Min CSP on Four Elements: Moving Beyond Submodularity

Peter Jonsson[1][*], Fredrik Kuivinen[2], and Johan Thapper[3][**]

[1] Department of Computer and Information Science,
Linköpings universitet,
SE-581 83 Linköping, Sweden
`petej@ida.liu.se`
[2] `frekui@gmail.com`
[3] École polytechnique, Laboratoire d'informatique (LIX),
91128 Palaiseau Cedex, France
`thapper@lix.polytechnique.fr`

Abstract. We report new results on the complexity of the valued constraint satisfaction problem (VCSP). Under the unique games conjecture, the approximability of finite-valued VCSP is fairly well-understood. However, there is yet no characterisation of VCSPs that can be solved exactly in polynomial time. This is unsatisfactory, since such results are interesting from a combinatorial optimisation perspective; there are deep connections with, for instance, submodular and bisubmodular minimisation. We consider the Min and Max CSP problems (*i.e.* where the cost functions only attain values in $\{0, 1\}$) over four-element domains and identify all tractable fragments. Similar classifications were previously known for two- and three-element domains. In the process, we introduce a new class of tractable VCSPs based on a generalisation of submodularity. We also extend and modify a graph-based technique by Kolmogorov and Živný (originally introduced by Takhanov) for efficiently obtaining hardness results in our setting. This allow us to prove the result without relying on computer-assisted case analyses (which otherwise are fairly common when studying the complexity and approximability of VCSPs.) The hardness results are further simplified by the introduction of powerful reduction techniques.

Keywords: constraint satisfaction problems, combinatorial optimisation, computational complexity, submodularity

1 Introduction

This paper concerns the computational complexity of an optimisation problem with strong connections to the *constraint satisfaction problem* (CSP). An instance of the constraint satisfaction problem consists of a finite set of variables,

---

a set of values (the domain), and a finite set of constraints. The goal is to determine whether there is an assignment of values to the variables such that all the constraints are satisfied. CSPs provide a general framework for modelling a variety of combinatorial decision problems [6, 8].

Various optimisation variations of the constraint satisfaction framework have been proposed and many of them can be seen as special cases of the valued constraint satisfaction problem (VCSP), introduced by Schiex et al. [20]. This is an optimisation problem which is general enough to express such diverse problems as MAX CSP, where the goal is to maximise the number of satisfied constraints, and the minimum cost homomorphism problem (Min HOM), where all constraints must be satisfied, but each variable-value tuple in the assignment is given an independent cost. To accomplish this, instances of the VCSP assign costs (possibly infinite) to individual tuples of the constraints. It is then convenient to replace relations by *cost functions*, *i.e.* functions from tuples of the domain to some set of costs. This set of costs can be relatively general, but much is captured by using $\mathbb{Q}_{\geq 0} \cup \{\infty\}$, where $\mathbb{Q}_{\geq 0}$ denotes the set of non-negative rational numbers. We arrive at the following formal definition.

Definition 1. *Let D be a finite domain, and let Γ be a set of functions $f_i : D^{k_i} \to \mathbb{Q}_{\geq 0} \cup \{\infty\}$. By VCSP($\Gamma$) we denote the following minimisation problem:*

Instance: *A set of variables V, and a sum $\sum_{i=1}^{m} \varrho_i f_i(\boldsymbol{x_i})$, where $\varrho_i \in \mathbb{Q}_{\geq 0}$, $f_i \in \Gamma$, and $\boldsymbol{x_i}$ is a list of k_i variables from V.*
Solution: *A function $\sigma : V \to D$.*
Measure: *$m(\sigma) = \sum_{i=1}^{m} \varrho_i f_i(\sigma(\boldsymbol{x_i}))$, where $\sigma(\boldsymbol{x_i})$ is the list of elements from D obtained by applying σ component-wise to $\boldsymbol{x_i}$.*

The set Γ is often referred to as the *constraint language*. We will use Γ as our parameter throughout the paper. For instance, when we say that a class of VCSPs X is polynomial-time solvable, then we mean that VCSP(Γ) is polynomial-time solvable for every $\Gamma \in X$. Finite-valued functions, *i.e.* functions with a range in $\mathbb{Q}_{\geq 0}$, are sometimes called *soft constraints*. A prominent example is given by functions with a range in $\{0, 1\}$; they can be used to express instances of the well-known MIN CSP and MAX CSP problems (which, for instance, include MAX k-CUT, MAX k-SAT, and NEAREST CODEWORD as subproblems). On the other side we have *crisp constraints* which represent the standard type of CSP constraints. These can be expressed by cost functions taking values in $\{0, \infty\}$.

A systematic study of the computational complexity of the VCSP was initiated by Cohen et al. [4]; for instance, they prove a complexity dichotomy for VCSP(Γ) over two-element domains. This was the starting point for an intensive research effort leading to a large number of complexity results for VCSP: examples include complete classifications of conservative constraint languages (*i.e.* languages containing all unary cost functions) [7, 14, 13], $\{0, 1\}$ languages on three elements [11], languages containing a single $\{0, 1\}$ cost function [12], and arbitrary languages with $\{0, \infty\}$ cost functions [22]. We note that some of these results have been proved by computer-assisted search—something that drastically reduces the readability, and insight gained from the proofs. We also

note that there is no generally accepted conjecture stating which VCSPs are polynomial-time solvable.

The picture is clearer when considering the approximability of finite-valued VCSP. Raghavendra [19] have presented algorithms for approximating any finite-valued VCSP. These algorithms achieve an optimal approximation ratio for the constraint languages that cannot be solved to optimality in polynomial time, given that the unique games conjecture (UGC) is true. For the constraint languages that can be solved to optimality, one gets a PTAS from these algorithms. Furthermore, no characterisation of the set of constraint languages that can be solved to optimality follows from Raghavendra's result. Thus, Raghavendra's result does not imply the complexity results discussed above (not even conditionally under the UGC).

The goal of this paper is to prove a dichotomy result for VCSP with $\{0,1\}$ cost functions over four-element domains: we show that every such problem is either solvable in polynomial time or NP-hard. Such a dichotomy result is not known for CSPs on four-element domains (and, consequently, not for unrestricted VCSPs on four-element domains). Our result proves that, in contrast to the two-element, three-element, and conservative case, submodularity is not the only source of tractability. In order to outline the proof, let Γ denote a constraint language with $\{0,1\}$ cost functions over a four-element domain D. We will need two tractability results in our classification. The first one is well-known: if every function in Γ is submodular on a chain (*i.e.* a total ordering of D), then VCSP(Γ) is solvable in polynomial time. The second result is new and can be found in Section 3: we introduce *1-defect chain multimorphisms* and prove that if Γ has such a multimorphism, then VCSP(Γ) is tractable. A multimorphism is, loosely speaking, a pair of functions such that Γ satisfies certain invariance properties under them. The algorithm we present is based on a combination of submodular and bisubmodular minimisation [9, 17, 21].

The hardness part of the proof consists of four parts (Sections 4–7). We begin by introducing some tools in Section 4 and 5. Section 4 concerns the problem of adding (crisp) constant unary relations to Γ without changing the computational complexity of the resulting problem. The main tool for doing this is using the concept of *indicator problems* introduced by Jeavons et al. [10] (see also Cohen et al. [3]). Section 5 introduces a graph construction for studying Γ. In principle, this graph provides information about the complexity of VCSP(Γ) based on the two-element sublanguages of Γ. Similar graphs has been used repeatedly in the study of VCSP, *cf.* [1, 14, 22]. Equipped with these tools, we determine the complexity of VCSP(Γ) over a four-element domain in Section 6. The graph introduced in Section 5 allows us to prove that, when Γ is a *core* (*cf.* Section 4), VCSP(Γ) is polynomial-time solvable if and only if Γ is submodular on a chain or Γ has a 1-defect chain multimorphism (Theorem 26). Some proofs of intermediate results are deferred to Section 7.

2 Preliminaries

Throughout this paper, we will assume that Γ is a finite set of $\{0,1\}$-valued functions. By Min $\mathrm{CSP}(\Gamma)$ we denote the problem $\mathrm{VCSP}(\Gamma)$. It turns out to be convenient to introduce a generalisation of this problem in which we allow additional constraints on the solutions. From a VCSP perspective, this means that we allow *crisp* as well as $\{0,1\}$-valued cost functions. To make the distinction clear, and since we will not be using any *mixed* cost functions, we represent the crisp constraints with relations instead of $\{0,\infty\}$-valued cost functions.

Definition 2. *Let Γ be a set of $\{0,1\}$-valued functions on a domain D, and let Δ be a set of finitary relations on D. By Min $\mathrm{CSP}(\Gamma, \Delta)$ we denote the following minimisation problem:*

Instance: *A Min $\mathrm{CSP}(\Gamma)$-instance $\mathcal{I}$, and a finite set of constraint applications $\{(\boldsymbol{y_j}; R_j)\}$, where $R_j \in \Delta$ and $\boldsymbol{y_j}$ is a matching list of variables from V.*
Solution: *A solution σ to $\mathcal{I}$ such that $\sigma(\boldsymbol{y_j}) \in R_j$ for all j.*
Measure: *The measure of σ as a solution to $\mathcal{I}$.*

We will generally omit the parenthesis surrounding singletons in unary relations, as in the following definition: let $\mathcal{C}_D = \{\{d\} \mid d \in D\}$ be the set of constant unary relations over D.

2.1 Expressive power and weighted pp-definitions

It is often possible to enrich a set of functions Γ without changing the computational complexity of Min CSP. In this paper, we will make use two distinct, but related notions aimed at this purpose.

Definition 3. *Let $\mathcal{I}$ be an instance of Min $\mathrm{CSP}(\Gamma, \Delta)$, and let $\boldsymbol{x} = (x_1, \ldots, x_s)$ be a sequence of distinct variables from $V(\mathcal{I})$. Let*

$$\pi_{\boldsymbol{x}} \mathrm{Optsol}(\mathcal{I}) = \{(\sigma(x_1), \ldots, \sigma(x_s)) \mid \sigma \text{ is an optimal solution to } \mathcal{I}\},$$

i.e. the projection of the set of optimal solutions onto $\boldsymbol{x}$. We say that such a relation has a weighted pp-definition in (Γ, Δ). Let $\langle \Gamma, \Delta \rangle_w$ denote the set of relations which have a weighted pp-definition in (Γ, Δ).

For an instance $\mathcal{J}$ of Min CSP, we define $\mathsf{Opt}(\mathcal{J})$ to be the optimal value of a solution to $\mathcal{J}$, and to be undefined if no solution exists. The following definition is a variation of the concept of the *expressive power* of a valued constraint language, see for example Cohen et al. [4].

Definition 4. *Let $\mathcal{I}$ be an instance of Min $\mathrm{CSP}(\Gamma, \Delta)$, and let $\boldsymbol{x} = (x_1, \ldots, x_k)$ be a sequence of distinct variables from $V(\mathcal{I})$. Define the function $\mathcal{I}_{\boldsymbol{x}} : D^k \to \mathbb{Q}_{\geq 0}$ by letting $\mathcal{I}_{\boldsymbol{x}}(a_1, \ldots, a_k) = \mathsf{Opt}(\mathcal{I} \cup \{(x_i; \{a_i\}) \mid 1 \leq i \leq k\})$. We say that $\mathcal{I}_{\boldsymbol{x}}$ is expressible over (Γ, Δ). Let $\langle \Gamma, \Delta \rangle_{fn}$ denote the set of total functions expressible over (Γ, Δ).*

Proposition 5. *Let $\Gamma' \subseteq \langle \Gamma, \Delta \rangle_{fn}$ and $\Delta' \subseteq \langle \Gamma, \Delta \rangle_w$ be finite sets. Then,* MIN CSP(Γ', Δ') *is polynomial-time reducible to* MIN CSP(Γ, Δ).

