\gamma(x, y) = 0$ if $x = y$ and $\gamma(x, y) = 1$ if $x \neq y$. Then the problem of finding an optimal Min-Cut in a graph $G = (V, E)$ is equivalent to solving an instance $I = (V, D, \phi_I)$ of $\text{VCSP}_s(\{\gamma\})$ such that

$$\phi_I(x_1, \ldots, x_n) = \sum_{\{u,v\}\in E} w(u,v) \cdot \gamma(u,v)\,. \tag{5}$$

Note that the two non-surjective assignments to I correspond to sets $\emptyset$ and V, which are not admissible solutions to the Min-Cut problem. Since every instance of $\text{VCSP}_s(\{\gamma\})$ can be straightforwardly translated to a Min-Cut instance, and the Min-Cut problem is solvable in polynomial time (say, by a reduction to the (s, t)-Min-Cut problem, though other algorithms exist [41]), language $\{\gamma\}$ is s-tractable.

The computational complexity of $\text{VCSP}(\Gamma)$ and $\text{VCSP}_s(\Gamma)$ is closely related. Namely, $\text{VCSP}(\Gamma)$ is polynomial-time reducible to $\text{VCSP}_s(\Gamma)$ (see Lemma 2.5), i.e., any intractable language is also s-intractable. Let $C_D = \{\rho_d \mid d \in D\}$, where we define $\rho_d : D \to \overline{\mathbb{Q}}$ by $\rho_d(x) = 0$ if $x = d$ and $\rho_d(x) = \infty$ if $x \neq d$; these unary weighted relations are called *constants*. Conversely, $\text{VCSP}_s(\Gamma)$ is polynomial-time reducible to $\text{VCSP}(\Gamma \cup C_D)$ (see Lemma 2.6), i.e., any tractable language containing constants C_D is also s-tractable. In the case of Boolean $\{0, \infty\}$-valued languages, Schaefer's dichotomy involves six tractable classes. Four of them include constants, and hence they are s-tractable. Creignou and Hébrard [14] showed that the remaining two classes (0-valid and 1-valid[2]) are s-intractable, thus obtaining a dichotomy classification of Boolean $\{0, \infty\}$-valued languages in the surjective setting.

Contributions

Complexity classification. As our main contribution, we establish a dichotomy classification of all Boolean ($\overline{\mathbb{Q}}$-valued) languages in the surjective setting, which extends the classification from [14]. Let $D = \{0, 1\}$. Six of the eight tractable classes of Boolean languages identified by Cohen et al. [12] include constants C_D, and thus are also s-tractable. We show that languages in the remaining two classes (0-optimal and 1-optimal[3]) are s-tractable if, for every weighted relation, the set of feasible tuples and the set of optimal tuples are *essentially downsets* (in the 0-optimal case; see Definition 3.8) or *essentially upsets* (in the 1-optimal case); otherwise, they are s-intractable.

Somewhat surprisingly, such languages are s-tractable regardless of the remaining (i.e., finite but non-optimal) values. Those values must, however, bear on the time bound of any polynomial-time algorithm solving surjective VCSPs over such languages (unless P = NP). In particular, we give an example of an infinite language Γ that is s-tractable (i.e., $\text{VCSP}_s(\Gamma')$ can be solved in polynomial time for every finite $\Gamma' \subseteq \Gamma$) but $\text{VCSP}_s(\Gamma)$ is NP-hard (see Example 3.12). This is quite unusual; all known tractable classes of VCSPs are in fact *globally tractable*, which means that $\text{VCSP}(\Gamma')$ is solvable by the same polynomial-time algorithm for every finite subset Γ' of a tractable language Γ, and hence $\text{VCSP}(\Gamma)$ is also polynomial-time solvable [7]. To capture this distinction, our main result (Theorem 3.2) gives a classification in terms of *global* s-tractability,[4] from which a classification for s-tractability easily follows (see Remark 2). We call the condition that describes the borderline

[2] A $\{0, \infty\}$-valued weighted relation is 0-valid (1-valid) if it assigns value 0 to the all-zero (all-one) tuple.

[3] A weighted relation is 0-optimal (1-optimal) if the all-zero (all-one) tuple minimises it.

[4] Weighted relations in an instance are assumed to be represented explicitly (by tables of values). We only consider languages of bounded arity; this restriction is vital in some of our proofs. Also, unbounded arity presents new challenges to complexity classification. For example, explicitly representing a weighted relation of an arity that is super-logarithmic in the number of variables requires super-polynomial space.

of global s-tractability in the 0-optimal case *EDS* (see Definition 3.1), drawing a parallel to the corresponding condition for s-tractability, which involves essentially downsets. The 1-optimal case is analogous (one only needs to exchange the roles of labels 0 and 1).

Tractability. While 0-optimal and 1-optimal languages are trivially tractable for VCSPs, the algorithm for surjective VCSPs over the newly identified class of languages is nontrivial and constitutes our second main contribution. The global s-tractability part of our result is established by a reduction from $\overline{\mathbb{Q}}$-valued $\mathrm{VCSP}_{\mathrm{s}}$ to the *generalised Min-Cut* problem (defined in Section 5), for which we require to find all α-optimal solutions in polynomial time, where α is a constant depending on the valued constraint language. The generalised Min-Cut problem consists in minimising an objective function $f + g$, where f is a superadditive set function given by an oracle and g is a cut function (same as in the Min-Cut problem); see Section 5 for the details. We prove that the running time of our algorithm is roughly $O\left(n^{20\alpha}\right)$, thus improving on the bound of $O\left(n^{3^{3\alpha}}\right)$ established in [43] (one of the two extended conference abstracts of this paper) for the special case of $\{0, 1\}$-valued languages.

Hardness. The hardness part of our result is proved by analysing weighted relations that can be obtained from a language using gadgets that preserve (global) s-tractability. Since not all standard gadgets have this property (in particular, minimisation over a variable may affect the surjectivity of a solution), we cannot employ the algebraic approach [11]. Instead, we define a collection of operations that form building blocks of gadgets preserving tractability in the surjective setting (see Definition 2.9 and Lemma 2.10). Such gadgets apply to non-Boolean domains as well, and may be useful in future work on non-Boolean surjective VCSPs. Another important ingredient of our proof is the NP-hardness of the *Minimum Distance* problem [45], which to the best of our knowledge has not previously appeared in the literature on exact solvability of (V)CSPs.

Approximability. By a simple reduction, our main result implies a complexity classification of the approximability of maximising $\mathbb{Q}_{\geq 0}$-valued surjective VCSPs (see Theorem 3.16).

Enumeration. For the globally s-tractable languages, we also show that *all* optimal solutions can be enumerated with polynomial delay [44] (see Theorem 3.4). While this is an easy observation for the already known globally s-tractable languages (since constants C_D allow for a standard self-reduction technique), we prove the same result for the newly discovered classes of languages, which do *not* include constants C_D.

Related work

Recent years have seen some remarkable progress on the computational complexity of CSPs and VCSPs parametrised by the (valued) constraint language. We highlight the resolution of the "bounded width conjecture" [2] and the result that a dichotomy for CSPs, conjectured in [18] and recently established by two independent proofs [6, 48], implies a dichotomy for VCSPs [31, 32]. All this work is for arbitrary (i.e., not necessarily Boolean) finite domains and relies on the algebraic approach initiated in [7] and nicely described in a survey [3].

One of the important aspects of the algebraic approach is the assumption that constants C_D are present in (valued) constraint languages. (This is without loss of generality with respect to polynomial-time solvability.) In the surjective setting, it is the lack of constants that makes it difficult, if not impossible, to employ the algebraic approach. Chen made the first step in this direction [9] but it is not clear how to take his result (for CSPs) further.

For a binary (unweighted) relation γ, $\mathrm{VCSP}_{\mathrm{s}}(\{\gamma\})$ has been studied under the name of surjective γ-Colouring [4, 24, 25, 35] and vertex-compaction [47]. We remark that our notion of surjectivity is global. For the γ-Colouring problem, a local version of surjectivity has also been studied [19, 20]. This version corresponds to finding a graph homomorphism such that the neighbourhood of every vertex v is mapped surjectively onto the neighbourhood of the image of v.

Under the assumption of the unique games conjecture [30], Raghavendra has shown that the optimal approximation ratio for maximising $\mathbb{Q}_{\geq 0}$-valued VCSPs is achieved by the basic semidefinite programming relaxation [37, 38].

Bach and Zhou have shown that any Max-CSP that is solvable in polynomial time in the non-surjective setting admits a PTAS in the surjective setting, and that any Max-CSP that is APX-hard in the non-surjective setting remains APX-hard in the surjective setting [1].

2 PRELIMINARIES

2.1 Weighted relations and VCSPs

We work in the arithmetic model of computation, i.e., every number is represented in constant space, and basic arithmetic operations take constant time. Let $\overline{\mathbb{Q}} = \mathbb{Q} \cup \{\infty\}$ denote the set of extended rationals. For any $c \in \overline{\mathbb{Q}}$, we define $c \leq \infty$ and $\infty + c = c + \infty = \infty$. If $c \geq 0$, we define $c \cdot \infty = \infty \cdot c = \infty$. We leave the result of multiplying ∞ undefined for $c < 0$.

For any integer $n \geq 1$, let $[n] = \{1, \ldots, n\}$.

Definition 2.1. Let $r \geq 1$ be an integer. An r-ary *weighted relation* over D is a mapping $\gamma : D^r \to \overline{\mathbb{Q}}$; the *arity* of γ equals $\mathrm{ar}(\gamma) = r$. We denote by $\mathrm{Feas}(\gamma)$ the underlying *feasibility relation* of γ, i.e.

$$\mathrm{Feas}(\gamma) = \{\mathbf{x} \in D^r \mid \gamma(\mathbf{x}) < \infty\} \ . \tag{6}$$

We denote by $\mathrm{Opt}(\gamma)$ the relation consisting of the minimal-valued tuples, i.e.

$$\mathrm{Opt}(\gamma) = \{\mathbf{x} \in \mathrm{Feas}(\gamma) \mid \gamma(\mathbf{x}) \leq \gamma(\mathbf{y}) \text{ for every } \mathbf{y} \in D^r\} \ . \tag{7}$$

A weighted relation γ is called *crisp* if $\mathrm{Feas}(\gamma) = \mathrm{Opt}(\gamma)$. In other words, there exists a constant $c \in \mathbb{Q}$ such that $\gamma(\mathbf{x}) = c$ for all $\mathbf{x} \in \mathrm{Feas}(\gamma)$ and $\gamma(\mathbf{x}) = \infty$ for all $\mathbf{x} \in D^r \setminus \mathrm{Feas}(\gamma)$.

Weighted relations that differ only by a constant are considered equivalent, as adding a rational constant to a weighted relation changes the value of every solution to the VCSP by the same amount. Therefore, a crisp weighted relation γ can be equated with the relation $\mathrm{Feas}(\gamma)$. Conversely, a relation ρ can be seen as a crisp weighted relation γ_c with $\mathrm{Feas}(\gamma_c) = \rho$ and the codomain equal to $\{c, \infty\}$ for some $c \in \mathbb{Q}$. Unless stated otherwise, we choose $c = 0$.

Definition 2.2. We denote by $\rho_=$ the binary equality relation $\{(d, d) \mid d \in D\}$. For any $d \in D$, we denote by ρ_d the unary relation $\{(d)\}$, which is called a *constant*. We denote the set of constants on D by $C_D = \{\rho_d \mid d \in D\}$.

For any relation ρ, we denote by $\mathrm{Soft}(\rho)$ the *soft variant* of ρ defined by $\mathrm{Soft}(\rho)(\mathbf{x}) = 0$ if $\mathbf{x} \in \rho$ and $\mathrm{Soft}(\rho)(\mathbf{x}) = 1$ otherwise.

Definition 2.3. A *constraint language* (or simply a *language*) over D is a (possibly infinite) set of weighted relations over D.

In this paper, we only consider languages of bounded arity. Note that a crisp language is of a bounded arity if and only if it is finite.

Definition 2.4. A language Γ is called *s-tractable* if, for every finite $\Gamma' \subseteq \Gamma$, $\mathrm{VCSP_s}(\Gamma')$ can be solved in polynomial time. If $\mathrm{VCSP_s}(\Gamma)$ can be solved in polynomial time, language Γ is called *globally s-tractable*.

If there exists a finite $\Gamma' \subseteq \Gamma$ such that $\mathrm{VCSP_s}(\Gamma')$ is NP-hard, language Γ is called *s-intractable*. If $\mathrm{VCSP_s}(\Gamma)$ is NP-hard, language Γ is called *globally s-intractable*.

Note that a globally s-tractable language is s-tractable, and an s-intractable language is globally s-intractable.

Lemmas 2.5 and 2.6 establish a relation between the complexity of the VCSP and $\mathrm{VCSP_s}$. We denote by $\leq_p$ the standard polynomial-time Turing reduction.

Lemma 2.5. *For any constraint language* Γ,

$$\mathrm{VCSP}(\Gamma) \leq_p \mathrm{VCSP_s}(\Gamma)\,. \tag{8}$$

Proof. Given an instance I of $\mathrm{VCSP}(\Gamma)$, we construct an instance I' of $\mathrm{VCSP_s}(\Gamma)$ by adding $|D|$ extra variables. Any solution to I can be extended to a surjective solution to I' of the same value and, conversely, any (surjective) solution to I' induces a solution to I of the same value. □

Lemma 2.6. *For any constraint language* Γ,

$$\mathrm{VCSP_s}(\Gamma) \leq_p \mathrm{VCSP}(\Gamma \cup \mathcal{C}_D)\,. \tag{9}$$

Proof. Given an instance $I = (V, D, \phi_I)$ of $\mathrm{VCSP_s}(\Gamma)$, we iterate through all $O\left(|V|^{|D|}\right)$ injective mappings $f : D \to V$. For each mapping f, we construct an instance I'_f of $\mathrm{VCSP}(\Gamma \cup \mathcal{C}_D)$ by adding constraints $\rho_d(f(d))$ for all $d \in D$. The additional constraints guarantee that only surjective solutions to I'_f are feasible. Conversely, any surjective solution to I is a feasible solution to I'_f for some mapping f. Therefore, a solution of the smallest value among optimal solutions to I'_f for all f is an optimal surjective solution to I. □

Corollary 2.7. *Any (globally) tractable language* Γ *with* $\mathcal{C}_D \subseteq \Gamma$ *is also (globally) s-tractable.*

Now we define a few operations on weighted relations that occur throughout the paper.

Definition 2.8. Let γ be an r-ary weighted relation.

- *Addition of a rational constant*: For any $c \in \mathbb{Q}$, $\gamma + c = \gamma'$ such that $\gamma'(\mathbf{x}) = \gamma(\mathbf{x}) + c$.
- *Non-negative scaling*: For any $c \in \mathbb{Q}_{\geq 0}$, $c \cdot \gamma = \gamma'$ such that $\gamma'(\mathbf{x}) = c \cdot \gamma(\mathbf{x})$. Note that $0 \cdot \gamma = \mathrm{Feas}(\gamma)$.
- *Coordinate mapping*: For any arity r' and mapping $f : [r] \to [r']$, $f(\gamma) = \gamma'$ such that $\gamma'(x_1, \ldots, x_{r'}) = \gamma\left(x_{f(1)}, \ldots, x_{f(r)}\right)$.
- *Minimisation*: For any $i \in [r]$, the minimisation of γ at coordinate i results in γ' such that $\gamma'(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_r) = \min_{x_i \in D} \gamma(x_1, \ldots, x_r)$.
- *Pinning*: For any $d \in D$ and $i \in [r]$, the pinning of γ to label d at coordinate i results in γ' such that $\gamma'(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_r) = \gamma(x_1, \ldots, x_{i-1}, d, x_{i+1}, \ldots, x_r)$. A pinning to label d is called a *d-pinning*.
- *Addition*: For any weighted relations γ_1, γ_2 with $\mathrm{ar}(\gamma_1) = \mathrm{ar}(\gamma_2)$, $\gamma_1 + \gamma_2 = \gamma'$ such that $\gamma'(\mathbf{x}) = \gamma_1(\mathbf{x}) + \gamma_2(\mathbf{x})$.

We extend operations on weighted relations to languages in the natural way, e.g., $\mathrm{Feas}(\Gamma) = \{\mathrm{Feas}(\gamma) \mid \gamma \in \Gamma\}$.

A weighted relational clone [11] is a language closed under certain operations (e.g., non-negative scaling and minimisation) that preserve the tractability of languages in the following sense: The VCSP over the smallest weighted relational clone containing a language Γ can be reduced in polynomial time to $\mathrm{VCSP}(\Gamma)$. Weighted relational clones are characterised by their weighted polymorphisms (a generalisation of multimorphisms defined in Definition 2.12), which enables the employment of tools from universal algebra in the effort to obtain a complexity classification of languages.

