

Computational Complexity of the Extended Minimum Cost Homomorphism Problem on Three-Element Domains

Hannes Uppman*

Department of Computer and Information Science, Linköping University,
Linköping, Sweden
hannes.uppman@liu.se

Abstract

In this paper we study the computational complexity of the extended *minimum cost homomorphism problem* (Min-Cost-Hom) as a function of a constraint language, i.e. a set of constraint relations and cost functions that are allowed to appear in instances. A wide range of natural combinatorial optimisation problems can be expressed as extended Min-Cost-Homs and a classification of their complexity would be highly desirable, both from a direct, applied point of view as well as from a theoretical perspective.

The extended Min-Cost-Hom can be understood either as a flexible optimisation version of the *constraint satisfaction problem* (CSP) or a restriction of the (general-valued) *valued constraint satisfaction problem* (VCSP). Other optimisation versions of CSPs such as the *minimum solution problem* (Min-Sol) and the *minimum ones problem* (Min-Ones) are special cases of the extended Min-Cost-Hom.

The study of VCSPs has recently seen remarkable progress. A complete classification for the complexity of finite-valued languages on arbitrary finite domains has been obtained Thapper and Živný [STOC'13]. However, understanding the complexity of languages that are not finite-valued appears to be more difficult. The extended Min-Cost-Hom allows us to study problematic languages of this type without having to deal with the full generality of the VCSP. A recent classification for the complexity of three-element Min-Sol, Uppman [ICALP'13], takes a step in this direction. In this paper we generalise this result considerably by determining the complexity of three-element extended Min-Cost-Hom.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes, F.2.2 Nonnumerical Algorithms and Problems, F.4.1 Mathematical Logic

Keywords and phrases Complexity, Optimisation, Constraint Satisfaction

Digital Object Identifier 10.4230/LIPIcs.STACS.2014.651

1 Introduction

The *constraint satisfaction problem* (CSP) is a decision problem where an instance consists of a set of variables, a set of values, and a collection of constraints expressed over the variables. The objective is to determine if it is possible to assign values to the variables in such a way that all constraints are satisfied simultaneously. In general the constraint satisfaction problem is NP-complete. However, by only allowing constraint-relations from a fixed constraint language Γ one can obtain tractable fragments. A famous conjecture by Feder and Vardi [7]

* Partially supported by the National Graduate School in Computer Science (CUGS), Sweden.

predicts that this restricted problem, denoted $\text{CSP}(\Gamma)$, is either (depending on Γ) in P or is NP-complete.

In this paper we will study an optimisation version of the CSP. Several such variants have been investigated in the literature. Examples are: the *min ones problem* (Min-Ones) [17], the *minimum solution problem* (Min-Sol) [14] and the *valued constraint satisfaction problem* (VCSP) [18]. The problem we will work with is called the extended *minimum cost homomorphism problem* (Min-Cost-Hom). The “unextended” version of this problem was, motivated by a problem in defence logistics, introduced in [9] and studied in a series of papers before its complexity was completely characterised in [20]. The extended version of the problem was introduced in [21] and differs from the original version in that it is parametrised not only by a set of allowed constraint relations, but also by a set of allowed cost functions (a formal definition is given in Section 2).

The extended Min-Cost-Hom provide a more general framework than both Min-Ones and Min-Sol; a problem of one of the latter types is also an extended Min-Cost-Hom. The VCSP-framework on the other hand is more general than the extended Min-Cost-Hom. In fact, we can describe every extended Min-Cost-Hom as a VCSP for a constraint language in which every cost function is either $\{0, \infty\}$ -valued or unary. The extended Min-Cost-Hom captures, despite this restriction, a wealth of combinatorial optimisation problems arising in a broad range of fields.

The study of VCSPs has recently seen remarkable progress; Thapper and Živný [22] described when a certain linear programming relaxation solves instances of the problem, Kolmogorov [15] simplified this description for finite-valued languages, Huber, Krokhin and Powell [10] classified all finite-valued languages on three-element domains, and Thapper and Živný [23] found a complete classification of the complexity for finite-valued languages on arbitrary finite domains.

Most of the classifications that have been obtained concerns finite-valued constraint languages ([22] mentioned above being a notable exception). Understanding the complexity of general languages appears to be more difficult. Extended Min-Cost-Homs allows us to study languages of this type without having to deal with with the full generality of the VCSP. Using techniques of the so called algebraic approach (see e.g. [2, 3, 11]), and building on results by Takhanov [20, 21] and Thapper and Živný [22, 23] we could in [24] take a step in this direction by proving a classification for the complexity of Min-Sol on the three-element domain. In this paper we generalise these results to the extended Min-Cost-Hom. Namely, we prove the following theorem.

► **Theorem 1.** *Let (Γ, Δ) be a finite language on a three-element domain D and define $\Gamma^+ = \Gamma \cup \{\{d\} : d \in D\} \cup \{\{x : \nu(x) < \infty\} : \nu \in \Delta\}$. If (Γ, Δ) is a core, then one of the following is true.*

- (Γ^+, Δ) is of semilattice type (Definition 5) and $\text{Min-Cost-Hom}(\Gamma^+, \Delta)$ is in PO.
- (Γ^+, Δ) is of tournament pair type (Definition 14) and $\text{Min-Cost-Hom}(\Gamma^+, \Delta)$ is in PO.
- $\text{Min-Cost-Hom}(\Gamma, \Delta)$ is NP-hard.

If (Γ, Δ) is of semilattice type, then $\text{Min-Cost-Hom}(\Gamma, \Delta)$ can be solved efficiently by linear programming [22]. If (Γ, Δ) is of tournament pair type we show how to reduce the problem to one demonstrated to be tractable in [20, 21]. Also in this case the underlying algorithmic technique is linear programming.

We define cores in Section 5. Theorem 1 combined with the following result, which follows from [23, Lemma 2.4], yields a full classification for the extended Min-Cost-Hom on three-element domains.

► **Proposition 2.** *If (Γ', Δ') is a core of (Γ, Δ) then $\text{Min-Cost-Hom}(\Gamma, \Delta)$ and $\text{Min-Cost-Hom}(\Gamma', \Delta')$ are polynomial-time inter-reducible.*

To obtain the classification we apply tools from the algebraic approach, and, following Thapper and Živný, we make repeated use of Motzkin’s Theorem. Our tractability results are formulated and proved for arbitrary finite domains and are therefore not restricted to the three-element case. Many of the tools we derive to aid in proving our main theorem are also effective on domains of size larger than three. One example is that we show that a relation fails to be in the wpp-closure of a language only if some fractional polymorphism of the language does not preserve the relation (Proposition 20). This complements results in [3]. Another example is that we show that all constants can be added to a core language without significantly changing the complexity of the associated extended Min-Cost-Hom (Proposition 34). This complements results in [23].