Proof. The reduction from MIN CSP(Γ', Δ') to MIN CSP(Γ, Δ') is a special case of Theorem 3.4 in [4]. We allow weights as a part of our instances, but this makes no essential difference.

For the remaining part, we will assume that $\Delta' \setminus \Delta$ contains a single relation $R = \pi_{\boldsymbol{x}} \text{Optsol}(\mathcal{J})$. The case when $\Delta' \setminus \Delta = \{R_1, \ldots, R_k\}$, for $k > 1$ can be handled by eliminating one relation at a time using the same argument. Let $\mathcal{I}'$ be an instance of MIN CSP(Γ, Δ'). For each application $(\boldsymbol{u_i}; R)$, $i = 1, \ldots, t$, we create a copy $\mathcal{J}_i$ of $\mathcal{J}$ in which the variables $\boldsymbol{x}$ have been replaced by $\boldsymbol{u_i}$. We now create an instance $\mathcal{I}$ of MIN CSP(Γ, Δ) as follows: let $V(\mathcal{I}) = (\bigcup_{i=1}^{t} V(\mathcal{J}_i)) \cup V(\mathcal{I}')$, $S(\mathcal{I}) = S(\mathcal{I}') + M \cdot \sum_{i=1}^{t} S(\mathcal{J}_i)$, and let the set of constraint applications of $\mathcal{I}$ consist of all applications from $\mathcal{I}'$ apart from those involving the relation R, and all applications from $\mathcal{J}_i$, $i = 1, \ldots, t$. We will choose M large enough, so that if $\mathcal{I}'$ is satisfiable, then in any optimal solution σ to $\mathcal{I}$, the restriction of σ to the set $V(\mathcal{J}_i)$ is forced to be an optimal solution to the instance $\mathcal{J}_i$. It then follows that $\sigma(\boldsymbol{u_i}) \in R$, so we can recover an optimal solution to $\mathcal{I}'$ from σ. The value of M is chosen as follows: if all solutions to $\mathcal{J}$ have the same measure, we can let $M = 0$. Otherwise, let $\delta > 0$ be the minimal difference in measure between a sub-optimal solution, and an optimal solution to $\mathcal{J}$. Assume that $S(\mathcal{I}') = \sum_{i=1}^{m} \varrho_i f_i(\boldsymbol{x_i})$, and let $U = \sum_{i=1}^{m} \varrho_i$. Note that if σ is any solution to the instance obtained from $\mathcal{I}'$ by removing all constraint applications, then $m(\sigma) \leq U$. We can then let $M = (U + 1)/\delta$; the representation size of M is linearly bounded in the size of the instance $\mathcal{I}'$. It is easy to check that if $\mathcal{I}$ is unsatisfiable, or if $\text{Opt}(\mathcal{I}) > U + M \cdot t \cdot \text{Opt}(\mathcal{J})$, then $\mathcal{I}'$ is unsatisfiable. Otherwise $\text{Opt}(\mathcal{I}') = \text{Opt}(\mathcal{I}) - M \cdot t \cdot \text{Opt}(\mathcal{J})$. $\qquad \square$

2.2 Multimorphisms and submodularity

We now turn our attention to *multimorphisms* and tractable minimisation problems. Let D be a finite set. Let $f : D^k \to D$ be a function, and let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \in D^n$, with components $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{in})$. Then, we let $f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$ denote the n-tuple $(f(x_{11}, \ldots, x_{k1}), \ldots, f(x_{1n}, \ldots, x_{kn}))$.

A (binary) *multimorphism* of Γ is a pair of functions $f, g : D^2 \to D$ such that for any $h \in \Gamma$, and matching tuples $\boldsymbol{x}$ and $\boldsymbol{y}$,

$$h(f(\boldsymbol{x}, \boldsymbol{y})) + h(g(\boldsymbol{x}, \boldsymbol{y})) \leq h(\boldsymbol{x}) + h(\boldsymbol{y}). \tag{1}$$

The concept of multimorphisms was introduced by Cohen et al. [4] as an extension of the notion of *polymorphisms* to the analysis of the VCSP problem.

Definition 6 (Multimorphism Function Minimisation). *Let X be a finite set of triples $(D_i; f_i, g_i)$, where D_i is a finite set and f_i, g_i are functions mapping D_i^2 to D_i. MFM(X) is a minimisation problem with*

Instance: *A positive integer n, a function $j : [n] \to [|X|]$, and a function $h : D \to \mathbb{Z}$ where $D = \prod_{i=1}^{n} D_{j(i)}$. Furthermore,*

$$h(\boldsymbol{x}) + h(\boldsymbol{y}) \geq h(f_{j(1)}(x_1, y_1), f_{j(2)}(x_2, y_2), \ldots, f_{j(n)}(x_n, y_n)) +$$
$$h(g_{j(1)}(x_1, y_1), g_{j(2)}(x_2, y_2), \ldots, g_{j(n)}(x_n, y_n))$$

for all $\boldsymbol{x}, \boldsymbol{y} \in D$. The function h is given to the algorithm as an oracle, i.e., for any $\boldsymbol{x} \in D$ we can query the oracle to obtain $h(\boldsymbol{x})$ in unit time.
Solution: *A tuple $\boldsymbol{x} \in D$.*
Measure: *The value of $h(\boldsymbol{x})$.*

For a finite set X we say that $\mathrm{MFM}(X)$ is *oracle-tractable* if it can be solved in time $O(n^c)$ for some constant c. It is not hard to see that if (f, g) is a multimorphism of Γ, and $\mathrm{MFM}(D; f, g)$ is oracle-tractable, then $\mathrm{MIN} \; \mathrm{CSP}(\Gamma)$ is tractable.

We now give two examples of oracle-tractable problems. A partial order on D is called a *lattice* if every pair of elements $a, b \in D$ has a greatest lower bound $a \wedge b$ (meet) and a least upper bound $a \vee b$ (join). A *chain* on D is a lattice which is also a total order.

For $i = 1, \ldots, n$, let L_i be a lattice on D_i. The *product lattice* $L_1 \times \cdots \times L_n$ is defined on the set $D_1 \times \cdots \times D_n$ by extending the meet and join component-wise: for $\boldsymbol{a} = (a_1, \ldots, a_n)$ and $\boldsymbol{b} = (b_1, \ldots, b_n)$, let $\boldsymbol{a} \wedge \boldsymbol{b} = (a_1 \wedge b_1, \ldots, a_n \wedge b_n)$, and let $\boldsymbol{a} \vee \boldsymbol{b} = (a_1 \vee b_1, \ldots, a_n \vee b_n)$.

A function $f : D^k \to \mathbb{Z}$ is called *submodular* on the lattice $L = (D; \wedge, \vee)$ if

$$f(\boldsymbol{a} \wedge \boldsymbol{b}) + f(\boldsymbol{a} \vee \boldsymbol{b}) \leq f(\boldsymbol{a}) + f(\boldsymbol{b})$$

for all $\boldsymbol{a}, \boldsymbol{b} \in D^k$. A set of functions Γ is said to be submodular on L if every function in Γ is submodular on L. This is equivalent to $(\wedge, \vee)$ being a multimorphism of Γ. It follows from known algorithms for submodular function minimisation that $\mathrm{MFM}(X)$ is oracle-tractable for any finite set X of finite *distributive lattices* (*e.g.* chains) [9, 21].

The second example is strongly related to submodularity, but here we use a partial order that is not a lattice to define the multimorphism. Let $D = \{0, 1, 2\}$, and define the functions $u, v : D^2 \to D$ by letting $u(x, y) = \min\{x, y\}$, $v(x, y) = \max\{x, y\}$ if $\{x, y\} \neq \{1, 2\}$, and $u(x, y) = v(x, y) = 0$ otherwise. We say that a function $h : D^k \to \mathbb{Z}$ is *bisubmodular* if h has the multimorphism (u, v). It is possible to minimise a k-ary bisubmodular function in time polynomial in k, provided that evaluating h on a tuple is a primitive operation [17].

3 A New Tractable Class

In this section, we introduce a new multimorphism which ensures tractability for MIN CSP (and more generally for VCSP).

Definition 7. *Let b and c be two distinct elements in D. Let $(D; <)$ be a partial order which relates all pairs of elements except for b and c. Assume that $f, g : D^2 \to D$ are two commutative functions satisfying the following conditions:*

- If $\{x,y\} \neq \{b,c\}$, *then* $f(x,y) = x \wedge y$ *and* $g(x,y) = x \vee y$.
- If $\{x,y\} = \{b,c\}$, *then* $\{f(x,y), g(x,y)\} \cap \{x,y\} = \varnothing$, *and* $f(x,y) < g(x,y)$.

We call $(D; f, g)$ *a* 1-defect chain *(over* $(D; <)$*), and say that* $\{b, c\}$ *is the* defect *of* $(D; f, g)$. *If a function has the multimorphism* (f, g), *then we also say that* (f, g) *is a* 1-defect chain multimorphism.

Three types of 1-defect chains are shown in Fig. 1(a–c). Note this is not an exhaustive list, *e.g.* for $|D| > 4$, there are 1-defect chains similar to Fig. 1(b), but with $f(b, c) < g(b, c) < b, c$. When $|D| = 4$, type (b) is precisely the product lattice shown in Fig. 1(d). We denote this lattice by L_{ad}
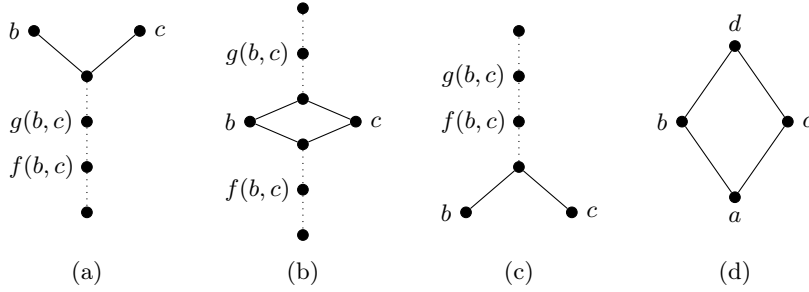

Fig. 1. Three types of 1-defect multimorphisms with defect $\{b, c\}$. (a) $f(b, c) < g(b, c) < b, c$. (b) $f(b, c) < b, c < g(b, c)$. (c) $b, c < f(b, c) < g(b, c)$. (d) The Hasse diagram of the lattice L_{ad}, a special case of (b).