In the surjective setting, however, minimisation may not preserve the tractability of languages, and thus we need to define a language closure that excludes this operation. Consequently, we are unable to use the algebraic approach in our proofs in Section 4.

Definition 2.9. A constraint language Γ is called *closed* if it is closed under addition, coordinate mapping, non-negative scaling, addition of a rational constant, operation Opt, and, for all $d \in D$ such that $\rho_d \in \Gamma$, d-pinning.

We define Γ^* to be the smallest closed language containing Γ.

Now we show that these closure operations preserve the complexity of the $\mathrm{VCSP_s}$. Note that we require a language to be closed under d-pinning *only* if it contains ρ_d.

Lemma 2.10. *For any constraint language* Γ*,*

$$\mathrm{VCSP_s}(\Gamma^*) \leq_p \mathrm{VCSP_s}(\Gamma) . \tag{10}$$

Proof. For most of the closure operations, standard reductions for the VCSP apply to the surjective setting as well. Let $\gamma_1, \gamma_2 \in \Gamma$ be weighted relations with $\mathrm{ar}(\gamma_1) = \mathrm{ar}(\gamma_2)$, and let $\gamma' = \gamma_1 + \gamma_2$. Then $\mathrm{VCSP_s}(\Gamma \cup \{\gamma'\}) \leq_p \mathrm{VCSP_s}(\Gamma)$, as any constraint of the form $w \cdot \gamma'(\mathbf{x})$ can be replaced with a pair of constraints $w \cdot \gamma_1(\mathbf{x})$, $w \cdot \gamma_2(\mathbf{x})$. Similarly, let $\gamma \in \Gamma$ and $\gamma' = f(\gamma)$ where $f : [\mathrm{ar}(\gamma)] \to [\mathrm{ar}(\gamma')]$; then any constraint of the form $w \cdot \gamma'\left(x_1, \ldots, x_{\mathrm{ar}(\gamma')}\right)$ can be replaced with a constraint $w \cdot \gamma\left(x_{f(1)}, \ldots, x_{f(\mathrm{ar}(\gamma))}\right)$. Non-negative scaling can be achieved by scaling the weight of affected constraints. Addition of a rational constant changes the value of every solution by the same amount, and thus it can be ignored.

Now we show that $\mathrm{VCSP_s}(\Gamma \cup \{\mathrm{Opt}(\gamma)\}) \leq_p \mathrm{VCSP_s}(\Gamma)$ for any $\gamma \in \Gamma$. Let I be an instance of $\mathrm{VCSP_s}(\Gamma \cup \{\mathrm{Opt}(\gamma)\})$. Without loss of generality, assume that the minimum values assigned by γ and $\mathrm{Opt}(\gamma)$ equal 0 and all weighted relations in I assign non-negative values (this can be achieved by adding rational constants). We may also assume that γ is not crisp (otherwise $\mathrm{Opt}(\gamma) = \gamma$). Let m denote the smallest positive value assigned by γ, and let M be an upper bound on the value of any feasible solution to I (e.g., the weighted sum of the maximum finite values assigned by the constraints of I). We replace every constraint of the form $w \cdot \mathrm{Opt}(\gamma)(\mathbf{x})$ in I with a constraint $(M/m + 1) \cdot \gamma(\mathbf{x})$ to obtain an instance $I' \in \mathrm{VCSP_s}(\Gamma)$. Any feasible solution to instance I gets assigned the same value by I'. Any infeasible solution to instance I is either infeasible for I' as well, or it incurs an infinite value from a constraint of the form $w \cdot \mathrm{Opt}(\gamma)(\mathbf{x})$ in I and thus a value of at least $(M/m + 1) \cdot m > M$ in I'. Therefore, an optimal solution to I' is optimal for I as well.

In the case of pinning, we need a different reduction as the standard one relies on minimisation. Suppose that $\rho_d \in \Gamma$. Let γ' be a d-pinning of a weighted relation $\gamma \in \Gamma$; without loss of generality, let it be a pinning at the first coordinate. We show that $\mathrm{VCSP_s}(\Gamma \cup \{\gamma'\}) \leq_p \mathrm{VCSP_s}(\Gamma)$. Let $I = (V, D, \phi_I)$ be an instance of $\mathrm{VCSP_s}(\Gamma \cup \{\gamma'\})$ with $V = \{x_1, \ldots, x_n\}$. In a surjective solution to I, at least one variable is assigned label d, but we do not a priori know which one. For every $i \in [n]$, we construct an instance $I_i = (V, D, \phi_{I_i})$ of $\mathrm{VCSP_s}(\Gamma)$ by replacing all constraints of the form $\gamma'(\mathbf{x})$ with $\gamma(x_i, \mathbf{x})$ and adding a constraint $\rho_d(x_i)$ to force variable x_i to take label d. A solution of the smallest value among optimal solutions to $I_1, \ldots, I_n$ is an optimal solution to I. □

2.2 Polymorphisms and multimorphisms

For any $r \geq 1$ and a k-ary operation $h : D^k \to D$, we extend h to r-tuples over D by applying it componentwise. Namely, for $\mathbf{x}_1, \ldots, \mathbf{x}_k \in D^r$ where $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,r})$, we define $h(\mathbf{x}_1, \ldots, \mathbf{x}_k) \in D^r$ by

$$h\left(\mathbf{x}_1, \ldots, \mathbf{x}_k\right) = \left(h\left(x_{1,1}, \ldots, x_{k,1}\right), \ldots, h\left(x_{1,r}, \ldots, x_{k,r}\right)\right) . \tag{11}$$

The following notion is at the heart of the algebraic approach to decision CSPs [7].

Definition 2.11. Let γ be a weighted relation on D. A k-ary operation $h : D^k \to D$ is a *polymorphism* of γ (and γ is *invariant under* or *admits* h) if, for every $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \operatorname{Feas}(\gamma)$, we have $h(\mathbf{x}_1, \ldots, \mathbf{x}_k) \in \operatorname{Feas}(\gamma)$. We say that h is a polymorphism of a language Γ if it is a polymorphism of every $\gamma \in \Gamma$.

The following notion, which involves a collection of k k-ary polymorphisms, plays an important role in the complexity classification of Boolean valued constraint languages [12], as we will see in Theorem 2.14 in Section 2.3.

Definition 2.12. Let γ be a weighted relation on D. A list $\langle h_1, \ldots, h_k \rangle$ of k-ary polymorphisms of γ is a k-ary *multimorphism* of γ (and γ *admits* $\langle h_1, \ldots, h_k \rangle$) if, for every $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \operatorname{Feas}(\gamma)$, we have

$$\sum_{i=1}^{k} \gamma(h_i(\mathbf{x}_1, \ldots, \mathbf{x}_k)) \leq \sum_{i=1}^{k} \gamma(\mathbf{x}_i) .$$

$\langle h_1, \ldots, h_k \rangle$ is a multimorphism of a language Γ if it is a multimorphism of every $\gamma \in \Gamma$.

The operations in Definition 2.9 preserve multimorphisms [12, 22], i.e., any multimorphism of a language Γ is also a multimorphism of Γ^*. Consequently, all polymorphisms of a crisp weighted relation are preserved.

2.3 Boolean VCSPs

In the rest of the paper, we consider only Boolean languages (i.e., $D = \{0, 1\}$), unless explicitly mentioned otherwise. For any arity $r \geq 1$, we denote by $\mathbf{0}^r$ ($\mathbf{1}^r$) the zero (one) r-tuple. For r-tuples $\mathbf{x} = (x_1, \ldots, x_r)$ and $\mathbf{y} = (y_1, \ldots, y_r) \in D^r$, we define $\mathbf{x} \leq \mathbf{y}$ if and only if $x_i \leq y_i$ for all $i \in [r]$ (where $0 < 1$). We also define the following operations on D:

- For any $a \in D$, c_a is the constant unary operation such that $c_a(x) = a$ for all $x \in D$.
- Operation $\neg$ is the unary negation, i.e. $\neg(0) = 1$ and $\neg(1) = 0$. For a weighted relation γ, we define $\neg(\gamma)$ to be the weighted relation $\neg(\gamma)(\mathbf{x}) = \gamma(\neg(\mathbf{x}))$. For a language Γ, we define $\neg(\Gamma) = \{\neg(\gamma) \mid \gamma \in \Gamma\}$. Note that $\neg(\Gamma)$ can be obtained from Γ simply by exchanging the labels $\{0, 1\}$, and hence has the same complexity as Γ.
- Binary operation $\oplus$ is the addition modulo 2 operation. In this case, we use the infix notation, i.e., $0 \oplus 0 = 0 = 1 \oplus 1$ and $0 \oplus 1 = 1 = 1 \oplus 0$.
- Binary operation min (max) returns the smaller (larger) of its two arguments with respect to the order $0 < 1$.
- Binary operation sub (for subtraction) is defined as $\operatorname{sub}(x, y) = \min(x, \neg y)$.
- Ternary operation Mn (for minority) is the unique ternary operation on D satisfying

  $$\operatorname{Mn}(x, x, y) = \operatorname{Mn}(x, y, x) = \operatorname{Mn}(y, x, x) = y$$

  for all $x, y \in D$.
- Ternary operation Mj (for majority) is the unique ternary operation on D satisfying

  $$\operatorname{Mj}(x, x, y) = \operatorname{Mj}(x, y, x) = \operatorname{Mj}(y, x, x) = x$$

  for all $x, y \in D$.

Lemma 2.13. *If a weighted relation admits polymorphism* sub*, then it also admits polymorphisms* c_0 *and* min.

Proof. For every $x, y \in D$, it holds $c_0(x) = 0 = \operatorname{sub}(x, x)$ and $\min(x, y) = \operatorname{sub}(x, \operatorname{sub}(x, y))$. □

Cohen et al. [12] established a complexity classification of Boolean constraint languages.

Theorem 2.14 ([12, Theorem 7.1]). *Let* Γ *be a Boolean* $\overline{\mathbb{Q}}$*-valued language. Then* Γ *is tractable if it admits any the following eight multimorphisms:* $\langle c_0\rangle$, $\langle c_1\rangle$, $\langle \min, \min\rangle$, $\langle \max, \max\rangle$, $\langle \min, \max\rangle$, $\langle \mathrm{Mn}, \mathrm{Mn}, \mathrm{Mn}\rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mj}\rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn}\rangle$*. Otherwise,* Γ *is intractable.*

Note that multimorphism $\langle \min, \max\rangle$ corresponds to submodularity [40]. Constants $C_D = \{\rho_0, \rho_1\}$ admit multimorphisms $\langle \min, \min\rangle$, $\langle \max, \max\rangle$, $\langle \min, \max\rangle$, $\langle \mathrm{Mn}, \mathrm{Mn}, \mathrm{Mn}\rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mj}\rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn}\rangle$; hence, these six classes of languages are s-tractable by Lemma 2.6. However, ρ_0 does not admit $\langle c_1\rangle$ and ρ_1 does not admit $\langle c_0\rangle$.

Remark 1. Although Theorem 2.14 is stated only for the weaker notion of tractability (i.e., for finite languages) in [12], the proofs there actually establish the same classification for the stronger notion of global tractability as well.

In particular, all the tractable classes (characterised by the eight multimorphisms) are globally tractable. Conversely, any globally intractable language is also intractable: If a language Γ does not admit any of the eight multimorphisms, then there exists a finite subset $\Gamma' \subseteq \Gamma$ with $|\Gamma'| \leq 8$ that does not admit any of the eight multimorphisms (since a single weighted relation suffices to violate a multimorphism).

We note that Theorem 2.14 is a generalisation of Schaefer's classification of $\{0, \infty\}$-valued constraint languages [39] and Creignou's classification of $\{0, 1\}$-valued constraint languages [13]. In particular, Theorem 2.14 implies the following classification of $\mathbb{Q}$-valued languages.

Theorem 2.15 ([12, Corollary 7.11]). *Let* Γ *be a Boolean* $\mathbb{Q}$*-valued language. Then* Γ *is tractable if it admits any of the following multimorphisms:* $\langle c_0\rangle$, $\langle c_1\rangle$, $\langle \min, \max\rangle$*. Otherwise,* Γ *is intractable.*

Creignou and Hébrard [14] established a complexity classification of Boolean $\{0, \infty\}$-valued languages in the surjective setting.

Theorem 2.16 ([14]). *Let* Γ *be a Boolean* $\{0, \infty\}$*-valued language. Then* Γ *is s-tractable if it is invariant under any of the following operations:* min, max, Mn, Mj*. Otherwise,* Γ *is s-intractable.*

3 RESULTS

We present our results in three parts: Section 3.1 defines the EDS property and states the main classification theorem, Section 3.2 focuses on finite EDS languages, and Section 3.3 gives a classification in terms of approximability for the surjective Max-VCSP.

3.1 Boolean surjective VCSPs

We first define the property EDS (which stands for *essentially a downset*, see Definition 3.8) characterising the newly discovered tractable class of weighted relations.

Definition 3.1. For any $\alpha \geq 1$, an r-ary weighted relation γ is α*-EDS* if, for every $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\gamma)$, it holds $\mathbf{0}^r \in \mathrm{Feas}(\gamma)$ and

$$\alpha \cdot (\gamma(\mathbf{x}) + \gamma(\mathbf{y}) - 2 \cdot \gamma(\mathbf{0}^r)) \geq \gamma(\mathrm{sub}(\mathbf{x}, \mathbf{y})) - \gamma(\mathbf{0}^r)\,. \tag{12}$$

A weighted relation is *EDS* if it is α-EDS for some $\alpha \geq 1$. A language is EDS if there exists $\alpha \geq 1$ such that every weighted relation in the language is α-EDS.

Although this definition does not involve the notion of polymorphisms, it is stated in a similar vein. Let h be a binary operation defined by $h(x, y) = 0$; then the requirement "for every $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\gamma)$, it holds $\mathbf{0}^r \in \mathrm{Feas}(\gamma)$" translates to "$\gamma$ is invariant under h" (or, equivalently, "γ is invariant under c_0").[5][6] In the case of $\alpha = 1$, inequality (12) translates to that of admitting multimorphism $\langle \mathrm{sub}, h\rangle$.

[5]In fact, any EDS weighted relation admits multimorphism $\langle c_0\rangle$ (see Lemma 3.5).

[6]Note that the unary empty relation $\rho_\emptyset$ is vacuously α-EDS for all $\alpha \geq 1$, as $\mathrm{Feas}(\rho_\emptyset) = \emptyset$.

For more intuition behind this notion in the general case, see the corresponding definition of EDS for set functions (Definition 5.13) in Section 5.3. Finite EDS languages admit a simpler equivalent definition, see Corollary 3.11.

The following classification of $\overline{\mathbb{Q}}$-valued languages is our main result.

Theorem 3.2. *Let Γ be a Boolean $\overline{\mathbb{Q}}$-valued language. Then Γ is globally s-tractable if it is EDS, or $\neg(\Gamma)$ is EDS, or Γ admits any of the following multimorphisms:* $\langle \min, \min \rangle$, $\langle \max, \max \rangle$, $\langle \min, \max \rangle$, $\langle \mathrm{Mn}, \mathrm{Mn}, \mathrm{Mn} \rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mj} \rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn} \rangle$. *Otherwise, Γ is globally s-intractable.*

Proof. The global s-tractability of languages admitting any of the six multimorphisms in the statement of the theorem follows from Theorem 2.14 (see Remark 1) by Lemma 2.6. The global s-tractability of EDS languages (whether Γ or $\neg(\Gamma)$, which is symmetric) follows from Theorem 5.18, proved in Section 5. Finally, the global s-intractability of the remaining languages follows from Theorem 4.12, proved in Section 4. □

Remark 2. Theorem 3.2 gives us also a classification in terms of s-tractability. As noted in Section 2.1, any globally s-tractable language is s-tractable. Consider now a globally s-intractable language Γ. It does not admit any of the six multimorphisms, and hence there exists a finite subset of Γ that does not admit them either (see Remark 1). If there exists a finite subset $\Gamma' \subseteq \Gamma$ such that neither Γ' nor $\neg(\Gamma')$ is EDS, then Γ is s-intractable; otherwise Γ is s-tractable. Equivalently (by Corollary 3.11), Γ is s-intractable if neither $\mathrm{Feas}(\Gamma) \cup \mathrm{Opt}(\Gamma)$ nor $\mathrm{Feas}(\neg(\Gamma)) \cup \mathrm{Opt}(\neg(\Gamma))$ admit polymorphism sub, and it is s-tractable otherwise.

To see how EDS languages fit into the classification of $\{0, \infty\}$-valued languages established in Theorem 2.16, note the following. Any $\{0, \infty\}$-valued language of bounded arity is finite. By Corollary 3.11, any EDS $\{0, \infty\}$-valued language admits polymorphism sub, and hence also polymorphism min (by Lemma 2.13).