The rest of the paper is organised as follows. In Section 2 we define some fundamental concepts, in Section 3 we state and prove tractability results, in Section 4 we collect a number of results that will be used later on (these might also be useful on domain of larger size), in Section 5 we define cores [23] and prove a related result, in Section 6 we focus on the three-element domain and establish our main result; that core languages that are not tractable by the results in Section 3 are in fact NP-hard.

A longer version of this paper, containing complete proofs, is available at <http://arxiv.org/abs/1308.1394>.

2 Preliminaries

Let D be a finite set. The pair (Γ, Δ) is called a finite *language* if Γ is a finite set of finitary relations on D and Δ is a finite set of functions $D \rightarrow \mathbb{Q}_{\geq 0} \cup \{\infty\}$. For every finite language (Γ, Δ) we define the optimisation problem $\text{Min-Cost-Hom}(\Gamma, \Delta)$ as follows.

Instance: A triple (V, C, w) where

- V is a set of variables,
- C is a set of Γ -allowed constraints, i.e. a set of pairs (s, R) where the constraint-scope s is a tuple of variables, and the constraint-relation R is a member of Γ of the same arity as s ,
- w is a weight function $V \times \Delta \rightarrow \mathbb{Q}_{\geq 0}$.

Solution: A function $\varphi : V \rightarrow D$ s.t. for every $(s, R) \in C$ it holds that $\varphi(s) \in R$, where φ is applied component-wise.

Measure: The measure of a solution φ is $m(\varphi) = \sum_{v \in V} \sum_{\nu \in \Delta} w(v, \nu) \nu(\varphi(v))$. For every function $\varphi : V \rightarrow D$ that is not a solution we define $m(\varphi) = \infty$.

The objective is to find a solution φ that minimises $m(\varphi)$.

For an instance I we let $\text{Sol}(I)$ denote the set of all solutions, $\text{OptSol}(I)$ the set of all optimal solutions and $\text{Opt}(I)$ the measure of an optimal solution. If I is unsatisfiable we set $\text{Opt}(I) = \infty$. We define $0\infty = \infty 0 = 0$, $x \leq \infty$ and $x + \infty = \infty + x = \infty$ for all $x \in \mathbb{Q}_{\geq 0} \cup \{\infty\}$.

2.1 Names and Notation

A k -ary *operation* on D is a function $D^k \rightarrow D$ and a (unary) *cost function* on D is a function $D \rightarrow \mathbb{Q}_{\geq 0} \cup \{\infty\}$. The set of all operations on D is denoted \mathcal{O}_D . The i th projection operation will be denoted pr_i and the arity of a relation R is denoted $\text{ar}(R)$. We define $\binom{A}{2} = \{\{x, y\} \subseteq A : x \neq y\}$. For functions $f_1, \dots, f_k : A \rightarrow B$ and $g : B^k \rightarrow C$ we denote

by $g[f_1, \dots, f_k]$ the function $x \mapsto g(f_1(x), \dots, f_k(x))$ from A to C . For a binary operation f we define \bar{f} through $\bar{f}(x, y) = f(y, x)$. A k -ary operation f on D is called *conservative* if $f(x_1, \dots, x_k) \in \{x_1, \dots, x_k\}$ for every $x_1, \dots, x_k \in D$. A ternary operation m on D is called *arithmetical* if $m(x, y, y) = m(x, y, x) = m(y, y, x) = x$ for every $x, y \in D$. We say that an operation f on D is conservative (arithmetical) on $S \subseteq D$ if $f|_S$ is conservative (arithmetical). Similarly we say that f is conservative (arithmetical) on $\mathcal{S} \subseteq 2^D$ if $f|_S$ is conservative (arithmetical) for every $S \in \mathcal{S}$.

For a set A of operations (relations) we write $A^{(k)}$ for the set of all k -ary operations (relations) in A . For a set Γ of relations on D we use Γ^c to denote $\Gamma \cup \{\{d\} : d \in D\}$.

We use δ for the Kronecker delta function, i.e. $\delta_{x,y} = 1$ if $x = y$ and $\delta_{x,y} = 0$ otherwise. A *semilattice operation* is a binary operation that is idempotent, commutative and associative.

2.2 Polymorphisms

A function $f : D^m \rightarrow D$ is called a *polymorphism* of Γ if for every $R \in \Gamma$ and every $t_1, \dots, t_m \in R$ it holds that $f(t_1, \dots, t_m) \in R$, where f is applied component-wise. The set of all polymorphisms of Γ is denoted $\text{Pol}(\Gamma)$. A function $\omega : \text{Pol}^{(k)}(\Gamma) \rightarrow \mathbb{Q}_{\geq 0}$ is a k -ary *fractional polymorphism* [4] of (Γ, Δ) iff $\sum_{g \in \text{Pol}^{(k)}(\Gamma)} \omega(g) = 1$ and

$$\sum_{g \in \text{Pol}^{(k)}(\Gamma)} \omega(g) \nu(g(x_1, \dots, x_k)) \leq \frac{1}{k} \sum_{i=1}^k \nu(x_i) \text{ for every } \nu \in \Delta, x_1, \dots, x_k \in D.$$

The support of a fractional polymorphism ω , denoted $\text{supp}(\omega)$, is the set of polymorphisms for which ω is non-zero. The set of all fractional polymorphisms of (Γ, Δ) is denoted $\text{fPol}(\Gamma, \Delta)$.

► **Example 3.** The function pr_i is a trivial polymorphism for any set of relations Γ , and the function $f \mapsto \sum_{i=1}^k \frac{1}{k} \delta_{\text{pr}_i, f}$ is a k -ary fractional polymorphism of every language (Γ, Δ) .

2.3 Reductions

A relation R is called *pp-definable* in Γ iff there is an instance $I = (V, C)$ of $\text{CSP}(\Gamma)$ s.t. $R = \{(\varphi(v_1), \dots, \varphi(v_n)) : \varphi \in \text{Sol}(I)\}$ for some $v_1, \dots, v_n \in V$. The notation $\langle \Gamma \rangle$ is used for the set of all relations that are pp-definable in Γ . Similarly; R is called *weighted pp-definable* (wpp-definable) in (Γ, Δ) iff there is an instance $I = (V, C, w)$ of $\text{Min-Cost-Hom}(\Gamma, \Delta)$ s.t. $R = \{(\varphi(v_1), \dots, \varphi(v_n)) : \varphi \in \text{OptSol}(I)\}$ for some $v_1, \dots, v_n \in V$. We use $\langle \Gamma, \Delta \rangle_w$ to denote the set of all such relations. A function $\nu : D \rightarrow \mathbb{Q}_{\geq 0} \cup \{\infty\}$ is called *expressible* in (Γ, Δ) iff there is an instance $I = (V, C, w)$ of $\text{Min-Cost-Hom}(\Gamma, \Delta)$ and $v \in V$ s.t. $\nu(x) = \min\{m(\varphi) : \varphi : V \rightarrow D, \varphi(v) = x\}$. The set of all cost functions expressible in (Γ, Δ) is denoted $\langle \Gamma, \Delta \rangle_e$. We use $\text{Feas}(\Delta)$ for the set $\{x : \nu(x) < \infty\} : \nu \in \Delta$.

