# 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 is fairly common when studying VCSPs). The hardness results are further simplified by the introduction of powerful reduction techniques.

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

## 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 wide variety of combinatorial decision problems [4].

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). 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. We have the following formal definition.

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

The set $\Gamma$ is often referred to as the *constraint language.* We parameterise problems with constraint languages throughout the paper. For instance, when we say that a class of VCSPs $X$ is polynomial-time solvable, then we mean that VCSP($\Gamma$) 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. [2]. This led 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) [5, 9], $\{0, 1\}$ languages on three elements [8], and the MIN HOM problem [16]. 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 concerning the approximability of finite-valued VCSP. Raghavendra [14] has 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. 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 study VCSPs 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 $\Gamma$ denote a constraint language with $\{0,1\}$ cost functions over a four-element domain $D$. We will need one new tractability result for our classification; this result can be found in Section 3 and our algorithm is based on a combination of submodular and bisubmodular minimisation [6, 12, 15]. The hardness proof consists of three parts. Section 4 concerns the problem of adding (crisp) constant unary relations to $\Gamma$ without changing the computational complexity of the resulting problem, and Section 5 introduces a graph construction for studying $\Gamma$. This graph provides information about the complexity of VCSP($\Gamma$) based on the two-element sublanguages of $\Gamma$. Similar graphs have been used repeatedly in the study of VCSP, cf. [9, 16]. Equipped with these tools, we prove our main classification result, Theorem 18, in Section 6. Due to space constraints, some proofs have been left out. A full version of this paper can be downloaded from `http://arxiv.org/abs/1102.2880`

## 2 Preliminaries

Throughout this paper, we will assume that $\Gamma$ is a finite set of {0,1}-valued functions. By Min CSP($\Gamma$) we denote the problem VCSP($\Gamma$). Note that Min CSP($\Gamma$) is polynomial-time equivalent to Max CSP($\{1 - f | f \in \Gamma\}$). This implies, for instance, that the dichotomy theorem for Max CSP over domains over size three also can be viewed as a dichotomy result for Min CSP. It turns out to be convenient to work with a slightly more general problem, in which we allow additional crisp constraints on the solutions.

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

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

### 2.1 Weighted pp-definitions and expressive power

We continue by defining two closure operators that are useful in studying the complexity of Min CSP. Let $\mathcal{I}$ be an instance of Min CSP($\Gamma, \Delta$), and let $\mathbf{x} = (x_1, \ldots, x_s)$ be a sequence of distinct variables from $V(\mathcal{I})$. Let $\pi_{\mathbf{x}}\mathrm{Optsol}(\mathcal{I})$ denote the set $\{(\sigma(x_1), \ldots, \sigma(x_s)) \mid \sigma \text{ is an optimal solution to } \mathcal{I}\}$, i.e. the projection of the set of optimal solutions onto $\mathbf{x}$. We say that such a relation *has a weighted pp-definition in* ($\Gamma, \Delta$). Let $\langle \Gamma, \Delta \rangle_w$ denote the set of relations which have a weighted pp-definition in ($\Gamma, \Delta$). 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. [2]. Define the function $\mathcal{I}_{\mathbf{x}} : D^k \to \mathbb{Q}_{\geq 0}$ by letting $\mathcal{I}_{\mathbf{x}}(a_1, \ldots, a_k) = \mathsf{Opt}(\mathcal{I} \cup \{(x_i; \{a_i\}) \mid 1 \leq i \leq k\})$. We say that $\mathcal{I}_{\mathbf{x}}$ is *expressible over* $(\Gamma, \Delta)$. Let $\langle \Gamma, \Delta \rangle_{fn}$ denote the set of *total functions* expressible over $(\Gamma, \Delta)$.

**Proposition 3.** *Let $\Gamma' \subseteq \langle \Gamma, \Delta \rangle_{fn}$ and $\Delta' \subseteq \langle \Gamma, \Delta \rangle_w$ be finite sets. Then,* Min CSP$(\Gamma', \Delta')$ *is polynomial-time reducible to* Min CSP$(\Gamma, \Delta)$.