For $\mathbb{Q}$-valued languages, Theorem 3.2 gives a tighter classification: the only reasons for global s-tractability are EDS and submodularity.

Theorem 3.3. *Let Γ be a Boolean $\mathbb{Q}$-valued language. Then Γ is globally s-tractable if it is EDS, or $\neg(\Gamma)$ is EDS, or Γ admits the $\langle \min, \max \rangle$ multimorphism. Otherwise, Γ is globally s-intractable.*

Proof. We need to show that in the case of $\mathbb{Q}$-valued languages, the remaining globally s-tractable classes from Theorem 3.2 (which are characterised by polymorphisms $\langle \min, \min \rangle$, $\langle \max, \max \rangle$, $\langle \mathrm{Mn}, \mathrm{Mn}, \mathrm{Mn} \rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mj} \rangle$, and $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn} \rangle$) collapse.

If a $\mathbb{Q}$-valued r-ary weighted relation γ admits the $\langle \min, \min \rangle$ multimorphism, then it holds $\gamma(\mathbf{x}) \geq \gamma(\mathbf{y})$ for all $\mathbf{x} \geq \mathbf{y}$. This implies that, for all $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\gamma)$, it holds $\gamma(\mathbf{x}) \geq \gamma(\mathrm{sub}(\mathbf{x}, \mathbf{y}))$ and $\gamma(\mathbf{y}) \geq \gamma(\mathbf{0}^r)$. Hence, γ is 1-EDS. If γ admits the $\langle \max, \max \rangle$ multimorphism, then $\neg(\gamma)$ admits the $\langle \min, \min \rangle$ multimorphism. Therefore, if a $\mathbb{Q}$-valued language Γ admits $\langle \min, \min \rangle$ or $\langle \max, \max \rangle$ as a multimorphism, then Γ or $\neg(\Gamma)$ is EDS.

Multimorphisms $\langle \mathrm{Mn}, \mathrm{Mn}, \mathrm{Mn} \rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mj} \rangle$, and $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn} \rangle$ are all covered by the $\langle \min, \max \rangle$ multimorphism: Weighted relations that admit $\langle \mathrm{Mn}, \mathrm{Mn}, \mathrm{Mn} \rangle$ or $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mj} \rangle$ as a multimorphism are crisp [12, Propositions 6.20 and 6.22], and hence, in the $\mathbb{Q}$-valued case, they are constant functions. $\mathbb{Q}$-valued weighted relations that admit the $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn} \rangle$ multimorphism are modular [12, Corollary 6.26], and hence they are submodular. □

Enumerating all optimal solutions to an instance with polynomial delay is a fundamental problem [27, 46] studied in the context of CSP [8, 16]. An algorithm outputting a sequence of solutions works with polynomial delay if the time it takes to output the first solution as well as the time it takes between every two consecutive solutions is bounded by a polynomial in the input size.

It is known that, for a tractable constraint language Γ that includes constants C_D, one can enumerate all optimal solutions with polynomial delay [10]. Our results imply that the newly discovered globally s-tractable EDS languages enjoy the same property (despite *not* including constants).

Theorem 3.4. *Let Γ be a Boolean $\overline{\mathbb{Q}}$-valued language. If Γ is globally s-tractable then there is a polynomial-delay algorithm that enumerates all optimal solutions to any instance of* $\mathrm{VCSP_s}(\Gamma)$.

The theorem is proved in Section 5.3.

3.2 Finite EDS languages

The EDS property can be described in a simpler way for languages of finite size; see the following observation and Corollary 3.11.

Observation 1. *A language of finite size is EDS if and only if it consists of EDS weighted relations.*

In the following we prove several useful properties EDS weighted relations.

Lemma 3.5. *Any EDS weighted relation admits multimorphism $\langle c_0 \rangle$.*

Proof. Let γ be an r-ary α-EDS weighted relation. For any $\mathbf{x} \in \mathrm{Feas}(\gamma)$, it holds $\mathbf{0}^r \in \mathrm{Feas}(\gamma)$ and

$$\alpha \cdot (2 \cdot \gamma(\mathbf{x}) - 2 \cdot \gamma(\mathbf{0}^r)) \geq \gamma(\mathrm{sub}(\mathbf{x}, \mathbf{x})) - \gamma(\mathbf{0}^r) = 0 \tag{13}$$

as $\mathrm{sub}(\mathbf{x}, \mathbf{x}) = \mathbf{0}^r$, and therefore $\gamma(\mathbf{x}) \geq \gamma(\mathbf{0}^r)$. □

Lemma 3.6. *A crisp weighted relation is EDS if and only if it admits polymorphism* sub.

Proof. Any EDS weighted relation admits polymorphism sub. For the converse implication, note that any crisp weighted relation that admits polymorphism sub (and thus, by Lemma 2.13, also polymorphism c_0) satisfies (12) for any $\alpha \geq 1$. □

Lemma 3.7. *A weighted relation γ is EDS if and only if both* $\mathrm{Feas}(\gamma)$ *and* $\mathrm{Opt}(\gamma)$ *are EDS.*

Proof. Let γ be an r-ary α-EDS weighted relation. For any $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\gamma)$, it holds $\mathbf{0}^r \in \mathrm{Feas}(\gamma)$ and

$$\infty > \alpha \cdot (\gamma(\mathbf{x}) + \gamma(\mathbf{y}) - 2 \cdot \gamma(\mathbf{0}^r)) \geq \gamma(\mathrm{sub}(\mathbf{x}, \mathbf{y})) - \gamma(\mathbf{0}^r)\,, \tag{14}$$

and hence $\mathrm{sub}(\mathbf{x}, \mathbf{y}) \in \mathrm{Feas}(\gamma)$. By Lemma 3.6, $\mathrm{Feas}(\gamma)$ is EDS. Similarly, for any $\mathbf{x}, \mathbf{y} \in \mathrm{Opt}(\gamma)$, it holds $\mathbf{0}^r \in \mathrm{Opt}(\gamma)$ (by Lemma 3.5) and

$$0 = \alpha \cdot (\gamma(\mathbf{x}) + \gamma(\mathbf{y}) - 2 \cdot \gamma(\mathbf{0}^r)) \geq \gamma(\mathrm{sub}(\mathbf{x}, \mathbf{y})) - \gamma(\mathbf{0}^r)\,; \tag{15}$$

therefore $\mathrm{sub}(\mathbf{x}, \mathbf{y}) \in \mathrm{Opt}(\gamma)$ and $\mathrm{Opt}(\gamma)$ is EDS.

To prove the converse implication, let us assume that $\mathrm{Feas}(\gamma)$, $\mathrm{Opt}(\gamma)$ are EDS and consider any $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\gamma)$. As $\mathrm{Opt}(\gamma)$ admits polymorphism c_0, it holds $\mathbf{0}^r \in \mathrm{Opt}(\gamma) \subseteq \mathrm{Feas}(\gamma)$. Therefore, the left-hand side of (12) is non-negative. Moreover, if it equals 0, then $\mathbf{x}, \mathbf{y} \in \mathrm{Opt}(\gamma)$, and hence $\mathrm{sub}(\mathbf{x}, \mathbf{y}) \in \mathrm{Opt}(\gamma)$ and the right-hand side equals 0 as well. Therefore, (12) holds for large enough α, as there are only finitely many choices of $\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\gamma)$. □

We show that relations invariant under sub have a simple structure.

Definition 3.8. An r-ary relation ρ is a *downset* if, for any r-tuples $\mathbf{x}, \mathbf{y}$ such that $\mathbf{x} \geq \mathbf{y}$ and $\mathbf{x} \in \rho$, it holds $\mathbf{y} \in \rho$.

An r-ary relation ρ is *essentially a downset* if it can be written as a conjunction of a downset and binary equality relations. Formally, there exists a downset ρ' with $\mathrm{ar}(\rho') = r' \leq r$, a permutation π of $[r]$, and indices $a_{r'+1}, \ldots, a_r \in \{\pi(1), \ldots, \pi(r')\}$ such that

$$\rho(x_1, \ldots, x_r) = \rho'\left(x_{\pi(1)}, \ldots, x_{\pi(r')}\right) + \sum_{i=r'+1}^{r} \rho_{=}\left(x_{\pi(i)}, x_{a_i}\right) . \tag{16}$$

(Note that addition of crisp weighted relations corresponds to conjunction.) In other words, removing duplicate coordinates[7] of ρ results in a downset.

Example 3.9. Relation $\rho' = \{(0, 0), (0, 1), (1, 0)\}$ is a downset, while $\rho = \{(0, 0, 0), (0, 1, 1), (1, 0, 0)\}$ is only essentially a downset (as $\rho(x, y, z) = \rho'(x, y) + \rho_{=}(y, z)$).

Lemma 3.10. *A relation is essentially a downset if and only if it admits polymorphism* sub.

Proof. For any r-ary relation ρ that is essentially a downset and $\mathbf{x}, \mathbf{y} \in \rho$, we prove that $\mathbf{z} = \mathrm{sub}(\mathbf{x}, \mathbf{y}) \in \rho$. Let $\mathbf{x} = (x_1, \ldots, x_r)$, $\mathbf{y} = (y_1, \ldots, y_r)$, $\mathbf{z} = (z_1, \ldots, z_r)$. It holds $\mathbf{x} \geq \mathbf{z}$. Moreover, for any coordinates i, j such that $x_i = x_j$ and $y_i = y_j$, it holds $z_i = z_j$. Since ρ can be written as a sum of a downset and equality relations, we have $\mathbf{z} \in \rho$.

We prove the converse implication by contradiction. Suppose that ρ is a smallest-arity relation that admits polymorphism sub but is not essentially a downset; let us denote its arity by r. If there are distinct coordinates i, j such that $z_i = z_j$ for all $\mathbf{z} = (z_1, \ldots, z_r) \in \rho$, identifying these coordinates yields an $(r-1)$-ary relation ρ' such that ρ can be written as the sum of ρ' and a binary equality relation. However, ρ' also admits sub, and hence is essentially a downset by the choice of ρ, which implies that ρ is essentially a downset as well. Therefore, for any distinct coordinates i, j, there exists $\mathbf{z}^{(i,j)} \in \rho$ with $z_i^{(i,j)} \neq z_j^{(i,j)}$.

As ρ is not a downset, for some r-tuples $\mathbf{x}, \mathbf{y}$ it holds $\mathbf{x} \geq \mathbf{y}$, $\mathbf{x} \in \rho$, $\mathbf{y} \notin \rho$. We may assume without loss of generality that, for some $n \in [r]$, the set of coordinates with label 1 equals $[n]$ for $\mathbf{x}$ and $[n-1]$ for $\mathbf{y}$. Let $\mathbf{e} = (e_1, \ldots, e_r) \in \rho$ be a tuple with the smallest number of coordinates labelled 1 such that $e_n = 1$. We claim that $e_i = 0$ for all $i \neq n$: Otherwise, either $\mathrm{sub}\left(\mathbf{e}, \mathbf{z}^{(i,n)}\right) = \min\left(\mathbf{e}, \neg(\mathbf{z}^{(i,n)})\right)$ or $\mathrm{sub}\left(\mathbf{e}, \mathrm{sub}\left(\mathbf{e}, \mathbf{z}^{(i,n)}\right)\right) = \min\left(\mathbf{e}, \mathbf{z}^{(i,n)}\right)$ contradicts the minimality of $\mathbf{e}$. But then $\mathrm{sub}(\mathbf{x}, \mathbf{e}) = \mathbf{y} \in \rho$, which is a contradiction. □

Corollary 3.11. *Let Γ be a finite language. The following conditions are equivalent.*

(1) Language Γ is EDS.
(2) For every $\gamma \in \Gamma$, weighted relation γ is EDS.
(3) For every $\gamma \in \Gamma$, both $\mathrm{Feas}(\gamma)$ *and* $\mathrm{Opt}(\gamma)$ *admit polymorphism* sub.
(4) For every $\gamma \in \Gamma$, both $\mathrm{Feas}(\gamma)$ *and* $\mathrm{Opt}(\gamma)$ *are essentially downsets.*

Remark 3. In [23], a weighted relation γ is called PDS if both $\mathrm{Feas}(\gamma)$ and $\mathrm{Opt}(\gamma)$ are essentially downsets. For a $\{0, 1\}$-valued weighted relation, this condition is equivalent to that of being almost-min-min [43]. By Corollary 3.11, PDS and EDS are equivalent concepts for languages of finite size.

As we show in the following example, there exists an infinite non-EDS language Γ such that every finite subset $\Gamma' \subseteq \Gamma$ is EDS. Hence, Γ is s-tractable, although it is globally s-intractable ($\mathrm{VCSP_s}(\Gamma)$ is NP-hard).

[7] A coordinate i is a duplicate of a coordinate j if, for every $(x_1, \ldots, x_r) \in \rho$, it holds $x_i = x_j$.

Example 3.12. For any $w \in \mathbb{Z}_{\geq 1}$, we define a ternary weighted relation μ_w on $D = \{0,1\}$ by

$$\mu_w(x,y,z) = \begin{cases} 2 & \text{if } z = 1 \text{ and } x = y, \\ 1 & \text{if } z = 1 \text{ and } x \neq y, \\ 0 & \text{if } z = 0 \text{ and } x = y = 0, \\ w & \text{otherwise.} \end{cases} \tag{17}$$

Note that $\mathrm{Feas}(\mu_w) = D^3$ and $\mathrm{Opt}(\mu_w) = \{(0,0,0)\}$ are downsets, and hence μ_w is EDS. However, it is not α-EDS for any $\alpha < w/2$: For $\mathbf{x} = (0,1,1)$, $\mathbf{y} = (1,0,1)$, we have $\mu_w(\mathbf{x}) + \mu_w(\mathbf{y}) = 2$ but $\mu_w(\mathrm{sub}(\mathbf{x},\mathbf{y})) = \mu_w(0,1,0) = w$. Language $\Gamma = \{\mu_w \mid w \in \mathbb{Z}_{\geq 1}\}$ is therefore not EDS.

By our classification (Theorem 3.2), language Γ is globally s-intractable; here we show it directly by a reduction from the NP-hard Max-Cut problem. Given an undirected graph $G = (V,E)$ with no isolated vertices, we construct a $\mathrm{VCSP_s}(\Gamma)$ instance I as follows. Let $w = 2|E| + 1$. We introduce a corresponding variable for every vertex in V, and add a special variable z. For every edge $\{x,y\} \in E$, we impose a constraint $\mu_w(x,y,z)$.

Cuts in G are in one-to-one correspondence with assignments to I satisfying $z = 1$. In particular, a cut of size k corresponds to an assignment to I with value $k + 2(|E| - k) = 2|E| - k$. Any surjective assignment with $z = 0$ is of value at least $w > 2|E| - k$. Thus, solving I amounts to solving Max-Cut in G.

3.3 Approximability of maximising surjective VCSP

Although the VCSP is commonly defined with a minimisation objective, it is easy to see that, for exact solvability, its maximisation variant is essentially an identical problem: Minimising a $\mathbb{Q}$-valued function ϕ_I corresponds to maximising $-\phi_I$. When studying approximability, however, the two variants vastly differ (see [34] for a survey).

We focus on maximisation of the $\mathbb{Q}_{\geq 0}$-valued VCSP. This problem generalises the Max-CSP, in which the objective is to maximise the number of satisfied constraints; in particular, the Max-CSP corresponds to maximisation of the $\{0,1\}$-valued VCSP. The complexity of exactly maximising the $\mathbb{Q}_{\geq 0}$-valued VCSP was established by Thapper and Živný [42]. Raghavendra [37] showed that, assuming the unique games conjecture, the basic semidefinite programming relaxation achieves the optimal approximation ratio for the problem. In this section, we consider approximate maximisation of the *surjective* $\mathbb{Q}_{\geq 0}$-valued VCSP.

Definition 3.13. An *instance* $I = (V, D, \phi_I)$ of the Max-VCSP on domain D is given by a finite set of variables $V = \{x_1, \ldots, x_n\}$ and an objective function $\phi_I : D^n \to \mathbb{Q}_{\geq 0}$ expressed as a weighted sum of *constraints* over V, i.e.,

$$\phi_I(x_1, \ldots, x_n) = \sum_{i=1}^{q} w_i \cdot \gamma_i(\mathbf{x}_i), \tag{18}$$

where γ_i is a $\mathbb{Q}_{\geq 0}$-valued weighted relation, $w_i \in \mathbb{Q}_{\geq 0}$ is the *weight* and $\mathbf{x}_i \in V^{\mathrm{ar}(\gamma_i)}$ the *scope* of the ith constraint.

Given an instance I, the goal is to find an assignment $s : V \to D$ of domain labels to the variables that *maximises* ϕ_I. We denote the maximum value of the objective function by opt_I. For any $r \in [0,1]$, an assignment s is an *r-approximate solution* to I if $\phi_I(s) \geq r \cdot \mathrm{opt}_I$.

