Beyond Boolean Surjective VCSPs

Gregor Matl
Department of Informatics, Technical University of Munich, Germany
matlg@in.tum.de

Stanislav Živný
Department of Computer Science, University of Oxford, UK
standa.zivny@cs.ox.ac.uk

──────── **Abstract** ────────

Fulla, Uppman, and Živný [ACM ToCT'18] established a dichotomy theorem for Boolean surjective general-valued constraint satisfaction problems (VCSPs), i.e., VCSPs on two-element domains in which both labels have to be used in a solution. This result, in addition to identifying the complexity frontier, features the discovery of a new non-trivial tractable case (called EDS) that does not appear in the non-surjective setting.

In this work, we go beyond Boolean domains. As our main result, we introduce a generalisation of EDS to arbitrary finite domains called SEDS (similar to EDS) and establish a conditional complexity classification of SEDS VCSPs based on a reduction to smaller domains. This gives a complete classification of SEDS VCSPs on three-element domains. The basis of our tractability result is a natural generalisation of the Min-Cut problem, in which only solutions of certain size (given by a lower and upper bound) are permitted. We show that all near-optimal solutions to this problem can be enumerated in polynomial time, which might be of independent interest.

1 Introduction

Constraint satisfaction problems (CSPs) are fundamental computer science problems studied in artificial intelligence, logic (as model checking of the positive primitive fragment of first-order logic), graph theory (as homomorphisms between relational structures), and databases (as conjunctive queries) [13]. A vast generalisation of CSPs is that of general-valued CSPs (VCSPs) [21], see also [6]. Recent years have seen some remarkable progress on our understanding of the computational complexity of CSPs and VCSPs, as will be discussed later in related work. We start with a few definitions to state existing as well as our new results.

We consider regular, surjective and lower-bounded VCSPs on the extended rationals $\overline{\mathbb{Q}} = \mathbb{Q} \cup \{\infty\}$. An *instance* $I = (V, D, \phi_I)$ of either of these problems 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* $\phi_I : D^n \to \overline{\mathbb{Q}}$. The objective function is of the form

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

where $t \in \mathbb{N}$ and, for each $1 \le i \le t$, $\gamma_i : D^{\mathrm{ar}(\gamma_i)} \to \overline{\mathbb{Q}}$ is a *weighted relation* of arity $\mathrm{ar}(\gamma_i) \in \mathbb{N}$, $w_i \in \mathbb{Q}_{\ge 0}$ is a *weight* and $\boldsymbol{x}_i \in V^{\mathrm{ar}(\gamma_i)}$ is a tuple of variables from V called the *scope* of γ_i.

Regular, surjective and lower-bounded VCSPs differ only in their solution space, although this makes a big difference in complexity. If I is an instance of a regular VCSP, an *assignment* is a map $s : V \to D$ assigning a label from D to each variable. In the surjective setting, only a surjective map $s : V \to D$ is an assignment. For lower-bounded VCSPs, a lower bound $l : D \to \mathbb{N}_0$ is provided and an assignment is a map $s : V \to D$ such that $\left| s^{-1}(d) \right| \ge l(d)$ for every label $d \in D$. In other words, a lower bound $l(d)$ on the number of occurrences of each label $d \in D$ is imposed. The *value* of an assignment s is given by $\phi_I(s(x_1), \ldots, s(x_n))$. An assignment is called *feasible* if its value is finite, and is called *optimal* if it is of minimal value among all assignments for the instance. The objective is to obtain an optimal assignment.

While finding an optimal assignment is NP-hard in general, valued constraint languages impose a natural restriction on the types of instances that are allowed. A *valued constraint language*, or simply a *language*, is a possibly infinite set of weighted relations. In this paper, we only consider languages of bounded arity, that is languages admitting a fixed upper bound on the arity of all weighted relations contained in them. Weighted relations in any VCSP instance will be stored explicitly.

We denote the class of regular VCSP instances with objective functions using only weighted relations from a language Γ by $\mathrm{VCSP}(\Gamma)$. Similarly, $\mathrm{VCSP}_s(\Gamma)$ is the class of surjective VCSP instances with weighted relations from Γ and, for some lower bound $l : D \to \mathbb{N}_0$, $\mathrm{VCSP}_l(\Gamma)$ is the class of lower-bounded VCSP instances over Γ with bound l.

A language Γ is *globally tractable* if there is a polynomial-time algorithm for solving each instance of $\mathrm{VCSP}(\Gamma)$, or *globally intractable* if $\mathrm{VCSP}(\Gamma)$ is NP-hard. Analogously, Γ is *globally s-tractable* if there is a polynomial-time algorithm for $\mathrm{VCSP}_s(\Gamma)$, or *globally s-intractable* if $\mathrm{VCSP}_s(\Gamma)$ is NP-hard. And Γ is *globally ℓ-tractable* if $\mathrm{VCSP}_l(\Gamma)$ is solvable in polynomial time for every fixed lower bound $l : D \to \mathbb{N}_0$, or *globally ℓ-intractable* if $\mathrm{VCSP}_l(\Gamma)$ is NP-hard for at least one fixed lower bound $l : D \to \mathbb{N}_0$. Thus, global ℓ-tractability implies global s-tractability, and global s-intractability implies global ℓ-intractability.

The following examples show how well-studied variants of the MIN-CUT problem can be modelled in the VCSP frameworks we have defined.

▶ **Example 1** (*r*-TERMINAL MIN-CUT)**.** Given a graph $G = (V, E)$ with non-negative edge weights $w : E \to \mathbb{Q}_{\ge 0}$ and designated terminal vertices $s_1, \ldots, s_r \in V$, the *r*-TERMINAL MIN-CUT problem asks to partition V into subsets $X_1, \ldots, X_r$ such that $s_d \in X_d$ for all $d \in [r] := \{1, \ldots, r\}$, while the accumulated weight of all edges going between distinct sets X_i and X_j is minimised. For $r = 2$, this problem is also known as the (s, t)-MIN-CUT problem.

We show how this problem can be represented as a regular VCSP. Let $\gamma_{r\text{-cut}}$ denote the binary weighted relation defined for $x, y \in [r]$ by $\gamma_{r\text{-cut}}(x, y) = 0$ if $x = y$ and $\gamma_{r\text{-cut}}(x, y) = 1$ otherwise. Furthermore, for each label $d \in [r]$, let ρ_d denote the constant relation given by $\rho_d(d) = 0$ and $\rho_d(x) = \infty$ for $d \ne x \in [r]$. Let $\Gamma_{r\text{-cut}} = \{\gamma_{r\text{-cut}}, \rho_1, \ldots, \rho_r\}$.

Finding an optimal r-terminal cut is equivalent to solving the VCSP $(\Gamma_{r\text{-cut}})$ instance $I = (V, [r], \phi)$ with objective function

$$\phi(x_1, \dots, x_n) = \rho_1(s_1) + \cdots + \rho_r(s_r) + \sum_{(u,v) \in E} w(u,v) \cdot \gamma_{r\text{-cut}}(u,v).$$

To see this, observe that there is a correspondence between feasible assignments $s : V \to [r]$ and r-terminal cuts $X_1, \dots, X_r$ by setting $X_d = \{v \in V : s(v) = d\}$, with the objective value remaining equal. Hence, an optimal assignment induces an optimal cut.