**Proof (sketch):** The reduction from Min CSP$(\Gamma', \Delta')$ to Min CSP$(\Gamma, \Delta')$ is a special case of Theorem 3.4 in [2]. We allow weights as a part of our instances, but this makes no essential difference. To prove that there is a polynomial-time transformation from Min CSP$(\Gamma, \Delta')$ to Min CSP$(\Gamma, \Delta)$, we need a way to 'force' constraints in $\Delta' \setminus \Delta$ to hold in every optimal solution. This can quite easily be guaranteed by using large weights, and one sees that the representation size of these weights needs to grow only linearly in the size of the instance. □

### 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 $\mathbf{x}_1, \ldots, \mathbf{x}_k \in D^n$, with components $\mathbf{x}_i = (x_{i1}, \ldots, x_{in})$. Then, we let $f(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ denote the $n$-tuple $(f(x_{11}, \ldots, x_{k1}), \ldots, f(x_{1n}, \ldots, x_{kn}))$. A *multimorphism* [2] of $\Gamma$ is a pair of functions $f, g : D^2 \to D$ such that for any $h \in \Gamma$, and matching tuples $\mathbf{x}$ and $\mathbf{y}$, $h(f(\mathbf{x}, \mathbf{y})) + h(g(\mathbf{x}, \mathbf{y})) \leq h(\mathbf{x}) + h(\mathbf{y})$.

**Definition 4 (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,*

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

*for all $\mathbf{x}, \mathbf{y} \in D$. The function $h$ is given to the algorithm as an oracle, i.e., for any $\mathbf{x} \in D$ we can query the oracle to obtain $h(\mathbf{x})$ in unit time.*

**Solution:** *A tuple $\mathbf{x} \in D$.*

**Measure:** *The value of $h(\mathbf{x})$.*

For a finite set $X$ we say that 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 $\Gamma$, and MFM$(D; f, g)$ is oracle-tractable, then Min 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.

A function $f : D^k \to \mathbb{Z}$ is called *submodular* on the lattice $L = (D; \wedge, \vee)$ if $f(\mathbf{a} \wedge \mathbf{b}) + f(\mathbf{a} \vee \mathbf{b}) \leq f(\mathbf{a}) + f(\mathbf{b})$ for all $\mathbf{a}, \mathbf{b} \in D^k$. A set of functions $\Gamma$ is said to be submodular on $L$ if every function in $\Gamma$ is submodular on $L$. This is equivalent to $(\wedge, \vee)$ being a multimorphism of $\Gamma$. It follows from known algorithms for submodular function minimisation that MFM($X$) is oracle-tractable for any finite set $X$ of finite *distributive lattices* (e.g. chains) [6, 15].

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. A function $h : D^k \to \mathbb{Z}$ is *bisubmodular* if $h$ has the multimorphism $(u, v)$. The main result of [12] implies that MFM($\{D; u, v\}$) is oracle-tractable.

## 3 A New Tractable Class

In this section, we introduce a new class of multimorphisms which ensures tractability for Min CSP (and more generally for VCSP).

**Definition 5.** *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 $L_{ad}$.

*Example 6.* 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. $g_1(c, d) = c$):

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

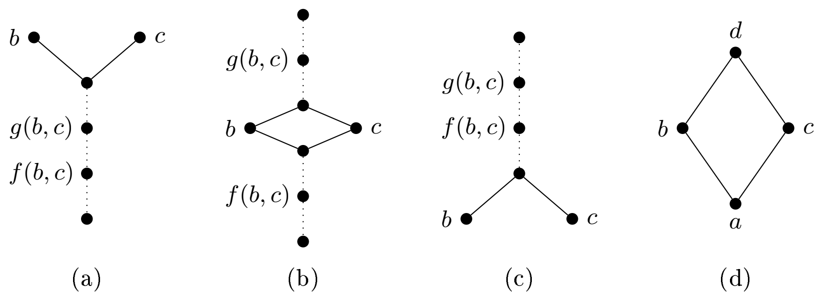


**Fig. 1.** Three types of 1-defect multimorphisms with defect $\{b, c\}$. (a) $f(b, c) < g(b, c) < b, c$. (b) $f(b, c) < b, c < g(b, c)$. (c) $b, c < f(b, c) < g(b, c)$. (d) The Hasse diagram of the lattice $L_{ad}$, a special case of (b).