Example 8. Let $D = \{a, b, c, d\}$, and assume that $(D; f, g)$ is a 1-defect chain, with defect $\{b, c\}$, and that $a = f(b, c), d = g(b, c)$. If $a < b, c < d$, then f and g are the meet and join of L_{ad}, *cf.* Fig. 1(d). When $a < d < b, c$ we have the situation in Fig. 1(a), and when $b, c < a < d$ we have the situation in Fig. 1(c). In the two latter cases, f and g are given by the two following multimorphisms (rows and columns are listed in the order a, b, c, d, *e.g.* $g_1(c, d) = c$):

$$
f_1 : \begin{matrix} a\ a\ a\ a \\ a\ b\ a\ d \\ a\ a\ c\ d \\ a\ d\ d\ d \end{matrix}
\qquad
g_1 : \begin{matrix} a\ b\ c\ d \\ b\ b\ d\ b \\ c\ d\ c\ c \\ d\ b\ c\ d \end{matrix}
\qquad
f_2 : \begin{matrix} a\ b\ c\ a \\ b\ b\ a\ b \\ c\ a\ c\ c \\ a\ b\ c\ d \end{matrix}
\qquad
g_2 : \begin{matrix} a\ a\ a\ d \\ a\ b\ d\ d \\ a\ d\ c\ d \\ d\ d\ d\ d \end{matrix}
$$

The proof of tractability for languages with 1-defect chain multimorphisms is inspired by Krokhin and Larose's [15] result on maximising supermodular functions on Mal'tsev products of lattices. First we will need some notation and a general lemma on oracle-tractability of MFM problems.

For an equivalence relation θ on D we use $x[\theta]$ to denote the equivalence class containing $x \in D$. The relation θ is a *congruence* on $(D; f, g)$, if $f(x_1, y_1)[\theta] = f(x_2, y_2)[\theta]$ and $g(x_1, y_1)[\theta] = g(x_2, y_2)[\theta]$ whenever $x_1[\theta] = x_2[\theta]$ and $y_1[\theta] =$

$y_2[\theta]$. We use D/θ to denote the set $\{x[\theta] \mid x \in D\}$ and $f/\theta : (D/\theta)^2 \to D/\theta$ to denote the function $(x[\theta], y[\theta]) \mapsto f(x, y)[\theta]$.

Lemma 9. *Let f, g be two functions that map D^2 to D. If there is a congruence relation θ on $(D; f, g)$ such that 1) $MFM(D/\theta; f/\theta, g/\theta)$ is oracle-tractable; and 2) $MFM(\{(X; f|_X, g|_X) \mid X \in D/\theta\})$ is oracle-tractable, then $MFM(D; f, g)$ is oracle-tractable.*

Proof. Let $h : D^n \to \mathbb{Z}$ be the function we want to minimise. We define a new function $h' : (D/\theta)^n \to \mathbb{Z}$ by

$$h'(z_1, z_2, \ldots, z_n) = \min_{x_i \in z_i} h(x_1, x_2, \ldots, x_n).$$

It is clear that $\min_{\boldsymbol{z} \in (D/\theta)^n} h'(\boldsymbol{z}) = \min_{\boldsymbol{x} \in D^n} h(\boldsymbol{x})$. By assumption 2 in the statement of the lemma we can compute h' given $z_1, z_2, \ldots, z_n$. To simplify the notation we let $u = f/\theta$ and $v = g/\theta$. We will now prove that h' is an instance of $MFM(D/\theta; u, v)$.

Let $\boldsymbol{x}, \boldsymbol{y} \in D^k$ and choose $x_i' \in x_i[\theta]$ and $y_i' \in y_i[\theta]$ so that $h'(\boldsymbol{x}[\theta]) = h(\boldsymbol{x}')$ and $h'(\boldsymbol{y}[\theta]) = h(\boldsymbol{y}')$. We then have

$$h'(\boldsymbol{x}[\theta]) + h'(\boldsymbol{y}[\theta]) = h(\boldsymbol{x}') + h(\boldsymbol{y}') \tag{2}$$
$$\geq h(f(\boldsymbol{x}', \boldsymbol{y}')) + h(g(\boldsymbol{x}', \boldsymbol{y}')) \tag{3}$$
$$\geq h'(f(\boldsymbol{x}', \boldsymbol{y}')[\theta]) + h'(g(\boldsymbol{x}', \boldsymbol{y}')[\theta]) \tag{4}$$
$$= h'(f(\boldsymbol{x}, \boldsymbol{y})[\theta]) + h'(g(\boldsymbol{x}, \boldsymbol{y})[\theta])) \tag{5}$$
$$= h'(u(\boldsymbol{x}[\theta], \boldsymbol{y}[\theta])) + h'(v(\boldsymbol{x}[\theta], \boldsymbol{y}[\theta])). \tag{6}$$

Here (2) follows from our choice of $\boldsymbol{x}'$ and $\boldsymbol{y}'$, (3) follows from the fact that h is an instance of $MFM(D; f, g)$, (4) follows from the definition of h', and finally (5) and (6) follows as θ is a congruence relation of f and g. Hence, h' is an instance of $MFM(D/\theta; u, v)$ and can be minimised in polynomial time by the first assumption in the lemma. $\square$

Armed with this lemma and the oracle-tractability of submodular and bisubmodular functions described in the previous section, we can now present a new tractable class of MIN CSP-problems.

Proposition 10. *If Γ has a 1-defect chain multimorphism, then MIN CSP(Γ) is tractable.*

Proof. Assume that Γ has a 1-defect chain multimorphism (f, g) over $(D; <)$ with defect $\{b, c\}$. We prove that $MFM(D; f, g)$ is oracle-tractable.

Assume that b and c are maximal elements, *i.e.* $x < b, c$ for all $x \in D \backslash \{b, c\}$. In this case the equivalence relation θ with classes $A = D \backslash \{b, c\}$, $B = \{b\}$, $C = \{c\}$ is a congruence relation of $(D; f, g)$. Furthermore, $MFM(\{A, B, C\}; f/\theta, g/\theta)$ and $MFM(A; f|_A, g|_A)$ are oracle-tractable [17, 21]. It now follows from Lemma 9 that $MFM(D; f, g)$ is oracle-tractable. The same argument works for the case when b and c are minimal elements.

If $f(b,c) < g(b,c) < b,c$, but b and c are not maximal, then we can use the congruence relation θ' with classes $A = \{x \mid x \leq b \text{ or } x \leq c\}$ and $B = D \setminus A$. Here, $(\{A,B\}; f/\theta', g/\theta')$ and $(B; f|_B, g|_B)$ are chains, and $(A; f|_A, g|_A)$ is a 1-defect chain of the previous type. One can show that when $\mathrm{MFM}(X)$ and $\mathrm{MFM}(Y)$ are both oracle-tractable, then so is $\mathrm{MFM}(X \cup Y)$. Combining this with the technique used above, we can now solve the minimisation problem. An analogous construction works in the case when $b,c < f(b,c), g(b,c)$, using the congruence consisting of the class $\{x \mid x \geq b \text{ or } x \geq c\}$ and its complement. Finally, when $f(b,c) < b,c < g(b,c)$, we can use the congruence relation θ'' with classes $B = \{x \mid x \leq b\}$ and $C = \{x \mid x \geq c\}$. Here, $(\{B,C\}, f/\theta'', g/\theta'')$, $(B, f|_B, g|_B)$, and $(C, f|_C, g|_C)$ are all chains and thus the MFM problem for these triples is oracle-tractable [21]. □

We now turn to prove a different property of functions with 1-defect chain multimorphisms. It is based on the following result for submodular functions on chains, which was derived by Queyranne et al. [18] from earlier work by Topkis [23] (See also Burkard et al. [2]). This formulation is due to Deineko et al. [7]:

Lemma 11. *A function $f : D^k \to \mathbb{Z}$ is submodular on a chain $(D; \wedge, \vee)$ if and only if the following holds: every binary function obtained from f by replacing any given $k - 2$ variables by any constants is submodular on this chain.*

It is straightforward to extend this lemma to products of chains, such as L_{ad}. Here, we outline the proof of the corresponding property for arbitrary 1-defect chains, which will be needed in Section 6. We will use the following observation.

Definition 12. *A binary operation $f : D^2 \to D$ is called a 2-semilattice if it is idempotent, commutative, and $f(f(x,y), x) = f(x,y)$ for all $x, y \in D$.*

Proposition 13. *Let $(D; f, g)$ be a 1-defect chain with a defect on $\{b, c\}$.*

1. *If $f(b,c) < b,c$, then f is a 2-semilattice and $g(f(x,y), x) = x$ for $x, y \in D$.*
2. *If $g(b,c) > b,c$, then g is a 2-semilattice and $f(g(x,y), x) = x$ for $x, y \in D$.*
3. *For $\boldsymbol{x}, \boldsymbol{y} \in \{b,c\}^k$, we have $\{f(f(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{x}), g(f(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{x})\} = \{f(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{x}\}$ and $\{g(g(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{x}), f(g(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{x})\} = \{g(\boldsymbol{x}, \boldsymbol{y}), \boldsymbol{x}\}$.*

Proof. For $\{x, y\} \neq \{b, c\}$, the equalities $f(f(x,y), x) = f(x,y)$ and $g(f(x,y), x) = x$ follow from the underlying partial order. Assume instead that $\{x, y\} = \{b, c\}$, and that $f(x,y) < x, y$. Since $\{f(x,y), x\} \neq \{b, c\}$, we have that $f(f(x,y), x)$ is the greatest lower bound of $f(x,y)$ and x, which is $f(x,y)$. We also have that $g(f(x,y), x)$ is the lowest upper bound of $f(x,y)$ and x, which is x. An analogous argument proves (2).

The first equality of (3) follows from (1) if $f(b,c) < b,c$, and the second equality follows from (2) if $g(b,c) > b,c$. At least one of $f(b,c) < b,c$ and $g(b,c) > b,c$ holds. If both holds, there is nothing to prove, so assume that $f(b,c) < b,c$, but $g(b,c) < b,c$. We then have $g(g(x,y), x) = x$ and $f(g(x,y), x) = g(x,y)$ for $\{x,y\} = \{b,c\}$, so the second equality of (3) also holds. The remaining case follows similarly. □

9

Lemma 14. *A function $h : D^k \to \mathbb{Z}$, $k \geq 2$, has the 1-defect chain multimorphism (f, g) if and only if every binary function obtained from h by replacing any given $k - 2$ variables by any constants has the multimorphism (f, g).*

Proof. Let $\{b, c\}$ be the defect of (f, g). We prove the statement for the case $f(b, c) < b, c$. The other case follows analogously.

Every function obtained from h by fixing a number of variables is clearly invariant under every multimorphism of h.

For the opposite direction, assume that h does not have the multimorphism (f, g). We want to prove that there exist vectors $\boldsymbol{x}, \boldsymbol{y} \in D^k$ such that

$$h(\boldsymbol{x}) + h(\boldsymbol{y}) < h(f(\boldsymbol{x}, \boldsymbol{y})) + h(g(\boldsymbol{x}, \boldsymbol{y})), \tag{7}$$

with $d_H(\boldsymbol{x}, \boldsymbol{y}) = 2$, where d_H denotes the *Hamming distance* on D^k, *i.e.* the number of coordinates in which $\boldsymbol{x}$ and $\boldsymbol{y}$ differ.

Assume to the contrary that the result does not hold. We can then choose a function h of minimal arity such that

$$\min\{\, d_H(\boldsymbol{x}, \boldsymbol{y}) \mid \boldsymbol{x} \text{ and } \boldsymbol{y} \text{ satisfy } (7)\,\} > 2.$$

The arity of h must in fact be equal to the least $d_H(\boldsymbol{x}, \boldsymbol{y})$; otherwise, we could obtain a function h' of strictly smaller arity by fixing the variables in h on which $\boldsymbol{x}$ and $\boldsymbol{y}$ agree. This would contradict the minimality in the choice of h.

