The complexity of finite-valued CSPs

JOHAN THAPPER, Université Paris-Est Marne-la-Vallée, France
STANISLAV ŽIVNÝ, University of Oxford, United Kingdom

We study the computational complexity of exact minimisation of rational-valued discrete functions. Let Γ be a set of rational-valued functions on a fixed finite domain; such a set is called a *finite-valued constraint language*. The valued constraint satisfaction problem, VCSP(Γ), is the problem of minimising a function given as a sum of functions from Γ. We establish a dichotomy theorem with respect to exact solvability for *all* finite-valued constraint languages defined on domains of *arbitrary* finite size.

We show that every constraint language Γ either admits a binary symmetric fractional polymorphism in which case the basic linear programming relaxation solves any instance of VCSP(Γ) exactly, or Γ satisfies a simple hardness condition that allows for a polynomial-time reduction from Max-Cut to VCSP(Γ).

Categories and Subject Descriptors: F.2.0 [**Analysis of Algorithms and Problem Complexity**]: General

General Terms: Algorithms, Theory

Additional Key Words and Phrases: discrete optimisation, complexity, dichotomy, valued constraint satisfaction problems

ACM Reference Format:
Thapper, J. and Živný, S. 2013. The complexity of finite-valued VCSPs. J. ACM 1, 1, Article 1 (May 2014), 33 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

In this paper we study the following problem: what classes of discrete extensionally-represented functions can be minimised exactly in polynomial time? Such problems can be readily described as (finite-)valued constraint satisfaction problems. We provide a complete answer to this question for rational-valued functions defined on arbitrary finite domains.

The constraint satisfaction problem, or CSP for short, provides a common framework for many theoretical and practical problems in computer science. Problems that can be cast in the CSP framework have been studied in several contexts of computer science including artificial intelligence [Dechter 2003], database theory [Kolaitis and Vardi 2000], and graph theory [Hell and Nešetřil 2004; 2008]. A CSP instance can informally be described as a set of variables to be assigned values from the domains of the variables

An extended abstract of this work appeared in the *Proceedings of the 45th ACM Symposium on the Theory of Computing (STOC)*, pp. 695-704, 2013 [Thapper and Živný 2013].
Johan Thapper was partially supported by the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013 Grant Agreement no. 257039).
Stanislav Živný was supported by a Royal Society University Research Fellowship.
Author's addresses: J. Thapper, LIGM, Université Paris-Est Marne-la-Vallée, Cité Descartes, 5, Boulevard Descartes Champs-sur-Marne 77454 Marne-la-Vallée Cedex 2, France; S. Živný, Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, OX1 3QD Oxford, United Kingdom.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

so that all constraints are satisfied [Montanari 1974]. The CSP is NP-complete in general and thus we are interested in restrictions which give rise to tractable classes of problems. Following [Feder and Vardi 1998], we restrict the constraint language, that is, all constraint relations in a given instance must belong to a fixed, finite set of relations on the domain. The most successful approach to classifying language-restricted CSPs is the so-called algebraic approach [Jeavons et al. 1997; Jeavons 1998; Bulatov et al. 2005], which has led to several complexity classifications [Bulatov 2006; 2011; Barto et al. 2009; Barto 2011] and algorithmic characterisations [Barto and Kozik 2014; Idziak et al. 2010; Berman et al. 2010] going beyond the seminal work of Schaefer on Boolean CSPs [Schaefer 1978].

Several natural optimisation variants of CSPs have been studied in the literature such as Max-CSP, where the goal is to maximise the number of satisfied constraints (or, equivalently, minimise the number of unsatisfied constraints) [Cohen et al. 2005; Creignou et al. 2001; Jonsson et al. 2006; Jonsson et al. 2011; Deineko et al. 2008], and Max-Ones [Creignou et al. 2001; Jonsson et al. 2008] and Min-Cost-Hom [Takhanov 2010a; 2010b; Uppman 2013; 2014], where all constraints have to be satisfied and some additional function of the assignment is optimised. The most general variant is the valued constraint satisfaction problem, or VCSP for short, which deals with both feasibility and optimisation [Cohen et al. 2006a; Živný 2012]. A valued constraint language Γ is a set of functions on a fixed domain and a VCSP instance over Γ is given by a sum of functions from Γ with the goal to minimise the sum. The VCSP framework is very robust and has also been studied under different names such as Min-Sum problems, Gibbs energy minimisation, Markov Random Fields, Conditional Random Fields and others in different contexts in computer science [Lauritzen 1996; Wainwright and Jordan 2008; Crama and Hammer 2011]. The VCSP in its full generality considers functions with the range being the rationals with positive infinity [Cohen et al. 2006a]; this includes both CSPs (feasibility) and Max-CSPs (optimisation) as special cases where the range of the functions is $\{0, \infty\}$ and $\{0, 1\}$, respectively. In this work we will focus on finite-valued VCSPs, that is, the range of the functions is the set of rationals. Finite-valued CSPs capture optimisation problems. (Finite-valued CSPs are called generalised CSPs in the approximation community [Raghavendra 2008].)