The r-Terminal Min-Cut problem can be solved in polynomial time if $r = 2$, but it is NP-hard for any $r \geq 3$ [8]. Since every VCSP $(\Gamma_{r\text{-cut}})$ instance can also be reduced to an instance of the r-Terminal Min-Cut problem, the language $\Gamma_{r\text{-cut}}$ is globally tractable if $r = 2$ and globally intractable for $r \geq 3$. $\diamond$

▶ **Example 2** (r-Way Min-Cut). Without setting out any terminals, the r-Way Min-Cut problem asks to partition V into non-empty subsets $X_1, \dots, X_r$ such that weight of the induced cut is minimised. Finding an optimal r-way min-cut is equivalent to solving the VCSP$_s(\{\gamma_{r\text{-cut}}\})$ instance $I = (V, [r], \phi)$ with objective function

$$\phi(x_1, \dots, x_r) = \sum_{(u,v) \in E} w(u,v) \cdot \gamma_{r\text{-cut}}(u,v).$$

The r-Way Min-Cut problem can be solved in polynomial time for every fixed integer r [11]. Since every VCSP$_s(\{\gamma_{r\text{-cut}}\})$ instance can be reduced to a r-Way Min-Cut problem as well, the language $\{\gamma_{r\text{-cut}}\}$ is globally s-tractable. $\diamond$

For a fixed $l : D \to V$, VCSP$_l(\{\gamma_{r\text{-cut}}\})$ allows to model a generalisation of the r-Way Min-Cut problem where a partition $X_1, \dots, X_r$ of V minimising the induced cut is sought under the condition that $|X_d| \geq l(d)$ for every $d \in D$. As far as we know, the complexity of both VCSP$_l(\{\gamma_{r\text{-cut}}\})$ and the lower-bounded r-Way Min-Cut problem is unknown.

Related Work

Early results on CSPs include the fundamental results of Schaefer on Boolean CSPs [20] and of Hell and Nešetřil on graph CSPs [12]. The computational complexity of CSPs has drawn a lot of attention following the seminal paper of Feder and Vardi [9]. Using the algebraic approach [15, 3], the complexity of CSPs on finite domains was resolved in two independent papers by Bulatov [4] and Zhuk [24]. The computational complexity of the problem of minimising the number of unsatisfied constraints (and more generally rational-valued weighted relations) was obtained by Thapper and Živný in [23]. Finally, the computational complexity of general-valued CSPs on finite domains was obtained by the work of Kozik and Ochremiak [18] and Kolmogorov, Krokhin, and Rolínek [16].

In addition to constraints that apply locally to the variables specified as arguments, forms of VCSPs have been considered where global conditions are imposed. Among those are CSPs with global cardinality constraints, or CCSPs, where it is specified how often exactly each label has to occur in an assignment. A dichotomy theorem for CCSPs on finite domains was established by Bulatov and Marx [5].

Surjective VCSPs, which can be seen as imposing a global condition as well, have been studied by Fulla, Uppman, and Živný [10], following earlier results on CSPs by Creignou and Hébrard [7] and Bodirsky, Kára, and Martin [1]. Unfortunately, the algebraic approach that has proved pivotal in the understanding of the computational complexity of regular CSPs and VCSPs is not applicable in the surjective setting.

The following two facts are easy to show (see, e.g, [10]): (i) intractable languages are also *s*-intractable; (ii) a tractable language Γ is also *s*-tractable *if* Γ includes all constant relations. Consequently, new *s*-tractable languages can only occur (if at all) as subsets of tractable languages that do not contain all constant relations. [10] identified the first example of such languages. In particular, [10] identified languages on the Boolean domain that are essentially a downset, or EDS, as a new class of efficiently solvable problems and, in doing so, provided a classification of surjective VCSPs on the Boolean domain.

The tractability result of EDS languages is based on the Generalised Min-Cut (GMC) problem for graphs, also introduced in [10]. In a GMC instance, the goal is to find a non-trivial subset of the vertices such that the weight of the induced cut and a superadditive set function are minimised simultaneously. [10] showed how the objective function of surjective VCSPs that are EDS can be approximated by an instance of the GMC problem. In addition, they provided a polynomial-time algorithm to enumerate all solutions to the GMC problem that are optimal up to a constant factor, which in combination results in an efficient algorithm for surjective VCSPs that are EDS.

Contributions

This paper extends the class EDS to arbitrary finite domains. We introduce a class SIM of languages that exhibit properties similar to Boolean languages. Based on this class, we define the class SEDS as a natural extension of EDS and classify languages from this extension based on two criteria. Firstly, we give a subclass SDS of SEDS that guarantees global ℓ-tractability without additional requirements. Secondly, we prove that the complexity of lower-bounded VCSPs over any remaining SEDS languages is equivalent to the complexity over a particular language on a smaller domain, which can be constructed by including all possible ways to assign a certain label (formally defined in Section 3). This is illustrated in Figure 1 (left).
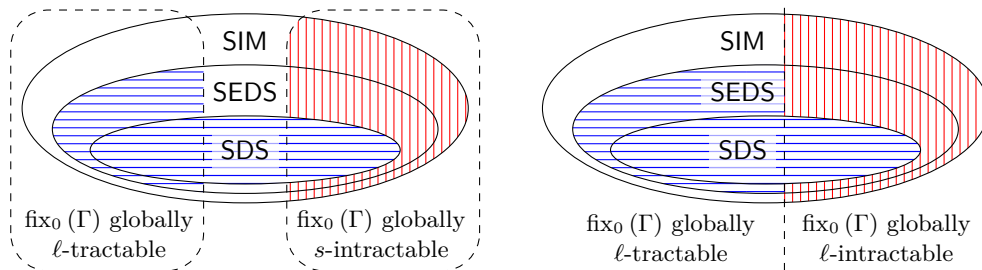
Figure 1 Classification of SEDS languages on arbitrary finite domains (left) and on three-element domains (right). A language Γ is globally ℓ-tractable when marked by horizontal (blue) lines and globally *s*-intractable when marked by vertical (red) lines, depending on the language fix_0 (Γ) on a smaller domain. (Recall that global *s*-intractability implies global ℓ-intractability.) In case of three-element domains, the Boolean language fix_0 (Γ) is either globally ℓ-tractable or globally ℓ-intractable, while this is not known for larger domains.

One implication of our results is a dichotomy theorem for lower-bounded VCSPs on the Boolean domain; every Boolean language is either globally ℓ-tractable or globally ℓ-intractable. Although lower-bounded VCSPs are more general than surjective VCSPs, this classification coincides with the dichotomy theorem for surjective VCSPs given by [10]. (Details are given in the full version of this paper [19].)

In addition, combining our reduction of SEDS languages to a smaller domain and the dichotomy theorem for the Boolean domain leads to a classification of all SEDS languages on three-element domains with respect to ℓ-tractability, which is featured on the right-hand side of Figure 1.