An assignment s is *surjective* if its image equals D. We denote the maximum objective value of surjective assignments by s-opt_I. For any $r \in (0,1]$, a surjective assignment s is an *r-approximate surjective solution* to I if $\phi_I(s) \geq r \cdot \text{s-opt}_I$.

We denote by $\text{Max-VCSP}_\mathrm{s}(\Gamma)$ the surjective Max-VCSP problem on instances over a language Γ.

Following the standard definitions, we say that Max-VCSP$_s(\Gamma)$ belongs to APX if, for some $r \in (0, 1]$, there exists a polynomial-time algorithm that finds an r-approximate surjective solution to every Max-VCSP$_s(\Gamma)$ instance. If such an algorithm exists for every $r < 1$, we say that the problem admits a polynomial-time approximation scheme (PTAS). Max-VCSP$_s(\Gamma)$ is APX-hard if there exists a PTAS reduction (an approximation-preserving reduction, see [15]) from every problem in APX to Max-VCSP$_s(\Gamma)$.

First, we prove that a polynomial-time algorithm for exactly maximising the $\mathbb{Q}_{\geq 0}$-valued VCSP over a language Γ implies a PTAS for Max-VCSP$_s(\Gamma)$. Second, we establish a complexity classification of Boolean languages in Theorem 3.16.

Lemma 3.14. *Let Γ be a $\mathbb{Q}_{\geq 0}$-valued language and $r, \epsilon \in \mathbb{R}$ such that $0 < \epsilon \leq r \leq 1$. There is a polynomial-time algorithm that, given a Max-VCSP instance $I = (V, D, \phi_I)$ over Γ and an r-approximate solution s to I, outputs an $(r - \epsilon)$-approximate* surjective *solution s' to I.*

Proof. Let $a_{\max}$ denote the maximum arity of weighted relations in Γ, and n the number of variables of I. If $n < \frac{r \cdot |D| \cdot a_{\max}}{\epsilon}$, we find an optimal surjective assignment to I by trying all $O(|D|^n)$ assignments.

Otherwise, we modify the given assignment s in order to obtain a surjective assignment s'. For any variable $x \in V$, let $B_x \subseteq [q]$ be the set of indices of constraints in whose scopes x appears. We define the contribution of x by

$$c(x) = \sum_{i \in B_x} w_i \cdot \gamma_i(s(\mathbf{x}_i)) \,. \tag{19}$$

It follows that the total contribution of all variables is at most $a_{\max} \cdot \phi_I(s)$.

Let U be a set of $|D|$ variables with the smallest contribution. We assign to them labels D bijectively. The resulting assignment s' is surjective, and it holds

$$\begin{aligned}
\phi_I(s') &\geq \phi_I(s) - \sum_{x \in U} c(x) && (20)\\
&\geq \phi_I(s) - \frac{|D|}{n} \cdot a_{\max} \cdot \phi_I(s) && (21)\\
&\geq \left(1 - \frac{|D|}{n} \cdot a_{\max}\right) \cdot r \cdot \mathrm{opt}_I && (22)\\
&\geq (r - \epsilon) \cdot \text{s-opt}_I \,. && (23)
\end{aligned}$$

□

Applying this lemma to an *optimal* solution to an Max-VCSP instance (i.e., $r = 1$) gives us the following corollary.

Corollary 3.15. *If the* Max-VCSP *over a $\mathbb{Q}_{\geq 0}$-valued language Γ is solvable in polynomial time, then there is a PTAS for* Max-VCSP$_s(\Gamma)$.

Finally, we classify Boolean $\mathbb{Q}_{\geq 0}$-valued languages by the complexity of the corresponding Max-VCSP$_s$. Since multimorphisms and the EDS property are defined in the context of minimisation, the following theorem applies them to language $-\Gamma$ instead of Γ (where $-\Gamma = \{-\gamma \mid \gamma \in \Gamma\}$ and $(-\gamma)(\mathbf{x}) = -\gamma(\mathbf{x})$).

Theorem 3.16. *Let Γ be a Boolean $\mathbb{Q}_{\geq 0}$-valued language. Then*

(1) Max-VCSP$_s(\Gamma)$ *is solvable exactly in polynomial time if $-\Gamma$ is EDS, or $-(\neg(\Gamma))$ is EDS, or $-\Gamma$ admits the $\langle \min, \max \rangle$ multimorphism;*

(2) otherwise it is NP-hard to solve exactly, but

(a) *it is in PTAS if* $-\Gamma$ *admits* $\langle c_0\rangle$ *or* $\langle c_1\rangle$,
(b) *and is APX-hard otherwise.*

Proof. Theorem 3.3 implies the case (1) and NP-hardness in the case (2). Case (2a) follows from Theorem 2.15 and Corollary 3.15. By Theorem 2.15, if $-\Gamma$ does not admit either of $\langle c_0\rangle$, $\langle c_1\rangle$ and $\langle \min, \max\rangle$, then Max-VCSP($\Gamma$) is NP-hard. The proof of Theorem 2.15 in [12] actually establishes that Max-VCSP(Γ) is APX-hard. By the approximation-preserving reduction in the proof of Lemma 2.5, this implies that Max-VCSP$_s$(Γ) is APX-hard as well. □

Theorem 3.16 generalises the result of Bach and Zhou [1, Theorem 16] in two respects. Firstly, we classify all $\mathbb{Q}_{\geq 0}$-valued languages as opposed to $\{0, 1\}$-valued languages. Secondly, we classify constraint languages as being in P, in PTAS, or being APX-hard; [1] only distinguishes admitting a PTAS versus being APX-hard. Finally, the main technical component of Theorem 3.16, Lemma 3.14, has a slightly simpler proof compared to [1].

4 HARDNESS PROOFS

Consider a Boolean language Γ over $D = \{0, 1\}$ that admits multimorphism $\langle c_0\rangle$ (the case of multimorphism $\langle c_1\rangle$ is symmetric), but does not admit any of the following multimorphisms: $\langle \min, \min\rangle$, $\langle \max, \max\rangle$, $\langle \min, \max\rangle$, $\langle \mathrm{Mn}, \mathrm{Mn}, \mathrm{Mn}\rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mj}\rangle$, $\langle \mathrm{Mj}, \mathrm{Mj}, \mathrm{Mn}\rangle$. Suppose that Γ is not EDS. We prove that VCSP$_s$(Γ) is NP-hard, i.e., Γ is globally s-intractable.

We start by showing that there exists a relation such that it is not invariant under sub and it can be added to Γ without changing the complexity of VCSP$_s$(Γ) (see Corollary 4.4). For finite Γ, this follows simply from Corollary 3.11 and Lemma 2.10, as there exists $\gamma \in \Gamma$ such that Feas(γ) or Opt(γ) is not invariant under sub. In general, however, a different argument is necessary. We prove it by showing that Γ contains weighted relations arbitrarily "similar" to a relation which is not invariant under sub, and that this relation may be thus added to Γ.

Definition 4.1. For any $\alpha \geq 1$, an r-ary weighted relation γ is α*-crisp* if its image $\gamma(D^r)$ lies in $[0, 1] \cup (\alpha, \infty]$. We will denote by $\mathrm{Round}_\alpha(\gamma)$ the r-ary relation defined as

$$\mathrm{Round}_\alpha(\gamma)(\mathbf{x}) = \begin{cases} 0 & \text{if } \gamma(\mathbf{x}) \in [0, 1], \\ \infty & \text{if } \gamma(\mathbf{x}) \in (\alpha, \infty]. \end{cases} \tag{24}$$

Note that an α-crisp weighted relation is α'-crisp for any $\alpha' \leq \alpha$. Moreover, a crisp weighted relation ρ is α-crisp for any $\alpha \geq 1$, and $\mathrm{Round}_\alpha(\rho) = \rho$.

Lemma 4.2. *Let* Γ *be a language and* ρ *a relation such that, for any* $\alpha \geq 1$, *there exists an* α*-crisp weighted relation* $\gamma \in \Gamma$ *with* $\mathrm{Round}_\alpha(\gamma) = \rho$. *Then* VCSP$_s$($\Gamma \cup \{\rho\}$) $\leq_p$ VCSP$_s$(Γ).

Proof. Let I be an instance of VCSP$_s$($\Gamma \cup \{\rho\}$) with k constraints that apply relation ρ. By scaling and adding rational constants to weighted relations in I, we ensure that all the assigned values are non-negative integers. Let M be an upper bound on the maximum value of a feasible solution to I (e.g., the weighted sum of the maximum finite values assigned by the constraints of I). Let $\gamma \in \Gamma$ be a $M \cdot (k + 1)$-crisp weighted relation such that $\mathrm{Round}_{M\cdot(k+1)}(\gamma) = \rho$. In each constraint applying relation ρ, we replace it by γ with weight $1/(k + 1)$, and thus obtain an instance of VCSP$_s$(Γ). Since γ is $M \cdot (k + 1)$-crisp, the value of any feasible assignment increases by at most $k/(k + 1) < 1$, and the value of any infeasible assignment becomes larger than M. □

Lemma 4.3. *Let* Γ *be a language such that it admits multimorphism* $\langle c_0\rangle$ *but is not EDS. Then there exists a relation* ρ *that is invariant under* c_0 *but not under* sub *and, for any* $\alpha \geq 1$, *there exists an* α*-crisp weighted relation* $\gamma \in \Gamma^*$ *with* $\mathrm{Round}_\alpha(\gamma) = \rho$.

Proof. We will show that for any $\alpha \geq 1$, there exists an α-crisp weighted relation $\gamma \in \Gamma^*$ such that $\mathrm{Round}_\alpha(\gamma)$ is a relation of arity at most 4 that is invariant under c_0 but not under sub. As there are only finitely many such relations, the claim of the lemma will follow.

Language Γ^* admits multimorphism $\langle c_0 \rangle$ as well but is not EDS; in particular, it is not α^{17}-EDS. Therefore, there exists an r-ary weighted relation $\gamma \in \Gamma^*$ and $\mathbf{u}, \mathbf{v} \in \mathrm{Feas}(\gamma)$ such that $\gamma(\mathbf{0}^r) = 0$ (as Γ^* is closed under adding rational constants) and

$$0 \leq \alpha^{17} \cdot (\gamma(\mathbf{u}) + \gamma(\mathbf{v})) < \gamma(\mathrm{sub}(\mathbf{u}, \mathbf{v})) . \tag{25}$$

We may assume that there are no distinct coordinates i, j where $u_i = u_j$ and $v_i = v_j$ (otherwise we identify them), and hence $r \leq 4$. As Γ^* is closed under scaling, we may also assume that $\gamma(\mathbf{u}), \gamma(\mathbf{v}) \leq 1$ and $\gamma(\mathrm{sub}(\mathbf{u}, \mathbf{v})) > \alpha^{17}$.

Let us consider, for any $0 \leq i \leq 16$, the intersection of the image $\gamma(D^r)$ with the interval $\left(\alpha^i, \alpha^{i+1}\right]$. Since $|D^r| \leq 2^4 = 16$, the intersection is empty for some i. Scaling γ by $1/\alpha^i$ then yields an α-crisp weighted relation $\gamma' \in \Gamma^*$ such that $\mathrm{Round}_\alpha(\gamma')$ is invariant under c_0 but not under sub, as $\gamma'(\mathbf{0}^r), \gamma'(\mathbf{u}), \gamma'(\mathbf{v}) \leq 1$ and $\gamma'(\mathrm{sub}(\mathbf{u}, \mathbf{v})) > \alpha$. ◻

Corollary 4.4. *Let Γ be a language such that it admits multimorphism $\langle c_0 \rangle$ but is not EDS. Then* $\mathrm{VCSP_s}(\Gamma \cup \{\rho\}) \leq_p \mathrm{VCSP_s}(\Gamma)$ *for some relation ρ that is invariant under c_0 but not under* sub.

Proof. By Lemmas 4.3 and 4.2, we have that $\mathrm{VCSP_s}(\Gamma^* \cup \{\rho\}) \leq_p \mathrm{VCSP_s}(\Gamma^*)$ for some relation ρ that is invariant under c_0 but not under sub. By Lemma 2.10, it holds $\mathrm{VCSP_s}(\Gamma^*) \leq_p \mathrm{VCSP_s}(\Gamma)$. ◻

We define weighted relations $\gamma_0 = \mathrm{Soft}(\rho_0)$, $\gamma_1 = \mathrm{Soft}(\rho_1)$, and $\gamma_= = \mathrm{Soft}(\rho_=)$; a binary relation $\rho_\leq = \{(0,0), (0,1), (1,1)\}$, and, for $r \in \{3,4\}$, an r-ary relation

$$A_r = \left\{ (x_1, \ldots, x_r) \in \{0,1\}^r \;\middle|\; \sum_{i=1}^{r} x_i \equiv 0 \pmod 2 \right\} . \tag{26}$$

Assuming that Γ does not admit polymorphism sub, we prove that $\mathrm{VCSP_s}(\Gamma)$ is NP-hard (see Lemma 4.11). The proof makes use of several sources of hardness. More specifically, we show that at least one of the following cases applies:

- $\mathrm{VCSP_s}(\mathrm{Feas}(\Gamma) \cup \mathrm{Opt}(\Gamma))$ is NP-hard (by the classification of $\{0,\infty\}$-valued languages, see Theorem 2.16).
- $\mathrm{VCSP}(\Gamma \cup \mathcal{C}_D)$ reduces to $\mathrm{VCSP_s}(\Gamma)$. In particular, it holds $\rho_\leq \in \Gamma^*$, which can be used to simulate constants (see Lemma 4.9). The intractability of $\mathrm{VCSP}(\Gamma \cup \mathcal{C}_D)$ follows from Theorem 2.14.
- The NP-hard Minimum Distance problem [45] reduces to $\mathrm{VCSP_s}(\Gamma)$. In particular, it holds $\{A_3, \gamma_0\} \subseteq \Gamma^*$ or $\{A_4, \gamma_=\} \subseteq \Gamma^*$; the reduction from the Minimum Distance problem to these languages is given in Lemma 4.10.

Before proving Lemma 4.11, we need a few auxiliary lemmas to establish the existence of certain weighted relations in Γ^*.

Lemma 4.5. *Let ρ be a relation invariant under c_0 but not under $\neg$. Then $\rho_0 \in \{\rho\}^*$ or $\rho_\leq \in \{\rho\}^*$.*

Proof. Let r denote the arity of ρ. There exists an r-tuple $\mathbf{u} \in \rho$ such that $\neg(\mathbf{u}) \notin \rho$. If $\mathbf{1}^r \notin \rho$, we obtain ρ_0 by identifying all coordinates of ρ. Otherwise, we obtain $\rho_\leq$ by identifying all coordinates where $u_i = 0$ and identifying all coordinates where $u_i = 1$. ◻

Lemma 4.6. *Let γ be a non-crisp weighted relation such that it admits multimorphism $\langle c_0 \rangle$. Then $\gamma_0 \in \{\gamma, \rho_0\}^*$. If in addition* $\mathrm{Feas}(\gamma)$ *and* $\mathrm{Opt}(\gamma)$ *are invariant under $\neg$, then $\gamma_= \in \{\gamma\}^*$.*

Proof. Let r denote the arity of γ. There exists an r-tuple $\mathbf{u}$ such that $\gamma(\mathbf{0}^r) < \gamma(\mathbf{u}) < \infty$. By 0-pinning at all coordinates where $u_i = 0$ and identifying all coordinates where $u_i = 1$, we obtain a unary weighted relation $\gamma' \in \{\gamma, \rho_0\}^*$ such that $\gamma'(0) < \gamma'(1) < \infty$. From it, we can obtain γ_0 by adding a rational constant and scaling, as $\gamma_0 = \frac{\gamma' - \gamma'(0)}{\gamma'(1) - \gamma'(0)}$.