What makes these closures interesting is the following result, see e.g. [4, 5, 13].

► **Theorem 4.** Let $\Gamma' \subseteq \langle \Gamma, \Delta \rangle_w$ and $\Delta' \subseteq \langle \Gamma, \Delta \rangle_e$ be finite sets. Then, it holds that $\text{Min-Cost-Hom}(\Gamma' \cup \text{Feas}(\Delta'), \Delta')$ is polynomial-time reducible to $\text{Min-Cost-Hom}(\Gamma, \Delta)$.

3 Tractable languages

We will make use of two tractability results. The first follows from a theorem by Thapper and Živný [22, Theorem 4.1 (see remarks in Section 5)].

► **Definition 5.** We say that a finite language (Γ, Δ) is of *semilattice type* if there exists $\omega \in \text{fPol}^{(2)}(\Gamma, \Delta)$ with $f \in \text{supp}(\omega)$ s.t. f is a semilattice operation.

► **Theorem 6.** *If (Γ, Δ) is a finite language of semilattice type, then $\text{Min-Cost-Hom}(\Gamma, \Delta)$ is in PO.*

► **Example 7.** Let (Γ, Δ) be a language on a totally ordered domain D that admits the binary fractional polymorphism $f \mapsto \frac{1}{2}\delta_{\min, f} + \frac{1}{2}\delta_{\max, f}$. Certainly min is a semilattice operation, so by Theorem 6 it follows that $\text{Min-Cost-Hom}(\Gamma, \Delta)$ is in PO.

We remark that the theorem in [22] from which Theorem 6 follows is very capable; it explains the tractability of every finite-valued VCSP that is not NP-hard [23].

The second tractability result generalises a family of languages that Takhanov has proved tractable [20, 21]. The particular formulation we will use here is a bit more general than a version we previously used in [24, Theorem 8].

To state the result we need to introduce a few concepts. A central observation is given by the following lemma. The result follows immediately from the definition of fractional polymorphisms and the measure function m . We omit the proof.

► **Lemma 8.** *If (Γ, Δ) admits a k -ary fractional polymorphism ω and I is an instance of $\text{Min-Cost-Hom}(\Gamma, \Delta)$ with $\varphi_1, \dots, \varphi_k \in \text{Sol}(I)$, then $f[\varphi_1, \dots, \varphi_k] \in \text{Sol}(I)$ for every $f \in \text{supp}(\omega)$ and*

$$\sum_{f \in \text{Pol}^{(k)}(\Gamma)} \omega(f)m(f[\varphi_1, \dots, \varphi_k]) \leq \frac{1}{k} \sum_{i=1}^k m(\varphi_k).$$

► **Example 9.** Consider again Example 7. It follows from Lemma 8 that, for any instance $I = (V, C, w)$ and any $\varphi_1, \varphi_2 : V \rightarrow D$, we have $m(\min[\varphi_1, \varphi_2]) + m(\max[\varphi_1, \varphi_2]) \leq m(\varphi_1) + m(\varphi_2)$. Functions of this kind are called *submodular* and are central characters in the field of discrete optimisation, see e.g. [8].

The following definition establishes some convenient notation.

► **Definition 10.** For functions $\omega \in \text{fPol}^{(k)}(\Gamma, \Delta)$ and $x \in D, y \in D^k$ we define $W_x^\omega(y) = \sum_{f \in \text{Pol}^{(k)}(\Gamma): f(y)=x} \omega(f)$. When there is no risk of confusion we drop the superscript and simply write $W_x(y)$.

For an instance I of $\text{Min-Cost-Hom}(\Gamma, \Delta)$, a variable v and a value d we use $\text{Opt}(I, v \rightarrow d)$ to denote the optimal measure of a solution to I that maps v to d , i.e. $\min\{m(\varphi) : \varphi \in \text{Sol}(I), \varphi(v) = x\}$. Using these definitions we obtain the following corollary of Lemma 8.

► **Lemma 11.** *Let $I = (V, C, w)$ be an instance of $\text{Min-Cost-Hom}(\Gamma, \Delta)$ and $v \in V$ be s.t. $\{a_1, \dots, a_k\} \subseteq \{\varphi(v) : \varphi \in \text{Sol}(I)\}$. If (Γ, Δ) admits a k -ary fractional polymorphism ω , then*

$$\sum_{d \in D} W_d(a_1, \dots, a_k) \text{Opt}(I, v \rightarrow d) \leq \frac{1}{k} \sum_{i=1}^k \text{Opt}(I, v \rightarrow a_i).$$

► **Definition 12.** We say that $S \subseteq D$ is *shrinkable* to $S \setminus \{x\}$ in (Γ, Δ) if (Γ, Δ) admits a sequence of fractional polymorphisms $\omega_1, \dots, \omega_m$ and tuples $a^1 \in S^{k_1}, \dots, a^m \in S^{k_m}$ s.t. whenever $I = (V, C, w)$ is an instance of $\text{Min-Cost-Hom}(\Gamma, \Delta)$ with $v \in V$ s.t. $S \subseteq \{\varphi(v) : \varphi \in \text{Sol}(I)\}$ it holds that the system of inequalities we obtain from Lemma 11 applied to ω_i and a^i , for $i \in [m]$, implies that

$$\sum_{i=1}^n t_i \text{Opt}(I, v \rightarrow s_i) \leq \text{Opt}(I, v \rightarrow x)$$

for some integer n , some $t_1, \dots, t_n \in \mathbb{Q}_{\geq 0}$ s.t. $\sum_{i=1}^n t_i = 1$, and some $s_1, \dots, s_n \in S \setminus \{x\}$.

If S is shrinkable to S' and S' is shrinkable to S'' , then we say that S is shrinkable to S'' .

So, if S is shrinkable to $S \setminus \{x\}$ in (Γ, Δ) there is a set of fractional polymorphisms of (Γ, Δ) with which we can prove the existence of some $s \in S \setminus \{x\}$ s.t. if I is an instance of Min-Cost-Hom (Γ, Δ) , $v \in V$ and $S \subseteq \{\varphi(v) : \varphi \in \text{Sol}(I)\}$, then $\text{Opt}(I, v \rightarrow s) \leq \text{Opt}(I, v \rightarrow x)$.

► **Example 13.** Consider the language (Γ, \emptyset) on the domain D . Let $\{a_1, \dots, a_m\} \subseteq D$. It is not hard to see that $\omega : f \mapsto \sum_{i=1}^{m-1} \frac{1}{m-1} \delta_{\text{pr}_i, f}$ is in $\text{fPol}^{(m)}(\Gamma, \emptyset)$. Hence, ω and (a_1, \dots, a_m) certifies that $\{a_1, \dots, a_m\}$ is shrinkable to $\{a_1, \dots, a_{m-1}\}$.