The proof of tractability for languages with 1-defect chain multimorphisms is inspired by Krokhin and Larose's [10] 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 $\theta$ on $D$ we use $x[\theta]$ to denote the equivalence class containing $x \in D$. The relation $\theta$ 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 $D/\theta$ 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 7.** *Let $f, g$ be two functions that map $D^2$ to $D$. If there is a congruence relation $\theta$ 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 $\min_{\mathbf{z} \in (D/\theta)^n} h'(\mathbf{z}) = \min_{\mathbf{x} \in D^n} h(\mathbf{x})$. 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 $\mathbf{x}, \mathbf{y} \in D^k$ and choose $x'_i \in x_i[\theta]$ and $y'_i \in y_i[\theta]$ so that $h'(\mathbf{x}[\theta]) = h(\mathbf{x}')$ and $h'(\mathbf{y}[\theta]) = h(\mathbf{y}')$. We then have

$$\begin{aligned} h'(\mathbf{x}[\theta]) + h'(\mathbf{y}[\theta]) &= h(\mathbf{x}') + h(\mathbf{y}') && (1)\\ &\geq h(f(\mathbf{x}', \mathbf{y}')) + h(g(\mathbf{x}', \mathbf{y}')) && (2)\\ &\geq h'(f(\mathbf{x}', \mathbf{y}')[\theta]) + h'(g(\mathbf{x}', \mathbf{y}')[\theta]) && (3)\\ &= h'(f(\mathbf{x}, \mathbf{y})[\theta]) + h'(g(\mathbf{x}, \mathbf{y})[\theta])) && (4)\\ &= h'(u(\mathbf{x}[\theta], \mathbf{y}[\theta])) + h'(v(\mathbf{x}[\theta], \mathbf{y}[\theta])). && (5) \end{aligned}$$

Here (1) follows from our choice of $\mathbf{x}'$ and $\mathbf{y}'$, (2) follows from the fact that $h$ is an instance of MFM($D; f, g$), (3) follows from the definition of $h'$, and finally (4) and (5) follows as $\theta$ 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. □

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 8.** *If $\Gamma$ has a 1-defect chain multimorphism, then* MIN CSP($\Gamma$) *is tractable.*

*Proof.* Assume that $\Gamma$ 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 $\theta$ with classes $A = D \setminus \{b, c\}$, $B = \{b\}$, $C = \{c\}$ 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 [12, 15]. It now follows from Lemma 7 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 $\theta'$ 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 MFM($X$) and MFM($Y$) are both oracle-tractable, then so is MFM($X \cup Y$). Combining this with the technique used above, we can now solve the minimisation problem. An analoguous 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 $\theta''$ 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 [15]. □

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. [13].

**Lemma 9.** *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.

**Lemma 10.** *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 (sketch):** 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 $\mathbf{x}, \mathbf{y} \in D^k$ such that

$$h(\mathbf{x}) + h(\mathbf{y}) < h(f(\mathbf{x}, \mathbf{y})) + h(g(\mathbf{x}, \mathbf{y})), \tag{6}$$

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

Assume to the contrary that the result does not hold. We can then choose a function $h$ of minimal arity such that $\min\{\, d_H(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \text{ and } \mathbf{y} \text{ satisfy (6)} \,\} > 2$. The arity of $h$ must in fact be equal to the least $d_H(\mathbf{x}, \mathbf{y})$. Otherwise, we could obtain a function $h'$ of strictly smaller arity by fixing the variables in $h$ on which $\mathbf{x}$ and $\mathbf{y}$ agree. This would contradict the minimality in the choice of $h$.

This means that for any vectors which share an element in some coordinate, the reverse (non-strict) inequality holds in (6). It is possible to combine such inequalities to prove that there are $\mathbf{x}$ and $\mathbf{y}$, with $d_H(\mathbf{x}, \mathbf{y}) = k$, and satisfying (6), such that $\{x_i, y_i\} \neq \{b, c\}$ for all $i$, where $\{b, c\}$ is the defect of $(f, g)$.