If $\mathrm{Feas}(\gamma)$ and $\mathrm{Opt}(\gamma)$ are invariant under $\neg$, it holds $\gamma(\mathbf{1}^r) = \gamma(\mathbf{0}^r)$ and $\gamma(\mathbf{0}^r) < \gamma(\neg(\mathbf{u})) < \infty$. By identifying all coordinates where $u_i = 0$ and identifying all coordinates where $u_i = 1$, we obtain a binary weighted relation $\gamma' \in \{\gamma\}^*$. Consider $\gamma'' \in \{\gamma\}^*$ defined as $\gamma''(x, y) = \gamma'(x, y) + \gamma'(y, x)$. It holds $\gamma''(0, 0) = \gamma''(1, 1) < \gamma''(0, 1) = \gamma''(1, 0) < \infty$. From it, we can obtain $\gamma_=$ by adding a rational constant and scaling. □

Lemma 4.7. *Let ρ be a relation invariant under c_0, $\neg$, and* Mn, *but not under* sub. *Then $A_4 \in \{\rho\}^*$.*

Proof. Let ρ' be a smallest-arity relation in $\{\rho\}^*$ that is not invariant under sub, and denote its arity by r. As $\mathbf{0}^r \in \rho'$ and $\mathrm{Mn}(\mathbf{x}, \mathbf{y}, \mathbf{0}^r) = \mathbf{x} \oplus \mathbf{y}$, relation ρ' is closed under the $\oplus$ operation. Let $\mathbf{u}, \mathbf{v} \in \rho'$ be r-tuples such that $\mathrm{sub}(\mathbf{u}, \mathbf{v}) \notin \rho'$. There are no distinct coordinates i, j where $u_i = u_j$ and $v_i = v_j$, otherwise we could identify them to obtain an $(r-1)$-ary relation not invariant under sub. For any $a, b \in \{0, 1\}$, there is a coordinate i where $u_i = a$ and $v_i = b$, otherwise $\mathrm{sub}(\mathbf{u}, \mathbf{v})$ would be equal to $\neg(\mathbf{v})$, $\mathbf{u} \oplus \mathbf{v}$, $\mathbf{0}^r$, or $\mathbf{u}$ respectively, which would imply $\mathrm{sub}(\mathbf{u}, \mathbf{v}) \in \rho'$. Therefore, $r = 4$, and we may assume without loss of generality that $\mathbf{u} = (0, 0, 1, 1)$, $\mathbf{v} = (0, 1, 0, 1)$. As

$$
\begin{aligned}
\mathrm{sub}(\mathbf{u}, \mathbf{v}) &= (0, 0, 1, 0) \\
&= (0, 0, 0, 1) \oplus \mathbf{u} \\
&= (0, 1, 0, 0) \oplus (\mathbf{u} \oplus \mathbf{v}) \\
&= (1, 0, 0, 0) \oplus \neg(\mathbf{v}),
\end{aligned}
$$

it holds $(0, 0, 0, 1), (0, 0, 1, 0), (0, 1, 0, 0), (1, 0, 0, 0) \notin \rho'$. Since ρ' is closed under $\neg$, we have $\rho' = A_4$. □

Lemma 4.8. *Let ρ be a relation invariant under c_0 but not under* sub. *If ρ is invariant under* Mn, *then $A_3 \in \{\rho, \rho_0\}^*$. If ρ is invariant under* min *or* max, *then $\rho_\leq \in \{\rho, \rho_0\}^*$.*

Proof. Let ρ' be a smallest-arity relation in $\{\rho, \rho_0\}^*$ that is not invariant under sub, and denote its arity by r. Let $\mathbf{u}, \mathbf{v} \in \rho'$ be r-tuples such that $\mathrm{sub}(\mathbf{u}, \mathbf{v}) \notin \rho'$. There are no distinct coordinates i, j where $u_i = u_j$ and $v_i = v_j$, otherwise we could identify them to obtain an $(r-1)$-ary relation not invariant under sub. For any $b \in \{0, 1\}$, there is a coordinate i where $u_i = 1$ and $v_i = b$, otherwise $\mathrm{sub}(\mathbf{u}, \mathbf{v})$ would be equal to $\mathbf{0}^r$ or $\mathbf{u}$ respectively, which would imply $\mathrm{sub}(\mathbf{u}, \mathbf{v}) \in \rho'$. However, there is no coordinate i where $u_i = v_i = 0$, otherwise we could obtain an $(r-1)$-ary relation not invariant under sub by 0-pinning ρ' at coordinate i. Therefore, $r = 2$ or $r = 3$. If $r = 2$, we have $\rho_\leq \in \{\rho, \rho_0\}^*$, and ρ is not invariant under Mn (as neither is $\rho_\leq$).

If $r = 3$, we may assume without loss of generality that $\mathbf{u} = (0, 1, 1)$ and $\mathbf{v} = (1, 0, 1)$. Relation ρ is not invariant under min, otherwise it would hold $\min(\mathbf{u}, \mathbf{v}) = (0, 0, 1) \in \rho'$ and we could obtain a binary relation not invariant under sub by 0-pinning ρ' at the first coordinate. Similarly, relation ρ is not invariant under max, otherwise it would hold $\max(\mathbf{u}, \mathbf{v}) = (1, 1, 1) \in \rho'$ and we could obtain a binary relation not invariant under sub by identifying the first and third coordinate. Finally, assume that relation ρ is invariant under Mn. Then ρ' is also closed under the $\oplus$ operation, as $\mathrm{Mn}(\mathbf{x}, \mathbf{y}, \mathbf{0}^r) = \mathbf{x} \oplus \mathbf{y}$, and we have $\mathbf{u} \oplus \mathbf{v} = (1, 1, 0) \in \rho'$. Since $\mathrm{sub}(\mathbf{u}, \mathbf{v}) = (0, 1, 0) = (0, 0, 1) \oplus \mathbf{u} = (1, 1, 1) \oplus \mathbf{v} = (1, 0, 0) \oplus (\mathbf{u} \oplus \mathbf{v})$, it holds $(0, 0, 1), (1, 1, 1), (1, 0, 0) \notin \rho'$, and therefore $\rho' = A_3$. □

Lemma 4.9. *If $\rho_\leq \in \Gamma$, then* $\mathrm{VCSP}(\Gamma \cup C_D) \leq_p \mathrm{VCSP_s}(\Gamma)$.

Proof. For a given instance of $\mathrm{VCSP}(\Gamma \cup \{\rho_0, \rho_1\})$ with variables V, we construct an instance of $\mathrm{VCSP}_s(\Gamma)$ as follows: We introduce new variables y_0, y_1 and impose constraints $\rho_\leq(y_0, x)$, $\rho_\leq(x, y_1)$ for all $x \in V$ to ensure that $y_0 = 0, y_1 = 1$ in any feasible surjective assignment. Then we replace each constraint of the form $\rho_0(x)$ with $\rho_\leq(x, y_0)$ and each constraint of the form $\rho_1(x)$ with $\rho_\leq(y_1, x)$. □

Lemma 4.10. *Languages $\{A_3, \gamma_0\}$ and $\{A_4, \gamma_=\}$ are both s-intractable.*

Proof. First we show a reduction from the optimisation variant of the Minimum Distance problem, which is NP-hard [45], to $\mathrm{VCSP}_s(\{A_3, \gamma_0\})$. A problem instance is given as an $m \times n$ matrix H over the field $D = \{0, 1\}$, and the objective is to find a non-zero vector $\mathbf{x} = (x_1, \dots, x_n) \in D^n$ satisfying $H \cdot \mathbf{x} = \mathbf{0}^m$ with the minimum weight (i.e. $\sum_{i=1}^n x_i$).

Note that $\rho_0 = \mathrm{Opt}(\gamma_0)$, and therefore we may use relation ρ_0 as well (by Lemma 2.10). We construct a VCSP_s instance I as follows: Let $x_1, \dots, x_n$ be variables corresponding to the elements of the sought vector $\mathbf{x}$. The requirement $H \cdot \mathbf{x} = \mathbf{0}^m$ can be seen as a system of m linear equations, each in the form $\bigoplus_{i=1}^k x_{a_i} = 0$ for a set $\{a_1, \dots, a_k\} \subseteq [n]$ (the set may differ for each equation). We encode such an equation by introducing new variables $y_0, \dots, y_k$ and imposing constraints $\rho_0(y_0)$, $A_3(y_{i-1}, x_{a_i}, y_i)$ for all $i \in [k]$, and $\rho_0(y_k)$. These ensure that each variable y_j is assigned the value of the prefix sum $\bigoplus_{i=1}^j x_{a_i}$, and that the total sum equals 0. Finally, we encode the objective function of the Minimum Distance problem by imposing constraints $\gamma_0(x_1), \dots, \gamma_0(x_n)$.

Every vector $\mathbf{x} \in D^n$ satisfying $H \cdot \mathbf{x} = \mathbf{0}^m$ corresponds to a feasible assignment to I. If $\mathbf{x}$ is non-zero, the corresponding assignment is surjective, as at least one of variables $x_1, \dots, x_n$ gets label 1 and, for every equation, variable y_0 gets label 0. Conversely, if a feasible assignment to I is surjective, then it corresponds to a non-zero vector $\mathbf{x}$ (labelling all variables $x_1, \dots, x_n$ with 0 implies that all the prefix sums y_j equal 0 as well). The objective value of the assignment corresponding to a vector $\mathbf{x}$ equals the weight of $\mathbf{x}$, and hence finding an optimal surjective solution to I solves the Minimum Distance problem.

Finally, we show that the language $\{A_4, \gamma_=\}$ is s-intractable by a reduction from $\mathrm{VCSP}_s(\{A_3, \gamma_0\})$ to $\mathrm{VCSP}_s(\{A_4, \gamma_=\})$. Given an instance I, we construct an instance I' by introducing a new variable w and replacing each constraint of the form $A_3(x, y, z)$ with $A_4(x, y, z, w)$ and each constraint of the form $\gamma_0(x)$ with $\gamma_=(x, w)$. Any surjective assignment to I can be extended to a surjective assignment to I' of the same objective value by labelling w with 0. Conversely, consider a feasible surjective assignment s' to I'; we may assume $s'(w) = 0$ since language $\{A_4, \gamma_=\}$ admits multimorphism $\langle\neg\rangle$. Restricting s' to the variables of I gives us a surjective assignment to I of the same objective value. Note that if s' assigns label 1 to all the variables except w, its restriction will not be surjective; however, such s' violates constraints $\rho_0(y_0)$ and thus is not feasible. □

Lemma 4.11. *Let Γ be a language such that it admits multimorphism $\langle c_0 \rangle$ but not polymorphism* sub. *If $\Gamma \cup C_D$ is intractable, then $\mathrm{VCSP}_s(\Gamma)$ is NP-hard.*

Proof. Let $\Phi = \mathrm{Feas}(\Gamma) \cup \mathrm{Opt}(\Gamma) \subseteq \Gamma^*$. Suppose that Φ does not admit any of the following polymorphisms: min, max, Mn, and Mj. By the classification of $\{0, \infty\}$-valued languages (see Theorem 2.16), Φ is s-intractable. Hence, $\mathrm{VCSP}_s(\Gamma)$ is NP-hard by Lemma 2.10. In the rest of the proof, we assume that Φ admits at least one of polymorphisms min, max, Mn, and Mj. Note that Φ admits polymorphism c_0 but not polymorphism sub. Since $\min(x, y) = \mathrm{Mj}(x, y, 0)$, we may assume that Φ admits at least one of polymorphisms min, max, and Mn.

Suppose that Φ admits polymorphism $\neg$. Then it does not admit min, as $\mathrm{sub}(x, y) = \min(x, \neg y)$, nor it admits max, as $\min(x, y) = \neg \max(\neg x, \neg y)$. Therefore, Φ admits polymorphism Mn. If Γ is crisp, then language $\Gamma \cup \{\rho_0, \rho_1\}$ admits multimorphism $\langle \mathrm{Mn}, \mathrm{Mn}, \mathrm{Mn} \rangle$ and thus is tractable by Theorem 2.14, which contradicts an assumption of the lemma. Hence, Γ is not crisp. By Lemmas 4.7 and 4.6, we have $\{A_4, \gamma_=\} \subseteq \Gamma^*$. Therefore, $\mathrm{VCSP}_s(\Gamma)$ is NP-hard by Lemma 4.10.

If Φ does not admit polymorphism $\neg$, then, by Lemma 4.5, we have $\rho_0 \in \Gamma^*$ or $\rho_\leq \in \Gamma^*$. If $\rho_\leq \in \Gamma^*$, $\mathrm{VCSP_s}(\Gamma)$ is NP-hard by Lemma 4.9 and we are done; in the rest of the proof we assume that $\rho_\leq \notin \Gamma^*$ and hence $\rho_0 \in \Gamma^*$. If Φ admits polymorphism min or max, we get $\rho_\leq \in \Gamma^*$ by Lemma 4.8, which is a contradiction. Therefore, Φ admits Mn, and thus Γ is not crisp (by the same argument as in the previous paragraph). By Lemmas 4.8 and 4.6, we have $\{A_3, \gamma_0\} \subseteq \Gamma^*$. Therefore, $\mathrm{VCSP_s}(\Gamma)$ is NP-hard by Lemma 4.10. □

Theorem 4.12. *Let Γ be a language such that it is not EDS, $\neg(\Gamma)$ is not EDS, and Γ does not admit any of the following multimorphisms:* $\langle\min,\min\rangle$, $\langle\max,\max\rangle$, $\langle\min,\max\rangle$, $\langle\mathrm{Mn},\mathrm{Mn},\mathrm{Mn}\rangle$, $\langle\mathrm{Mj},\mathrm{Mj},\mathrm{Mj}\rangle$, $\langle\mathrm{Mj},\mathrm{Mj},\mathrm{Mn}\rangle$. *Then* $\mathrm{VCSP_s}(\Gamma)$ *is NP-hard.*

Proof. If Γ does not admit at least one of multimorphisms $\langle c_0\rangle$ and $\langle c_1\rangle$, it is intractable by Theorem 2.14, and hence $\mathrm{VCSP_s}(\Gamma)$ is NP-hard by Lemma 2.5. Language $\Gamma \cup C_D$ is, by the same theorem, intractable. We may assume that Γ admits multimorphism $\langle c_0\rangle$; if it does not, we consider $\neg(\Gamma)$ instead. By Corollary 4.4 and Lemma 4.11, $\mathrm{VCSP_s}(\Gamma)$ is NP-hard. □

5 TRACTABILITY OF EDS LANGUAGES

We prove that EDS languages are globally s-tractable by a reduction to a generalised variant of the Min-Cut problem. The problem is defined in Section 5.1, its tractability is established in Section 5.2, and the reduction is stated in Section 5.3.

5.1 Generalised Min-Cut problem

Let V be a finite set. A *set function* on V is a function $\gamma : 2^V \to \mathbb{Q}_{\geq 0} \cup \{\infty\}$ with $\gamma(\emptyset) = 0$.

Definition 5.1. A set function $\gamma : 2^V \to \mathbb{Q}_{\geq 0} \cup \{\infty\}$ is *symmetric* if $\gamma(X) = \gamma(V \setminus X)$ for all $X \subseteq V$; it is *increasing* if $\gamma(X) \leq \gamma(Y)$ for all $X \subseteq Y \subseteq V$; it is *superadditive* if

$$\gamma(X) + \gamma(Y) \leq \gamma(X \cup Y) \tag{27}$$

for all disjoint $X, Y \subseteq V$; it is *posimodular* if

$$\gamma(X) + \gamma(Y) \geq \gamma(X \setminus Y) + \gamma(Y \setminus X) \tag{28}$$

for all $X, Y \subseteq V$; and it is *submodular* if

$$\gamma(X) + \gamma(Y) \geq \gamma(X \cap Y) + \gamma(X \cup Y) \tag{29}$$

for all $X, Y \subseteq V$.

Note that any superadditive set function is also increasing, as for all $X \subseteq Y \subseteq V$ it holds $\gamma(X) \leq \gamma(X) + \gamma(Y \setminus X) \leq \gamma(Y)$ by superadditivity. In the case of symmetric set functions, submodularity implies posimodularity, as

$$\begin{aligned} \gamma(X) + \gamma(Y) &= \gamma(X) + \gamma(V \setminus Y) && (30)\\ &\geq \gamma(X \cap (V \setminus Y)) + \gamma(X \cup (V \setminus Y)) && (31)\\ &= \gamma(X \setminus Y) + \gamma(V \setminus (Y \setminus X)) && (32)\\ &= \gamma(X \setminus Y) + \gamma(Y \setminus X)\,. && (33) \end{aligned}$$

and, similarly, posimodularity implies submodularity.

Example 5.2. Let V be a finite set and $T \subseteq V$ a non-empty subset. We define a set function γ on V by $\gamma(X) = 1$ if $T \subseteq X$ and $\gamma(X) = 0$ otherwise. Intuitively, this corresponds to a soft NAND constraint imposed on variables T. The set function γ is superadditive, and hence also increasing.

We now formally define the Min-Cut problem.

Definition 5.3. An instance of the *Min-Cut* (MC) problem is given by an undirected graph $G = (V, E)$ with edge weights $w : E \to \mathbb{Q}_{\geq 0} \cup \{\infty\}$. The objective function g of the MC problem is a set function on V defined by

$$g(X) = \sum_{|X \cap \{u,v\}|=1} w(u, v) . \tag{34}$$

Function g is a well-known example of a submodular function. Since it is symmetric, it is also posimodular.

A *solution* to the MC problem is a set X such that $\emptyset \subsetneq X \subsetneq V$. Note that a cut $(X, V \setminus X)$ corresponds to two solutions, namely X and $V \setminus X$. An *optimal* solution is a solution with the minimum objective value among all solutions. A *minimal* optimal solution is an optimal solution with no proper subset being an optimal solution.