Given the generality of the VCSP, it is not surprising that only few complexity classifications are known. In the general-valued case (that is, when the range of the functions is the rationals with positive infinity), only constraint languages on a two-element domain [Cohen et al. 2006a; Creed and Živný 2011] and conservative (containing all $\{0, 1\}$-valued unary functions) constraint languages [Kolmogorov and Živný 2013] have been completely classified with respect to exact solvability. In the finite-valued case, constraint languages on two-element domains [Cohen et al. 2006a], three-element domains [Huber et al. 2014], and conservative constraint languages [Kolmogorov and Živný 2013] have been completely classified with respect to exact solvability. In the special case of $\{0, 1\}$-valued constraint languages, which correspond to Max-CSPs, constraint languages on two-element domains [Creignou 1995], three-element domains [Jonsson et al. 2006], four-element domains [Jonsson et al. 2011], and conservative (containing all $\{0, 1\}$-valued unary functions) constraint languages [Deineko et al. 2008] have been classified with respect to exact solvability. Generalising the algebraic approach to CSPs [Bulatov et al. 2005], algebraic properties called multimorphisms [Cohen et al. 2006a], fractional polymorphisms [Cohen et al. 2006b], and weighted polymorphisms [Cohen et al. 2013] have been invented for the study of the computational complexity of classes of VCSPs.

1.1. Contribution

We study the computational complexity of finite-valued constraint languages on arbitrary finite domains. We characterise all tractable finite-valued constraint languages as those admitting a binary symmetric fractional polymorphism. Tractability follows from the results in [Thapper and Živný 2012; Kolmogorov 2013] (see also [Kolmogorov et al. 2015], which is an extended version of [Thapper and Živný 2012] and [Kolmogorov 2013]) that show that all instances over such constraint languages are solvable by the basic linear programming relaxation (BLP). In the other direction, we show that instances over constraint languages not admitting such a fractional polymorphism are NP-hard by a reduction from Max-Cut [Garey and Johnson 1979].

THEOREM 1.1. *Let D be an arbitrary finite set and let Γ be a finite-valued constraint language defined on D. VCSP(Γ) is tractable if, and only if, the BLP solves VCSP(Γ). Otherwise, VCSP(Γ) is NP-hard.*

An explicit hardness condition is given in Theorem 3.4.

Our results generalise *all previous partial classifications* of finite-valued constraint languages: the classifications of $\{0,1\}$-valued constraint languages on two-element, three-element, and four-element domains obtained in [Creignou 1995; Creignou et al. 2001], [Jonsson et al. 2006], and [Jonsson et al. 2011], respectively; the classification of $\{0,1\}$-valued constraint languages containing all unary functions obtained in [Deineko et al. 2008]; the classifications of finite-valued constraint languages on two-element and three-element domains obtained in [Cohen et al. 2006a] and [Huber et al. 2014], respectively; the classification of finite-valued constraint languages containing all $\{0,1\}$-valued unary functions obtained in [Kolmogorov and Živný 2013]; and the classification of Min-0-Ext problems obtained in [Hirai 2015].

Our results demonstrate that (i) a binary symmetric fractional polymorphism is sufficient for characterising tractability, and (ii) only cores and constants are required for the hardness condition (details are explained in Section 2). This is in contrast with ordinary CSPs (that is, the decision problems), where the hardness condition also requires an equivalence relation and the conjectured tractable cases are characterised by polymorphisms of arity higher than two [Bulatov et al. 2005].

Another problem tackled here is referred to, in [Creignou et al. 2001], as the *meta problem*: given a finite-valued constraint language Γ, decide whether it gives rise to a tractable class VCSP(Γ). We show that the meta problem is solvable in polynomial time when the constraint language Γ is assumed to be a core. However, we also show that deciding whether Γ is a core is co-NP-complete and that deciding whether a given Γ' is a core of Γ is DP-complete. In particular, all considered meta problems are decidable.