The foundation of our results is an extension of the Generalised Min-Cut problem that might be of independent interest. Given integers $p, q \in \mathbb{N}_0$, a graph with non-negative edge weights and a superadditive set function defined on its vertices, the goal in the Bounded Generalised Min-Cut problem is, just like in the GMC problem, to find a subset of the vertices such that the sum of the induced cut and the superadditive set function evaluated on it are minimal among all possible solutions. The solution space, however, is restricted to subsets containing at least q and at most all but p vertices.

If an optimal solution has value 0, there can be exponentially many optimal solution, e.g. when there are no edges and the superadditive function always evaluates to 0. Our main algorithmic result is that, for all other instances and any constant bounds $p, q \in \mathbb{N}_0$, all solutions that are optimal up to a constant factor can be enumerated in polynomial time (and thus, in particular, there are only polynomially many of them).

2   The Bounded Generalised Min-Cut Problem

We begin by presenting our algorithm for the Bounded Generalised Min-Cut problem. The problem is based on the notion of superadditive set functions, which we define first.

▶ **Definition 3.** *A set function on a finite set V is a function $f : 2^V \to \overline{\mathbb{Q}}$ defined on subsets of V; it is normalised if it satisfies $f(\emptyset) = 0$ and $f(X) \geq 0$ for all $X \subseteq V$.*

A set function f on V is increasing if it is normalised and $f(X) \leq f(Y)$ for all $X \subseteq Y \subseteq V$. It is superadditive if it is normalised and, for all disjoint $X, Y \subseteq V$, it holds that

$$f(X) + f(Y) \leq f(X \cup Y). \tag{SUP}$$

Since equation (SUP) implies that $f(X) \leq f(X) + f(Y \backslash X) \leq f(Y)$ for all $X \subseteq Y \subseteq V$, every superadditive set function must also be increasing.

▶ **Definition 4.** *For $q, p \in \mathbb{N}_0$, the Bounded Generalised Min-Cut problem with lower bound q an upper bound p is denoted by GMC_q^p.*

A GMC_q^p instance h 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. For $X \subseteq V$, let $w(X) = \sum_{|\{u,v\} \cap X|=1} w(\{u,v\})$ denote the weight of the cut induced by X.

A solution for instance h is any set $X \subseteq V$ such that $|X| \geq q$ and $|X| \leq |V| - p$. The objective is to minimise the value $h(X) = f(X) + w(X)$. A solution X is optimal if the value $h(X)$ is minimal among all solutions for this instance. We denote the value of an optimal solution by λ. For any $\alpha \geq 1$, a solution X is α-optimal if $h(X) \leq \alpha\lambda$.

The Generalised Min-Cut problem, simply denoted by GMC, is the Bounded Generalised Min-Cut problem with lower and upper bound 1. All α-optimal solutions of a GMC instance can be enumerated in polynomial time according to [10, Theorem 5.11], which we restate here.

▶ **Theorem 5.** *For any instance h of the GMC problem on n vertices with optimal value $0 < \lambda < \infty$ and any constant $\alpha \in \mathbb{N}$, 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.*

We will assume that all edges are positive-valued, as they can be ignored otherwise. To simplify the problem further, observe that it can be determined in polynomial time whether the optimal value of a GMC_q^p instance is $\lambda = 0$ or $\lambda = \infty$. If $\lambda = 0$, an optimal solution can

be found by checking all connected components, because a solution of value 0 cannot cut any edges and because the superadditive set function f is increasing. Moreover, in order to determine whether $\lambda = \infty$ it is sufficient to check all solutions of size q. When these solutions all have infinite value, each one must either contain an edge of infinite weight or the superadditive set function must evaluate to infinity. In either case, all supersets will have infinite value as well, implying $\lambda = \infty$.

Consequently, our goal is to provide a polynomial-time algorithm for enumerating near-optimal solutions in the case that the optimal value is both positive and finite. Before doing so, we give two auxiliary lemmas. The first one is based on [10, Lemma 5.6].

▶ **Lemma 6.** *For any $p, q \in \mathbb{N}_0$, any GMC_q^p instance h on a graph $G = (V, E)$ and any subset $V' \subseteq V$, there is a GMC_q^p instance h' on the induced subgraph $G[V']$ that preserves the objective value of all solutions $X \subseteq V'$. In particular, any α-optimal solution X of h such that $X \subseteq V'$ is α-optimal for h' 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

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

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

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

The next lemma, which is based on [10, Lemma 5.10], can be deduced from the superadditivity of f and the posimodularity of the cut function w.

▶ **Lemma 7.** *Let h be a GMC_q^p instance over vertices V with optimal value λ and let $X, Y \subseteq V$ such that $h(X) \leq \alpha\lambda$ and $w(Y) \leq \beta\lambda$ for some $\alpha \geq 1$ and $\beta \geq 0$. Then it holds*

$$h(X \setminus Y) + h(X \cap Y) < (\alpha + 2\beta)\lambda.$$

We now proceed with our main algorithmic result. We only sketch the proof here but full details are given in the full version [19].

▶ **Theorem 8.** *For some constant $q \geq 2$, let h be a GMC_q^1 instance on a graph $G = (V, E)$ of size $n = |V|$ with optimal value $0 < \lambda < \infty$. Let $Y \cup Z = V$ be a partition of V and let $Y_1 \cup \cdots \cup Y_k = Y$ for some $k \in \mathbb{N}_0$ be a partition of Y satisfying $0 < |Y_i| < q$ and $h(Y_i) \leq \frac{\lambda}{3q}$ for all $1 \leq i \leq k$.*

Then for every constant $\alpha \geq 1$, at most $\frac{|Z|}{n} \cdot n^{\tau(q,\alpha)}$ α-optimal solutions $X \subseteq V$ of h satisfy $|X \cap Y| < q$, where $\tau(q, \alpha) = 60q\alpha + 41q + 7$. These solutions can all be enumerated in polynomial time.

Note that with $Y = \emptyset$ and $Z = V$, this theorem states for any GMC_q^1 instance that the number of α-optimal solutions is bounded by $n^{\tau(q,\alpha)}$.

Proof sketch. We give a proof by induction over $n + \frac{|Z|}{n+1}$. For $n \leq q$ or $Z = \emptyset$, there are no solutions of the described form and hence, the statement holds.

Now, fix some $n > q$, some GMC_q^1 instance h on a graph $G = (V, E)$ of size n with optimum value $0 < \lambda < \infty$ and partitions $Y \cup Z = V$ and $Y_1 \cup \cdots \cup Y_k = Y$ as described. By the induction hypothesis, we can assume that the theorem holds for every graph of size $n' < n$ as well as for every partition $\tilde{Y} \cup \tilde{Z} = V$ of graph G satisfying $\left|\tilde{Z}\right| < |Z|$.

According to Lemma 6, there exists a GMC_q^1 instance h_Z on the induced subgraph $G[Z]$ that preserves the objective value of every solution $X' \subsetneq Z$ with respect to h. In the following, we treat h_Z as a GMC instance (i.e. with lower bound 1). Let λ_Z denote the optimal value of h_Z and let $Y_{k+1} \subsetneq Z$ be an optimal solution of h_Z, i.e. $h_Z(Y_{k+1}) = \lambda_Z$.