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 $\varphi(\mathbf{z}) = (\varphi_1(z_1), \ldots, \varphi_k(z_k))$, and let $\mathbf{x}', \mathbf{y}' \in (D')^k$ be such that $\varphi(\mathbf{x}') = \mathbf{x}$ and $\varphi(\mathbf{y}') = \mathbf{y}$. Define $h'(\mathbf{z}') = h(\varphi(\mathbf{z}'))$. Then, $h'(\mathbf{x}') + h'(\mathbf{y}') = h(\mathbf{x}) + h(\mathbf{y}) < h(f(\mathbf{x}, \mathbf{y})) + h(g(\mathbf{x}, \mathbf{y})) = h'(f'(\mathbf{x}', \mathbf{y}')) + h'(g'(\mathbf{x}', \mathbf{y}'))$. It follows that $h'$ is not submodular on $(D', f', g')$. By Lemma 9, there are elements $\mathbf{z}', \mathbf{w}' \in (D')^k$ with $d_H(\mathbf{z}', \mathbf{w}') = 2$ such that $h'(\mathbf{z}') + h'(\mathbf{w}') < h'(f'(\mathbf{z}', \mathbf{w}')) + h'(g'(\mathbf{z}', \mathbf{w}'))$. Hence, $h(\varphi(\mathbf{z}')) + h(\varphi(\mathbf{w}')) = h'(\mathbf{z}') + h'(\mathbf{w}') < h'(f'(\mathbf{z}', \mathbf{w}')) + h'(g'(\mathbf{z}', \mathbf{w}')) = h(f(\varphi(\mathbf{z}'), \varphi(\mathbf{w}'))) + h(g(\varphi(\mathbf{z}'), \varphi(\mathbf{w}')))$, and we have $d_H(\varphi(\mathbf{z}'), \varphi(\mathbf{w}')) = 2$. This contradicts the original choice of $h$. □

## 4 Endomorphisms, Cores and Constants

In this section, we show that under a natural condition, it is possible to add constant unary relations to $\Gamma$ 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 $\Gamma$ if it is an endomorphism of each function in $\Gamma$. A set of functions, $\Gamma$, is said to be a *core* if all of its endomorphisms are injective. The idea is that if $\Gamma$ is not a core, then we can apply a non-injective endomorphism to every function in $\Gamma$, and obtain a polynomial-time equivalent problem on a strictly smaller domain. We can then use results previously obtained for smaller domains [2, 8]. Thus, we can restrict our attention to the case when $\Gamma$ is a core.

The set of all endomorphisms of $\Gamma$ is denoted by $\mathrm{End}\,(\Gamma)$. Recall that a bijective endomorphism is called an *automorphism* and that the automorphisms of $\Gamma$ form a group under composition.

Jeavons et al. [7] 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. Let $\Gamma$ 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 $\mathbf{a} = (a_1, \dots, a_k) \in D^k$, let $\mathbf{x_a} = (x_{a_1}, \dots x_{a_k}) \in X_D^k$. The indicator problem $\mathcal{IP}(\Gamma)$ is defined as the instance of Min CSP$(\Gamma)$ with variables $X_D$, and sum $\sum_{f_i \in \Gamma} \sum_{\mathbf{a} \in f_i^{-1}(0)} f_i(\mathbf{x_a})$, where $k_i$ is the arity of the function $f_i$.

Let $\mathcal{C}_D = \{\{d\} \mid d \in D\}$ be the set of constant unary relations over $D$. The proof of the following result follows the lines of similar results for related problems, such as the CSP decision problem.

**Proposition 11.** *Let $\Gamma$ be a core over $D$. Then,* Min CSP$(\Gamma, \mathcal{C}_D)$ *is polynomial-time reducible to* Min CSP$(\Gamma)$.

*Proof.* Let $\iota : D \to X_D$ be the function defined by $\iota(d) = x_d$. Theorem 3.5 in [7] implies the following property of $\mathcal{IP}(\Gamma)$: the set of optimal solutions to $\mathcal{IP}(\Gamma)$ is equal to $\{\sigma : X_D \to D \mid \sigma \circ \iota \in \mathrm{End}\,(\Gamma)\}$.