A finite-valued constraint language Γ is called *tractable* if every finite subset $\Gamma' \subseteq \Gamma$ gives rise to a tractable class VCSP(Γ'). However, in principle, the algorithms solving VCSP(Γ') for different finite subsets of Γ could be quite different. If there exists a uniform polynomial-time algorithm for VCSP(Γ) then we say that Γ is *globally tractable*. In the case of ordinary CSPs (that is, decision problems), in all known cases every tractable constraint language is also globally tractable. Our results show that this holds in general for finite-valued constraint languages: all tractable infinite constraint languages are globally tractable, using the BLP relaxation, and all other constraint languages are NP-hard. We therefore derive a dichotomy result also for *infinite* finite-valued constraint languages.

The proof of our main result is a combination of various techniques. We elaborate on a slightly different, but equivalent, notion of core for finite-valued constraint languages from that used in [Huber et al. 2014]. We introduce the idea of studying expressible unary functions by encoding them in hyperplane arrangements. We also use the idea introduced in [Kolmogorov 2013] of working with *generalised fractional polymorphisms* but derive the necessary technical machinery using a Markov chain argument. This also provides natural way to derive the main result from [Kolmogorov 2013] which says that having a binary symmetric fractional polymorphism implies having symmetric fractional polymorphisms of all arities.

Since the announcement of our results in the conference version of this article [Thapper and Živný 2013], the techniques presented here have proved essential in recent complexity classifications of Min-Sol problems and Min-Cost-Hom problems, which are special cases of VCSPs [Uppman 2013; 2014].

1.2. Related work

Apart from language-based restrictions on (V)CSPs, also structure-based restrictions [Grohe 2007; Marx 2013; Gottlob et al. 2009; Färnqvist 2012] and hybrid restrictions [Cooper and Živný 2011; 2012] have been studied. Not only exact solvability, but also approximability of Max-CSPs and VCSPs has attracted a lot of attention [Creignou et al. 2001; Khanna et al. 2001; Håstad 2001; 2008; Guruswami and Raghavendra 2008; Jonsson et al. 2009]. Moreover, the robust approximability of Max-CSPs has also been studied [Kun et al. 2012; Barto and Kozik 2012; Dalmau and Krokhin 2013]. Under the assumption of the *unique games conjecture* [Khot 2010], Raghavendra has shown that the basic semidefinite programming (SDP) relaxation solves all tractable finite-valued CSPs (without a characterisation of the tractable cases) [Raghavendra 2008]. Moreover, Chapters 6 and 7 of [Raghavendra 2009] imply that if a finite-valued constraint language Γ admits a cyclic fractional polymorphism of some arity $k \geq 2$ then the basic SDP relaxation solves any VCSP instance over Γ. Our results show, assuming $\mathrm{P} \neq \mathrm{NP}$, that for exact solvability the BLP relaxation suffices.

2. PRELIMINARIES

We use the following notation: any name with a bar denotes a tuple. We denote by x_i the ith component of a tuple $\bar{x}$. Superscripts are used for collections of tuples; e.g., we write x_i^j for the ith component of the jth tuple $\bar{x}^j$.

2.1. Valued CSPs

Let D be a finite set called the *domain*. We denote by $\mathbb{Q}_{>0}$, $\mathbb{Q}_{\geq 0}$, and $\mathbb{Q}$, respectively, the set of positive rational numbers, nonnegative rational numbers, and rational numbers. A *(cost) function* is any function $f : D^m \to \mathbb{Q}$, where $m = ar(f)$ is the *arity* of f. A *valued constraint language* Γ is a set of cost functions. Unless specifically said otherwise, we assume that all constraint languages under consideration are *finite*. Valued constraint languages consisting of $\mathbb{Q}$-valued cost functions that do not take on infinite costs are called *finite-valued* constraint languages in the literature and this is the term we used in the abstract and introduction. Since we exclusively study finite-valued constraint languages, for simplicity we omit the words "valued" and "finite-valued" and in the rest of the paper we say simply "constraint language".