We will first show that it is possible to choose $\boldsymbol{x}$ and $\boldsymbol{y}$ so that $\{x_i, y_i\} \neq \{b, c\}$ for all i. Let $k_1, k_2 \geq 1$ so that $k_1 + k_2 = k$, and let $(\boldsymbol{x_1}; \boldsymbol{x_2}), (\boldsymbol{y_1}; \boldsymbol{y_2}) \in D^k$ be two vectors with $d_H((\boldsymbol{x_1}; \boldsymbol{x_2}), (\boldsymbol{y_1}; \boldsymbol{y_2})) = k$, satisfying (7). Now, assume that $(\boldsymbol{x_1}; \boldsymbol{x_2}), (\boldsymbol{y_1}; \boldsymbol{y_2}) \in \{b, c\}^k$. We then have

$$h(\boldsymbol{x_1}; \boldsymbol{x_2}) + h(\boldsymbol{y_1}; \boldsymbol{y_2}) < h(f(\boldsymbol{x_1}, \boldsymbol{y_1}); f(\boldsymbol{x_2}; \boldsymbol{y_2})) + h(g(\boldsymbol{x_1}, \boldsymbol{y_1}); g(\boldsymbol{x_2}, \boldsymbol{y_2}))$$

Since both $d_H((\boldsymbol{x_1}; \boldsymbol{x_2}), (\boldsymbol{x_1}; \boldsymbol{y_2}))$ and $d_H((\boldsymbol{y_1}; \boldsymbol{y_2}), (\boldsymbol{x_1}; \boldsymbol{y_2}))$ are strictly less that the arity of h, we have by assumption

$$h(\boldsymbol{x_1}; \boldsymbol{x_2}) + h(\boldsymbol{x_1}; \boldsymbol{y_2}) \geq h(\boldsymbol{x_1}; f(\boldsymbol{x_2}, \boldsymbol{y_2})) + h(\boldsymbol{x_1}; g(\boldsymbol{x_2}, \boldsymbol{y_2})), \text{ and}$$

$$h(\boldsymbol{y_1}; \boldsymbol{y_2}) + h(\boldsymbol{x_1}; \boldsymbol{y_2}) \geq h(f(\boldsymbol{x_1}, \boldsymbol{y_1}); \boldsymbol{y_2}) + h(g(\boldsymbol{x_1}, \boldsymbol{y_1}); \boldsymbol{y_2}).$$

By combining these inequalities, we get

$$h(\boldsymbol{x_1}; f(\boldsymbol{x_2}, \boldsymbol{y_2})) + h(f(\boldsymbol{x_1}, \boldsymbol{y_1}); \boldsymbol{y_2}) + h(\boldsymbol{x_1}; g(\boldsymbol{x_2}, \boldsymbol{y_2})) + h(g(\boldsymbol{x_1}, \boldsymbol{y_1}); \boldsymbol{y_2})$$

$$< h(f(\boldsymbol{x_1}, \boldsymbol{y_1}); f(\boldsymbol{x_2}; \boldsymbol{y_2})) + h(\boldsymbol{x_1}, \boldsymbol{y_2}) + h(g(\boldsymbol{x_1}, \boldsymbol{y_1}); g(\boldsymbol{x_2}, \boldsymbol{y_2})) + h(\boldsymbol{x_1}, \boldsymbol{y_2}).$$

Let $\boldsymbol{x} = (\boldsymbol{x_1}; f(\boldsymbol{x_2}, \boldsymbol{y_2}))$, $\boldsymbol{y} = (f(\boldsymbol{x_1}, \boldsymbol{y_1}); \boldsymbol{y_2})$, $\boldsymbol{x}' = (\boldsymbol{x_1}; g(\boldsymbol{x_2}, \boldsymbol{y_2}))$, and $\boldsymbol{y}' = (g(\boldsymbol{x_1}, \boldsymbol{y_1}); \boldsymbol{y_2})$. By Proposition 13(3), we have

$$\{f(\boldsymbol{x}, \boldsymbol{y}), g(\boldsymbol{x}, \boldsymbol{y})\} = \{(\boldsymbol{x_1}; \boldsymbol{y_2}), (f(\boldsymbol{x_1}, \boldsymbol{y_1}); f(\boldsymbol{x_2}, \boldsymbol{y_2}))\}, \text{ and}$$

$$\{f(\boldsymbol{x}', \boldsymbol{y}'), g(\boldsymbol{x}', \boldsymbol{y}')\} = \{(\boldsymbol{x_1}; \boldsymbol{y_2}), (g(\boldsymbol{x_1}, \boldsymbol{y_1}); g(\boldsymbol{x_2}, \boldsymbol{y_2}))\}.$$

Hence, we can rewrite the previous inequality:

$$h(\boldsymbol{x}) + h(\boldsymbol{y}) + h(\boldsymbol{x}') + h(\boldsymbol{y}')$$

10

$$< h(f(\boldsymbol{x}, \boldsymbol{y})) + h(g(\boldsymbol{x}, \boldsymbol{y})) + h(f(\boldsymbol{x}', \boldsymbol{y}')) + h(g(\boldsymbol{x}', \boldsymbol{y}')).$$

It follows that either the pair $\boldsymbol{x}$ and $\boldsymbol{y}$, or the pair $\boldsymbol{x}'$ and $\boldsymbol{y}'$ satisfies condition (7). Furthermore, $\{x_i, y_i\} \neq \{b, c\}$ and $\{x_i', y_i'\} \neq \{b, c\}$, for all i.

If instead we have vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ satisfying (7) such that $\{x_i, y_i\} \neq \{b, c\}$ for *some*, but not all i, then we proceed as follows. Note that $\{x_i, y_i\} \neq \{b, c\}$ implies $\{f(x_i, y_i), g(x_i, y_i)\} = \{x_i, y_i\}$. Without loss of generality, we may therefore assume that $\boldsymbol{x} = (\boldsymbol{x_1}; \boldsymbol{x_2}), \boldsymbol{y} = (\boldsymbol{y_1}; \boldsymbol{y_2}) \in D^k$, with $\boldsymbol{x_1}, \boldsymbol{y_1} \in D^{k_1}$ for $k_1 \geq 1$, are such that $f(\boldsymbol{x_1}, \boldsymbol{y_1}) = \boldsymbol{x_1}$ and $g(\boldsymbol{x_1}, \boldsymbol{y_1}) = \boldsymbol{y_1}$, possibly by first exchanging $\boldsymbol{x}$ and $\boldsymbol{y}$. For these vectors, condition (7) now reads:

$$h(\boldsymbol{x_1}; \boldsymbol{x_2}) + h(\boldsymbol{y_1}; \boldsymbol{y_2}) < h(\boldsymbol{x_1}; f(\boldsymbol{x_2}, \boldsymbol{y_2})) + h(\boldsymbol{y_1}; g(\boldsymbol{x_2}, \boldsymbol{y_2})).$$

Due to the minimality of h's arity, we must have

$$h(\boldsymbol{y_1}; \boldsymbol{x_2}) + h(\boldsymbol{y_1}; \boldsymbol{y_2}) \geq h(\boldsymbol{y_1}; f(\boldsymbol{x_2}, \boldsymbol{y_2})) + h(\boldsymbol{y_1}; g(\boldsymbol{x_2}, \boldsymbol{y_2})).$$

We therefore have

$$h(\boldsymbol{x_1}; \boldsymbol{x_2}) + h(\boldsymbol{y_1}; f(\boldsymbol{x_2}, \boldsymbol{y_2})) < h(\boldsymbol{x_1}; f(\boldsymbol{x_2}, \boldsymbol{y_2})) + h(\boldsymbol{y_1}; \boldsymbol{x_2}).$$

Let $\boldsymbol{x} = (\boldsymbol{x_1}; \boldsymbol{x_2})$ and $\boldsymbol{y} = (\boldsymbol{y_1}; f(\boldsymbol{x_2}, \boldsymbol{y_2}))$. By Proposition 13(1), f is a 2-semilattice, so we have $f(f(\boldsymbol{x_2}, \boldsymbol{y_2}), \boldsymbol{x_2}) = f(\boldsymbol{x_2}, \boldsymbol{y_2})$, and thus

$$(\boldsymbol{x_1}; f(\boldsymbol{x_2}; \boldsymbol{y_2})) = (\boldsymbol{x_1}; f(f(\boldsymbol{x_2}, \boldsymbol{y_2}), \boldsymbol{x_2}))$$
$$= f((\boldsymbol{y_1}; f(\boldsymbol{x_2}, \boldsymbol{y_2})), (\boldsymbol{x_1}; \boldsymbol{x_2})) = f(\boldsymbol{y}, \boldsymbol{x}).$$

Furthermore, $g(f(\boldsymbol{x_2}, \boldsymbol{y_2}), \boldsymbol{x_2}) = \boldsymbol{x_2}$, so

$$(\boldsymbol{y_1}; \boldsymbol{x_2}) = (\boldsymbol{y_1}; g(f(\boldsymbol{x_2}, \boldsymbol{y_2}), \boldsymbol{x_2})) = g((\boldsymbol{y_1}; f(\boldsymbol{x_2}, \boldsymbol{y_2})), (\boldsymbol{x_1}; \boldsymbol{x_2})) = g(\boldsymbol{y}, \boldsymbol{x}).$$

We therefore conclude that

$$h(\boldsymbol{x}) + h(\boldsymbol{y}) < h(f(\boldsymbol{x}, \boldsymbol{y})) + h(g(\boldsymbol{x}, \boldsymbol{y})),$$

so that condition (7) holds for $\boldsymbol{x}$ and $\boldsymbol{y}$ with $\{x_i, y_i\} \neq \{b, c\}$ for all i. From now on, we assume that $\boldsymbol{x}$ and $\boldsymbol{y}$ are chosen in this way.

Let $D' = D \setminus \{b, c\} \cup \{B\}$. For each i, let $\varphi_i : D' \to D$ be an injection which fixes $D \setminus \{b, c\}$, and sends B to b or c in such a way that $\{x_i, y_i\} \subseteq \varphi_i(D)$. Let $(D'; f', g')$ be the chain defined by $x <' y$ if $x, y \neq B$ and $x < y$, $x <' B$ if $x < b, c$, and $B <' y$ if $b, c < y$. Then, $\varphi_i(f'(x, y)) = f(\varphi_i(x), \varphi_i(y))$, and $\varphi_i(g'(x, y)) = g(\varphi_i(x), \varphi_i(y))$, for all i. Let $\varphi(\boldsymbol{z}) = (\varphi_1(z_1), \ldots, \varphi_k(z_k))$, and let $\boldsymbol{x}', \boldsymbol{y}' \in (D')^k$ be such that $\varphi(\boldsymbol{x}') = \boldsymbol{x}$ and $\varphi(\boldsymbol{y}') = \boldsymbol{y}$. Define $h'(\boldsymbol{z}') = h(\varphi(\boldsymbol{z}'))$. Then,

$$h'(\boldsymbol{x}') + h'(\boldsymbol{y}') = h(\boldsymbol{x}) + h(\boldsymbol{y}) < h(f(\boldsymbol{x}, \boldsymbol{y})) + h(g(\boldsymbol{x}, \boldsymbol{y}))$$
$$= h'(f'(\boldsymbol{x}', \boldsymbol{y}')) + h'(g'(\boldsymbol{x}', \boldsymbol{y}')).$$

It follows that h' is not submodular on (D', f', g'). By Lemma 11, there are elements $\boldsymbol{z}', \boldsymbol{w}' \in (D')^k$ with $d_H(\boldsymbol{z}', \boldsymbol{w}') = 2$ such that $h'(\boldsymbol{z}') + h'(\boldsymbol{w}') < h'(f'(\boldsymbol{z}', \boldsymbol{w}')) + h'(g'(\boldsymbol{z}', \boldsymbol{w}'))$. Hence,

$$h(\varphi(\boldsymbol{z}')) + h(\varphi(\boldsymbol{w}')) = h'(\boldsymbol{z}') + h'(\boldsymbol{w}') < h'(f'(\boldsymbol{z}', \boldsymbol{w}')) + h'(g'(\boldsymbol{z}', \boldsymbol{w}'))$$
$$= h(f(\varphi(\boldsymbol{z}'), \varphi(\boldsymbol{w}'))) + h(g(\varphi(\boldsymbol{z}'), \varphi(\boldsymbol{w}'))),$$

and $d_H(\varphi(\boldsymbol{z}'), \varphi(\boldsymbol{w}')) = 2$. This contradicts the original choice of h. $\square$

4 Endomorphisms, cores and constants

In this section, we show that under a natural condition, it is possible to add constant unary relations to Γ without changing the computational complexity of the corresponding MIN CSP-problem. Let $h : D^k \to \{0, 1\}$. A function $g : D \to D$ is called an *endomorphism of h* if for every k-tuple $(x_1, \ldots, x_k) \in D^k$, it holds that $h(x_1, \ldots, x_k) = 0 \implies h(g(x_1), \ldots, g(x_k)) = 0$. The function g is an endomorphism of Γ if it is an endomorphism of each function in Γ. The set of all endomorphisms of Γ is denoted by $\text{End}(\Gamma)$. A bijective endomorphism is called an *automorphism*. The automorphisms of Γ form a group under composition.