We can now define the second family of tractable languages.

► **Definition 14.** A finite language (Γ, Δ) on the domain D is said to be of *tournament pair type* if $\Gamma = \Gamma^c$, $\text{CSP}(\Gamma)$ is in P and there exists $\mathcal{F} \subseteq \langle \Gamma, \Delta \rangle_w^{(1)}$, $\mathcal{A} \subseteq \binom{D}{2}$, $f_1, f_2 \in \text{Pol}^{(2)}(\Gamma)$ and $g \in \text{Pol}^{(3)}(\Gamma)$ s.t. the following holds.

- If $\{a, b\} \subseteq B$ for some $B \in \mathcal{F}$, and $\{a, b\} \notin \mathcal{A}$, then $f_1|_{\{a,b\}}$ and $f_2|_{\{a,b\}}$ are projections and $g|_{\{a,b\}}$ is arithmetical.
- If $\{a, b\} \subseteq B$ for some $B \in \mathcal{F}$, and $\{a, b\} \in \mathcal{A}$, then $f_1|_{\{a,b\}}$ and $f_2|_{\{a,b\}}$ are different idempotent, conservative and commutative operations.
- Every $S \in \langle \Gamma, \Delta \rangle_w^{(1)} \setminus \mathcal{F}$ is shrinkable to some $S' \in \mathcal{F}$.
- g is idempotent on every set in \mathcal{F} and conservative on every set in $\mathcal{F} \setminus \mathcal{A}$.

► **Theorem 15.** *If (Γ, Δ) is a finite language of tournament pair type, then Min-Cost-Hom (Γ, Δ) is in PO.*

Proof sketch. Given an instance I of Min-Cost-Hom (Γ, Δ) we can, since $\text{CSP}(\Gamma^c)$ is in P , compute for every variable v the set $D_v = \{\varphi(v) : \varphi \in \text{Sol}(I)\}$. From the definition of shrinkable sets it is immediate that if D_v is shrinkable to $S \in \langle \Gamma, \Delta \rangle_w$, then we can add the constraint (v, S) to I without deteriorating the measure of an optimal solution. We can repeat this procedure until D_v is in \mathcal{F} for every variable v .

It is known, see [24, Proof of Theorem 8], that from f_1, f_2, g one can construct (by superposition) operations f'_1, f'_2, g' that in addition to the conditions of the theorem also satisfy the following stronger properties:

- If $\{a, b\} \subseteq B$ for some $B \in \mathcal{F}$ and $\{a, b\} \notin \mathcal{A}$, then $f'_1|_{\{a,b\}} = f'_2|_{\{a,b\}} = \text{pr}_1$.
- The operation g' is idempotent and conservative on every set in \mathcal{F} .

Clearly $f'_1, f'_2, g' \in \text{Pol}(\Gamma)$. Note that f'_1, f'_2, g' preserves every unary relation $S \subseteq B$ for $B \in \mathcal{F}$. The result therefore follows from a reduction to the conservative, multi-sorted version of the problem and a result due to Takhanov for this variant [21, Theorem 23]. ◀

► **Example 16.** Consider again Min-Cost-Hom (Γ, \emptyset) . We saw in Example 13 that for every $\{x\} \subseteq X \subseteq D$ it holds that X is shrinkable to $\{x\}$. Hence, if $\Gamma^c = \Gamma$ and $\text{CSP}(\Gamma)$ is in P it follows from Theorem 15 that Min-Cost-Hom (Γ, \emptyset) is in PO. This of course is no surprise as Min-Cost-Hom (Γ, \emptyset) essentially is the same problem as $\text{CSP}(\Gamma)$.

4 Tools

Here we establish a collection of results that are used to prove the results in the last two sections. We hope this will provide an overview of the kind of techniques that are used to prove our main theorem.

Several of the results are proved with the help of the following classical theorem, see e.g. [19, p. 94].

► **Theorem 17 (Motzkin's Transposition Theorem).** *For any $A \in \mathbb{Q}^{m \times n}$, $B \in \mathbb{Q}^{p \times n}$, $b \in \mathbb{Q}^m$ and $c \in \mathbb{Q}^p$, exactly one of the following holds:*

- $Ax \leq b, Bx < c$ for some $x \in \mathbb{Q}^n$
- $A^T y + B^T z = 0$ and $(b^T y + c^T z < 0$ or $b^T y + c^T z = 0$ and $z \neq 0$) for some $y \in \mathbb{Q}_{\geq 0}^m$ and $z \in \mathbb{Q}_{\geq 0}^p$

The first result concerns a slight generalisation of the concept of dominating fractional polymorphisms [24].

► **Definition 18.** Let $k \geq 2$ and $a \in D^{k-1}, b \in D$ be s.t. a_1, \dots, a_{k-1}, b are distinct elements. A fractional polymorphism $\omega \in \text{fPol}^{(k)}(\Gamma, \Delta)$ is called (a_1, \dots, a_{k-1}, b) -dominating if $W_{a_j}^\omega(a_1, \dots, a_{k-1}, b) \geq \frac{1}{k}$ for every $j \in [k-1]$ and $\frac{1}{k} > W_b^\omega(a_1, \dots, a_{k-1}, b)$.

► **Proposition 19.** Let (Γ, Δ) be a finite language on a finite set D . Let $k \geq 2$ and $a \in D^{k-1}, b \in D$ be s.t. a_1, \dots, a_{k-1}, b are distinct. If (Γ, Δ) does not admit a fractional polymorphism that is (a_1, \dots, a_{k-1}, b) -dominating, then $\langle \Gamma, \Delta \rangle_e$ contains a unary function ν that satisfies $\infty > \nu(a_1), \dots, \nu(a_{k-1}), \nu(b)$ and $\nu(c) > \nu(b)$ for every $c \in D \setminus \{b\}$.

Using similar arguments we can also prove the following characterisation of which relations that are wpp-definable in (Γ, Δ) .

► **Proposition 20.** Let (Γ, Δ) be a finite language on a finite set D and let $\emptyset \neq R = \{t_1, \dots, t_k\} \subseteq D^n$. Exactly one of the following is true.

1. There exists $\omega \in \text{fPol}^{(k)}(\Gamma, \Delta)$ with $f \in \text{supp}(\omega)$ s.t. $f(t_1, \dots, t_k) \notin \{t_1, \dots, t_k\}$.
2. It holds that $R \in \langle \Gamma, \Delta \rangle_w$.

From Proposition 20 we can quickly derive a number of useful results.

► **Corollary 21.** Let (Γ, Δ) be a finite language on a finite set D . For any fixed k the set of wpp-definable k -ary relations, $\langle \Gamma, \Delta \rangle_w^{(k)}$, can be computed.