Consider any α-optimal solution $X \subseteq V$ of h satisfying $|X \cap Y| < q$. For some integer t, let $i_1, \ldots, i_t$ denote indices such that $X \cap Y = X \cap (Y_{i_1} \cup \cdots \cup Y_{i_t})$, i.e. such that X has vertices only in $Y_{i_1}, \ldots, Y_{i_t}$ and Z. Since $|X \cap Y| < q$, we require that $t < q$. Let $U = Y_{i_1} \cup \cdots \cup Y_{i_t}$. We distinguish two cases, which are illustrated in Figure 2.
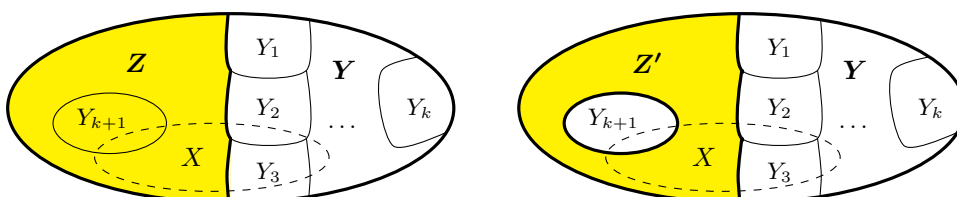
Figure 2 Given a partition $V = Y \cup Z$ with $Y = Y_1 \cup \cdots \cup Y_k$ of the vertices of a GMC_q^1 instance h, we want to find every solution X such that $h(X) \le \alpha\lambda$ and $|X \cap Y| < q$. Consider the GMC instance h_Z on $G[Z]$ with optimal solution Y_{k+1}. If $h(Y_{k+1}) \ge \frac{\lambda}{3q}$, $X \cap Z$ must be a near-optimal solution of h (left, first case). Otherwise, we apply the induction hypothesis either on the partition $V = Z' \cup (Y_1 \cup \cdots \cup Y_{k+1})$ or on the subgraph $G[Z']$, where $Z' = Z \backslash Y_{k+1}$ (right, second case).

In the first case, we assume that $\lambda_Z \ge \frac{\lambda}{3q}$. From our assumption that $h(Y_i) \le \frac{\lambda}{3q}$ for all $1 \le i \le k$, it can be deduced that $w(U) < \frac{\lambda}{3}$. By Lemma 7 with $\beta = \frac{1}{3}$, it must hold that

$$h(X \backslash U) + h(X \cap U) \le \left(\alpha + \frac{2}{3}\right)\lambda.$$

Since $X \cap Z = X \backslash U$, our assumption $\lambda_Z \ge \frac{\lambda}{3q}$ then implies that $h(X \cap Z) \le (3q\alpha + 2q)\lambda_Z$. Hence, if $X \cap Z \subsetneq Z$, then $X \cap Z$ is a $(3q\alpha + 2q)$-optimal solution of h_Z when treated as a GMC instance. The number of choices for $X \cap Z$ can then be bounded by Theorem 5.

At the same time, there are less than n^q ways to pick at most $q - 1$ vertices from Y and therefore less than n^q choices for $X \cap Y$. By pairing up all choices for $X \cap Z$ with those for $X \cap Y$, we can conclude that there are at most $\frac{1}{n} \cdot n^{\tau(q,\alpha)}$ choices for X in this case.

In the second case, we assume $\lambda_Z \le \frac{\lambda}{3q}$. Note that $h_Z(Y_{k+1}) = \lambda_Z < \lambda$ implies $|Y_{k+1}| < q$. Let $Y' = Y \cup Y_{k+1}$, $Z' = Z \backslash Y_{k+1}$ and $U' = U \cup Y_{k+1}$.

If $|X \cap Y'| < q$, we can apply the induction hypothesis for instance h with the partitions $Y' \cup Z' = V$ and $Y' = Y_1 \cup \cdots \cup Y_k \cup Y_{k+1}$. Consequently, the number of such solutions is at most $\frac{|Z'|}{n} \cdot n^{\tau(\alpha)} \le \frac{|Z|-1}{n} \cdot n^{\tau(\alpha)}$.

Therefore, we now assume that $|X \cap Y'| \ge q$ and need to show that there are at most $\frac{1}{n} \cdot n^{\tau(\alpha)}$ choices for X in this case. Given that $\lambda_Z \le \frac{\lambda}{3q}$, we can deduce that $w(U') \le \frac{\lambda}{3}$. Thus, Lemma 7 implies that

$$h(X \backslash U') + h(X \cap U') \le \left(\alpha + \frac{2}{3}\right)\lambda.$$

Being a solution of h, the set $X \cap U' = X \cap Y'$ must have value $h(X \cap Q') \geq \lambda$. It therefore holds that

$$h(X \cap Z') = h(X \backslash U') \leq \left(\alpha + \frac{2}{3}\right)\lambda - h(X \cap Q') \leq \left(\alpha - \frac{1}{3}\right)\lambda.$$

Let $h_{Z'}$ denote the GMC_q^1 instance on the induced subgraph $G[Z']$ that preserves the value of h as detailed in Lemma 6. There are roughly n^q choices for X such that $|X \cap Z'| < q$ or $X = Z'$. Otherwise, the set $X \cap Z'$ must be an $\left(\alpha - \frac{1}{3}\right)$-optimal solution of $h_{Z'}$. By applying the induction hypothesis on $h_{Z'}$ with the trivial partition $\emptyset \cup Z' = Z'$, the number of $\left(\alpha - \frac{1}{3}\right)$-optimal solutions can be bounded by $n^{\tau\left(q,\alpha-\frac{1}{3}\right)}$.

Next, we limit the number of choices for $X \cap Y'$. Since X contains at most $q - 1$ vertices from Y (less than n^q choices) and since Y_{k+1} contains at most $q - 1$ vertices (at most 2^{q-1} choices), the number of possible choices for $X \cap Y'$ is limited by $n^q \cdot 2^{q-1} \leq n^{2q}$.

Pairing up each possible choice for $X \cap Z'$ with each choice for $X \cap Y'$ gives a total of at most $\frac{1}{n} \cdot n^{\tau(q,\alpha)}$ solutions, as required.

A polynomial-time algorithm to enumerate all such solutions follows immediately from these calculations. To see this, note that only the second case is defined recursively. Therefore, checking both the first and the second case does not increase the overall complexity of $n^{O(q+\alpha)}$. In particular, it is not necessary to know the value λ beforehand. ◀

▶ **Corollary 9.** *For any $p, q \in \mathbb{N}_0$ and $\alpha \geq 1$, where q and α are constants, and for any GMC_q^p instance h with optimal value $0 < \lambda < \infty$, all α-optimal solutions can be enumerated in polynomial time.*

Proof. Let $h = f + w$ be a GMC_q^p instance with $0 < \lambda < \infty$. First, we assume that $p \geq 1$ and $q \geq 2$. The superadditive set function

$$f'(X) = \begin{cases} \infty & \text{if } |X| > |V| - p \\ f(X) & \text{otherwise} \end{cases}$$

defines a GMC_q^1 instance $h' = f' + w$ where every solution $X \subseteq V$ of size $|X| > |V| - p$ is infeasible so that the set of feasible solutions and their values are identical for h and h'. Therefore, it is sufficient to enumerate all α-optimal solutions of h', which can be accomplished in polynomial time according to Theorem 8

If $p = 0$ or $q < 2$, there are up to $|V| + 2$ additional solutions that can all be checked in polynomial time. ◀

3 Extending EDS to Larger Domains

In this section, we formally introduce the classes SIM, SEDS and SDS. In order to simplify our notation, we will subsequently always consider the $(k + 1)$-element domain $D = \{0, 1, \ldots, k\}$ for some integer k. Any other domain of size $k + 1$ can simply be relabelled without affecting its properties. One label from the domain will play a special role; without loss of generality (due to relabellings), it will be 0.