Definition 2.1. An instance I of the *valued constraint satisfaction problem*, or VCSP for short, is given by the set $V = \{x_1, \ldots, x_n\}$ of variables and the objective function $f_I(x_1, \ldots, x_n) = \sum_{i=1}^{q} w_i \cdot f_i(\bar{x}^i)$ where, for every $1 \leq i \leq q$, $f_i : D^{ar(f_i)} \to \mathbb{Q}$, $\bar{x}^i \in V^{ar(f_i)}$, and $w_i \in \mathbb{Q}_{\geq 0}$ is a weight. The functions f_i are extensionally represented, i.e., given by a table of costs for all possible $|D|^{ar(f_i)}$ assignments. A solution to I is a function $h : V \to D$, its measure given by $\sum_{i=1}^{q} w_i \cdot f_i(h(\bar{x}^i))$, where h is applied componentwise. The goal is to find a solution of minimum measure.

We denote by VCSP(Γ) the class of all instances in which all functions are from Γ. The minimum measure of a solution to an instance $I \in$ VCSP(Γ) is denoted by $\mathrm{Opt}_\Gamma(I)$. A constraint language Γ is called *tractable* if, for any finite $\Gamma' \subseteq \Gamma$, VCSP(Γ') is tractable, that is, a solution of measure $\mathrm{Opt}_\Gamma(I)$ can be found for any instance $I \in$ VCSP(Γ') in polynomial time; Γ is called *NP-hard* if VCSP(Γ') is NP-hard for some finite $\Gamma' \subseteq \Gamma$. Moreover, Γ is called *globally tractable* if there is a uniform algorithm for VCSP(Γ).

2.2. Expressive power

Definition 2.2. For a constraint language Γ, we let $\langle\Gamma\rangle$ be the set of all functions $f(x_1, \dots, x_m)$ such that for some instance $I \in$ VCSP(Γ) with objective function $f_I(x_1, \dots, x_m, x_{m+1}, \dots, x_n)$, we have

$$f(x_1, \dots, x_m) \;=\; \min_{x_{m+1},\dots,x_n} f_I(x_1, \dots, x_m, x_{m+1}, \dots, x_n)\,.$$

We then say that Γ *expresses* f and call $\langle\Gamma\rangle$ the *expressive power* of Γ.

In other words, $\langle\Gamma\rangle$ is the closure of Γ under addition, multiplication by nonnegative constants, and minimisation over extra variables. For two functions f and f', we write $f \equiv f'$ if $f = a \cdot f' + b$ for some $a \in \mathbb{Q}_{>0}$ and $b \in \mathbb{Q}$, i.e., if f can be obtained from f' by *scaling* and *translation*. For a constraint language Γ, let $\Gamma_\equiv = \{f \mid f \equiv f' \text{ for some } f' \in \Gamma\}$. It has been shown that with respect to exact solvability, we only need to consider constraint languages closed under expressibility, scaling, and translation:

THEOREM 2.3 ([COHEN ET AL. 2006A]). *Let Γ be a constraint language and Γ' a finite set such that $\Gamma' \subseteq \langle\Gamma\rangle_\equiv$. Then VCSP($\Gamma'$) polynomial-time reduces to VCSP(Γ).*

We define the following condition:

There exist distinct $a, b \in D$ such that $\langle\Gamma\rangle$ contains a binary function f with $\operatorname{argmin} f = \{(a,b),(b,a)\}$. **(MC)**

A slightly different condition[1] was formulated in [Huber et al. 2014]:

There exist distinct $a, b \in D$ such that $\langle\Gamma\rangle$ contains a unary function u with $\operatorname{argmin} u = \{a,b\}$ *and a binary function f with* $f(a,b) = f(b,a) < f(a,a) = f(b,b)$. **(MC′)**

Observe that (MC′) implies (MC). In fact, we will now prove that the two conditions are equivalent.

LEMMA 2.4. *For any constraint language Γ, (MC) holds if, and only, if (MC′) holds.*

PROOF. We need to prove that (MC) implies (MC′). Let Γ be a constraint language with a function $f \in \langle\Gamma\rangle$ such that $\operatorname{argmin} f = \{(a,b),(b,a)\}$. Note that $u(x) = \min_y f(x,y)$ is a unary function with $\operatorname{argmin} u = \{a,b\}$. If $f(a,a) = f(b,b)$, then u and f satisfy (MC′). Otherwise, assume without loss of generality that $f(a,b) = f(b,a) = 0$, $f(x,y) \geq 1$ for $\{x,y\} \neq \{a,b\}$, and that $f(a,a) < f(b,b)$. Let $K = \max_x f(a,a) - f(x,x)$, and define $u'(x) = \min_y K \cdot u(y) + f(y,y) + f(x,y)$. Note that $u(x) = 0$ for $x = a, b$ and $u(x) \geq 1$ otherwise. Also note that $\min_y f(y,y) = f(a,a) - K$. The three arguments in the following $\min$-expressions correspond to the cases $y \notin \{a,b\}$, $y = a$, and $y = b$, respectively.