Definition 15. *A set of functions, Γ, is said to be a* core *if all of its endomorphisms are injective.*

The idea is that if Γ it not a core, then we can apply a non-injective endomorphism to every function in Γ, and obtain a polynomial-time equivalent problem on a strictly smaller domain. We can then use results previously obtained for smaller domains [4, 11]. Thus, we can restrict our attention to the case when Γ is a core.

Jeavons et al. [10] defined the notion of an *indicator problem of order k* for CSPs. We will exploit indicator problems of order 1 here, adapted to the setting of MIN CSP.

Definition 16. *Let Γ be a finite set of $\{0, 1\}$-valued functions over D. Let X_D denote the set containing a variable x_d for each $d \in D$, and for $\boldsymbol{a} = (a_1, \ldots, a_k) \in D^k$, let $\boldsymbol{x_a} = (x_{a_1}, \ldots x_{a_k}) \in X_D^k$. The indicator problem $\mathcal{IP}(\Gamma)$ is defined as the instance of MIN CSP(Γ) with variables X_D, and sum $\sum_{f_i \in \Gamma} \sum_{\boldsymbol{a} \in f_i^{-1}(0)} f_i(\boldsymbol{x_a})$, where k_i is the arity of the function f_i.*

Let $\iota : D \to X_D$ be the function defined by $\iota(d) = x_d$. Theorem 3.5 in [10] implies the following property of $\mathcal{IP}(\Gamma)$:

Proposition 17. *For any finite set of functions, Γ, the set of optimal solutions to $\mathcal{IP}(\Gamma)$ is equal to $\{\sigma : X_D \to D \mid \sigma \circ \iota \in \text{End}(\Gamma)\}$.*

The proof of the following result follows the lines of similar results for related problems, such as the CSP decision problem.

Proposition 18. *Let Γ be a core over D. Then, MIN CSP$(\Gamma, \mathcal{C}_D)$ is polynomial-time reducible to MIN CSP(Γ).*

Proof. Let $\mathcal{J}$ be an instance of MIN CSP$(\Gamma, \mathcal{C}_D)$. The only way for $\mathcal{J}$ to be unsatisfiable is if it contains two contradicting constraint applications $(y; \{a\})$ and $(y; \{b\})$, with $a \neq b$. This is easily checked in polynomial time.

Otherwise, Let $\boldsymbol{x}$ be a list of the variables X_D, and let $R = \pi_{\boldsymbol{x}} \text{Optsol}(\mathcal{IP}(\Gamma))$. Now modify $\mathcal{J}$ to an instance $\mathcal{J}'$ of MIN CSP(Γ, R) as follows. Add the variables in X_D to $V(\mathcal{J}')$, and add the constraint application $(\boldsymbol{x}; R)$. Furthermore, remove each constraint $(y; \{a\})$, and replace y by x_a throughout the instance. Let σ' be

an optimal solution to $\mathcal{J}'$. Since Γ is a core, $g = \sigma'|_{X_D} \circ \iota$ is an automorphism of Γ, and so is its inverse, g^{-1}. Hence, $\sigma = g^{-1} \circ \sigma'$ is also an optimal solution to $\mathcal{J}'$. From σ we easily recover a solution to $\mathcal{J}$ of equal measure, and conversely, any solution to $\mathcal{J}$ can be interpreted as a solution to $\mathcal{J}'$. It follows that we have a reduction from MIN $\mathrm{CSP}(\Gamma, \mathcal{C}_D)$ to MIN $\mathrm{CSP}(\Gamma, R)$. By Proposition 5, we finally have a reduction from MIN $\mathrm{CSP}(\Gamma, R)$ to MIN $\mathrm{CSP}(\Gamma)$. $\square$

For $a, b \in D$, let $e_{ab} : D \to D$ denote the function $e_{ab}(a) = b$ and $e_{ab}(x) = x$ for $x \neq a$.

Lemma 19. *If $e_{ab} \notin \mathrm{End}(\Gamma)$, then $\langle \Gamma, \mathcal{C}_D \rangle_{fn}$ contains a unary $\{0,1\}$-valued function u such that $u(a) = 0$ and $u(b) = 1$.*

Proof. Let $h : D^k \to \{0,1\}$ be a function in Γ, and $a_1, \ldots, a_k \in D$ be elements such that $h(a_1, \ldots, a_k) = 0$, but $h(e_{ab}(a_1), \ldots, e_{ab}(a_k)) = 1$.

Let $\mathcal{J}$ be the instance of MIN $\mathrm{CSP}(\Gamma, \mathcal{C}_D)$ with variables $V(\mathcal{J}) = X_D$, sum $S(\mathcal{J}) = h(x_{a_1}, \ldots, x_{a_k})$, and constraint applications $(x_d; \{d\})$ for $d \neq a$. Then, $u = \mathcal{J}_{x_a}$ is a unary function in $\langle \Gamma, \mathcal{C}_D \rangle_{fn}$, with $u(a) = 0$ and $u(b) = 1$. $\square$

5   A Graph of Partial Multimorphisms

Let Γ be a core over D. In this section, we define a graph $G = (V, E)$ which encodes either the **NP**-hardness of MIN $\mathrm{CSP}(\Gamma, \mathcal{C}_D)$ or provides a multimorphism for the binary functions in $\langle \Gamma, \mathcal{C}_D \rangle_{fn}$. The graph is a variation of a graph defined by Kolmogorov and Živný [14], with changes made to accommodate for additional multimorphisms.

Let V be the set of partial functions $(f, g) : D^2 \to D^2$ such that

- f and g are defined on a subset $\{a, b\} \subseteq D$;
- f and g are idempotent and commutative; and
- $\{f(a,b), g(a,b)\} = \{a,b\}$ or $\{f(a,b), g(a,b)\} \cap \{a,b\} = \varnothing$.

We do allow $a = b$ in the definition of V, *i.e.* there is precisely one vertex for each singleton in D. For $a, b \in D$, we let $G[a, b]$ denote the graph induced by the set of vertices defined on $\{a, b\}$. Let $(f_1, g_1) \in G[a_1, b_1]$ and $(f_2, g_2) \in G[a_2, b_2]$. There is an edge in E between (f_1, g_1) and (f_2, g_2) if there is a binary function $h \in \langle \Gamma, \mathcal{C}_D \rangle_{fn}$ such that

$$\min\{h(a_1, a_2) + h(b_1, b_2), h(a_1, b_2) + h(b_1, a_2)\} < $$
$$h(f_1(a_1, b_1), f_2(a_2, b_2)) + h(g_1(a_1, b_1), g_2(a_2, b_2)). \quad (8)$$

The following lemma shows how G can be used to construct multimorphisms of binary functions in $\langle \Gamma, \mathcal{C}_D \rangle_{fn}$:

Lemma 20. *Let $I \subseteq V$ be an independent set in G with precisely one vertex $(f_{\{x,y\}}, g_{\{x,y\}})$ from each subgraph $G[x, y]$. Then, every binary function $h \in \langle \Gamma, \mathcal{C}_D \rangle_{fn}$ has the multimorphism (f, g) defined by $f(x, y) = f_{\{x,y\}}(x, y)$ and $g(x, y) = g_{\{x,y\}}(x, y)$.*

Proof. Assume to the contrary that (f, g) is not a multimorphism of h. Then, there are tuples $(a_1, a_2), (b_1, b_2) \in D^2$ such that

$$h(a_1, a_2) + h(b_1, b_2) < h(f(a_1, b_1), f(a_2, b_2)) + h(g(a_1, b_1), g(a_2, b_2)).$$

But this would imply that $\{(f_{\{a_1,b_1\}}, g_{\{a_1,b_1\}}), (f_{\{a_2,b_2\}}, g_{\{a_2,b_2\}})\} \in E$, which is a contradiction since I is an independent set. $\qquad\square$

For distinct $a, b \in D$, let $\overrightarrow{ab}$ denote the vertex $(f, g) \in G[a, b]$ such that $f(a, b) = f(b, a) = a$ and $g(a, b) = g(b, a) = b$. We say that such a vertex is *conservative*. Let V' denote the set of all conservative vertices, and let $G' = G[V']$ be the subgraph of G induced by V'. Let $V'_\Gamma \subseteq V'$ be the set of vertices $\overrightarrow{xy}$ such that $\{x, y\} \in \langle \Gamma, \mathcal{C}_D \rangle_w$. For conservative vertices $\overrightarrow{a_1 b_1}$ and $\overrightarrow{a_2 b_2}$, condition (8) reduces to:

$$h(a_1, b_2) + h(b_1, a_2) < h(a_1, a_2) + h(b_1, b_2). \tag{9}$$

For a vertex $x = (f, g)$, we let $\overline{x}$ denote the vertex (g, f). It follows immediately from (8) that $\{x, y\} \in E$ iff $\{\overline{x}, \overline{y}\} \in E$. Next, we prove a number of basic properties of the graph G.