Proof sketch. This is immediate from Proposition 20; we can find all polymorphisms of arities $1, \dots, |D|^k$ and then, for every $R \subseteq D^k$, solve a linear program. ◀

► **Corollary 22.** Let (Γ, Δ) be a finite language on a finite set D and let $\{a, b\} \subseteq D$. If there is $\nu \in \langle \Gamma, \Delta \rangle_e$ and $A \subseteq D$ s.t. $\{a, b\} \subseteq A, A \in \langle \Gamma, \Delta \rangle_w, \nu(a) < \nu(b) < \infty$ and $\nu(b) \leq \nu(x)$ for any $x \in A \setminus \{a, b\}$, then one of the following is true.

1. $\{a, b\} \in \langle \Gamma, \Delta \rangle_w$
2. There is $\omega \in \text{fPol}^{(2)}(\Gamma, \Delta)$ that is (a, b) -dominating.

Proof. Assume (1) does not hold. By Proposition 20 there must exist some $\omega \in \text{fPol}^{(2)}(\Gamma, \Delta)$ with $f \in \text{supp}(\omega)$ s.t. $f(a, b) \notin \{a, b\}$. It is not hard to see that in this case, because of ν , the fractional polymorphism ω must be (a, b) -dominating. Hence, (2) must be true. ◀

► **Corollary 23.** Let (Γ, Δ) be a finite language on a finite set D and let $\{a_1, \dots, a_k\} \subseteq D$. One of the following is true.

1. There is $\omega \in \text{fPol}^{(k)}(\Gamma, \Delta)$ and $i \in [k]$ s.t. ω is $(a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_k, a_i)$ -dominating.
2. For every $i \in [k]$ there is $j \in [k] \setminus \{i\}$ s.t. $\{a_i, a_j\} \in \langle \Gamma, \Delta \rangle_w$.

Proof. Assume (1) is false. By Proposition 19, for any $i \in [k]$, there is $\nu_i \in \langle \Gamma, \Delta \rangle_e$ s.t. $\arg \min_{x \in D} \nu_i(x) = \{a_i\}$ and $\nu_i(x) < \infty$ if $x \in \{a_1, \dots, a_k\}$. Let $i \in [m]$. Pick j s.t. $\nu_i(a_j) = \min\{\nu_i(x) : x \in \{a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_k\}\}$.

Note that there is no $\psi \in \text{fPol}^{(2)}(\Gamma, \Delta)$ that is (a_i, a_j) -dominating; if there was then

$$f \mapsto \sum_{i=1}^{k-2} \frac{1}{k} \delta_{\text{pr}_i, f} + \sum_{g \in \text{supp}(\psi)} \frac{2}{k} \psi(g) \delta_{g[\text{pr}_{k-1}, \text{pr}_k], f}$$

would be $(b_1, \dots, b_{k-2}, a_i, a_j)$ -dominating for $b_1, \dots, b_{k-2} \in D$. Hence, by Corollary 22, we have $\{a_i, a_j\} \in \langle \Gamma, \Delta \rangle_w$. Since the choice of i was arbitrary (2) must be true. \blacktriangleleft

The generalised min-closed languages were introduced by Jonsson, Kuivinen and Nordh [12] and defined as sets of relations preserved by a particular type of binary operation. Kuivinen [16, Section 5.5] provides an alternative characterisation of the languages as those preserved by a so called min set function.

A *set function* [6] is a function $f : 2^D \setminus \{\emptyset\} \rightarrow D$. A ν -*min set function* [16] is a set function f satisfying $\nu(f(X)) \leq \min\{\nu(x) : x \in X\}$ for every $X \in 2^D \setminus \{\emptyset\}$. The following proposition, which is a variant of [16, Theorem 5.18], will later prove to be useful.

► **Proposition 24.** *Let $(\Gamma, \{\nu\})$ be a finite language s.t. $\langle \Gamma, \{\nu\} \rangle_w^{(1)} \subseteq \Gamma$. The following are equivalent:*

1. Γ is preserved by a ν -min set function,
2. Γ is preserved by a set function f s.t. $\nu(f(X)) = \min\{\nu(x) : x \in \bigcap_{Y \in \langle \Gamma \rangle: Y \supseteq X} Y\}$ for every $X \in 2^D \setminus \{\emptyset\}$,
3. Γ is preserved by a set function and for every $R \in \langle \Gamma \rangle$ it holds that

$$R \cap (\arg \min_{x \in \text{pr}_1(R)} \nu(x) \times \dots \times \arg \min_{x \in \text{pr}_{\text{ar}(R)}(R)} \nu(x)) \neq \emptyset.$$

Furthermore, if ν is injective, then the following condition is equivalent to the ones above.

4. For every $R \in \langle \Gamma \rangle$ it holds that

$$R \cap (\arg \min_{x \in \text{pr}_1(R)} \nu(x) \times \dots \times \arg \min_{x \in \text{pr}_{\text{ar}(R)}(R)} \nu(x)) \neq \emptyset.$$

Let $\nu : D \rightarrow \mathbb{Q}_{\geq 0}$ be injective. We call the binary relation R a *cross* (with respect to ν) iff $|R| \geq 2$ and there are $\alpha_1, \alpha_2 \in \mathbb{Q}_{> 0}$ s.t. $\alpha_1 \nu(t_1) + \alpha_2 \nu(t_2) = 1$ for every $t \in R$. The following lemma is a generalisation of [24, Lemma 25].

► **Lemma 25.** *Let $\nu : D \rightarrow \mathbb{Q}_{\geq 0}$ be injective. If Γ is not preserved by a ν -min set function, then $\langle \Gamma, \Delta \rangle_w$ contains a cross.*

Proof. Let \min_ν be the unique set function satisfying $\{\min_\nu(X)\} = \arg \min_{x \in X} \nu(x)$ for every $X \subseteq D$. If Γ is not preserved by a ν -min set function, then Proposition 24 implies that there is $R \in \langle \Gamma \rangle$ s.t. $(\min_\nu(\text{pr}_1(R)), \dots, \min_\nu(\text{pr}_{\text{ar}(R)}(R))) \notin R$.

In fact, there must be a binary relation in $\langle \Gamma \rangle$ of this kind. To see this let $R \in \langle \Gamma \rangle$ be a k -ary relation s.t. $(\min_\nu(\text{pr}_1(R)), \dots, \min_\nu(\text{pr}_k(R))) \notin R$ and s.t. that every relation $R' \in \langle \Gamma \rangle$ of smaller arity satisfies $(\min_\nu(\text{pr}_1(R')), \dots, \min_\nu(\text{pr}_{\text{ar}(R')}(R'))) \in R'$. This means that there is $t^1 \in R$ s.t. $t_i^1 = \min_\nu(\text{pr}_i(R))$ for $i \in [k] \setminus \{1\}$, otherwise $\text{pr}_{2, \dots, \text{ar}(R)}(R)$ contradicts the minimality of k . Similarly there is $t^2 \in R$ s.t. $t_i^2 = \min_\nu(\text{pr}_i(R))$ for $i \in [k] \setminus \{2\}$. This means that $R' = \{(x, y) : (x, y, \min_\nu(\text{pr}_3(R)), \dots, \min_\nu(\text{pr}_k(R))) \in R\}$ is a non-empty relation of arity 2 s.t. $(\min_\nu(\text{pr}_1(R')), \min_\nu(\text{pr}_2(R'))) \notin R'$. Hence, $k = 2$.