Note that any two different minimal optimal solutions X, Y must be disjoint, otherwise $X \setminus Y$ or $Y \setminus X$ would be a smaller optimal solution (by the posimodularity of g).

Although the definition allows infinite weight edges, those can be easily eliminated by identifying their endpoints, and so we may assume that all edge weights are finite. Edges with weight 0 are conventionally disregarded.

Finally, we define the Generalised Min-Cut problem, which further generalises the problem introduced in [43].

Definition 5.4. An instance J of the *Generalised Min-Cut* (GMC) problem is given by an undirected graph $G = (V, E)$ with edge weights $w : E \to \mathbb{Q}_{\geq 0} \cup \{\infty\}$, and an oracle defining a superadditive set function f on V. The objective function the GMC problem is a set function on V defined by $J(X) = f(X) + g(X)$, where g is the objective function of the underlying Min-Cut problem on G.

A *solution* to the GMC problem is a set X such that $\emptyset \subsetneq X \subsetneq V$. An *optimal* solution is a solution with the minimum objective value among all solutions. We denote this minimum objective value by λ. For any $\alpha \geq 1$, an α*-optimal* solution is a solution X such that $J(X) \leq \alpha\lambda$.

We show in Theorem 5.11 that, in the case of $0 < \lambda < \infty$ and a fixed $\alpha \geq 1$, there are only polynomially many α-optimal solutions and they can be found in polynomial time.

5.2 Tractability of the Generalised Min-Cut problem

In this section, we present a polynomial-time algorithm that solves the Generalised Min-Cut problem. We assume that $w(u, v) \in \mathbb{Q}_{>0}$ for all edges (u, v).

Lemma 5.5. *There is a polynomial-time algorithm that, given an instance J of the GMC problem, either finds a solution X with $J(X) = \lambda = 0$, or determines that $\lambda = \infty$, or determines that $0 < \lambda < \infty$.*

Proof. A solution X with $J(X) = f(X) + g(X) = 0$ satisfies $f(X) = g(X) = 0$, and hence it does not cut any edge. Since the set function f is increasing, we may assume that X is a single connected component. The algorithm simply tries each connected component as a solution, which takes a linear number of queries to the oracle for f.

The case of $\lambda = \infty$ occurs only if $f(X) = \infty$ for all solutions X. Since f is increasing, it is sufficient to check all solutions of size 1. □

In view of Lemma 5.5, we can assume that $0 < \lambda < \infty$. Our goal is to show that, for a given $\alpha \geq 1$, all α-optimal solutions to a GMC instance can be found in polynomial time. This is proved in Theorem 5.11; before that we need to prove several auxiliary lemmas on properties of the MC and GMC problems.

Lemma 5.6. *For any instance J of the GMC problem on a graph $G = (V, E)$ and any non-empty set $V' \subseteq V$, there is an instance J' on the induced subgraph $G[V']$ that preserves the objective value of all solutions $X \subsetneq V'$. In particular, any α-optimal solution X of J such that $X \subsetneq V'$ is α-optimal for J' as well.*

Proof. Edges with exactly one endpoint in V' need to be taken into account separately because they do not appear in the induced subgraph. We accomplish that by defining the new set function f' by

$$f'(X) = f(X) + \sum_{u \in X} \sum_{v \in V \setminus V'} w(u, v) \tag{35}$$

for all $X \subseteq V'$. By the construction, f' is superadditive, and the objective value $J'(X)$ for any $X \subsetneq V'$ equals $J(X)$.

Note that the minimum objective value for J' is greater than or equal to the minimum objective value for J. Therefore, any solution $X \subsetneq V'$ that is α-optimal for J is also α-optimal for J'. □

Lemma 5.7. *Let X be an optimal solution to an instance of the GMC problem over vertices V with $\lambda < \infty$, and Y a minimal optimal solution to the underlying MC problem. Then $X \subseteq Y$, $X \subseteq V \setminus Y$, or X is an optimal solution to the underlying MC problem.*

Proof. Assume that $X \nsubseteq Y$ and $X \nsubseteq V \setminus Y$. If $Y \subseteq X$, we have $f(Y) \leq f(X)$ as f is increasing, and hence $f(Y) + g(Y) \leq f(X) + g(X) < \infty$. Therefore, Y is optimal for the GMC problem and X is optimal for the MC problem. In the rest, we assume that $Y \nsubseteq X$.

By the posimodularity of g we have $g(X) + g(Y) \geq g(X \setminus Y) + g(Y \setminus X)$. Since $Y \setminus X$ is a proper non-empty subset of Y, it holds $g(Y \setminus X) > g(Y)$, and hence $g(X) > g(X \setminus Y)$. But then $f(X) + g(X) > f(X \setminus Y) + g(X \setminus Y)$ as $\infty > f(X) \geq f(X \setminus Y)$. Set $X \setminus Y$ is non-empty, and therefore contradicts the optimality of X. □

The following lemma relates the number of optimal solutions and the number of minimal optimal solutions to the MC problem. Note that this bound is tight for (unweighted) paths and cycles with at most one path attached to each vertex.

Lemma 5.8. *For any instance of the MC problem on a connected graph with $n \geq 2$ vertices and p minimal optimal solutions, there are at most $p(p-1) + 2(n-p)$ optimal solutions.*

We prove the lemma by induction on n, closely following the proof that establishes the cactus representation of minimum cuts in [21]. We note that the cactus representation could be applied directly to obtain a weaker bound of $p(p-1) + O(n)$ but we do not know how to achieve the exact bound using it.

Proof. For $n = 2$, the lemma holds as there are exactly two solutions and both are minimal optimal. Assume $n \geq 3$. We denote the number of optimal solutions by s. A solution X is called a *star* if $|X| = 1$ or $|X| = n - 1$, otherwise it is called *proper*.

First we consider the case where every optimal solution is a star. Let us denote the minimum cuts by $(\{v_1\}, V \setminus \{v_1\}), \ldots, (\{v_h\}, V \setminus \{v_h\})$. If $h = 1$, then we have $s = p = 2$ and the bound holds. Otherwise, there are $2h$ optimal solutions but only h of them are minimal (i.e., $\{v_1\}, \ldots, \{v_h\}$). Hence,

$$p(p-1) + 2(n-p) = 2h + (h-1) \cdot (h-2) - 2 + 2(n-h) \tag{36}$$

$$\geq 2h = s \tag{37}$$

as it holds $n \geq h \geq 2$ and $n \geq 3$.

From now on we assume that there is a proper optimal solution, and hence $n \geq 4$. We say that solutions X, Y *cross* if none of $X \setminus Y$, $Y \setminus X$, $X \cap Y$, $V \setminus (X \cup Y)$ is empty. Note that only proper solutions might cross. If every proper optimal solution is crossed by some optimal solution, then the graph is a cycle with edges of equal weight [21, Lemma 7.1.3]. In that case, there are $n(n-1)$ optimal solutions (all sets of contiguous vertices except for $\emptyset$ and V) and n minimal optimal solutions (all singletons), and therefore the bound holds.

Finally, assume that there is a proper optimal solution that is not crossed by any optimal solution, and denote the corresponding minimum cut by (V_1, V_2). For any optimal solution X, it must hold either $X \subseteq V_1$, $V_1 \subseteq X$, $X \subseteq V_2$, or $V_2 \subseteq X$. For $i \in \{1, 2\}$, let G_i be the result of shrinking V_i into a new vertex t_i so that the weight of any edge (t_i, v) for $v \in V \setminus V_i$ equals the sum of weights of edges (u, v) for $u \in V_i$. Denote by n_i, p_i, and s_i the number of vertices, minimal optimal solutions, and optimal solutions to G_i. It holds $n = n_1 + n_2 - 2$. Consider any solution X' of G_i: If $t_i \notin X'$, it corresponds to a solution $X = X'$ of the original graph G; otherwise it corresponds to $X = X' \setminus \{t_i\} \cup V_i$. In both cases, the objective values of X' and X in their respective problem instances are equal. Therefore, any optimal solution X of G such that $X \subseteq V_2$ or $V_1 \subseteq X$ corresponds to an optimal solution to G_1, and any optimal solution to G such that $X \subseteq V_1$ or $V_2 \subseteq X$ corresponds to an optimal solution in G_2. Hence, $p = p_1 + p_2 - 2$ and $s = s_1 + s_2 - 2$, as only solutions V_1 and V_2 satisfy both conditions simultaneously. By the inductive hypothesis, we get

$$
\begin{aligned}
p(p-1) + 2(n-p) &= p_1(p_1 - 1) + 2(n_1 - p_1) + p_2(p_2 - 1) + 2(n_2 - p_2) \\
&\quad + 2(p_1 - 2) \cdot (p_2 - 2) - 2 && (38) \\
&\geq s_1 + s_2 - 2 + 2(p_1 - 2) \cdot (p_2 - 2) && (39) \\
&\geq s && (40)
\end{aligned}
$$

as it holds $p_1, p_2 \geq 2$. □

Lemma 5.9. *For any instance of the GMC problem on n vertices with $0 < \lambda < \infty$, the number of optimal solutions is at most $n(n-1)$. There is an algorithm that finds all of them in polynomial time.*

Note that the bound of $n(n-1)$ optimal solutions precisely matches the known upper bound of $\binom{n}{2}$ for the number of minimum cuts [29]; the bound is tight for cycles.

Proof. Let $t(n)$ denote the maximum number of optimal solutions for such instances on n vertices. We prove the bound by induction on n. If $n = 1$, there are no solutions and hence $t(1) = 0$. For $n \geq 2$, let $Y_1, \ldots, Y_p$ be the minimal optimal solutions to the underlying MC problem. As there exists at least one minimum cut and the minimal optimal solutions are all disjoint, it holds $2 \leq p \leq n$.

First, suppose that $\bigcup Y_i = V$. By Lemma 5.7, any optimal solution to the GMC problem is either a proper subset of some Y_i or an optimal solution to the underlying MC problem. Restricting solutions to a proper subset of Y_i is, by Lemma 5.6, equivalent to considering a GMC problem instance on vertices Y_i, and hence the number of such optimal solutions is bounded by $t(|Y_i|) \leq |Y_i| \cdot (|Y_i| - 1)$. Since it holds $\sum |Y_i| = n$ and $|Y_i| \geq 1$ for all i, the sum $\sum |Y_i| \cdot (|Y_i| - 1)$ is maximised when $p - 1$ of the sets Y_i are singletons and the size of the remaining one equals $n - p + 1$. If the graph is connected, then, by Lemma 5.8, there are at most $p(p-1) + 2(n-p)$ optimal solutions to the underlying MC problem. Adding these upper bounds we get

$$\begin{aligned}
& p(p-1) + 2(n-p) + \sum_{i=1}^{p} |Y_i| \cdot (|Y_i| - 1) && (41)\\
& \leq p(p-1) + 2(n-p) + (p-1) \cdot 1 \cdot 0 + (n-p+1) \cdot (n-p) && (42)\\
& = n(n-1) - 2(p-2) \cdot (n-p) && (43)\\
& \leq n(n-1)\,. && (44)
\end{aligned}$$

If the graph is disconnected, the sets $Y_1, \ldots, Y_p$ are precisely its connected components. The optimal solutions to the underlying MC problem are precisely unions of connected components (with the exception of $\emptyset$ and V), which means that there can be exponentially many of them. However, only the sets $Y_1, \ldots, Y_p$ themselves can be optimal solutions to the GMC problem: We have $0 < \lambda \leq f(Y_i) + g(Y_i) = f(Y_i)$. Since f is superadditive, it holds

$$f(Y_{i_1} \cup \cdots \cup Y_{i_k}) \geq f(Y_{i_1}) + \cdots + f(Y_{i_k}) \geq k\lambda \tag{45}$$

for any distinct $i_1, \ldots, i_k$, and hence no union of two or more connected components can be an optimal solution to the GMC problem. This gives us an upper bound of $p \leq p(p-1) + 2(n-p)$, and the rest follows as in the previous case.

Finally, suppose that $\bigcup Y_i \neq V$, and hence the graph is connected. Let $Z = V \setminus \bigcup Y_i$. By Lemma 5.7, any optimal solution to the GMC problem is a proper subset of some Y_i, a proper subset of Z, set Z itself, or an optimal solution to the underlying MC problem. Similarly as before, we get an upper bound of

$$\begin{aligned}
& p(p-1) + 2(n-p) + \sum_{i=1}^{p} |Y_i| \cdot (|Y_i| - 1) + |Z| \cdot (|Z| - 1) + 1 && (46)\\
& \leq p(p-1) + 2(n-p) + p \cdot 1 \cdot 0 + (n-p) \cdot (n-p-1) + 1 && (47)\\
& = n(n-1) - 2(p-1) \cdot (n-p) + 1 && (48)\\
& \leq n(n-1)\,. && (49)
\end{aligned}$$

Using a procedure generating all minimum cuts [46], it is straightforward to turn the above proof into a recursive algorithm that finds all optimal solutions in polynomial time. □

Lemma 5.10. *Let $\alpha, \beta \geq 1$. Let X be an α-optimal solution to an instance J of the GMC problem over vertices V with $0 < \lambda < \infty$, and Y an optimal solution to the underlying MC problem. If $g(Y) < \lambda/\beta$, then*

$$J(X \setminus Y) + J(X \cap Y) < \left(\alpha + \frac{2}{\beta}\right)\lambda\,; \tag{50}$$

if $g(Y) \geq \lambda/\beta$, then X is an $\alpha\beta$-optimal solution to the underlying MC problem.

Proof. If $g(Y) \geq \lambda/\beta$, it holds $g(X) \leq J(X) \leq \alpha\lambda \leq \alpha\beta \cdot g(Y)$, and hence X is an $\alpha\beta$-optimal solution to the underlying MC problem. In the rest we assume that $g(Y) < \lambda/\beta$.

Since g is posimodular, we have

$$g(X) + g(Y) \geq g(X \setminus Y) + g(Y \setminus X) \tag{51}$$

$$g(Y) + g(Y \setminus X) \geq g(X \cap Y) + g(\emptyset)\,, \tag{52}$$

and hence

$$g(X) + 2g(Y) \geq g(X \setminus Y) + g(X \cap Y)\,. \tag{53}$$

By superadditivity of f, it holds $f(X) \geq f(X \setminus Y) + f(X \cap Y)$. The claim then follows from the fact that $f(X) + g(X) + 2g(Y) < (\alpha + 2/\beta)\lambda$. □

Finally, we prove that α-optimal solutions to the GMC problem can be found in polynomial time.

Theorem 5.11. *For any instance J of the GMC problem on n vertices with $0 < \lambda < \infty$ and $\alpha \in \mathbb{Z}_{\geq 1}$, the number of α-optimal solutions is at most $n^{20\alpha-15}$. There is an algorithm that finds all of them in polynomial time.*

Note that for a cycle on n vertices, the number of α-optimal solutions to the MC problem is $\Theta(n^{2\alpha})$, and thus the exponent in our bound is asymptotically tight in α.

Proof. Let $\beta \in \mathbb{Z}_{\geq 3}$ be a parameter. Throughout the proof, we relax the integrality restriction on α and require only that $\alpha\beta$ is an integer. For $\alpha = 1$, the claim follows from Lemma 5.9, therefore we assume $\alpha \geq 1 + 1/\beta$ in the rest of the proof.

Define a linear function ℓ by

$$\ell(x) = \frac{2(\beta+1)}{\beta-2} \cdot (\beta x - 3) \ . \tag{54}$$

We prove that the number of α-optimal solutions is at most $n^{\ell(\alpha)}$; taking $\beta = 4$ then gives the claimed bound. Function ℓ was chosen as a slowest-growing function satisfying the following properties required in this proof: It holds $\ell(x) + \ell(y) \leq \ell(x + y - 3/\beta)$ for any x, y, and $\ell(x) \geq 2\beta x$ for any $x \geq 1 + 1/\beta$.

We prove the bound by induction on $n + \alpha\beta$. As it trivially holds for $n \leq 2$, we assume $n \geq 3$ in the rest of the proof. Let Y be an optimal solution to the underlying MC problem with $k = |Y| \leq n/2$. If $g(Y) \geq \lambda/\beta$ then, by Lemma 5.10, any α-optimal solution to the GMC problem is an $\alpha\beta$-optimal solution to the underlying MC problem. Since $g(Y) \geq \lambda/\beta > 0$, the graph is connected, and hence there are at most

$$2^{2\alpha\beta}\binom{n}{2\alpha\beta} \leq n^{2\alpha\beta} \leq n^{\ell(\alpha)} \tag{55}$$

such solutions by [29]. (In detail, [29, Theorem 6.2] shows that the number of $\alpha\beta$-optimal cuts in an n-vertex graph is $2^{2\alpha\beta-1}\binom{n}{2\alpha\beta}$, and every cut corresponds to two solutions.)