Lemma 21. *If $\{\overrightarrow{a_1 b_1}, \overrightarrow{a_2 b_2}\} \in E$, then there exists a function $h \in \langle \Gamma, \mathcal{C}_D \rangle_{fn}$ such that $h(a_1, b_2) = h(b_1, a_2) < h(a_1, a_2) = h(b_1, b_2)$.*

Proof. By definition of G, we can find $f \in \langle \Gamma, \mathcal{C}_D \rangle_{fn}$ such that

$$f(a_1, b_2) + f(b_1, a_2) < f(a_1, a_2) + f(b_1, b_2). \tag{10}$$

Since Γ is assumed to be a core, Lemma 19 is applicable for all choices of a and b. Using the unary functions obtained from this lemma, it is possible to ensure that the inequality in (10) holds for a function f with $f(a_1, b_2) = f(a_2, b_1)$. We will also assume that $f(a_1, a_2) \geq f(b_1, b_2)$ so that $\gamma = (f(a_1, a_2) - f(b_1, b_2))/2 \geq 0$. Let f_{a_1} and f_{a_2} be unary functions such that $f_{a_1}(a_1) < f_{a_1}(b_1)$ and $f_{a_2}(a_2) < f_{a_2}(b_2)$, and let $\alpha = f_{a_1}(b_1) - f_{a_1}(a_1)$ and $\beta = f_{a_2}(b_2) - f_{a_2}(a_2)$, and note that $\alpha, \beta > 0$. Now, define

$$h(x, y) = f(x, y) + \gamma \left(\alpha^{-1} f_{a_1}(x) + \beta^{-1} f_{a_2}(y) \right).$$

The function h satisfies the inequality $h(a_1, b_2) + h(b_1, a_2) < h(a_1, a_2) + h(b_1, b_2)$, and furthermore,

$$h(a_1, a_2) - h(b_1, b_2) = f(a_1, a_2) - f(b_1, b_2) +$$
$$+ \gamma \left(\frac{f_{a_1}(a_1) - f_{a_1}(b_1)}{\alpha} + \frac{f_{a_2}(a_2) - f_{a_2}(b_2)}{\beta} \right) =$$
$$= f(a_1, a_2) - f(b_1, b_2) + \gamma(-\alpha/\alpha - \beta/\beta) = 0,$$

and

$$h(a_1, b_2) - g(b_1, a_2) = f(a_1, b_2) - f(b_1, a_2) +$$

$$+ \gamma \left(\frac{f_{a_1}(a_1) - f_{a_1}(b_1)}{\alpha} + \frac{f_{a_2}(b_2) - f_{a_2}(a_2)}{\beta} \right) =$$

$$= 0 + \gamma(-\alpha/\alpha + \beta/\beta) = 0.$$

The lemma follows. $\qquad\square$

Lemma 22. *1. Let x, y, and z be conservative vertices, with $\{x, y\}$, $\{y, z\} \in E$, and assume that $y \in V'_\Gamma$. Then, $\{x, \overline{z}\} \in E$.*

2. For $n \geq 2$, let $(x_1, \ldots, x_n)$ be a path of conservative vertices in G, with $x_2, \ldots, x_{n-1} \in V'_\Gamma$. If n is even, then $\{x_1, x_n\} \in E$, otherwise $\{x_1, \overline{x_n}\} \in E$.

3. For $n \geq 3$, let $(x_1, \ldots, x_n, x_1)$ be an odd cycle of conservative vertices in G, with $x_2, \ldots, x_n \in V'_\Gamma$. Then, there is a loop on x_1.

4. If $\{\overrightarrow{a_1 b_1}, \overrightarrow{a_2 b_2}\} \in E$, then for each element $x \neq a_2, b_2$, either $\{\overrightarrow{a_1 b_1}, \overrightarrow{a_2 x}\} \in E$ or $\{\overrightarrow{a_1 b_1}, \overrightarrow{x b_2}\} \in E$.

5. If $\{\overrightarrow{xy}, \overrightarrow{yx}\}, \{\overrightarrow{yz}, \overrightarrow{zy}\} \in E$ and $\{\overrightarrow{xy}, \overrightarrow{yz}\} \notin E$, then $\{\overrightarrow{xy}, \overrightarrow{zx}\}, \{\overrightarrow{yz}, \overrightarrow{zx}\} \in E$.

6. If there is a loop on $\overrightarrow{xz}$, but $\overrightarrow{xy}$ and $\overrightarrow{yz}$ are loop-free, then $\{\overrightarrow{xy}, \overrightarrow{yz}\} \in E$.

Proof. Properties (1)–(3) are minor variations of Lemma 11(b) and (e) in [14]. We include the proofs here for completeness.

(1) Let $x = \overrightarrow{a_1 b_1}$, $y = \overrightarrow{a_2 b_2}$, and $z = \overrightarrow{a_3 b_3}$. By Lemma 21, we have $h_1, h_2 \in \langle \Gamma, \mathcal{C}_D \rangle_{fn}$ such that $\alpha_1 = h_1(a_1, b_2) = h_1(b_1, a_2) < h_1(a_1, a_2) = h_1(b_1, b_2) = \beta_1$ and $\alpha_2 = h_2(a_2, b_3) = h_2(b_2, a_3) < h_2(a_2, a_3) = h_2(b_2, b_3) = \beta_2$. Let $h'(u_1, u_3) = \min_{u_2 \in \{a_2, b_2\}} h_1(u_1, u_2) + h_2(u_2, u_3)$, which is in $\langle \Gamma, \mathcal{C}_D \rangle_{fn}$ since $y \in V'$. Now, $h'(a_1, b_3) + h'(b_1, a_3) = \min_{u_2, v_2 \in \{a_2, b_2\}} h_1(a_1, u_2) + h_1(b_1, v_2) + h_2(u_2, b_3) + h_2(v_2, a_3) = 2\min\{\alpha_1 + \beta_2, \alpha_2 + \beta_1\}$. We also have $h'(a_1, a_3) + h'(b_1, b_3) = \min_{u_2, v_2 \in \{a_2, b_2\}} h_1(a_1, u_2) + h_1(b_1, v_2) + h_2(u_2, a_3) + h_2(v_2, b_3) = 2(\alpha_1 + \beta_1)$. It follows that $h'(a_1, b_3) + h'(b_1, a_3) < h'(a_1, a_3) + h'(b_1, b_3)$, so $\{x, \overline{z}\} \in E$.

(2) and (3) These two properties follow by repeated application of (1), keeping in mind that $\{x, y\} \in E$ iff $\{\overline{x}, \overline{y}\} \in E$.

(4) By definition there exists a function $h \in \langle \Gamma, \mathcal{C}_D \rangle_{fn}$ such that $h(a_1, a_2) + h(b_1, b_2) > h(a_1, b_2) + h(b_1, a_2)$. If $h(a_1, a_2) + h(b_1, x) > h(a_1, x) + h(b_1, a_2)$, then we are in the first case. Otherwise, $h(a_1, a_2) + h(b_1, x) \leq h(a_1, x) + h(b_1, a_2)$, so $h(a_1, x) + h(b_1, b_2) = h(a_1, a_2) + h(b_1, b_2) + (h(a_1, x) - h(a_1, a_2)) > h(a_1, b_2) + h(b_1, a_2) + h(a_1, x) - h(a_1, a_2) \geq h(a_1, b_2) + (h(a_1, a_2) + h(b_1, x)) - h(a_1, a_2)$, which shows that we are in the second case.

(5) By (4), $\{\overrightarrow{xy}, \overrightarrow{yx}\} \in E$ implies $\{\overrightarrow{xy}, \overrightarrow{yz}\} \in E$ or $\{\overrightarrow{xy}, \overrightarrow{zx}\} \in E$. In the first case, we are done, so we assume that the latter holds. Again by (4), $\{\overrightarrow{yz}, \overrightarrow{zy}\} \in E$ implies $\{\overrightarrow{yz}, \overrightarrow{zx}\} \in E$ or $\{\overrightarrow{yz}, \overrightarrow{xy}\} \in E$. In the latter case, we are done, hence it follows that if $\{\overrightarrow{yz}, \overrightarrow{xy}\} \notin E$, then we have both $\{\overrightarrow{xy}, \overrightarrow{zx}\}$ and $\{\overrightarrow{yz}, \overrightarrow{zx}\}$ in E.

(6) By (4), $\{\overrightarrow{xz}, \overrightarrow{xz}\} \in E$ implies $\{\overrightarrow{xz}, \overrightarrow{xy}\} \in E$ or $\{\overrightarrow{xz}, \overrightarrow{yz}\} \in E$. In the first case, this in turn implies either $\{\overrightarrow{xy}, \overrightarrow{xy}\} \in E$ or $\{\overrightarrow{xy}, \overrightarrow{yz}\} \in E$. In the second case, it implies either $\{\overrightarrow{yz}, \overrightarrow{xy}\} \in E$ or $\{\overrightarrow{yz}, \overrightarrow{yz}\} \in E$. Hence, if both $\overrightarrow{xy}$ and $\overrightarrow{yz}$ are loop-free, then $\{\overrightarrow{xy}, \overrightarrow{yz}\} \in E$. $\qquad\square$

15

6 Classification for $|D| = 4$

We are now ready to derive a classification of the computational complexity of MIN CSP over a four-element domain. From here on, we assume that D is the domain $\{a, b, c, d\}$. First, we prove a result which describes the structure of the unary functions in $\langle \Gamma, \mathcal{C} \rangle_{fn}$, when Γ is a core.

Let $\Sigma = \{\{x, y\} \subseteq D \mid x \neq y\}$, $\Sigma_{ad} = \Sigma \setminus \{\{b, c\}\}$, $\Sigma_0 = \Sigma \setminus \{\{b, c\}, \{a, d\}\}$, and let $\Sigma_\Gamma = \langle \Gamma, \mathcal{C}_D \rangle_w \cap \Sigma$. For distinct $x, y \in D$, let $u_{xy}(z) = 0$ if $z \in \{x, y\}$, and $u_{xy}(z) = 1$ otherwise.

Proposition 23. *Let Γ be a core over $\{a, b, c, d\}$ and assume that $\{b, c\} \notin \Sigma_\Gamma$. Then, $\Sigma_0 \subseteq \Sigma_\Gamma$ and for all unary functions $u \in \langle \Gamma, \mathcal{C}_D \rangle_{fn}$, we have $u(a) + u(d) \leq u(b) + u(c)$. If $\Sigma_0 = \Sigma_\Gamma$, then $u(a) + u(d) = u(b) + u(c)$.*

Proof. Let $\mathcal{U}$ be the set of unary functions in $\langle \Gamma, \mathcal{C}_D \rangle_{fn}$. In order to simplify notation we will denote a unary function u by the vector $(u(a), u(b), u(c), u(d))$. To exclude the functions e_{ba}, e_{ca}, e_{bd}, and e_{cd} from the endomorphisms of Γ, Lemma 19 states that $\mathcal{U}$ must contain certain unary $\{0, 1\}$-valued functions. The following table lists the possibilities, provided that $\{b, c\} \notin \Sigma_\Gamma$, so in particular $u_{bc} = (1, 0, 0, 1) \notin \mathcal{U}$.

e_{ba}	e_{ca}	e_{bd}	e_{cd}
(1,0,0,0)	(1,0,0,0)	(0,0,0,1)	(0,0,0,1)
(1,0,1,0)	(1,1,0,0)	(0,0,1,1)	(0,1,0,1)
(1,0,1,1)	(1,1,0,1)	(1,0,1,1)	(1,1,0,1)

For each of the four functions e_{xy}, it is necessary that at least one of the three functions in the corresponding column is in $\mathcal{U}$. First assume that $(1, 0, 0, 0) \in \mathcal{U}$. We note that $(1, 0, 0, 0) + (0, 0, 0, 1) = u_{bc}$, so we conclude that $(0, 0, 0, 1) \notin \mathcal{U}$. Since e_{cd} is not an endomorphism of Γ, we must therefore either have $(0, 1, 0, 1)$ or $(1, 1, 0, 1)$ in $\mathcal{U}$. In the former case, we can add $(1, 0, 0, 0)$ to obtain $(1, 1, 0, 1)$, so we know that $(1, 1, 0, 1) \in \mathcal{U}$. By a similar argument, considering the function e_{bd}, we conclude that $(1, 0, 1, 1) \in \mathcal{U}$. Since $(1, 0, 1, 1) + (1, 1, 0, 1) = 1 + u_{bc}$, we have reached a contradiction. A similar argument shows that $(0, 0, 0, 1) \notin \mathcal{U}$.