Clearly we can choose α_1, α_2 s.t. $R'' = \arg \min_{(x, y) \in R} (\alpha_1 \nu(x) + \alpha_2 \nu(y))$ satisfies $|R''| \geq 2$, and $R'' \in \langle \Gamma, \Delta \rangle_w$ is a cross. \blacktriangleleft

To prove that a given language is computationally hard we make use of the following lemma which is an immediate consequence of [20, Theorem 3.1].

► **Lemma 26.** *If $\{a, b\} \in \Gamma$ and $\nu(a) < \nu(b) < \infty$, $\sigma(b) < \sigma(a) < \infty$ for some $\nu, \sigma \in \Delta$, then either*

- there exists $f_1, f_2 \in \text{Pol}^{(2)}(\Gamma)$ s.t. $f_1|_{\{a,b\}}$ and $f_2|_{\{a,b\}}$ are two different idempotent, commutative and conservative operations,
- there exists $g \in \text{Pol}^{(3)}(\Gamma)$ s.t. $g|_{\{a,b\}}$ is arithmetical, or
- $\text{Min-Cost-Hom}(\Gamma, \Delta)$ is NP-hard.

The following result by Takhanov [20, Theorem 5.4] shows how “partially arithmetical” polymorphisms (like the ones that we might get out of the previous lemma) can be stitched together.

► **Lemma 27.** *Let $\mathcal{C} \subseteq \binom{D}{2}$. If $\mathcal{C} \subseteq \Gamma$ and for each $\{a, b\} \in \mathcal{C}$ an operation in $\text{Pol}^{(3)}(\Gamma)$ is arithmetical on $\{a, b\}$, then there is an operation in $\text{Pol}^{(3)}(\Gamma)$ that is arithmetical on \mathcal{C} .*

The next lemma is a variation, see [24, Lemma 14], of a lemma by Thapper and Živný [23, Lemma 3.5]. It allows us to prove the existence of certain nontrivial fractional polymorphisms. We may also obtain this lemma as a simple corollary of Proposition 20.

► **Lemma 28.** *If $\{(a, b), (b, a)\} \notin \langle \Gamma, \Delta \rangle_w$, then for all $\sigma \in \langle \Gamma, \Delta \rangle_e$ there is $\omega \in \text{fPol}^{(2)}(\Gamma, \Delta)$ with $f \in \text{supp}(\omega)$ s.t. $\{f(a, b), f(b, a)\} \neq \{a, b\}$ and $\sigma(f(a, b)) + \sigma(f(b, a)) \leq \sigma(a) + \sigma(b)$.*

Finally, the following lemmas are used to “canonicalise” interesting fractional polymorphisms.

► **Definition 29.** Let $P \subseteq \mathcal{O}_D^{(2)}$. For a function $\omega : P \rightarrow \mathbb{Q}_{\geq 0}$ we define $\omega^2 : P \rightarrow \mathbb{Q}_{\geq 0}$ by $\omega^2(f) = \sum_{g, h \in P: g[h, \bar{h}] = f} \omega(g)\omega(h)$.

► **Lemma 30.** *If $\omega \in \text{fPol}^{(2)}(\Gamma, \Delta)$, then $\omega^2 \in \text{fPol}^{(2)}(\Gamma, \Delta)$.*

► **Lemma 31.** *Let $\beta : D^2 \rightarrow \mathbb{Q}_{\geq 0}$ and define $C_\omega(x) = \sum_{f \in \text{Pol}^{(2)}(\Gamma): f(x) = \bar{f}(x)} \omega(f)$ and $M(\omega) = \sum_{x \in D^2} C_\omega(x)$. Set $\Omega = \{\omega \in \text{fPol}^{(2)}(\Gamma, \Delta) : \forall s \in D^2, C_\omega(s) \geq \beta(s)\}$. If $\langle \Gamma, \Delta \rangle_w^{(1)} \subseteq \Gamma$, then either $\Omega = \emptyset$, or there is $\omega^* \in \Omega$ s.t. $M(\omega^*) = \sup_{\omega \in \Omega} M(\omega)$.*

► **Lemma 32.** *Let $\mathcal{S} \subseteq \binom{D}{2}$ and $\Pi = \{\omega \in \text{fPol}^{(2)}(\Gamma, \Delta) : \text{for all } S \in \mathcal{S} \text{ there exists } f \in \text{supp}(\omega) \text{ s.t. } f|_S \text{ is commutative}\}$. If $\langle \Gamma, \Delta \rangle_w^{(1)} \subseteq \Gamma$ and $\Pi \neq \emptyset$, then there is $\omega \in \Pi$ s.t. for every $f \in \text{supp}(\omega)$ and $x \in D^2$ it holds that $\{f(x), \bar{f}(x)\} \notin \mathcal{S}$.*

5 Cores

In this section we define cores and prove that one can add all constants to a language that is a core without making the associated extended Min-Cost-Hom much more difficult. We use a definition of cores from [23, Definition 3].

► **Definition 33.** A finite language (Γ, Δ) is a *core* iff for every $\omega \in \text{fPol}^{(1)}(\Gamma, \Delta)$ and every $f \in \text{supp}(\omega)$ it holds that f is injective. A language (Γ', Δ') is a core of another language (Γ, Δ) if (Γ', Δ') is a core and $(\Gamma', \Delta') = (\Gamma, \Delta)|_{g(D)}$ for some $\psi \in \text{fPol}^{(1)}(\Gamma, \Delta)$ and $g \in \text{supp}(\psi)$.

A result very similar to the following was given in [10, 23] for finite-valued languages.

► **Proposition 34.** *If (Γ, Δ) is a core, then $\text{Min-Cost-Hom}(\Gamma^c, \Delta)$ is polynomial-time reducible to $\text{Min-Cost-Hom}(\Gamma, \Delta)$.*

Proof sketch. We will show that $\text{Min-Cost-Hom}(\Gamma^c, \Delta)$ is polynomial-time reducible to $\text{Min-Cost-Hom}(\Gamma \cup \langle \Gamma, \Delta \rangle_w^{(D)}, \Delta)$. By Theorem 4 this is sufficient.

Assume $D = \{d_1, \dots, d_{|D|}\}$. Let $R = \{(d_1, \dots, d_{|D|})\}$ and let R' be the closure of R under the operations $f \in \text{supp}(\omega)$, $\omega \in \text{fPol}^{(1)}(\Gamma, \Delta)$.