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 $\mathbf{x}$ be a list of the variables $X_D$, and let $R = \pi_{\mathbf{x}}\mathrm{Optsol}(\mathcal{IP}(\Gamma))$. Now modify $\mathcal{J}$ to an instance $\mathcal{J}'$ of Min CSP$(\Gamma, R)$ as follows. Add the variables in $X_D$ to $V(\mathcal{J}')$, and add the constraint application $(\mathbf{x}; R)$. Furthermore, remove each constraint $(y; \{a\})$, and replace $y$ by $x_a$ throughout the instance. Let $\sigma'$ be an optimal solution to $\mathcal{J}'$. Since $\Gamma$ is a core, $g = \sigma'|_{X_D} \circ \iota$ is an automorphism of $\Gamma$, and so is its inverse, $g^{-1}$. Hence, $\sigma = g^{-1} \circ \sigma'$ is also an optimal solution to $\mathcal{J}'$. From $\sigma$ 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 CSP$(\Gamma, \mathcal{C}_D)$ to Min CSP$(\Gamma, R)$. By Proposition 3, we finally have a reduction from Min CSP$(\Gamma, R)$ to Min CSP$(\Gamma)$. □

For $a, b \in D$, let $e_{ab} : D \to D$ denote the function $e_{ab}(a) = b$ and $e_{ab}(x) = x$ for $x \neq a$. The proof of the following lemma is straightforward.

**Lemma 12.** *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 $u(b) = 1$.*

## 5 A Graph of Partial Multimorphisms

Let $\Gamma$ be a core over $D$. In this section, we define a graph $G = (V, E)$ which encodes either the **NP**-hardness of Min 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 [9].

Let $V$ be the set of partial functions $(f, g) : D^2 \to D^2$ such that (1) $f$ and $g$ are defined on a subset $\{a, b\} \subseteq D$; (2) $f$ and $g$ are idempotent and commutative; and (3) $\{f(a,b), g(a,b)\} = \{a, b\}$ or $\{f(a,b), g(a,b)\} \cap \{a, b\} = \varnothing$. We allow $a = b$

in the definition of $V$ so 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

$$\min\{h(a_1, a_2) + h(b_1, b_2), h(a_1, b_2) + h(b_1, a_2)\} < \\ h(f_1(a_1, b_1), f_2(a_2, b_2)) + h(g_1(a_1, b_1), g_2(a_2, b_2)). \quad (7)$$

We can now see how $G$ describes multimorphisms of binary functions in $\langle \Gamma, \mathcal{C}_D \rangle_{fn}$.

**Lemma 13.** *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. $\square$

For distinct $a, b \in D$, let $\overrightarrow{ab}$ denote the vertex $(f, g) \in G[a, b]$ such that $f(a, b) = a$ and $g(a, b) = b$. 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 $\overrightarrow{a_1b_1}$ and $\overrightarrow{a_2b_2}$, condition (7) reduces to:

$$h(a_1, b_2) + h(b_1, a_2) < h(a_1, a_2) + h(b_1, b_2). \quad (8)$$

For a vertex $x = (f, g)$, we let $\overline{x}$ denote the vertex $(g, f)$. It follows immediately from (7) that $\{x, y\} \in E$ iff $\{\overline{x}, \overline{y}\} \in E$. We also need to establish a number of additional properties of the graph $G$.

**Lemma 14.** *If $\{\overrightarrow{a_1b_1}, \overrightarrow{a_2b_2}\} \in E$, then there exists a function $h \in \langle \Gamma, \mathcal{C}_D \rangle_{fn}$ such that $h(a_1, b_2) = h(b_1, a_2) < h(a_1, a_2) = h(b_1, b_2)$.*

The proof of Lemma 14, and of properties (1–3) of the following lemma are very similar to the proof of Lemma 11 in Kolmogorov and Živný [9]. The main difference is that we do not have access to all unary functions, so we must be a bit more careful. Property (4) provides a way to deduce the existence of a set of neighbours for non-isolated conservative vertices; (5) and (6) follow from (4).