$$u'(x) \geq \min\{K + (f(a,a) - K) + 1, 0 + f(a,a) + 1, 0 + f(b,b) + 1\} > f(a,a) \qquad (x \neq a,b)$$
$$u'(a) \geq \min\{K + (f(a,a) - K) + 1, 0 + f(a,a) + f(a,a), 0 + f(b,b) + 0\} > f(a,a)$$
$$u'(b) \leq K \cdot f(a,b) + f(a,a) + f(b,a) = f(a,a)$$

[1]Condition (MC′) was called (MC) in [Huber et al. 2014].

Thus $\operatorname{argmin} u' = \{b\}$.

Now, let $\delta = f(b,b) - f(a,a) > 0$ and define

$$f'(x,y) = f(x,y) + \frac{\delta}{2}\frac{u'(x)+u'(y)}{u'(a)-u'(b)}.$$

We now verify that $f'(a,b) = f'(b,a) < f'(a,a) = f'(b,b)$:

$$\begin{aligned}
f'(a,a) - f'(a,b) &= f(a,a) + \frac{\delta}{2}\frac{u'(a)+u'(a)}{u'(a)-u'(b)} - f(a,b) - \frac{\delta}{2}\frac{u'(a)+u'(b)}{u'(a)-u'(b)} \\
&= f(a,a) + (f(b,b)-f(a,a))\frac{1}{2}\frac{u'(a)-u'(b)}{u'(a)-u'(b)} \\
&= \frac{1}{2}(f(a,a)+f(b,b)) > 0, \\
f'(a,a) - f'(b,b) &= f(a,a) + \frac{\delta}{2}\frac{u'(a)+u'(a)}{u'(a)-u'(b)} - f(b,b) - \frac{\delta}{2}\frac{u'(b)+u'(b)}{u'(a)-u'(b)} \\
&= f(a,a) - f(b,b) + (f(b,b)-f(a,a))\frac{u'(a)-u'(b)}{u'(a)-u'(b)} = 0, \\
f'(a,b) - f'(b,a) &= f(a,b) + \frac{\delta}{2}\frac{u'(a)+u'(b)}{u'(a)-u'(b)} - f(b,a) - \frac{\delta}{2}\frac{u'(b)+u'(a)}{u'(a)-u'(b)} = 0.
\end{aligned}$$

It follows that u and f' satisfy (MC′). □

It is known that condition (MC′) and thus, by Lemma 2.4, condition (MC) implies intractability (via a reduction from Max-Cut [Garey and Johnson 1979]):

LEMMA 2.5 ([COHEN ET AL. 2006A]). *If a constraint language* Γ *satisfies condition* (MC) *then* Γ *is NP-hard.*

2.3. Fractional polymorphisms

For a cost function f and $\bar{a}^1, \ldots, \bar{a}^m \in D^{ar(f)}$, let $f^m(\bar{a}^1, \ldots, \bar{a}^m) := \frac{1}{m}(f(\bar{a}^1) + \cdots + f(\bar{a}^m))$. An m-ary *operation* on D is a function $g : D^m \to D$. Let $\mathcal{O}_D^{(m)}$ denote the set of all m-ary operations on D. An m-ary *fractional operation* is a function $\omega : \mathcal{O}_D^{(m)} \to \mathbb{Q}_{\geq 0}$ such that $\|\omega\|_1 = 1$, where $\|\omega\|_1 := \sum_g \omega(g)$.[2] The set $\{g \mid \omega(g) > 0\}$ of operations is called the *support* of ω and is denoted by $\operatorname{supp}(\omega)$. For an operation g, we denote by χ_g the fractional operation that takes the value 1 on the operation g and 0 on all other operations.

A fractional operation ω is called an m-ary *fractional polymorphism* [Cohen et al. 2006b] of f if, for all $\bar{a}^1, \ldots, \bar{a}^m \in D^{ar(f)}$, it holds that

$$\sum_{g \in \mathcal{O}_D^{(m)}} \omega(g) f(g(\bar{a}^1, \ldots, \bar{a}^m)) \leq f^m(\bar{a}^1, \ldots, \bar{a}^m), \tag{1}$$

where the operations g are applied componentwise. If ω is a fractional polymorphism of f then we say that ω *improves* f and that f *admits* the fractional polymorphism ω.