Note that there is no $k > 1$, $\psi \in \text{fPol}^{(k)}(\Gamma, \Delta)$ and $g \in \text{supp}(\psi)$ s.t. g does not preserve R' . This follows from the fact that R' was generated from a single tuple. It is not hard to show that there is $\varpi \in \text{fPol}^{(1)}(\Gamma, \Delta)$ s.t. $R' = \{f(d_1, \dots, d_{|D|}) : f \in \text{supp}(\varpi)\}$. Assume that there is $s = f(t^1, \dots, t^k) \notin R'$ for some $f \in \text{supp}(\psi)$ and $t^1, \dots, t^k \in R'$. This means that we from ψ and ϖ can construct $\varpi' \in \text{fPol}^{(1)}(\Gamma, \Delta)$ with $f \in \text{supp}(\varpi')$ s.t. $s = f(d_1, \dots, d_{|D|})$, which is a contradiction.

From Proposition 20 it follows that $R' \in \langle \Gamma, \Delta \rangle_w$. Since (Γ, Δ) is a core, for every $\omega \in \text{fPol}^{(1)}(\Gamma, \Delta)$ and $f \in \text{supp}(\omega)$ we know that f is injective. Hence, every $t \in R'$ equals $(\pi(d_1), \dots, \pi(d_{|D|}))$ for some permutation π on D .

We now use a construction that is applied for the corresponding result for CSPs [2, Theorem 4.7]. Given an instance I of $\text{Min-Cost-Hom}(\Gamma^c, \Delta)$ we create an instance of I' of $\text{Min-Cost-Hom}(\Gamma \cup \langle \Gamma, \Delta \rangle_w^{(|D|)}, \Delta)$ from I by adding variables $v_{d_1}, \dots, v_{d_{|D|}}$ and replacing every constraint $(v, \{d_i\})$ with the constraint $((v, v_{d_i}), =)$. Finally we add the constraint $((v_{d_1}, \dots, v_{d_{|D|}}), R')$. If there is a solution to I , then there is also a solution to I' . And, if ψ is an optimal solution to I' , then $(\varphi(v_{d_1}), \dots, \varphi(v_{d_{|D|}})) = (\pi(d_1), \dots, \pi(d_{|D|}))$ for some permutation π on D and $\omega \in \text{fPol}^{(1)}(\Gamma, \Delta)$ s.t. $\pi \in \text{supp}(\omega)$. Hence $\pi^k \circ \psi$ is another optimal solution to I' , for any $k \geq 1$. In particular there is an optimal solution φ^* to I' s.t. $(\varphi^*(v_{d_1}), \dots, \varphi^*(v_{d_{|D|}})) = (d_1, \dots, d_{|D|})$. This allows us to recover an optimal solution to I . \blacktriangleleft

6 Proof of Theorem 1

In this section we prove our main result. To do this we rely of a few lemmas that are proved with the help of a fair bit of case analysis. For their proofs we refer the interested reader to the longer version of this paper.

Let A denote the following assumption: (Γ, Δ) is a finite language on $D = \{a, b, c\}$ s.t. $\Gamma^c \cup \text{Feas}(\Delta) \cup \langle \Gamma, \Delta \rangle_w^{(1)} \cup \langle \Gamma, \Delta \rangle_w^{(2)} \subseteq \Gamma$.

The supporting lemma below is used in the proofs of the results that follow.

► **Lemma 35.** *Assume A. If $\{a, b\} \notin \Gamma$, then either there is $\omega \in \text{fPol}^{(2)}(\Gamma, \Delta)$ that is (a, b) or (b, a) -dominating, or there are $\nu_a, \nu_b \in \langle \Gamma, \Delta \rangle_e$ s.t. $\nu_a(a) < \nu_a(c) < \nu_a(b)$ and $\nu_b(b) < \nu_b(c) < \nu_b(a)$.*

We are going to analyse a few different cases depending on the number of two-element subsets of the domain that is wpp-definable in (Γ, Δ) . The following lemma, which follows immediately from Corollary 23, connects this number to dominating fractional polymorphisms.

► **Lemma 36.** *Assume A. Either $|\Gamma \cap \binom{D}{2}| \geq 2$ or there is $\omega \in \text{fPol}^{(3)}(\Gamma, \Delta)$ and $a_1, a_2, a_3 \in D$ s.t. ω is (a_1, a_2, a_3) -dominating and $\{a_1, a_2, a_3\} = D$.*

To understand languages that admit a ternary dominating fractional polymorphism we use the following lemma.

► **Lemma 37.** *Assume A. If $\{a, b\} \notin \Gamma$ and there is $\omega \in \text{fPol}^{(3)}(\Gamma, \Delta)$ s.t. ω is (a, b, c) -dominating, then either $\{a, c\}, \{b, c\} \in \Gamma$, or (Γ, Δ) is of semilattice type or of tournament pair type, or $\text{Min-Cost-Hom}(\Gamma, \Delta)$ is NP-hard*

The following four lemmas are used to handle languages that contain two unary two-element relations.

► **Lemma 38.** *Assume A. If $\{a, c\}, \{c, b\} \in \Gamma$ and there is $\omega \in \text{fPol}^{(2)}(\Gamma, \Delta)$ that is (a, b) -dominating, then (Γ, Δ) is of tournament pair type or $\text{Min-Cost-Hom}(\Gamma, \Delta)$ is NP-hard.*

► **Lemma 39.** *Assume A. If $\{a, b\} \notin \Gamma$ and $\{a, c\}, \{c, b\} \in \Gamma$, then either $\{(a, c), (c, a)\} \in \Gamma$, $\{(b, c), (c, b)\} \in \Gamma$, or (Γ, Δ) is of semilattice type or of tournament pair type, or $\text{Min-Cost-Hom}(\Gamma, \Delta)$ is NP-hard*

► **Lemma 40.** *Assume A. If $\{a, b\} \notin \Gamma$, $\{a, c\}, \{c, b\} \in \Gamma$ and $\{(a, c), (c, a)\} \in \Gamma$ and $\{(b, c), (c, b)\} \notin \Gamma$, then (Γ, Δ) is of tournament pair type or $\text{Min-Cost-Hom}(\Gamma, \Delta)$ is NP-hard.*

► **Lemma 41.** *Assume A. If $\{a, b\} \notin \Gamma$ and $\{(a, c), (c, a)\}, \{(b, c), (c, b)\} \in \Gamma$, then (Γ, Δ) is of tournament pair type or $\text{Min-Cost-Hom}(\Gamma, \Delta)$ is NP-hard.*

We can now prove the main theorem.

Proof of Theorem 1. Let $\Gamma' = \langle \Gamma, \Delta \rangle_w^{(1)} \cup \langle \Gamma, \Delta \rangle_w^{(2)} \cup \Gamma^c \cup \text{Feas}(\Delta)$.