3.1 k-Set Functions

It will be convenient to go back and forth between weighted relations and k-set functions, which is, subject to a minor technical assumption, always possible.

▶ **Definition 10.** *Let $k \in \mathbb{N}$ and let V be a finite set. A k-set function on V is a function $f : (k+1)^V \to \overline{\mathbb{Q}}$ defined on k-tuples of pairwise disjoint subsets of V. A k-set function f over V is normalised if it satisfies $f(\emptyset, \ldots, \emptyset) = 0$ and $f(X_1, \ldots, X_k) \geq 0$ for all disjoint $X_1, \ldots, X_k \subseteq V$.*

Note that a 1-set function is simply a *set function* as defined in Section 2.

▶ **Definition 11.** *Let γ be an n-ary weighted relation on the $(k+1)$-element domain $D = \{0, 1, \ldots, k\}$, and let f be the k-set function on $V = [n]$ that is defined for disjoint sets $X_1, \ldots, X_k \subseteq V$ by $f(X_1, \ldots, X_k) = \gamma(\boldsymbol{x})$, where the i-th coordinate of $\boldsymbol{x}$ is given by $x_i = d$ if $i \in X_d$ for some $0 \neq d \in D$ and $x_i = 0$ otherwise. Then γ corresponds to f.*

Furthermore, we say that γ corresponds under normalisation to a k-set function if $\gamma(\boldsymbol{0}^n) < \infty$ and $\gamma(\boldsymbol{0}^n) \leq \gamma(\boldsymbol{x})$ for all $\boldsymbol{x} \in D^n$. In this case, the k-set function corresponding under normalisation to γ is the normalised k-set function corresponding to $\gamma - \gamma(\boldsymbol{0}^n)$, i.e. the weighted relation with offset such that the assignment $\boldsymbol{0}^n$ evaluates to 0.

According to this definition, there is a unique *k*-set function corresponding to every weighted relation on the $(k+1)$-element domain, and vice versa. Furthermore, assuming that $\gamma(\boldsymbol{0}^n) < \infty$, a weighted relation γ corresponds under normalisation to a *k*-set function precisely if it admits multimorphism $\langle c_0 \rangle$ (the definition of which can be found in [6]).

▶ **Definition 12.** *Let f be a k-set function and g a set function on V. We say that g α-approximates f if, for all disjoint $X_1, \ldots, X_k \subseteq V$, it holds that*

$$g(X_1 \cup \cdots \cup X_k) \leq f(X_1, \ldots, X_k) \leq \alpha \cdot g(X_1 \cup \cdots \cup X_k).$$

3.2 Fixing a Label: Reduced Languages

Reducing a language to a smaller domain by fixing all possible occurrences of a certain label, as defined subsequently, will be a central tool in our classification.

▶ **Definition 13.** *Let f be a k-set function on V, let $0 \leq d \leq k$ be a label from the domain and let $U \subseteq V$. Then $\mathrm{fix}_{d=U}[f]$ is the $(k-1)$-set function defined for disjoint sets $X_1, \ldots, X_{d-1}, X_{d+1}, \ldots, X_k \subseteq V \setminus U$ by*

$$\mathrm{fix}_{d=U}[f](X_1, \ldots, X_{d-1}, X_{d+1}, \ldots, X_k) = f(X_1, \ldots, X_{d-1}, U, X_{d+1}, \ldots, X_k).$$

Let γ be the weighted relation on domain D corresponding to f. Then $\mathrm{fix}_{d=U}[\gamma]$ denotes the weighted relation of arity $|V \setminus U|$ on domain $D \setminus \{d\}$ corresponding to $\mathrm{fix}_{d=U}[f]$.

In other words, $\mathrm{fix}_{d=U}[\gamma]$ takes an assignment from the domain $D \setminus \{d\}$ to all variables except for those with index in U, and evaluates it through γ by assigning label d to the remaining variables. In Definition 14, we generalise this concept in order to express the language that is generated by fixing every possible assignment of a certain label.

▶ **Definition 14.** *Let Γ be a language on domain D and let $d \in D$. For any $\gamma \in \Gamma$, let $\mathrm{fix}_d(\gamma) = \{\mathrm{fix}_{d=U}[\gamma] : U \subseteq V\}$. We define the language $\mathrm{fix}_d(\Gamma)$ on domain $D \setminus \{d\}$ by $\mathrm{fix}_d(\Gamma) = \bigcup_{\gamma \in \Gamma} \mathrm{fix}_d(\gamma)$.*

3.3 Extending EDS to Larger Domains

The class EDS, or *essentially a downset*, has been introduced in [10] for the Boolean domain.

▶ **Definition 15.** *For any $\alpha \geq 1$, a normalised set function f on V is α-EDS if, for all $X, Y \subseteq V$, it holds that*

$$f(X \backslash Y) \leq \alpha \cdot (f(X) + f(Y)).\tag{EDS}$$

A weighted relation is α-EDS if it corresponds under normalisation to a set function that is α-EDS. Moreover, a language Γ is EDS if there is some $\alpha \geq 1$ such that every weighted relation $\gamma \in \Gamma$ is α-EDS.

Fulla et al. showed [10] that EDS languages are globally s-tractable. We improve upon this result by proving that such languages are in fact globally ℓ-tractable, and we extend the idea of being essentially a downset to larger domains through the classes SIM, SEDS and SDS.

Intuitively, a language is SIM, or *similar to a Boolean language*, if, for every weighted relation, the value of any two assignments that assign label 0 to precisely the same set of variables is equal up to a constant factor.

▶ **Definition 16.** *Let f be a normalised k-set function on set V. For any $\alpha \geq 1$, f is called α-SIM if, for all disjoint $X_1, \ldots, X_k \subseteq V$ and all disjoint $Y_1, \ldots, Y_k \subseteq V$ such that $X_1 \cup \cdots \cup X_k = Y_1 \cup \cdots \cup Y_k$, it holds that*

$$f(X_1, \ldots, X_k) \leq \alpha \cdot f(Y_1, \ldots, Y_k).\tag{SIM}$$

A weighted relation is α-SIM if it corresponds under normalisation to a k-set function that is α-SIM. Moreover, a language Γ is SIM if there is some $\alpha \geq 1$ such that every weighted relation $\gamma \in \Gamma$ is α-SIM.

Note that every normalised set function is 1-SIM. Hence, EDS is a subclass of SIM. Going beyond the Boolean domain, the class SEDS of languages *similar to EDS* arises as a natural generalisation of EDS.