**Lemma 15.** *Let $x_1, \ldots, x_n$ be conservative vertices.*

1. *If $\{x_1, x_2\}$, $\{x_2, x_3\} \in E$ and $x_2 \in V'_\Gamma$, then $\{x_1, \overline{x_3}\} \in E$.*
2. *Let $(x_1, \ldots, x_n), n \geq 2$, be a path 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. *If $(x_1, \ldots, x_n, x_1)$, $n \geq 3$ is an odd cycle in $G$ and $x_2, \ldots, x_n \in V'_\Gamma$, then there is a loop on $x_1$.*
4. *If $\{\overrightarrow{a_1b_1}, \overrightarrow{a_2b_2}\} \in E$, then for each element $x \neq a_2, b_2$, either $\{\overrightarrow{a_1b_1}, \overrightarrow{a_2x}\} \in E$ or $\{\overrightarrow{a_1b_1}, \overrightarrow{xb_2}\} \in E$.*
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 $\{\overrightarrow{xy}, \overrightarrow{yz}\} \in E$.*

## 6 Classification for $|D| = 4$

We will now completely classify the 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 $\Gamma$ is a core. Let $\Sigma = \{\{x, y\} \subseteq D \mid x \neq y\}$, $\Sigma_0 = \Sigma \setminus \{\{b, c\}, \{a, d\}\}$, 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 16.** *Let $\Gamma$ 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 (sketch):** Let $\mathcal{U}$ be the set of unary functions in $\langle \Gamma, \mathcal{C}_D \rangle_{fn}$. If $\{b, c\} \notin \Sigma_\Gamma$, then $u_{bc} \notin \mathcal{U}$. Since $\Gamma$ is a core, $e_{ba}, e_{ca}, e_{bd}, e_{cd} \notin \mathrm{End}\,(\Gamma)$, so by Lemma 12, there must be a number of unary $\{0, 1\}$-valued functions in $\mathcal{U}$ to witness this. The set $\{u_{bd}, u_{cd}, u_{ab}, u_{ac}\}$ fulfils this condition, and due to the absence of $u_{bc}$, one can argue that this set must indeed lie entirely in $\mathcal{U}$. The last part of the proposition can be shown using the observation that this set can express every unary function $u$ such that $u(a) + u(d) = u(b) + u(c)$, and considering what happens when one adds a function $v$ with $v(a) + v(d) < v(b) + v(c)$. $\square$

It is possible to link properties of $G'$ to the existence of certain multimorphisms. Note that if $\{x, y\} \in \Sigma_\Gamma$, then $\overrightarrow{xy}, \overrightarrow{yx} \in V'_\Gamma$. Proposition 16 therefore gives us good control over the size of $V'_\Gamma$. In general $G[V'_\Gamma]$ needs to be bipartite unless Min CSP$(\Gamma)$ is **NP**-hard (cf. the proof of Theorem 18), so a lower bound on $\Sigma$ implies that a large induced subgraph of $G'$ needs to be bipartite. This connection is made formal by the following proposition, the proof of which is deferred to Appendix A.

**Proposition 17.** *Assume that $\Sigma_0 \subseteq \Sigma_\Gamma$. If $G'$ is bipartite, then the set of binary functions in $\langle \Gamma, \mathcal{C}_D \rangle_{fn}$ is submodular on a chain. If $G'$ is not bipartite but $G[V'_\Gamma]$ is, then the set of binary functions in $\langle \Gamma, \mathcal{C}_D \rangle_{fn}$ has a 1-defect chain multimorphism.*

We are now in a position to state and prove the main theorem.

**Theorem 18.** *Let $\Gamma$ be a core over $D = \{a, b, c, d\}$. If $\Gamma$ is submodular on a chain, or if $\Gamma$ has a 1-defect chain multimorphism, then* Min CSP$(\Gamma)$ *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 14 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 [2], the problem Min CSP$(\Gamma, \mathcal{C}_D)$ is **NP**-hard. By Proposition 11, Min CSP$(\Gamma, \mathcal{C}_D)$ reduces to Min CSP$(\Gamma)$. Hence, the latter problem is **NP**-hard as well.