Assume instead that $(1, 0, 0, 0), (0, 0, 0, 1) \notin \mathcal{U}$, $(1, 0, 1, 1) \in \mathcal{U}$. As noted above, we must have $(1, 1, 0, 1) \notin \mathcal{U}$, and consequently $(1, 1, 0, 0), (0, 1, 0, 1) \in \mathcal{U}$. But $(1, 1, 0, 0) + (0, 1, 0, 1) + 2 \cdot (1, 0, 1, 1) = 2 + u_{bc}$ so again we have a contradiction. Thus, the only possibility is that $\mathcal{U}_0 := \{u_{bd}, u_{cd}, u_{ab}, u_{ac}\} \subseteq \mathcal{U}$, so $\Sigma_0 \subseteq \Sigma_\Gamma$.

It is not hard to see that one can write every unary function u such that $u(a) + u(d) = u(b) + u(c)$ as a linear combination of functions from $\mathcal{U}_0$ with non-negative coefficients. We show that if $v \in \langle \Gamma, \mathcal{C}_D \rangle_{fn}$ is a unary function in such that $v(a) + v(d) < v(b) + v(c)$, then $\{a, d\} \in \Sigma_\Gamma$. The full statement follows similarly.

Let $\delta = (v(b) + v(c) - v(a) - v(d))/2 > 0$, and let $M = \max_{x \in D} v(x)$. Define $v'(x) = M - v(x)$ if $x = b, c$, and $v'(x) = M - v(x) + \delta$ otherwise. Then, $v'(a) + v'(d) = v'(b) + v'(c)$, and $M + \delta u_{ad} = v' + v$ can be written as a linear

combination of functions from $\mathcal{U}_0 \cup \{v\}$ with non-negative coefficients. Hence $M + \delta u_{ad} \in \langle \Gamma, \mathcal{C}_D \rangle_{fn}$, and $\{a, d\} \in \Sigma_\Gamma$. $\qquad \square$

We need the following two propositions in order to prove Theorem 26. Their proofs are deferred to the next section.

Proposition 24. *Assume that $\Sigma_0 \subseteq \Sigma_\Gamma$, and that G' is bipartite. Then, the set of binary functions in $\langle \Gamma, \mathcal{C}_D \rangle_{fn}$ is submodular on a chain.*

Proposition 25. *Assume that $\Sigma_0 \subseteq \Sigma_\Gamma$, that G' is not bipartite, but that $G[V'_\Gamma]$ is. Then, the set of binary functions in $\langle \Gamma, \mathcal{C}_D \rangle_{fn}$ has a 1-defect chain multimorphism.*

Theorem 26. *Let Γ be a core over D with $D = a, b, c, d$. If Γ is submodular on a chain, or if Γ has a 1-defect chain multimorphism, then MIN CSP(Γ) is tractable. Otherwise, it is **NP**-hard.*

Proof. Assume that $G[V'_\Gamma]$ has a loop on a vertex $\overrightarrow{xy}$. It then follows from Lemma 21 that there is a function $h \in \langle \Gamma, \mathcal{C}_D \rangle_{fn}$ such that $h(x, y) = h(y, x) < h(x, x) = h(y, y)$, and $\{x, y\} \in \langle \Gamma, \mathcal{C}_D \rangle_w$. By Proposition 5.1 in [4], MIN CSP$(\Gamma, \mathcal{C}_D)$ is **NP**-hard. By Proposition 18, MIN CSP$(\Gamma, \mathcal{C}_D)$ reduces to MIN CSP(Γ). Hence, the latter is also **NP**-hard.

If instead $G[V'_\Gamma]$ is loop-free, then it is bipartite, by Lemma 22(3). We may assume that $\Sigma_0 \subseteq \Sigma_\Gamma$: this is trivial if $\Sigma_\Gamma = \Sigma$. If Σ_Γ is strictly contained in Σ, then up to an automorphism we may assume that $\{b, c\} \notin \Sigma_\Gamma$, and the inclusion follows by Proposition 23. For a k-ary function $h \in \Gamma$, let $\Phi(h)$ be the set of binary which can be obtained from h by fixing at least $k - 2$ variables, and let Γ' be the union of $\Phi(h)$ over all $h \in \Gamma$.

Now, if G' is bipartite, then by Proposition 24, the set of binary functions in $\langle \Gamma, \mathcal{C}_D \rangle_{fn}$ is submodular on a chain. Since this set contains Γ', we may conclude, by Lemma 11, that Γ is submodular on this chain as well. It follows that MIN CSP(Γ) is tractable [9, 21].

Otherwise, G' is not bipartite, and by Proposition 25, the set of binary functions in $\langle \Gamma, \mathcal{C}_D \rangle_{fn}$ have a 1-defect chain multimorphism. Since this set contains Γ', we may conclude, by Lemma 14 this time, that Γ has a 1-defect chain multimorphism. It now follows from Proposition 10 that MIN CSP(Γ) is tractable. $\qquad \square$

7   Proofs of Propositions 24 and 25

Lemma 27. *If $\Sigma_0 \subseteq \Sigma_\Gamma$, and $x \in V'$ is not isolated in G', then $\{x, \overline{x}\} \in E$.*

Proof. By assumption, there is an edge $\{x, \overrightarrow{yz}\} \in E$. If $\{y, z\} \neq \{b, c\}, \{a, d\}$, then $\overrightarrow{yz} \in V'_\Gamma$ since $\Sigma_0 \subseteq \Sigma_\Gamma$. If instead $\{x, \overrightarrow{bc}\} \in E$, then it follows from Lemma 22(4) that either $\{x, \overrightarrow{ba}\} \in E$ or $\{x, \overrightarrow{ac}\} \in E$, and $\overrightarrow{ba}, \overrightarrow{ac} \in V'_\Gamma$ due to $\Sigma_0 \subseteq \Sigma_\Gamma$. In either case, $\{x, \overline{x}\} \in E$ follows from Lemma 22(1). $\qquad \square$

For an independent set I in G', let R_I denote the binary relation on D defined by $(x, y) \in R_I$ iff $\overrightarrow{xy} \in I$.

Proof (Proposition 24). Let $\{I, J\}$ be a 2-colouring of the subgraph of G' induced by the non-isolated vertices. We first show that R_I is a partial order on D. Let $(x, y), (y, z) \in R_I$. Then, $\overrightarrow{xy}$ and $\overrightarrow{yz}$ have the same colour in I, and it follows that $\{\overrightarrow{xy}, \overrightarrow{yz}\} \notin E$. Hence, by Lemma 22(5), we have $\{\overrightarrow{xy}, \overrightarrow{zx}\}, \{\overrightarrow{yz}, \overrightarrow{zx}\} \in E$. By Lemma 27, $\{\overrightarrow{zx}, \overrightarrow{xz}\} \in E$, so $\overrightarrow{xz} \in I$ and $(x, z) \in R_I$. Now, let $(D; <)$ be a linear extension of R_I, and let $I' \supseteq I$ be the corresponding subset of V'. The set I' is independent since I is independent and $I' \setminus I$ is a set of isolated vertices in G'. Since there are no edges from V' to the singleton vertices in G, we can add all of these to I' as well. Thus, by Lemma 20, every binary function in $\langle \Gamma, \mathcal{C}_D \rangle_{fn}$ is submodular on the chain $(D; \wedge, \vee)$, where $\wedge$ and $\vee$ are defined with respect to the total order $(D; <)$. $\qquad \square$

In the following, we will let (f, g) denote the vertex in G given by $f(b, c) = f(c, b) = a$ and $g(b, c) = g(c, b) = d$.

Lemma 28. *Assume that $\Sigma_\Gamma \subseteq \Sigma_{ad}$ and that there is an edge $\{(f, g), z\} \in E$, $z \in V'$. Then, $\{\overrightarrow{ab}, z\} \in E$ or $\{\overrightarrow{ac}, z\} \in E$, and $\{\overrightarrow{bd}, z\} \in E$ or $\{\overrightarrow{cd}, z\} \in E$.*

Proof. Let $z = \overrightarrow{xy}$. By definition, there exists a function $h \in \langle \Gamma, \mathcal{C}_D \rangle_{fn}$ such that $\min\{h(b, x) + h(c, y), h(c, x) + h(b, y)\} < h(a, x) + h(d, y)$. If $h(b, x) + h(c, y) < h(a, x) + h(d, y)$, then $h(a, x) + h(b, y) > (h(b, x) + h(c, y) - h(d, y)) + h(b, y) \geq h(b, x) + h(a, y) + h(d, y) - h(d, y)$ since $h(b, y) + h(c, y) \geq h(a, y) + h(d, y)$ by Proposition 23. Thus, $\{\overrightarrow{ab}, \overrightarrow{xy}\} \in E$. If $h(c, x) + h(b, y) < h(a, x) + h(d, y)$, then we obtain $\{\overrightarrow{ac}, \overrightarrow{xy}\} \in E$ following a similar argument, and the remaining two cases can be deduced in the same way.

Lemma 29. *If $\Sigma_\Gamma \subseteq \Sigma_0$, and there is a loop on $\overrightarrow{bc}$ or $\overrightarrow{ad}$, then there is a loop on at least one of the vertices $\overrightarrow{ab}$, $\overrightarrow{ac}$, $\overrightarrow{bd}$, $\overrightarrow{cd}$.*

Proof. Assume, without loss of generality, that there exists an $h \in \langle \Gamma, \mathcal{C}_D \rangle_{fn}$ such that $h(b, b) + h(c, c) > h(b, c) + h(c, b)$. By Proposition 23, $\Sigma_\Gamma \subseteq \Sigma_0$ implies the relations $h(b, b) + h(c, c) = h(a, a) + h(d, d)$, $h(b, b) + h(c, b) = h(a, b) + h(d, b)$, $h(b, c) + h(c, c) = h(a, c) + h(d, c)$, $h(b, b) + h(b, c) = h(b, a) + h(b, d)$, and $h(c, b) + h(c, c) = h(c, a) + h(c, d)$. It follows that $(h(a, a) + h(b, b)) + (h(a, a) + h(c, c)) + (h(b, b) + h(d, d)) + (h(c, c) + h(d, d)) > 2(h(b, b) + h(c, c) + h(b, c) + h(c, b)) = (h(a, b) + h(b, a)) + (h(a, c) + h(c, a)) + (h(b, d) + h(d, b)) + (h(c, d) + h(d, c))$, which implies that the inequality $h(x, x) + h(y, y) > h(x, y) + h(y, x)$ holds in at least one of the cases $\{x, y\} = \{a, b\}, \{a, c\}, \{b, d\}, \{c, d\}$. $\qquad \square$

Proof (Proposition 25). We follow a strategy similar to that of Proposition 24. However, instead of using G' we now consider the graph $G[V'_{ad} \cup \{(f, g), (g, f)\}]$, where $V'_{ad} = V' \setminus \{\overrightarrow{bc}, \overrightarrow{cb}\}$. First, we show that $G[V'_{ad}]$ is bipartite. If $\Sigma_\Gamma = \Sigma_{ad}$, then $G[V'_{ad}] = G[V'_\Gamma]$ is bipartite by assumption. Otherwise, $\Sigma_\Gamma = \Sigma_0$. Since

$G[V'_\Gamma] = G[V'_0]$ is loop-free, we know from Lemma 29 that there is no loop on $\overrightarrow{bc}$, nor on $\overrightarrow{ad}$. Thus, by Lemma 22(3), $G[V'_{ad}]$ is bipartite.