Note that if (Γ', Δ) is of semilattice type or of tournament pair type, then so is $(\Gamma^c \cup \text{Feas}(\Delta), \Delta)$. Furthermore, by Theorem 4 and Proposition 34 we know that $\text{Min-Cost-Hom}(\Gamma', \Delta)$ is polynomial time reducible to $\text{Min-Cost-Hom}(\Gamma, \Delta)$. Hence, if $\text{Min-Cost-Hom}(\Gamma', \Delta)$ is NP-hard, then also $\text{Min-Cost-Hom}(\Gamma, \Delta)$ is NP-hard.

Clearly, if $\text{CSP}(\Gamma')$ is NP-hard, then so is $\text{Min-Cost-Hom}(\Gamma', \Delta)$. And, if $\text{CSP}(\Gamma')$ is not NP-hard, then it is in P. This follows from [1].

If $|\binom{D}{2} \cap \Gamma'| = 3$, then (Γ', Δ) is of tournament pair type or $\text{Min-Cost-Hom}(\Gamma', \Delta)$ is NP-hard. This follows from [24, Theorem 12].

If $|\binom{D}{2} \cap \Gamma'| < 2$, then, by Lemma 36, we know that there is $\omega \in \text{fPol}^{(3)}(\Gamma', \Delta)$ that is (a_1, a_2, a_3) -dominating for some $\{a_1, a_2, a_3\} = D$. If $\{a_1, a_2\} \notin \Gamma'$, then by Lemma 37 we know that either $|\binom{D}{2} \cap \Gamma'| = 2$ (a contradiction) or (Γ', Δ) is of semilattice type or of tournament pair type, or $\text{Min-Cost-Hom}(\Gamma', \Delta)$ is NP-hard. Otherwise $\{a_1, a_2\} \in \Gamma'$. Since $|\binom{D}{2} \cap \Gamma'| < 2$ it must hold that $\{a_1, a_3\} \notin \Gamma'$ and $\{a_2, a_3\} \notin \Gamma'$. In this case, since $\{a_1, a_2, a_3\}$ is shrinkable to $\{a_1, a_2\}$, it holds that either (Γ', Δ) is of tournament pair type or $\text{Min-Cost-Hom}(\Gamma', \Delta)$ is NP-hard.

The only remaining case is $|\binom{D}{2} \cap \Gamma'| = 2$. In this case the result follows from Lemma 39, Lemma 40 and Lemma 41. ◀

Acknowledgements. I am thankful to Peter Jonsson for rewarding discussions and to the anonymous reviewers for their helpful comments.

References

- 1 Andrei Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *Journal of the ACM*, 53(1):66–120, 2006.
- 2 Andrei Bulatov, Peter Jeavons, and Andrei Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM Journal on Computing*, 34(3):720–742, 2005.
- 3 David Cohen, Martin Cooper, Páidí Creed, Peter Jeavons, and Stanislav Živný. An algebraic theory of complexity for discrete optimization. *SIAM Journal on Computing*, 42(5):1915–1939, 2013.
- 4 David Cohen, Martin Cooper, and Peter Jeavons. An algebraic characterisation of complexity for valued constraints. In *Proceedings of CP 2006*, volume 4204 of *Lecture Notes in Computer Science*, pages 107–121. Springer Berlin Heidelberg, 2006.
- 5 David Cohen, Martin Cooper, Peter Jeavons, and Andrei Krokhin. The complexity of soft constraint satisfaction. *Artificial Intelligence*, 170(11):983–1016, 2006.
- 6 Víctor Dalmau and Justin Pearson. Closure functions and width 1 problems. In *Proceedings of CP 1999*, volume 1713 of *Lecture Notes in Computer Science*, pages 159–173. Springer Berlin Heidelberg, 1999.

- 7 Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.
- 8 Satoru Fujishige. *Submodular Functions and Optimization*, volume 58 of *Annals of Discrete Mathematics*. Elsevier, 2005.
- 9 Gregory Gutin, Arash Rafiey, Anders Yeo, and Michael Tso. Level of repair analysis and minimum cost homomorphisms of graphs. *Discrete Applied Mathematics*, 154(6):881–889, 2006.
- 10 Anna Huber, Andrei Krokhin, and Robert Powell. Skew bisubmodularity and valued CSPs. In *Proceedings of SODA 2013*, pages 1296–1305, 2013.
- 11 Peter Jeavons, David Cohen, and Marc Gyssens. Closure properties of constraints. *Journal of the ACM*, 44(4):527–548, 1997.
- 12 Peter Jonsson, Fredrik Kuivinen, and Gustav Nordh. MAX ONES generalized to larger domains. *SIAM Journal on Computing*, 38(1):329–365, 2008.
- 13 Peter Jonsson, Fredrik Kuivinen, and Johan Thapper. Min CSP on four elements: Moving beyond submodularity. In *Proceedings of CP 2011*, volume 6876 of *Lecture Notes in Computer Science*, pages 438–453. Springer Berlin Heidelberg, 2011.
- 14 Peter Jonsson and Gustav Nordh. Introduction to the maximum solution problem. In *Complexity of Constraints*, volume 5250 of *Lecture Notes in Computer Science*, pages 255–282. Springer Berlin Heidelberg, 2008.
- 15 Vladimir Kolmogorov. The power of linear programming for finite-valued CSPs: A constructive characterization. In *Proceedings of ICALP 2013*, volume 7965 of *Lecture Notes in Computer Science*, pages 625–636. Springer Berlin Heidelberg, 2013.
- 16 Fredrik Kuivinen. *Algorithms and Hardness Results for Some Valued CSPs*. PhD thesis, Linköping University, The Institute of Technology, 2009.
- 17 Creignou Nadia, Khanna Sanjeev, and Sudan Madhu. *Complexity Classifications of Boolean Constraint Satisfaction Problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001.
- 18 Thomas Schiex, Hélène Fargier, and Gérard Verfaillie. Valued constraint satisfaction problems: Hard and easy problems. In *Proceedings of IJCAI 1995*, pages 631–637, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.
- 19 Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1986.
- 20 Rustem Takhanov. A dichotomy theorem for the general minimum cost homomorphism problem. In *Proceedings of STACS 2010*, number 5 in Leibniz International Proceedings in Informatics (LIPIcs), pages 657–668, Dagstuhl, Germany, 2010. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- 21 Rustem Takhanov. Extensions of the minimum cost homomorphism problem. In *Proceedings of COCOON 2010*, volume 6196 of *Lecture Notes in Computer Science*, pages 328–337. Springer Berlin Heidelberg, 2010.
- 22 Johan Thapper and Stanislav Živný. The power of linear programming for valued CSPs. In *Proceedings of FOCS 2012*, pages 669–678. IEEE Computer Society, 2012.
- 23 Johan Thapper and Stanislav Živný. The complexity of finite-valued CSPs. In *Proceedings of STOC 2013*, pages 695–704, New York, NY, USA, 2013. ACM.
- 24 Hannes Uppman. The complexity of three-element Min-Sol and conservative Min-Cost-Hom. In *Proceedings of ICALP 2013*, volume 7965 of *Lecture Notes in Computer Science*, pages 804–815. Springer Berlin Heidelberg, 2013.