If ω is a fractional polymorphism of every cost function in a constraint language Γ, then ω is called a fractional polymorphism of Γ, and we say that Γ admits the fractional polymorphism ω.

[2]In [Thapper and Živný 2013], fractional operations were defined without the requirement $\|\omega\|_1 = 1$ which was instead added to the definition of fractional polymorphisms. The present definition better matches the semantics of the qualifier "fractional".

It is known and easy to show that expressibility preserves fractional polymorphisms: if ω is a fractional polymorphism of Γ then ω is also a fractional polymorphism of $\langle\Gamma\rangle$ [Cohen et al. 2006b].

An operation g is *idempotent* if $g(x,\ldots,x)=x$. Let S_m be the symmetric group on $\{1,\ldots,m\}$. An m-ary operation g is *symmetric* if, for every permutation $\pi\in S_m$, we have $g(x_1,\ldots,x_m)=g(x_{\pi(1)},\ldots,x_{\pi(m)})$. An m-ary operation g is *cyclic* if $g(x_1,x_2,\ldots,x_m)=g(x_2,\ldots,x_m,x_1)$ for all $x_1,\ldots,x_m\in D$. Note that in the case of $m=2$, an operation is symmetric if, and only if, it is cyclic. A fractional operation is called idempotent, symmetric, or cyclic if all operations in its support are idempotent, symmetric, or cyclic, respectively.

A *mapping* of arity $m\to k$ on D is a function $\mathbf{g}:D^m\to D^k$. Let $\mathcal{O}_D^{(m\to k)}$ denote the set of all mappings of arity $m\to k$ on D. A *fractional mapping* (of arity $m\to k$) is a function $\rho:\mathcal{O}_D^{(m\to k)}\to\mathbb{Q}_{\geq 0}$ such that $\|\rho\|_1=1$, where $\|\rho\|_1:=\sum_{\mathbf{g}}\rho(\mathbf{g})$. A fractional mapping ρ is called a *generalised fractional polymorphism* (of arity $m\to k$) of f if, for all $\bar{a}^1,\ldots,\bar{a}^m\in D^{ar(f)}$, it holds that

$$\sum_{\mathbf{g}\in\mathcal{O}_D^{(m\to k)}}\rho(\mathbf{g})f^k(\mathbf{g}(\bar{a}^1,\ldots,\bar{a}^m))\leq f^m(\bar{a}^1,\ldots,\bar{a}^m). \tag{2}$$

As for ordinary fractional polymorphisms, we say that ρ is a generalised fractional polymorphism of a constraint language Γ if ρ is a generalised fractional polymorphism of every cost function from Γ and say that Γ admits ρ.

The definitions of the fractional mapping $\chi_{\mathbf{g}}$, given a mapping $\mathbf{g}$, and of the support $\mathrm{supp}(\rho)$ of a fractional mapping ρ are analogous to those for fractional operations.

A mapping $\mathbf{g}$ of arity $m\to k$ is *symmetric* if, for every permutation $\pi\in S_m$, we have $\mathbf{g}(x_1,\ldots,x_m)=\mathbf{g}(x_{\pi(1)},\ldots,x_{\pi(m)})$, and a fractional mapping is called symmetric if all mappings in its support are symmetric.

Note that a fractional polymorphism of arity m is the same as a generalised fractional polymorphism of arity $m\to 1$. In fact a fractional mapping of arity $m\to k$ is just a tuple of k fractional operations of arity $m\to 1$; however, this viewpoint, introduced in [Kolmogorov 2013], turns out to be very useful. For brevity, we will often omit the word "generalised" when no ambiguity can arise.

2.4. Cores

Let $S\subseteq D$. The *sub-language* $\Gamma[S]$ *of* Γ *induced by* S is the constraint language defined on domain S and containing the restriction of every function $f\in\Gamma$ onto S.

Definition 2.6. A constraint language Γ is a *core* if for every unary fractional polymorphism ω of Γ, $\mathrm{supp}(\omega)$ contains only injective operations. A constraint language Γ' is a *core of* Γ if Γ' is a core and $\Gamma'=\Gamma[g(D)]$ for some $g\in\mathrm{supp}(\omega)$ with ω a unary fractional polymorphism of Γ.

The following lemma implies that we may always assume that Γ is a core constraint language. It is an immediate consequence of Lemma 2.9 below.