Assume for the moment that the following holds:

$$\text{For } y \in D \setminus \{b, c\}, \text{ there is an odd path in } G[V'_{ad}] \text{ from } \overrightarrow{by} \text{ to } \overrightarrow{yc}. \tag{11}$$

Let $\{I, J\}$ be a 2-colouring of the subgraph of $G[V'_{ad}]$ induced by the non-isolated vertices. We claim that R_I is a partial order on D. Let $(x, y), (y, z) \in R_I$ and observe that (11) implies $\{x, z\} \neq \{b, c\}$. As in the proof of Proposition 24, we can argue that $\overrightarrow{xz}$ is connected by even paths to both $\overrightarrow{xy}$ and $\overrightarrow{yz}$. Since $\{x, z\} \neq \{b, c\}$, it follows that $(x, z) \in I$. Now take a transitive extension of R_I which orders all pairs of elements except for b and c, and let $I' \supseteq I$ be the corresponding subset of V'_{ad}. We can assume (possibly by swapping I and J) that $\overrightarrow{ad} \in I'$.

Next we show that $I' \cup \{(f, g)\}$ is independent. This will ensure that $f(b, c) = a < d = g(b, c)$ holds in the constructed multimorphism. If (f, g) is not connected to any vertex in V'_{ad}, then $I' \cup \{(f, g)\}$ is trivially independent. Otherwise, by Lemma 28, (11), and Lemma 27, we can show that from any $z \in V'_{ad}$ such that $\{(f, g), z\} \in E$, there are odd paths in $G[V'_{ad}]$ to each vertex in the set $S = \{\overrightarrow{ab}, \overrightarrow{ac}, \overrightarrow{bd}, \overrightarrow{cd}\}$. Since $G[V'_{ad}]$ is bipartite, it follows that $\{\overrightarrow{ab}, \overrightarrow{bd}\} \notin E$, so $\{\overrightarrow{ab}, \overrightarrow{da}\} \in E$ by Lemma 22(5). Hence, $I' = I = S \cup \{\overrightarrow{ad}\}$, and $z \notin I'$.

It remains to verify that $I' \cup \{(f, g)\}$ together with the singleton vertices in G also form an independent set, *i.e.* that there is no edge between a singleton and (f, g). But by condition (8) this is equivalent to saying that each row and column of every binary function in $\langle \Gamma, \mathcal{C}_D \rangle_{fn}$ is submodular on L_{ad}, which follows from Proposition 23. By Lemma 20, every binary function in $\langle \Gamma, \mathcal{C}_D \rangle_{fn}$ has the 1-defect chain multimorphism corresponding to $I' \cup \{(f, g)\}$.

Finally, we prove property (11). If $\Sigma_\Gamma = \Sigma_{ad}$, then by Lemma 22(3), and the fact that G' contains an odd cycle, we have a loop on $\overrightarrow{bc}$. Since $\overrightarrow{by}$ and $\overrightarrow{yc}$ are loop-free for $y \in D \setminus \{b, c\}$, we have $\{\overrightarrow{by}, \overrightarrow{yc}\} \in E$ by Lemma 22(6). Otherwise, $\Sigma_\Gamma = \Sigma_0$. We argued above that G' does not contain any loop in this case. Thus, by Lemma 22(3), every odd cycle C in G' must intersect both $\{\overrightarrow{bc}, \overrightarrow{cb}\}$ and $\{\overrightarrow{ad}, \overrightarrow{da}\}$. Now, by repeatedly applying Lemma 22(2) to C, we obtain a triangle on a subset of $\{\overrightarrow{bc}, \overrightarrow{cb}, \overrightarrow{ad}, \overrightarrow{da}\}$. By Lemma 27, we can conclude that G' in fact contains the complete graph on these four vertices. In particular, we have both $\{\overrightarrow{ad}, \overrightarrow{bc}\} \in E$ and $\{\overrightarrow{da}, \overrightarrow{bc}\} \in E$. By Lemma 22(4), we therefore have either $\{\overrightarrow{ad}, \overrightarrow{ba}\} \in E$ or $\{\overrightarrow{ad}, \overrightarrow{ac}\} \in E$, and furthermore, either $\{\overrightarrow{da}, \overrightarrow{ba}\} \in E$ or $\{\overrightarrow{da}, \overrightarrow{ac}\} \in E$. Since there is no loop on $\overrightarrow{ad}$, we conclude that either the path $(\overrightarrow{ba}, \overrightarrow{ad}, \overrightarrow{da}, \overrightarrow{ac})$ or the path $(\overrightarrow{ba}, \overrightarrow{da}, \overrightarrow{ad}, \overrightarrow{ac})$ is in $G[V'_{ad}]$. In the same way, we find an odd path from $\overrightarrow{bd}$ to $\overrightarrow{dc}$. $\qquad\square$

8   Discussion

We have presented a complete complexity classification for Min CSP over a four-element domain. More importantly, we have compiled a powerful set of tools which will allow further systematic study of this problem. In particular, we have shown that it is possible to add (crisp) constants to an arbitrary core, without changing the complexity of the problem. This result holds in the more general case of VCSP as well (although this requires a slightly different definition of endomorphisms), thus answering Question 4 in Živný [24]. We have also demonstrated that the techniques used by Krokhin and Larose [15] for lattices can be used effectively in the context of arbitrary algebras as well, and in doing so, we have given the first example of an instance where submodularity does not suffice as an origin of tractability for Min CSP. We hope that this insight will inspire an interest in the search for more tractable cases which are not explained by submodularity. Finally, we have shown that graph representations such as the one defined by Kolmogorov and Živný [14] can be used to great effect, even in non-conservative settings.

The curious readers may ask themselves several questions at this point, and the following one is particularly important: do 1-defect chain multimorphisms define genuinely new tractable classes? There is still a possibility that the tractability can be explained in terms of submodularity. We answer this question negatively with the following example.

Example 30. Consider the language $\Gamma = \{u_{bd}, u_{cd}, u_{ab}, u_{ac}, h\}$ where $h : D^2 \to \{0, 1\}$ is defined such that $h(x, y) = 1$ if and only if $x = c$ or $y = b$. Then, Γ is a core on $\{a, b, c, d\}$ but it is not submodular on any lattice. However, Γ has the 1-defect chain multimorphisms (f_1, g_1) and (f_2, g_2) from Example 8.

A related question is why bisubmodularity does not appear in the classification of Min CSP over domains of size three [11]. The reason is that for any cost function $h : \{0, 1, 2\}^k \to \{0, 1\}$ which is bisubmodular, the tuple $(0, 0, \ldots, 0)$ minimises h. It follows that any $\{0, 1\}$ constraint language over three elements which is bisubmodular is not a core.

There are several ways of extending this work, and one obvious way is to study VCSP instead of Min CSP. It is known that the *fractional polymorphisms* of the constraint language, introduced by Cohen et al. [3], characterise the complexity of this problem (see also [5]). Multimorphisms are a special case of such fractional polymorphisms. As in the case of Min CSP, it is currently not known if submodularity over every finite lattice implies tractability for VCSP. Distributive lattices imply tractability, and certain constructions on lattices preserve tractability (homomorphic images and Mal'tsev products) [15]. Furthermore, the five element modular non-distributive lattice (also known as the diamond) implies tractability for *unweighted* VCSP [16]. Finally, it is known that submodularity over finite modular lattices implies containment in **NP** ∩ **coNP** [16]. It is thus clear that in order to approach further classification of either Min CSP or VCSP, it will be necessary to study the complexity of minimising submodular cost functions over new finite lattices.

As a last note, we mention that it seems to be possible to adapt Proposition 24 to the classification in [14] of VCSP for conservative finite-valued languages. This would yield a simpler description of those tractable cases.

References

1. A. A. Bulatov. Tractable conservative constraint satisfaction problems. In *Proceedings of the 18th Annual IEEE Symposium on Logic in Computer Science (LICS-2003)*, pages 321–330, 2003.
2. R. E. Burkard, B. Klinz, and R. Rudolf. Perspectives of Monge properties in optimization. *Discrete Appl. Math.*, 70(2):95–161, 1996.
3. D. Cohen, M. Cooper, and P. Jeavons. An algebraic characterisation of complexity for valued constraints. In *Proceedings of the International Conference on Principles and Practice of Constraint Programming (CP-2006)*, pages 680–684, 2006.
4. D. Cohen, M. Cooper, P. Jeavons, and A. Krokhin. The complexity of soft constraint satisfaction. *Artificial Intelligence*, 170(11):983–1016, 2006.
5. D. A. Cohen, P. Creed, P. G. Jeavons, and S. Zivny. An algebraic theory of complexity for valued constraints: Establishing a galois connection. Technical Report RR-10-16, OUCL, November 2010.
6. N. Creignou, P. G. Kolaitis, and H. Vollmer, editors. *Complexity of Constraints: An Overview of Current Research Themes*, volume 5250 of *Lecture Notes in Computer Science*. Springer-Verlag, 2008.
7. V. Deineko, P. Jonsson, M. Klasson, and A. Krokhin. The approximability of MAX CSP with fixed-value constraints. *Journal of the ACM*, 55(4):1–37, 2008.
8. P. Hell and J. Nešetřil. Colouring, constraint satisfaction, and complexity. *Computer Science Review*, 2(3):143–163, 2008.
9. S. Iwata, L. Fleischer, and S. Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *J. ACM*, 48(4):761–777, 2001.
10. P. Jeavons, D. Cohen, and M. Gyssens. How to determine the expressive power of constraints. *Constraints*, 4(2):113–131, 1999.
11. P. Jonsson, M. Klasson, and A. Krokhin. The approximability of three-valued MAX CSP. *SIAM J. Comput.*, 35(6):1329–1349, 2006.
12. P. Jonsson and A. Krokhin. Maximum *H*-colourable subdigraphs and constraint optimization with arbitrary weights. *Journal of Computer and System Sciences*, 73(5):691–702, 2007.
13. V. Kolmogorov. A dichotomy theorem for conservative general-valued CSPs. *CoRR*, abs/1008.4035, 2010.
14. V. Kolmogorov and S. Živný. The complexity of conservative finite-valued CSPs. *CoRR*, abs/1008.1555, 2010.
15. A. Krokhin and B. Larose. Maximizing supermodular functions on product lattices, with application to maximum constraint satisfaction. *SIAM J. Discrete Math.*, 22(1):312–328, 2008.
16. F. Kuivinen. *Algorithms and Hardness Results for Some Valued CSPs*. PhD thesis, Linköping University, TCSLAB, 2009.
17. S. T. McCormick and S. Fujishige. Strongly polynomial and fully combinatorial algorithms for bisubmodular function minimization. *Math. Program.*, 122(1):87–120, 2010.
18. M. Queyranne, F. Spieksma, and F. Tardella. A general class of greedily solvable linear programs. *Math. Oper. Res.*, 23(4):892–908, 1998.

19. P. Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC-2008)*, pages 245–254, 2008.
20. T. Schiex, H. Fargier, and G. Verfaillie. Valued constraint satisfaction problems: Easy and hard problems. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-1995)*, pages 631–637, 1995.
21. A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *J. Comb. Theory, Ser. B*, 80(2):346–355, 2000.
22. R. Takhanov. A dichotomy theorem for the general minimum cost homomorphism problem. In *Proceedings of the 27th International Symposium on Theoretical Aspects of Computer Science (STACS-2010)*, pages 657–668, 2010.
23. D. M. Topkis. Minimizing a submodular function on a lattice. *Oper. Res.*, 26(2):475–496, 1978.
24. S. Živný. *The Complexity and Expressive Power of Valued Constraints*. PhD thesis, Oxford University Computing Laboratory, 2009.