▶ **Definition 17.** *For any $\alpha \geq 1$, a normalised k-set function f on V is α-SEDS if it is α-SIM and, for all disjoint $X_1, \ldots, X_k \subseteq V$ and all disjoint $Y_1, \ldots, Y_k \subseteq V$, it holds that*

$$f(X_1 \backslash Y_1, \ldots, X_k \backslash Y_k) \leq \alpha \cdot (f(X_1, \ldots, X_k) + f(Y_1, \ldots, Y_k)).\tag{SEDS}$$

A weighted relation is α-SEDS if it corresponds under normalisation to a k-set function that is α-SEDS. Moreover, a language Γ is SEDS if there is some $\alpha \geq 1$ such that every weighted relation $\gamma \in \Gamma$ is α-SEDS.

The class SDS, or *similar to a downset*, imposes a stricter requirement than SEDS. When any arguments of a weighted relation are changed to label 0, the value should decrease, stay equal or increase by at most a constant factor.

▶ **Definition 18.** *For any $\alpha \geq 1$, a normalised k-set function f on V is α-SDS if it is α-SIM and in addition, for all disjoint $X_1, \ldots, X_k, Y_1, \ldots, Y_k \subseteq V$, it holds that*

$$f(X_1, \ldots, X_k) \leq \alpha \cdot f(X_1 \cup Y_1, \ldots, X_k \cup Y_k).\tag{SDS}$$

A weighted relation is α-SDS if it corresponds under normalisation to a k-set function that is α-SDS, and a language Γ is SDS if there is some $\alpha \geq 1$ such that every weighted relation $\gamma \in \Gamma$ is α-SDS.

Note that SDS is a subclass of SEDS. To see this, consider any α-SDS k-set function f on V. Then it holds for all disjoint $X_1, \ldots, X_k \subseteq V$ and all disjoint $Y_1, \ldots, Y_k \subseteq V$ that

$$f(X_1 \backslash Y_1, \ldots, X_k \backslash Y_k) \leq \alpha \cdot f(X_1, \ldots, X_k) \leq \alpha \cdot (f(X_1, \ldots, X_k) + f(Y_1, \ldots, Y_k)),$$

proving that f is α-SEDS.

4   Classifying SEDS and SDS Languages

In this section, we first show that a SEDS language Γ is globally ℓ-tractable if it is SDS or if the reduced language $\mathrm{fix}_0\left(\Gamma\right)$ is globally ℓ-tractable. Afterwards, we prove global s-intractability of the remaining SEDS languages conditioned on global s-intractability of $\mathrm{fix}_0\left(\Gamma\right)$.

We begin by restating [10, Theorem 5.17] concerning EDS languages and then devise similar approximations for SEDS and SDS languages.

▶ **Theorem 19.** *For any α-EDS set function f on V, there exists a* GMC *instance h that $\alpha^{n+2}\left(n^3+2n\right)$-approximates f, where $n=|V|$.*

▶ **Lemma 20.** *For any α-SEDS k-set function f on V, there exists an α-EDS set function g that α^2-approximates f.*

Proof. We define the set function g on V by $g\left(X\right)=\frac{1}{\alpha}f\left(X,\emptyset,\dots,\emptyset\right)$. Observe that, since f is normalised, it holds $g\left(\emptyset\right)=f\left(\emptyset,\dots,\emptyset\right)=0$ and $g\left(X\right)=\frac{1}{\alpha}f\left(X,\emptyset,\dots,\emptyset\right)\geq 0$ for every $X\subseteq V$. Thus, g is normalised as well. In addition, for all $X,Y\subseteq V$, it holds that

$$\alpha\cdot\left(g\left(X\right)+g\left(Y\right)\right)=f\left(X,\emptyset,\dots,\emptyset\right)+f\left(Y,\emptyset,\dots,\emptyset\right)\geq\frac{1}{\alpha}\cdot f\left(X\backslash Y,\emptyset,\dots,\emptyset\right)=g\left(X\backslash Y\right),$$

where the second step uses equation (SEDS). Hence, g is α-EDS.

It remains to show that g α^2-approximates f. For this purpose, consider any disjoint $X_1,\dots,X_k\subseteq V$ and let $X=\bigcup_{i=1}^{k}X_i$ denote their union. Since f is α-SIM, it holds that

$$g\left(X\right)=\frac{1}{\alpha}f\left(X,\emptyset,\dots,\emptyset\right)\leq f\left(X_1,\dots,X_k\right)\leq\alpha\cdot f\left(X,\emptyset,\dots,\emptyset\right)=\alpha^2\cdot g\left(X\right).\qquad\blacktriangleleft$$

By combining Lemma 20 and Theorem 19, we can deduce the following result.

▶ **Theorem 21.** *For any α-SEDS k-set function f on V, there exists a* GMC *instance h that $\alpha^{n+4}\left(n^3+2n\right)$-approximates f, where $n=|V|$.*

For SDS languages, we can provide an even tighter result.

▶ **Theorem 22.** *For any α-SDS k-set function f on V, there exists a superadditive set function g that $n\alpha^{n+1}$-approximates f, where $n=|V|$.*

Based on these approximations, we now show our main tractability theorem, which in places closely follows the proof of [10, Theorem 5.18].

▶ **Theorem 23.** *Let Γ be a SEDS language. Then Γ is globally ℓ-tractable if it is SDS or if the reduced language $\mathrm{fix}_0\left(\Gamma\right)$ is globally ℓ-tractable.*

Proof. Let Γ be an SEDS language on domain D. Then every weighted relation $\gamma\in\Gamma$ corresponds under normalisation to a k-set function f_γ. Furthermore, weighted relations in Γ are of bounded arity. If Γ is SDS, Theorem 22 implies that for some $\alpha\in\mathbb{N}$, every such k-set function f_γ can be α-approximated by a superadditive set function h_γ. In the following, we treat h_γ as a GMC instance without any edge weights. If Γ is not SDS, we can still α-approximate every k-set function f_γ by a GMC instance h_γ according to Theorem 21, but there is no restriction on the edge weights.

Let $l:D\to\mathbb{N}_0$ be a fixed lower bound and consider any $\mathrm{VCSP}_l\left(\Gamma\right)$ instance I with objective function

$$\phi_I\left(x_1,\dots,x_n\right)=\sum_{i=1}^{t}w_i\cdot\gamma_i\left(\boldsymbol{x}^i\right).$$

Let f_I be the k-set function corresponding under normalisation to the objective function ϕ_I. We construct a GMC instance h that α-approximates f_I.

For $i \in [t]$, we relabel the vertices of h_{γ_i} to match the variables in the scope $\boldsymbol{x}^i$ of the i-th 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 h_{γ_i} and the superadditive set function by w_i (note that non-negative scaling preserves superadditivity). Instance h is then obtained by adding up GMC instances h_{γ_i} for all $i \in [t]$. In the following, we treat h as a $\mathrm{GMC}_{l^*}^{l(0)}$ instance, where $l^* = \sum_{i=1}^k l(i)$. Note that if Γ is SDS, h has zero edge weights.