From now on we assume that $g(Y) < \lambda/\beta$, and hence inequality (50) holds. Upper bounds in this case may be quite loose; in particular, we use the following inequalities:

$$(k/n)^{\ell(\alpha)} \leq (k/n)^{\ell(1+1/\beta)} = (k/n)^{2(\beta+1)} \leq (k/n)^8 \leq (k/n)(1/2)^7 = k/128n \tag{56}$$

$$(1/n)^{2\beta} \leq (1/n)^6 \leq (1/n)(1/3)^5 < 1/128n\,. \tag{57}$$

Consider any α-optimal solution to the GMC problem X.

If $X \subsetneq Y$, then, by Lemma 5.6, X is an α-optimal solution to an instance on vertices Y. By the induction hypothesis, there are at most $k^{\ell(\alpha)} \leq (k/128n) \cdot n^{\ell(\alpha)}$ such solutions.

Similarly, if $X \subsetneq V \setminus Y$, then X is an α-optimal solution to an instance on vertices $V \setminus Y$, and there are at most

$$(n-k)^{\ell(\alpha)} = (1-k/n)^{\ell(\alpha)} \cdot n^{\ell(\alpha)} \leq (1-k/n) \cdot n^{\ell(\alpha)} \tag{58}$$

such solutions.

If $Y \subsetneq X$, then $X \setminus Y$ is an $(\alpha - 1 + 2/\beta)$-optimal solution on vertices $V \setminus Y$ by (50) and the fact that $J(X \cap Y) \geq \lambda$. Similarly, if $V \setminus Y \subsetneq X$, then $X \cap Y$ is an $(\alpha - 1 + 2/\beta)$-optimal solution on vertices Y. In either case, we bound the number of such solutions depending on the value of α: For $\alpha < 2 - 2/\beta$, there are trivially none; for $\alpha = 2 - 2/\beta$, Lemma 5.9 gives a bound of $n(n-1) \leq n^{\ell(\alpha)-2\beta}$; and for $\alpha > 2 - 2/\beta$ we get an upper bound of $n^{\ell(\alpha-1+2/\beta)} \leq n^{\ell(\alpha)-2\beta}$ by the induction hypothesis. The number of solutions is thus at most $n^{\ell(\alpha)-2\beta} \leq (1/128n) \cdot n^{\ell(\alpha)}$ for any α.

Finally, we consider X such that $\emptyset \subsetneq X \setminus Y \subsetneq V \setminus Y$ and $\emptyset \subsetneq X \cap Y \subsetneq Y$, i.e., $X \setminus Y$ and $X \cap Y$ are solutions on vertices $V \setminus Y$ and Y respectively. Let i be the integer for which

$$\left(1 + \frac{i}{\beta}\right) \lambda \leq J(X \cap Y) < \left(1 + \frac{i+1}{\beta}\right) \lambda \,. \tag{59}$$

Then, by (50), it holds $J(X \setminus Y) < (\alpha - 1 - (i-2)/\beta)\lambda$. Therefore, $X \cap Y$ is a $(1 + (i+1)/\beta)$-optimal solution on vertices Y and $X \setminus Y$ is an $(\alpha - 1 - (i-2)/\beta)$-optimal solution on vertices $V \setminus Y$. Since $0 \leq i \leq (\alpha - 2)\beta + 1$, we can bound the number of such solutions by the induction hypothesis as at most

$$k^{\ell\left(1+\frac{i+1}{\beta}\right)} \cdot (n-k)^{\ell\left(\alpha-1-\frac{i-2}{\beta}\right)} \leq \left(\frac{k}{n}\right)^{\ell\left(1+\frac{i+1}{\beta}\right)} \cdot n^{\ell\left(1+\frac{i+1}{\beta}\right)+\ell\left(\alpha-1-\frac{i-2}{\beta}\right)} \tag{60}$$

$$\leq \left(\frac{k}{n}\right)^{2(\beta+1)} \cdot \frac{1}{2^i} \cdot n^{\ell(\alpha)} \,, \tag{61}$$

which is at most $2 \cdot (k/128n) \cdot n^{\ell(\alpha)}$ in total for all i.

By adding up the bounds we get that the number of α-optimal solutions is at most $n^{\ell(\alpha)}$. A polynomial-time algorithm that finds the α-optimal solutions follows from the above proof using a procedure generating all $\alpha\beta$-optimal cuts [46]. □

Remark 4. For our reduction from the $\mathrm{VCSP_s}$ over EDS languages, we need to find all α-optimal solutions to the GMC problem. However, if one is only interested in a single optimal solution, the presented algorithm can be easily adapted to an even more general problem.

Let f, g be set functions on V given by an oracle such that $f : 2^V \to \mathbb{Q}_{\geq 0} \cup \{\infty\}$ is increasing and $g : 2^V \to \mathbb{Q}_{\geq 0}$ satisfies the posimodularity and submodularity inequalities for intersecting pairs of sets (i.e. sets X, Y such that neither of $X \cap Y, X \setminus Y, Y \setminus X$ is empty). The objective is to minimise the sum of f and g.

The case when the optimum value $\lambda = \infty$ can be recognised by checking all solutions of size 1. Assuming $\lambda < \infty$, note that the proof of Lemma 5.7 works even for this more general problem. Let Y be a minimal optimal solution to g. It follows that there is an optimal solution X to $f + g$ such that $X \subseteq Y$, $X \subseteq V \setminus Y$, or X is itself a *minimal* optimal solution to g (as f is increasing). We can find all minimal optimal solutions to g in polynomial time [36, Theorem 10.11]. Restricting f, g to a subset of V preserves the required properties, and hence we can recursively solve the problem on Y and $V \setminus Y$. Therefore, an optimal solution to $f + g$ can be found in polynomial time.

5.3 Reduction to the Generalised Min-Cut problem

At the heart of our reduction is an observation that EDS weighted relations can be approximated by instances of the Generalised Min-Cut problem. We define this notion of approximability in Definition 5.14. In Theorem 5.15, we show how to approximate any EDS weighted relation with a constant factor. However, that construction does not yield a sufficient bound on the approximation factor; we present it only in order to provide some intuition for the more opaque construction in Theorem 5.17. Using that, we establish the global s-tractability of EDS languages in Theorem 5.18.

In this section, we equate weighted relations admitting multimorphism $\langle c_0 \rangle$ with set functions; the correspondence is formally stated in the following definition. Note that we may without loss of generality assume that the minimum assigned value equals 0, as adding a rational constant to a weighted relation preserves tractability.

Definition 5.12. Let γ be an r-ary weighted relation such that, for any r-tuple $\mathbf{x}$, $\gamma(\mathbf{x}) \geq \gamma(\mathbf{0}^r) = 0$. The *corresponding set function* γ' on $[r]$ is defined by $\gamma'(X) = \gamma(\mathbf{x})$ where $x_i = 1 \iff i \in X$.

The definition of α-EDS weighted relations then translates into the following:

Definition 5.13. For any $\alpha \geq 1$, a set function γ on V is *α-EDS* if, for every $X, Y \subseteq V$, it holds

$$\alpha \cdot (\gamma(X) + \gamma(Y)) \geq \gamma(X \setminus Y)\,. \tag{62}$$

Remark 5. Inequality (12) could be modified so that (62) becomes symmetric, say

$$\alpha \cdot (\gamma(X) + \gamma(Y)) \geq \gamma(X \setminus Y) + \gamma(Y \setminus X)\,. \tag{63}$$

It is easy to see that, although the set of α-EDS weighted relations for a fixed α would be different, this change would not affect the set of EDS weighted relations. We opt for the shorter, albeit asymmetric, definition.

Definition 5.14. Let J be an instance of the GMC problem on vertices V and γ a set function on V. For any $\alpha \geq 1$, we say that J *α-approximates* γ if, for all $X \subseteq V$,

$$J(X) \leq \gamma(X) \leq \alpha \cdot J(X)\,. \tag{64}$$

A set function is *α-approximable* if there exists a GMC instance that α-approximates it, and it is *approximable* if it is α-approximable for some $\alpha \geq 1$.

Theorem 5.15. *Any α-EDS set function is approximable.*

Proof. Let γ be an α-EDS set function on $[n]$ and γ' the corresponding n-ary weighted relation. By Corollary 3.11, both $\mathrm{Feas}(\gamma')$ and $\mathrm{Opt}(\gamma')$ are essentially downsets. The rest of the proof relies only on this property and does not depend on the value of α. The intuition behind our construction is that a downset can be represented by a superadditive function on $[n]$, and binary equality relations can be represented by edges.

There exist $A_{\mathrm{Feas}}, A_{\mathrm{Opt}} \subseteq [n]$, downsets $\mathcal{S}_{\mathrm{Feas}} \subseteq 2^{A_{\mathrm{Feas}}}$, $\mathcal{S}_{\mathrm{Opt}} \subseteq 2^{A_{\mathrm{Opt}}}$, and sets of pairs of distinct coordinates $E_{\mathrm{Feas}}, E_{\mathrm{Opt}}$ such that $|A_{\mathrm{Feas}}| + |E_{\mathrm{Feas}}| = |A_{\mathrm{Opt}}| + |E_{\mathrm{Opt}}| = n$ and

$$\gamma(X) < \infty \iff X \cap A_{\mathrm{Feas}} \in \mathcal{S}_{\mathrm{Feas}} \ \wedge\ |X \cap \{i,j\}| \neq 1 \text{ for all } \{i,j\} \in E_{\mathrm{Feas}} \tag{65}$$

$$\gamma(X) = 0 \iff X \cap A_{\mathrm{Opt}} \in \mathcal{S}_{\mathrm{Opt}} \ \wedge\ |X \cap \{i,j\}| \neq 1 \text{ for all } \{i,j\} \in E_{\mathrm{Opt}}\,. \tag{66}$$

We construct an instance J of the GMC problem on vertices $[n]$ as follows. Let $w_{\mathrm{Feas}}(i,j) = \infty$ if $\{i,j\} \in E_{\mathrm{Feas}}$ and $w_{\mathrm{Feas}}(i,j) = 0$ otherwise. Let $w_{\mathrm{Opt}}(i,j) = 1$ if $\{i,j\} \in E_{\mathrm{Opt}}$ and $w_{\mathrm{Opt}}(i,j) = 0$ otherwise. Then the weight of edge (i,j) is $w(i,j) = w_{\mathrm{Feas}}(i,j) + w_{\mathrm{Opt}}(i,j)$. Let f_{Feas} be a set function on $[n]$ defined by $f_{\mathrm{Feas}}(X) = 0$ if $X \cap A_{\mathrm{Feas}} \in \mathcal{S}_{\mathrm{Feas}}$ and $f_{\mathrm{Feas}}(X) = \infty$ otherwise; f_{Feas} is superadditive because $\mathcal{S}_{\mathrm{Feas}}$ is a downset. Let f_{Opt} be a set function on $[n]$ defined by $f_{\mathrm{Opt}}(X) = 0$ if $X \cap A_{\mathrm{Opt}} \in \mathcal{S}_{\mathrm{Opt}}$ and $f_{\mathrm{Opt}}(X) = |X \cap A_{\mathrm{Opt}}|$ otherwise; f_{Opt} is superadditive because $\mathcal{S}_{\mathrm{Opt}}$ is a downset. Then the superadditive function defining instance J is $f = f_{\mathrm{Feas}} + f_{\mathrm{Opt}}$.

By the construction, it holds $\gamma(X) < \infty \iff J(X) < \infty$ and $\gamma(X) = 0 \iff J(X) = 0$. Moreover, for any X such that $0 < J(X) < \infty$, it holds $1 \leq J(X) \leq n$. If the set

$$B = \{\gamma(X) \mid X \subseteq [n] \ \wedge\ 0 < \gamma(X) < \infty\} \tag{67}$$

is empty, then instance J 1-approximates γ; otherwise let $b_{\min}, b_{\max}$ denote the minimum and maximum of B. We scale the weights of the edges w and the superadditive function f by a factor of $b_{\min}/n$ to obtain an instance J' such that $J'(X) \leq \gamma(X)$ for all X. Instance J' then $(n \cdot b_{\max}/b_{\min})$-approximates γ. □

To establish the tractability of infinite EDS languages, we need a better bound on the approximability of α-EDS set functions than the one given in Theorem 5.15. This is achieved in Theorem 5.17, which we prove using the following technical lemma. We refer the reader to [17, Theorem 1.1] for an example of the application of this proof technique in a simpler setting.

Lemma 5.16. *Let γ be an α-EDS set function on V for some $\alpha \geq 1$. For any distinct $u, v \in V$, let $T_{\{u,v\}}$ be a subset of V such that $|T_{\{u,v\}} \cap \{u, v\}| = 1$. Then, for any $R \subseteq S \subseteq V$, it holds*

$$\alpha^{|S|+2} \cdot \left(\left(|S|^2 + 2 \right) \cdot \gamma(S) + \sum_{|R \cap \{u,v\}|=1} \gamma\left(T_{\{u,v\}}\right) \right) \geq \gamma(R) . \tag{68}$$

Proof. First, we show by induction that, for any $X, Y_1, \ldots, Y_n \subseteq V$, it holds

$$\alpha^n \cdot \left(\gamma(X) + \sum_{i=1}^{n} \gamma(Y_i) \right) \geq \gamma\left(X \setminus \bigcup_{i=1}^{n} Y_i \right) . \tag{69}$$

For $n = 1$, this is equivalent to (62). As for the inductive step, assume that (69) holds for $n \geq 1$. By the inductive hypothesis and (62), we get

$$\alpha^{n+1} \cdot \left(\gamma(X) + \sum_{i=1}^{n+1} \gamma(Y_i) \right) \geq \alpha \cdot \left(\gamma\left(X \setminus \bigcup_{i=1}^{n} Y_i \right) + \gamma(Y_{n+1}) \right) \tag{70}$$

$$\geq \gamma\left(X \setminus \bigcup_{i=1}^{n+1} Y_i \right) . \tag{71}$$

If $\gamma(S) = \infty$, the inequality claimed by this lemma trivially holds. In the rest of the proof, we assume $\gamma(S) < \infty$. For any $u \in R$, $v \in S \setminus R$, we define a set T'_{uv} such that $T'_{uv} \cap \{u, v\} = \{v\}$: If $v \in T_{\{u,v\}}$, let $T'_{uv} = T_{\{u,v\}}$; otherwise let $T'_{uv} = S \setminus T_{\{u,v\}}$. We claim that

$$\alpha \cdot \left(\gamma(S) + \gamma\left(T_{\{u,v\}}\right)\right) \geq \gamma(T'_{uv}) . \tag{72}$$

This is trivially true in the case of $T'_{uv} = T_{\{u,v\}}$, and it follows from (62) in the case of $T'_{uv} = S \setminus T_{\{u,v\}}$. By (72), it holds

$$\sum_{|R \cap \{u,v\}|=1} \gamma\left(T_{\{u,v\}}\right) \geq \sum_{u \in R} \sum_{v \in S \setminus R} \gamma\left(T_{\{u,v\}}\right) \tag{73}$$

$$\geq \frac{1}{\alpha} \sum_{u \in R} \sum_{v \in S \setminus R} \gamma(T'_{uv}) - |R| \cdot |S \setminus R| \cdot \gamma(S) . \tag{74}$$

For any $u \in R$, let

$$W_u = S \setminus \bigcup_{v \in S \setminus R} T'_{uv} . \tag{75}$$

By properties of T'_{uv}, it holds $u \in W_u \subseteq R$. Moreover, we have

$$\alpha^{|S \setminus R|} \cdot \left(\gamma(S) + \sum_{v \in S \setminus R} \gamma(T'_{uv}) \right) \geq \gamma(W_u) \tag{76}$$

by (69), which together with (74) gives us

$$\sum_{|R \cap \{u,v\}|=1} \gamma\left(T_{\{u,v\}}\right) \geq \frac{1}{\alpha^{|S \setminus R|+1}} \sum_{u \in R} \gamma(W_u) - |R| \cdot (|S \setminus R| + 1) \cdot \gamma(S) \tag{77}$$

$$\geq \frac{1}{\alpha^{|S \setminus R|+1}} \sum_{u \in R} \gamma(W_u) - |S|^2 \cdot \gamma(S) . \tag{78}$$

As it holds $\bigcup_{u \in R} W_u = R$, we have

$$\alpha^{|R|} \cdot \left(\gamma(S) + \sum_{u \in R} \gamma(W_u) \right) \geq \gamma(S \setminus R) , \tag{79}$$

and hence

$$\sum_{|R\cap\{u,v\}|=1} \gamma\left(T_{\{u,v\}}\right) \geq \frac{1}{\alpha^{|S|+1}} \cdot \gamma(S \setminus R) - \left(|S|^2 + 1\right) \cdot \gamma(S) \,. \tag{80}$$

As it holds $\alpha \cdot (\gamma(S) + \gamma(S \setminus R)) \geq \gamma(R)$, this proves the claimed inequality. □

THEOREM 5.17. *Any α-EDS set function on V is $\alpha^{n+2}\left(n^3 + 2n\right)$-approximable, where $n = |V|$.*

PROOF. Let γ be an α-EDS set function on V for some $\alpha \geq 1$. We construct an instance J of the GMC problem on vertices V such that it $\alpha^{n+2}\left(n^3 + 2n\right)$-approximates γ. The weight of edge (u,v) is

$$w(u,v) = \frac{1}{n^3 + 2n} \cdot \min\left\{\gamma(Z) \mid Z \subseteq V \wedge |Z \cap \{u,v\}| = 1\right\} \,. \tag{81}$$

Let f be a set function on V defined as

$$f(X) = \frac{|X|}{n^3 + 2n} \cdot \min\left\{\left(|Z|^2 + 2\right) \cdot \gamma(Z) \,\middle|\, X \subseteq Z \subseteq V\right\} \,. \tag{82}$$

We claim that f is a superadditive set function. As $\gamma(\emptyset) = 0$, it holds $f(\emptyset) = 0$. Consider any disjoint $X, Y \subseteq V$ and let $Z \supseteq X \cup Y$ be a minimiser in (82) for $f(X \cup Y)$. It holds $f(X) \leq |X| \cdot \left(|Z|^2 + 2\right) \cdot \gamma(Z) / \left(n^3 + 2n\right)$ and $f(Y) \leq |Y| \cdot \left(|Z|^2 + 2\right) \cdot \gamma(Z) / \left(n^3 + 2n\right)$, and hence

$$f(X) + f(Y) \leq \frac{|X \cup Y|}{n^3 + 2n} \cdot \left(|Z|^2 + 2\right) \cdot \gamma(Z) = f(X \cup Y) \,. \tag{83}$$

The edge weights w and superadditive set function f define the GMC instance J. Now we prove that it $\alpha^{n+2}\left(n^3 + 2n\right)$-approximates γ.