LEMMA 2.7. *If* Γ' *is a core of* Γ *then* $\mathrm{Opt}_\Gamma(I)=\mathrm{Opt}_{\Gamma'}(I')$ *for all instances* $I\in$ *VCSP*(Γ)*, where* I' *is obtained from* I *by substituting each function in* Γ *for its restriction in* Γ'.

We will need the following variation of Motzkin's transposition theorem.

LEMMA 2.8. *For any* $A\in\mathbb{Q}^{m\times n}$, $B\in\mathbb{Q}^{p\times n}$*, exactly one of the following holds:*

— $Ay>0$, $By\geq 0$*, for some* $y\in\mathbb{Q}^n_{\geq 0}$*; or*

— $A^\top z_1 + B^\top z_2 \leq 0$, *for some* $0 \neq z_1 \in \mathbb{Q}^m_{\geq 0}, z_2 \in \mathbb{Q}^p_{\geq 0}$.

PROOF. The following variation of Motzkin's transposition theorem is from [Schrijver 1986, Corollary 7.1k] with $b = c = 0$ and the matrices multiplied by -1: For any $A' \in \mathbb{Q}^{m' \times n'}$, $B' \in \mathbb{Q}^{p' \times n'}$, exactly one of the following holds:

(1) $A'y' > 0$, $B'y' \geq 0$, for some $y' \in \mathbb{Q}^{n'}$; or
(2) $A'^\top z'_1 + B'^\top z'_2 = 0$, for some $0 \neq z'_1 \in \mathbb{Q}^{m'}_{\geq 0}, z'_2 \in \mathbb{Q}^{p'}_{\geq 0}$.

Given A and B as in the statement of the lemma, set $n' = n$, $m' = m$, $p' = p + n$, $A' = A$ and $B' = \begin{pmatrix} B \\ I_{n\times n} \end{pmatrix}$, where $I_{n\times n} \in \mathbb{Q}^{n\times n}$ is the identity matrix.

Firstly, observe that (1), i.e., the existence of some $y' \in \mathbb{Q}^{n'}$ satisfying $A'y > 0$ and $B'y \geq 0$, is equivalent to the first case of the lemma, i.e., the existence of some $y \in \mathbb{Q}^n_{\geq 0}$ satisfying $Ay > 0$ and $By \geq 0$. Secondly, observe that (2), i.e., the existence of some $0 \neq z'_1 \in \mathbb{Q}^{m'}_{\geq 0}, z'_2 \in \mathbb{Q}^{p'}_{\geq 0}$ satisfying $A'^\top z'_1 + B'^\top z'_2 = 0$, is equivalent to the second case of the lemma, i.e., the existence of some $0 \neq z_1 \in \mathbb{Q}^m_{\geq 0}, z_2 \in \mathbb{Q}^p_{\geq 0}$ satisfying $A^\top z_1 + B^\top z_2 \leq 0$. To see this, note that the last n coordinates of z'_2 can be independently chosen, and therefore set to satisfy $A'^\top z'_1 + B'^\top z'_2 = 0$ as long as $A^\top z_1 + B^\top z_2 \leq 0$. This shows that (2) is implied by the second case of the lemma, and the other direction holds trivially. □

LEMMA 2.9. *For a constraint language* Γ*, and a unary operation* $g \in \mathcal{O}^{(1)}_D$*, the following are equivalent:*

(*1*) Γ *admits a unary fractional polymorphism* ω *with* $g \in \text{supp}(\omega)$.
(*2*) *For all instances* I *of VCSP*(Γ) *and all optimal solutions* s *to* I*,* $g \circ s$ *is also an optimal solution to* I.

PROOF. The first condition of the lemma holds if and only if the following system of linear inequalities is satisfiable:

$$\begin{aligned} \sum_{h\in\mathcal{O}^{(1)}_D} \omega(h) f(h(\bar{x})) \leq \|\omega\|_1 f(\bar{x}) \quad & \forall f \in \Gamma, \bar{x} \in D^{ar(f)} \\ \omega(g) > 0 \quad & \\ \omega(h) \geq 0 \quad & \forall h \in \mathcal{O}^{(1)}_D. \end{aligned} \tag{3}$$