Let $X_0, \ldots, X_k$ be a partition of $[n]$ such that $f_I(X_1, \ldots, X_k)$ is minimal among all partitions satisfying $|X_d| \geq l(d)$ for all $d \in D$. In other words, $X_0, \ldots, X_k$ corresponds to an optimal assignment for instance I. Let $X = \bigcup_{d=1}^k X_d$ denote all indices with non-zero labels. In addition, let $Y \subseteq [n]$ denote an optimal solution of the $\mathrm{GMC}_{l^*}^{l(0)}$ instance h and let $\lambda = h(Y)$.

Since $|Y| \geq l^*$, there must exist some partition $Y_1, \ldots, Y_k$ of Y such that $|Y_d| \geq l(d)$ for all $1 \leq d \leq k$. Because h α-approximates f_I, it holds that

$$\lambda \leq h(X) \leq f_I(X_1, \ldots, X_k) \leq f_I(Y_1, \ldots, Y_k) \leq \alpha \cdot h(Y) = \alpha \cdot \lambda.$$

Hence, X is an α-optimal solution of h.

As discussed in Section 2, it can be determined in polynomial time whether $\lambda = 0$, $\lambda = \infty$ or $0 < \lambda < \infty$. Furthermore, if $\lambda = 0$, a solution Z such that $h(Z) = 0$ can be found. Because Z must have size $|Z| \geq l^*$ as a solution of $\mathrm{GMC}_{l^*}^{l(0)}$ instance h, we can select some partition $Z_1, \ldots, Z_k$ of Z such that $|Z_d| \geq l(d)$ for all $1 \leq d \leq k$. Since h α-approximates f_I, it must hold $f_I(Z_1, \ldots, Z_k) \leq \alpha \cdot h(Z) = 0$, meaning that $Z_1, \ldots, Z_k$ represents an optimal assignment for instance I.

If $\lambda = \infty$, then obviously there are no feasible solutions.

Otherwise, it holds $0 < \lambda < \infty$. In this case, we distinguish whether Γ is SDS or $\mathrm{fix}_0(\Gamma)$ is globally ℓ-tractable.

First, we assume that Γ is SDS and hence, that h has zero edge weights. We claim that the size of X is bounded by a constant. For the sake of contradiction, assume that $|X| \geq (\alpha + 1) l^*$. Then there are disjoint subsets $Z_1, Z_2, \ldots, Z_{\alpha+1} \subseteq X$ such that $|Z_i| \geq l^*$ for all $1 \leq i \leq \alpha + 1$. Being a solution of h, every Z_i must have value at least $h(Z_i) \geq \lambda$. Based on the superadditivity of h, we arrive at the contradiction

$$(\alpha + 1) \cdot \lambda \leq h(Z_1) + \cdots + h(Z_{\alpha+1}) \leq h(X) \leq \alpha \cdot \lambda.$$

Thus, it must hold $|X| < (\alpha + 1) l^*$. This leaves less than $O\left(n^{(\alpha+1)l^*}\right)$ possible choices for X, each of which admits at most $O\left(k^{(\alpha+1)l^*}\right)$ partitions of the form $X_1 \cup \cdots \cup X_k = X$. By checking all of these, we can retrieve the sets $X_1, \ldots, X_k$ in polynomial time.

Now, we assume that $\mathrm{fix}_0(\Gamma)$ is globally ℓ-tractable. According to Corollary 9, there are only polynomially many α-optimal solutions of h, all of which can be computed in polynomial time. X must be among those solutions. By repeating the following process for all of them, we can assume that X is known, and so is $X_0 = [n] \setminus X$.

Let $D^* = D \setminus \{0\}$ and let $l_{\restriction_{D^*}} : D^* \to \mathbb{N}$ denote the restriction of l to D^*. We can efficiently find a minimal assignment for the $\mathrm{VCSP}_{l_{\restriction_{D^*}}}(\mathrm{fix}_0(\Gamma))$ instance $I_X = (X, D^*, \phi_X)$ with objective function $\phi_X = \mathrm{fix}_{0=X_0}[\phi_I]$. The sets $X_1, \ldots, X_k$ represent an assignment for I_X and, by assigning label 0 to the variables in X_0, every assignment for I_X induces an assignment for I with the same objective value. Thus, an optimal assignment for I_X induces an optimal assignment for I. ◀

▶ **Remark 24.** If Γ is SDS, the algorithm presented in Theorem 23 can in fact, for every fixed lower bound $l : D \to \mathbb{N}_0$ and every $\mathrm{VCSP}_l(\Gamma)$ instance I with optimal value $0 < \lambda < \infty$, enumerate all optimal solutions of I in polynomial time.

If Γ is SEDS and $\mathrm{fix}_0(\Gamma)$ is globally ℓ-tractable, this property holds true under the condition that for every $\mathrm{VCSP}_l(\mathrm{fix}_0(\Gamma))$ instance with optimal value $0 < \lambda < \infty$, all optimal solutions can be enumerated in polynomial time.

To complete our analysis of SEDS languages, we will now focus on the case that a language is not SDS and that $\mathrm{fix}_0(\Gamma)$ is globally s-intractable. Going even beyond SEDS, our main hardness result is that SIM languages are globally s-intractable under those circumstances. We only give a brief sketch of the proof here and provide the full proof in the full version of the paper [19].

▶ **Theorem 25.** *Let Γ be a valued constraint language over domain D that is SIM, but not SDS, and let $\mathrm{fix}_0(\Gamma)$ be globally s-intractable. Then Γ is globally s-intractable.*

Proof sketch. Since Γ is not SDS, there must exist a weighted relation $\gamma \in \Gamma$ of some arity r and disjoint $X_1, \ldots, X_k, Y_1, \ldots, Y_k \subseteq [r]$ such that, in violation of equation (SDS), the k-set function f corresponding under normalisation to γ satisfies

$$f(X_1, \ldots, X_k) \gg f(X_1 \cup Y_1, \ldots, X_k \cup Y_k).$$

Given any $\mathrm{VCSP}_s(\mathrm{fix}_0(\Gamma))$ instance $I^* = (V, D^*, \phi_I^*)$ on domain $D^* = D \setminus \{0\}$, we construct a $\mathrm{VCSP}_s(\Gamma)$ instance $I = (V \cup \{z\}, D, \phi_I)$ instance as follows. Let z denote an additional variable. The objective function ϕ_I consists of two components. The first component utilises the relation γ to ensure that any optimal assignment s for I must satisfy $s(z) = 0$ and $s(x) \neq 0$ for all $x \in V$. The second component models ϕ_I^* by replacing the constraints by corresponding weighted relations from Γ, where pinning values to label 0 is simulated by plugging in z.

This way, a solution for I^* can be obtained by solving I, thereby reducing $\mathrm{VCSP}_s(\mathrm{fix}_0(\Gamma))$ to $\mathrm{VCSP}_s(\Gamma)$.   ◀

On the Boolean domain, we obtain a complete classification of lower-bounded VCSPs, which coincides with the classification of Boolean surjective VCSPs provided by [10].

▶ **Theorem 26.** *Let Γ be a Boolean language. Then Γ is globally ℓ-tractable if and only it is globally s-tractable. Otherwise, Γ is globally ℓ-intractable.*

Moreover, we can now classify lower-bounded VCSPs over SEDS languages on three-element domains.