First, we show that $J(R) \leq \gamma(R)$ for all $R \subseteq V$. By (82), we have $f(R) \leq |R| \cdot \left(|R|^2 + 2\right) \cdot \gamma(R) / \left(n^3 + 2n\right)$. For any edge (u,v) cut by R (i.e. $|R \cap \{u,v\}| = 1$), it holds $w(u,v) \leq \gamma(R) / \left(n^3 + 2n\right)$ by (81), and hence $g(R) \leq |R| \cdot |V \setminus R| \cdot \gamma(R) / \left(n^3 + 2n\right)$. Together, this gives

$$J(R) = f(R) + g(R) \leq \frac{|R| \cdot \left(|R|^2 + |V \setminus R| + 2\right)}{n^3 + 2n} \cdot \gamma(R) \leq \gamma(R) \,. \tag{84}$$

Second, we show that $\alpha^{n+2}\left(n^3 + 2n\right) \cdot J(R) \geq \gamma(R)$ for all $R \subseteq V$. For $R = \emptyset$, the inequality holds, as $J(\emptyset) = \gamma(\emptyset) = 0$. Otherwise, let $S \supseteq R$ be a minimiser in (82) for $f(R)$, and $T_{\{u,v\}}$ a minimiser in (81) for any edge (u,v). It holds

$$\left(n^3 + 2n\right) \cdot f(R) = |R| \cdot \left(|S|^2 + 2\right) \cdot \gamma(S) \geq \left(|S|^2 + 2\right) \cdot \gamma(S) \tag{85}$$

$$\left(n^3 + 2n\right) \cdot g(R) = \sum_{|R\cap\{u,v\}=1|} \gamma\left(T_{\{u,v\}}\right) \,, \tag{86}$$

and therefore, by Lemma 5.16, $\alpha^{n+2}\left(n^3 + 2n\right) \cdot J(R) \geq \alpha^{|S|+2}\left(n^3 + 2n\right) \cdot J(R) \geq \gamma(R)$. □

THEOREM 5.18. *Any EDS language is globally s-tractable.*

PROOF. Let Γ be an EDS language and $\alpha' \geq 1$ such that every weighted relation in Γ is α'-EDS. Without loss of generality, we may assume that $\gamma\left(\mathbf{0}^{\mathrm{ar}(\gamma)}\right) = 0$ for every $\gamma \in \Gamma$, and hence identify weighted relations with their corresponding set functions. Weighted relations in Γ are of bounded arity and therefore, by Theorem 5.17, there exists α such that every $\gamma \in \Gamma$ is α-approximable. We will denote by J_γ a GMC instance that α-approximates γ.

Given a $\mathrm{VCSP_s}(\Gamma)$ instance I with an objective function

$$\phi_I'(x_1, \ldots, x_n) = \sum_{i=1}^{q} w_i \cdot \gamma_i(\mathbf{x}^i), \tag{87}$$

we denote by ϕ_I the corresponding set function and construct a GMC instance J that α-approximates ϕ_I. For $i \in [q]$, we relabel the vertices of J_{γ_i} to match the variables in the scope $\mathbf{x}^i$ of the ith constraint (i.e., vertex j is relabelled to x_j^i) and identify vertices in case of repeated variables. As the constraint is weighted by a non-negative factor w_i, we also scale the weights of the edges of J_{γ_i} and the superadditive function by w_i. (Note that non-negative scaling preserves superadditivity.) Instance J is then obtained by adding up GMC instances J_{γ_i} for all $i \in [q]$.

Let $\mathbf{x} \in D^n$ denote a surjective assignment minimising ϕ_I', $X \subseteq [n]$ the corresponding set $\{i \in [n] \mid x_i = 1\}$, $Y \subseteq [n]$ an optimal solution to J, and $\lambda = J(Y)$. Since J α-approximates ϕ_I, it holds

$$\lambda \leq J(X) \leq \phi_I(X) \leq \phi_I(Y) \leq \alpha \cdot J(Y) = \alpha\lambda, \tag{88}$$

and hence X is an α-optimal solution to J. By Lemma 5.5, we can determine whether $\lambda = 0$, in which case any optimal solution to J is also optimal for ϕ_I; and whether $\lambda = \infty$. If $0 < \lambda < \infty$, we find all α-optimal solutions by Theorem 5.11. □

We now prove Theorem 3.4.

Proof. We only need to prove the theorem in the case of an EDS language (whether Γ or $\neg(\Gamma)$, which is symmetric), as the remaining classes of globally s-tractable languages include constants C_D and thus admit a polynomial-delay algorithm using standard self-reduction techniques [10, 14].

Let Γ be an EDS language. As in the proof of Theorem 5.18, we may assume that every weighted relation in Γ assigns 0 as the minimum value. Given an instance of $\mathrm{VCSP_s}(\Gamma)$, we can determine in polynomial time, by Lemma 5.5, whether $\lambda = 0$, $0 < \lambda < \infty$, or $\lambda = \infty$. If $\lambda = 0$, then optimal solutions incur the minimum value from every constraint. By applying Opt to all constraints, we obtain a CSP instance invariant under min (by Lemma 2.13), and hence are able to enumerate all optimal solutions with a polynomial delay by the results in [14]. If $0 < \lambda < \infty$, then the claim follows from the proof of Theorem 5.18; moreover, the number of optimal solutions is polynomially bounded (see Theorem 5.11). Finally, the case $\lambda = \infty$ is trivial. □

6 CONCLUSIONS

We have established the complexity classification of surjective VCSPs on two-element domains. An obvious open problem is to consider surjective VCSPs on three-element domains. A complexity classification is known for $\{0, \infty\}$-valued languages [5] and $\mathbb{Q}$-valued languages [26] (the latter generalises the $\{0, 1\}$-valued case obtained in [28]). In fact, [31] implies a dichotomy for $\overline{\mathbb{Q}}$-valued languages on a three-element domain. However, all these results depend on the notion of core and the presence of constants C_D in the language, and thus it is unclear how to use them to obtain a complexity classification in the surjective setting. Moreover, one special case of the CSP on a three-element domain is the *3-No-Rainbow-Colouring* problem [4], whose complexity status is open.

ACKNOWLEDGEMENTS

We would like to thank Yuni Iwamasa, who prompted us to extend the complexity classification to languages of infinite size (and bounded arity). We also thank the anonymous reviewers of this paper and of the two extended abstracts [23, 43].

Peter Fulla and Stanislav Živný were supported by a Royal Society Research Grant. Stanislav Živný was supported by a Royal Society University Research Fellowship. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 714532). The paper reflects only the authors' views and not the views of the ERC or the European Commission. The European Union is not liable for any use that may be made of the information contained therein.

REFERENCES

[1] Walter Bach and Hang Zhou. Approximation for Maximum Surjective Constraint Satisfaction Problems. Technical report, October 2011. arXiv:1110.2953.

[2] Libor Barto and Marcin Kozik. Constraint Satisfaction Problems Solvable by Local Consistency Methods. *Journal of the ACM*, 61(1), 2014. Article No. 3.

[3] Libor Barto, Andrei Krokhin, and Ross Willard. Polymorphisms, and how to use them. In Krokhin and Živný [33], pages 1–44.

[4] Manuel Bodirsky, Jan Kára, and Barnaby Martin. The complexity of surjective homomorphism problems - a survey. *Discrete Applied Mathematics*, 160(12):1680–1690, 2012.

[5] Andrei Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *Journal of the ACM*, 53(1):66–120, 2006.

[6] Andrei Bulatov. A dichotomy theorem for nonuniform CSP. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS'17)*, pages 319–330. IEEE, 2017.

[7] Andrei Bulatov, Andrei Krokhin, and Peter Jeavons. Classifying the Complexity of Constraints using Finite Algebras. *SIAM Journal on Computing*, 34(3):720–742, 2005.

[8] Andrei A. Bulatov, Víctor Dalmau, Martin Grohe, and Dániel Marx. Enumerating homomorphisms. *Journal of Computer and System Sciences*, 78(2):638–650, 2012.

[9] Hubie Chen. An algebraic hardness criterion for surjective constraint satisfaction. *Algebra universalis*, 72(4):393–401, 2014.

[10] David A. Cohen. Tractable Decision for a Constraint Language Implies Tractable Search. *Constraints*, 9:219–229, 2004.

[11] David A. Cohen, Martin C. Cooper, Páidí Creed, Peter Jeavons, and Stanislav Živný. An algebraic theory of complexity for discrete optimisation. *SIAM Journal on Computing*, 42(5):915–1939, 2013.

[12] David A. Cohen, Martin C. Cooper, Peter G. Jeavons, and Andrei A. Krokhin. The Complexity of Soft Constraint Satisfaction. *Artificial Intelligence*, 170(11):983–1016, 2006.

[13] Nadia Creignou. A dichotomy theorem for maximum generalized satisfiability problems. *Journal of Computer and System Sciences*, 51(3):511–522, 1995.

[14] Nadia Creignou and Jean-Jacques Hébrard. On generating all solutions of generalized satisfiability problems. *Informatique Théorique et Applications*, 31(6):499–511, 1997.

[15] P. Crescenzi. A short guide to approximation preserving reductions. In *Proceedings of the 12th Annual IEEE Conference on Computational Complexity*, CCC '97, pages 262–, Washington, DC, USA, 1997. IEEE Computer Society.

[16] Rina Dechter and Alon Itai. Finding all solutions if you can find one. In *AAAI 1992 Workshop on Tractable Reasoning*, pages 35–39, 1992.

[17] Nikhil R. Devanur, Shaddin Dughmi, Roy Schwartz, Ankit Sharma, and Mohit Singh. On the approximation of submodular functions. April 2013. arXiv:1304.4948.

[18] Tomás Feder and Moshe Y. 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.

[19] Jiří Fiala and Jan Kratochvíl. Locally constrained graph homomorphisms - structure, complexity, and applications. *Computer Science Review*, 2(2):97–111, 2008.

[20] Jiří Fiala and Daniël Paulusma. A complete complexity classification of the role assignment problem. *Theoretical Computer Science*, 349(1):67–81, 2005.

[21] András Frank. *Connections in Combinatorial Optimization.* Oxford Lecture Series in Mathematics and Its Applications. OUP Oxford, 2011.

[22] Peter Fulla and Stanislav Živný. A Galois Connection for Valued Constraint Languages of Infinite Size. *ACM Transactions on Computation Theory*, 8(3), 2016. Article No. 9.

[23] Peter Fulla and Stanislav Živný. The complexity of Boolean surjective general-valued CSPs. In *Proceedings of the 42nd International Symposium on Mathematical Foundations of Computer Science (MFCS'17)*, 2017.

[24] Petr A. Golovach, Bernard Lidický, Barnaby Martin, and Daniël Paulusma. Finding vertex-surjective graph homomorphisms. *Acta Informatica*, 49(6):381–394, 2012.

[25] Petr A. Golovach, Daniël Paulusma, and Jian Song. Computing vertex-surjective homomorphisms to partially reflexive trees. *Theoretical Computer Science*, 457:86–100, 2012.

[26] Anna Huber, Andrei Krokhin, and Robert Powell. Skew bisubmodularity and valued CSPs. *SIAM Journal on Computing*, 43(3):1064–1084, 2014.

[27] David S. Johnson, Mihalis Yannakakis, and Christos H. Papadimitriou. On generating all maximal independent sets. *Information Processing Letters*, 27(3):119 – 123, 1988.

[28] Peter Jonsson, Mikael Klasson, and Andrei A. Krokhin. The approximability of three-valued MAX CSP. *SIAM Journal on Computing*, 35(6):1329–1349, 2006.

[29] David R. Karger. Global Min-cuts in RNC, and Other Ramifications of a Simple Min-Cut Algorithm. In *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'93)*, pages 21–30, 1993.

[30] Subhash Khot. On the unique games conjecture (invited survey). In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity (CCC'10)*, pages 99–121. IEEE Computer Society, 2010.

[31] Vladimir Kolmogorov, Andrei A. Krokhin, and Michal Rolínek. The Complexity of General-Valued CSPs. *SIAM Journal on Computing*, 46(3):1087–1110, 2017.

[32] Marcin Kozik and Joanna Ochremiak. Algebraic Properties of Valued Constraint Satisfaction Problem. In *Proceedings of the 42nd International Colloquium on Automata, Languages and Programming (ICALP'15)*, volume 9134 of *Lecture Notes in Computer Science*, pages 846–858. Springer, 2015.

[33] Andrei A. Krokhin and Stanislav Živný, editors. *The Constraint Satisfaction Problem: Complexity and Approximability*, volume 7 of *Dagstuhl Follow-Ups*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[34] Konstantin Makarychev and Yury Makarychev. Approximation Algorithms for CSPs. In Krokhin and Živný [33], pages 287–325.

[35] Barnaby Martin and Daniël Paulusma. The computational complexity of disconnected cut and $2K_2$-partition. *Journal of Combinatorial Theory, Series B*, 111:17–37, 2015.

[36] Hiroshi Nagamochi and Toshihide Ibaraki. *Algorithmic aspects of graph connectivity*, volume 123. Cambridge University Press New York, 2008.

[37] Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC'08)*, pages 245–254. ACM, 2008.

[38] Prasad Raghavendra. Approximating NP-hard Problems: Efficient Algorithms and their Limits. *PhD Thesis*, 2009.

[39] Thomas J. Schaefer. The Complexity of Satisfiability Problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC'78)*, pages 216–226. ACM, 1978.

[40] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer, 2003.

[41] Mechthild Stoer and Frank Wagner. A simple min-cut algorithm. *Journal of the ACM*, 44(4):585–591, 1997.

[42] Johan Thapper and Stanislav Živný. The complexity of finite-valued CSPs. *Journal of the ACM*, 63(4), 2016. Article No. 37.

[43] Hannes Uppman. Max-Sur-CSP on Two Elements. In *Proceedings of the 18th International Conference on Principles and Practice of Constraint Programming (CP'12)*, volume 7514 of *Lecture Notes in Computer Science*, pages 38–54. Springer, 2012.

[44] Leslie G. Valiant. The Complexity of Enumeration and Reliability Problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

[45] Alexander Vardy. Algorithmic Complexity in Coding Theory and the Minimum Distance Problem. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC'97)*, pages 92–109, New York, NY, USA, 1997. ACM.

[46] Vijay V. Vazirani and Mihalis Yannakakis. Suboptimal Cuts: Their Enumeration, Weight and Number (Extended Abstract). In *Proceedings of the 19th International Colloquium on Automata, Languages and Programming (ICALP'92)*, pages 366–377. Springer-Verlag, 1992.

[47] Narayan Vikas. Algorithms for partition of some class of graphs under compaction and vertex-compaction. *Algorithmica*, 67(2):180–206, 2013.

[48] Dmitriy Zhuk. The Proof of CSP Dichotomy Conjecture. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS'17)*, pages 331–342. IEEE, 2017.

Received November 2017; revised June 2018; accepted September 2018