If instead $G[V'_\Gamma]$ is loop-free, then it is bipartite, by Lemma 15(3). We may assume that $\Sigma_0 \subseteq \Sigma_\Gamma$: this is trivial if $\Sigma_\Gamma = \Sigma$. If $\Sigma_\Gamma$ is strictly contained in $\Sigma$, then up to an automorphism we may assume that $\{b,c\} \notin \Sigma_\Gamma$, and the inclusion follows by Proposition 16. For a $k$-ary function $h \in \Gamma$, let $\Phi(h)$ be the set of binary functions which can be obtained from $h$ by fixing $k-2$ variables, and let $\Gamma'$ be the union of $\Phi(h)$ over all $h \in \Gamma$.

Now, if $G'$ is bipartite, then by Proposition 17, the set of binary functions in $\langle \Gamma, \mathcal{C}_D \rangle_{fn}$ is submodular on a chain. Since this set contains $\Gamma'$, we may conclude, by Lemma 9, that $\Gamma$ is submodular on this chain as well. It follows that Min CSP$(\Gamma)$ is tractable [6, 15].

Otherwise, $G'$ is not bipartite, and by Proposition 17, the set of binary functions in $\langle \Gamma, \mathcal{C}_D \rangle_{fn}$ have a 1-defect chain multimorphism. Since this set contains $\Gamma'$, we may conclude, by Lemma 10 this time, that $\Gamma$ has a 1-defect chain multimorphism. It now follows from Proposition 8 that Min CSP$(\Gamma)$ is tractable. □

## 7 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 finite-valued VCSP as well, thus answering Question 4 in Živný [17]. We have also demonstrated that the techniques used by Krokhin and Larose [10] for lattices can be used effectively in the context of arbitrary algebras, and in doing so, we have given the first example of an instance of Min CSP where submodularity does not suffice to explain tractability. Finally, we have shown that graph representations such as the one defined by Kolmogorov and Živný [9] can be used to great effect, even in a non-conservative setting.

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 19.* 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 $y = b$. $\Gamma$ is a core on $\{a,b,c,d\}$ but it is not submodular on any lattice. However, $\Gamma$ have the 1-defect chain multimorphisms $(f_1, g_1)$ and $(f_2, g_2)$ from Example 6.

A related question is why bisubmodularity does not appear in the classification of Min CSP over domains of size three [8]. 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. [1], characterise the complexity of this problem (see also [3]). Multimorphisms are a special case of fractional polymorphisms. As for Min CSP, it is currently not known whether submodularity over every finite lattice implies tractability for VCSP. This is known to be true for distributive lattices, and for certain constructions on lattices, e.g. homomorphic images and Mal'tsev products [10]. The five element modular non-distributive lattice (also known as the diamond) implies tractability for *unweighted* VCSP [11]. Finally, it is known that submodularity over finite modular lattices implies containment in $\mathbf{NP} \cap \mathbf{coNP}$ [11]. 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.

## References


1. 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.
2. D. Cohen, M. Cooper, P. Jeavons, and A. Krokhin. The complexity of soft constraint satisfaction. *Artificial Intelligence*, 170(11):983–1016, 2006.
3. 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.
4. 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.
5. 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.
6. S. Iwata, L. Fleischer, and S. Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *J. ACM*, 48(4):761–777, 2001.
7. P. Jeavons, D. Cohen, and M. Gyssens. How to determine the expressive power of constraints. *Constraints*, 4(2):113–131, 1999.
8. P. Jonsson, M. Klasson, and A. Krokhin. The approximability of three-valued MAX CSP. *SIAM J. Comput.*, 35(6):1329–1349, 2006.
9. V. Kolmogorov and S. Živný. The complexity of conservative finite-valued CSPs. *CoRR*, abs/1008.1555, 2010.
10. 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.
11. F. Kuivinen. *Algorithms and Hardness Results for Some Valued CSPs*. PhD thesis, Linköping University, TCSLAB, 2009.