▶ **Theorem 27.** *Let Γ be a SEDS language on domain $D = \{0, 1, 2\}$. Then Γ is globally ℓ-tractable if it is SDS or if $\mathrm{fix}_0(\Gamma)$ is globally ℓ-tractable, and globally ℓ-intractable otherwise.*

Proof. If Γ is SDS or $\mathrm{fix}_0(\Gamma)$ globally ℓ-tractable, the statement follows from Theorem 23. Otherwise, $\mathrm{fix}_0(\Gamma)$ must be globally s-intractable by Theorem 26 and the dichotomy from [10, Theorem 3.2]. Hence, Γ is globally s-intractable by Theorem 25, which gives the result.   ◀

5   Conclusions

Based on the newly introduced Bounded Generalised Min-Cut problem and its tractability, which might be of independent interest, we have provided a conditional complexity classification of surjective and lower-bounded SEDS VCSPs on non-Boolean domains. Consequently, we obtained a dichotomy theorem with respect to ℓ-tractability for Boolean domains as well as, more interestingly, for SEDS languages on three-element domains.

While our results only pertain to languages that admit multimorphism $\langle c_d \rangle$ for some label d we expect our results and techniques to be useful in identifying new s-tractable and ℓ-tractable languages going beyond those admitting $\langle c_d \rangle$.

As mentioned in Section 1, globally tractable languages that include constant relations are also s-tractable. It is easy to show the same for global ℓ-tractability. For example, this shows that well-studied sources of tractability such as submodularity [22] and its generalisation k-submodularity [14], which are known to be globally tractable [17], are also globally ℓ-tractable.

What other non-Boolean languages are s-tractable and ℓ-tractable? Our results are a first step in this direction. In all cases we encountered global s-(in)tractability coincides with global ℓ-(in)tractability. We do not know whether this is true in general.

The natural next step is to consider languages on three-element domains. As is often the case in the (V)CSP literature, languages on three-element domains are significantly more complex than Boolean languages; for instance, compare two-element CSPs [20] and three-element CSPs [2]. As a concrete open problem (of a surjective VCSP on a three-element domain), the *3-No-Rainbow-Colouring* problem [1] asks to colour the vertices of a three-regular hypergraph such that every colour is used at least once, while no hyperedge attains all three colours. The complexity of this problem is open both the in the decision setting (is there a colouring) and also in the optimisation setting (what is the maximum number of properly colourable hyperedges/minimum number of improperly colourable hyperedges).

References

1   Manuel Bodirsky, Jan Kára, and Barnaby Martin. The complexity of surjective homomorphism problems – a survey. *Discrete Applied Mathematics*, 160(12):1680–1690, 2012. `doi:10.1016/j.dam.2012.03.029`.

2   Andrei Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *Journal of the ACM*, 53(1):66–120, 2006. `doi:10.1145/1120582.1120584`.

3   Andrei Bulatov, Peter Jeavons, and Andrei Krokhin. Classifying the Complexity of Constraints using Finite Algebras. *SIAM Journal on Computing*, 34(3):720–742, 2005. `doi:10.1137/S0097539700376676`.

4   Andrei A. Bulatov. A Dichotomy Theorem for Nonuniform CSPs. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS'17)*, pages 319–330, 2017. `doi:10.1109/FOCS.2017.37`.

5   Andrei A. Bulatov and Dániel Marx. The complexity of global cardinality constraints. *Logical Methods in Computer Science*, Volume 6, Issue 4, 2010. `doi:10.2168/LMCS-6(4:4)2010`.

6   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. `doi:10.1016/j.artint.2006.04.002`.

7   Nadia Creignou and Jean-Jacques Hébrard. On Generating All Solutions of Generalized Satisfiability Problems. *Informatique Théorique et Applications*, 31:499–511, November 1997. URL: `http://www.numdam.org/article/ITA_1997__31_6_499_0.pdf`.

8 Elias Dahlhaus, David S. Johnson, Christos H. Papadimitriou, Paul D. Seymour, and Mihalis Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23(4):864–894, 1994. `doi:10.1137/S0097539792225297`.

9 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. `doi:10.1137/S0097539794266766`.

10 Peter Fulla, Hannes Uppman, and Stanislav Živný. The complexity of Boolean surjective general-valued CSPs. *ACM Transactions on Computation Theory*, 11(1), 2018. Article No. 4. `doi:10.1145/3282429`.

11 Olivier Goldschmidt and Dorit S. Hochbaum. A Polynomial Algorithm for the k-cut Problem for Fixed k. *Mathematics of Operations Research*, 19(1):24–37, 1994. `doi:10.1287/moor.19.1.24`.

12 Pavol Hell and Jaroslav Nešetřil. On the complexity of H-coloring. *Journal of Combinatorial Theory, Series B*, 48(1):92–110, 1990. `doi:10.1016/0095-8956(90)90132-J`.

13 Pavol Hell and Jaroslav Nešetřil. Colouring, constraint satisfaction, and complexity. *Computer Science Review*, 2(3):143–163, 2008. `doi:10.1016/j.cosrev.2008.10.003`.

14 Anna Huber and Vladimir Kolmogorov. Towards Minimizing k-Submodular Functions. In *Proceedings of the 2nd International Symposium on Combinatorial Optimization (ISCO'12)*, volume 7422 of *Lecture Notes in Computer Science*, pages 451–462. Springer, 2012. `doi:10.1007/978-3-642-32147-4_40`.

15 Peter Jeavons, David Cohen, and Marc Gyssens. Closure Properties of Constraints. *Journal of the ACM*, 44(4):527–548, July 1997. `doi:10.1145/263867.263489`.

16 Vladimir Kolmogorov, Andrei Krokhin, and Michal Rolínek. The complexity of general-valued CSPs. *SIAM Journal on Computing*, 46(3):1087–1110, 2017. `doi:10.1137/16M1091836`.

17 Vladimir Kolmogorov, Johan Thapper, and Stanislav Živný. The power of linear programming for general-valued CSPs. *SIAM Journal on Computing*, 44(1):1–36, 2015. `doi:10.1137/130945648`.

18 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 Berlin Heidelberg, 2015. `doi:10.1007/978-3-662-47672-7_69`.

19 Gregor Matl and Stanislav Živný. Beyond Boolean Surjective VCSPs. Technical report, arXiv, January 2019. `arXiv:1901.07107`.

20 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. `doi:10.1145/800133.804350`.

21 Thomas Schiex, Hélène Fargier, and Gérard Verfaillie. Valued Constraint Satisfaction Problems: Hard and Easy Problems. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 631–637, 1995. URL: `http://ijcai.org/Proceedings/95-1/Papers/083.pdf`.

22 Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2003.

23 Johan Thapper and Stanislav Živný. The Complexity of Finite-Valued CSPs. *Journal of the ACM*, 63(4):37:1–37:33, September 2016. `doi:10.1145/2974019`.

24 D. Zhuk. A Proof of CSP Dichotomy Conjecture. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS'17)*, pages 331–342, 2017. `doi:10.1109/FOCS.2017.38`.