12. S. T. McCormick and S. Fujishige. Strongly polynomial and fully combinatorial algorithms for bisubmodular function minimization. *Math. Program.*, 122(1):87–120, 2010.
13. M. Queyranne, F. Spieksma, and F. Tardella. A general class of greedily solvable linear programs. *Math. Oper. Res.*, 23(4):892–908, 1998.
14. 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.
15. A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *J. Comb. Theory, Ser. B*, 80(2):346–355, 2000.
16. 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.
17. S. Živný. *The Complexity and Expressive Power of Valued Constraints*. PhD thesis, Oxford University Computing Laboratory, 2009.

# A Proof of Proposition 17

We will need three supporting lemmas, which are stated here without proofs. They follow without too much difficulty from the definition of the graph $G$, Lemma 15, and Proposition 16.

**Lemma 20.** *If $\Sigma_0 \subseteq \Sigma_\Gamma$, and $x \in V'$ is not isolated in $G'$, then $\{x, \overline{x}\} \in E$.*

**Lemma 21.** *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 $\{\overrightarrow{cd}, z\} \in E$.*

**Lemma 22.** *Assume that $\Sigma_\Gamma \subseteq \Sigma_0$. If there is a loop on $\overrightarrow{bc}$ or $\overrightarrow{ad}$, then there is a loop on at least one of the vertices $\overrightarrow{ab}$, $\overrightarrow{ac}$, $\overrightarrow{bd}$, $\overrightarrow{cd}$.*

**Proposition 17.** *Assume that $\Sigma_0 \subseteq \Sigma_\Gamma$. If $G'$ is bipartite, then the set of binary functions in $\langle \Gamma, \mathcal{C}_D \rangle_{fn}$ is submodular on a chain. If $G'$ is not bipartite but $G[V'_\Gamma]$ is, then the set of binary functions in $\langle \Gamma, \mathcal{C}_D \rangle_{fn}$ has a 1-defect chain multimorphism.*

*Proof.* We start by proving the case when $G'$ is bipartite. For an independent set $I$ in $G'$, let $R_I$ denote the binary relation on $D$ defined by $(x,y) \in R_I$ iff $\overrightarrow{xy} \in I$. 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 15(5), we have $\{\overrightarrow{xy}, \overrightarrow{zx}\}, \{\overrightarrow{yz}, \overrightarrow{zx}\} \in E$. By Lemma 20, $\{\overrightarrow{zx}, \overrightarrow{xz}\} \in E$, 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 13, 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 $(D; <)$.

Let $(f, g)$ denote the vertex in $G$ given by $f(b, c) = f(c, b) = a$ and $g(b, c) = g(c, b) = d$. We follow a similar strategy for the case when $G'$ is not bipartite. 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 22 that there is no loop on $\overrightarrow{bc}$, nor on $\overrightarrow{ad}$. Thus, by Lemma 15(3), $G[V'_{ad}]$ is bipartite.

Assume for the moment that the following holds:

$$\text{For } y \in D \setminus \{b, c\}, \text{ there is an odd path in } G[V'_{ad}] \text{ from } \overrightarrow{by} \text{ to } \overrightarrow{yc}. \tag{9}$$

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 (9) implies $\{x, z\} \neq \{b, c\}$. As in the case for bipartite $G'$, 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 21, (9), and Lemma 20, 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 15(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 (7) 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 16. By Lemma 13, 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 (9). If $\Sigma_\Gamma = \Sigma_{ad}$, then by Lemma 15(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 15(6). Otherwise, $\Sigma_\Gamma = \Sigma_0$. We argued above that $G'$ does not contain any loop in this case. Thus, by Lemma 15(3), every odd cycle $C$ in $G'$ must intersect both $\{\overrightarrow{bc}, \overrightarrow{cb}\}$ and $\{\overrightarrow{ad}, \overrightarrow{da}\}$. Now, by repeatedly applying Lemma 15(2) to $C$, we obtain a triangle on a subset of $\{\overrightarrow{bc}, \overrightarrow{cb}, \overrightarrow{ad}, \overrightarrow{da}\}$. By Lemma 20, 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 15(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 $\overrightarrow{dc}$. $\square$