According to Lemma 2.8, this is true if, and only if, the following system is unsatisfiable:

$$\begin{aligned} \sum_{f\in\Gamma,\bar{x}\in D^{ar(f)}} z_2(f,\bar{x})(f(\bar{x}) - f(h(\bar{x}))) \leq 0, \quad & \forall h \in \mathcal{O}^{(1)}_D, \\ z_1 + \sum_{f\in\Gamma,\bar{x}\in D^{ar(f)}} z_2(f,\bar{x})(f(\bar{x}) - f(g(\bar{x}))) \leq 0, \quad & \\ z_1 > 0, \quad & \\ z_2(f,\bar{x}) \geq 0, \quad & \forall f \in \Gamma, \bar{x} \in D^{ar(f)}. \end{aligned} \tag{4}$$

Let $V_D = \{v_a \mid a \in D\}$ and define $\iota : V_D \to D$ by $\iota(v_a) = a$. Then, (4) is unsatisfiable if, and only if, there is no instance J of VCSP(Γ), with variables $V(J) = V_D$ and objective function $f_J = \sum_{f,\bar{x}} z_2(f,\bar{x}) f(\iota^{-1}(\bar{x}))$ such that $g \circ \iota$ is a non-optimal solution.

It is clear that the second condition of the lemma implies that (4) is unsatisfiable. It remains to show the reverse implication. Let I be any instance of VCSP(Γ) and

$s : V(I) \to D$ any optimal solution to I. Construct an instance J of VCSP(Γ) with variables $V(J) = V_D$ by replacing each term $w_i \cdot f_i(\bar{x}^i)$ in f_I by the term $w_i \cdot f_i(\iota^{-1} \circ s(\bar{x}^i))$ in f_J. Since (4) is unsatisfiable, it follows that $g \circ \iota$ is an optimal solution to J, and hence that $g \circ s$ is an optimal solution to I. As I and s were chosen arbitrarily, this establishes the lemma. □

In [Huber et al. 2014], a constraint language Γ is defined to be a core if, for each $a \in D$, there is an instance I_a of VCSP(Γ) such that a appears in every optimal solution to I_a. We now show that this condition is equivalent to Definition 2.6.

LEMMA 2.10. *For a constraint language Γ, the following are equivalent:*

(*1*) *All unary fractional polymorphisms of Γ are injective.*
(*2*) *For each $a \in D$, there is an instance I_a of VCSP(Γ) such that a appears in every optimal solution to I_a.*

PROOF. First we show the implication $(2) \Rightarrow (1)$. Assume that (1) does not hold and let ω be a unary fractional polymorphism of Γ with a non-injective $g \in \text{supp}(\omega)$; that is, there is an $a \in D$ such that $a \notin g(D)$. Then, Lemma 2.9 implies that every instance of VCSP(Γ) has a solution where a does not appear, so (2) does not hold.

We now show $(1) \Rightarrow (2)$. By Lemma 2.9, condition (1) holds if, and only if, for every non-injective unary operation $g \in \mathcal{O}_D^{(1)}$, there exists an instance I_g of VCSP(Γ) and an optimal solution s_g to I_g such that $g \circ s_g$ is not an optimal solution to I_g. Let $f_{I_g} = \sum_i w_i \cdot f_i(\bar{x}^i)$ be the objective function of I_g, and, as in the proof of Lemma 2.9, construct an instance J_g with variables $V_D = \{v_a \mid a \in D\}$ and objective function $f_{J_g} = \sum_i w_i \cdot f_i(\iota^{-1}(\bar{x}^i))$, where $\iota : V_D \to D$ given by $\iota(v_a) = a$. Then, ι is an optimal solution to J_g, but $g \circ \iota$ is not. Let I be the instance with variables V_D and $f_I = \sum_g f_{J_g}$, where the sum is over all non-injective unary operations. Let s be an optimal solution to I. Note that s must also be an optimal solution to each instance J_g. Since $s \circ \iota^{-1}$ is a unary operation on D, it follows that s must be injective, hence for every $a \in D$, there is a $v \in V_D$ such that $s(v) = a$. We can therefore let $I_a := I$ for each $a \in D$. □

For a constraint language Γ, let Γ_c denote the set of all functions obtained from functions in Γ by fixing a (possibly empty) subset of the variables to domain values. We will use the following result, which says that we can restrict our attention to core constraint languages whose expressive powers contain certain unary functions.

PROPOSITION 2.11 ([HUBER ET AL. 2014]). *Let Γ be a core constraint language defined on a finite domain D.*

(*1*) *For each $a \in D$, $\langle \Gamma_c \rangle$ contains a unary function u_a such that* $\operatorname{argmin} u_a = a$.
(*2*) *Γ is NP-hard if, and only if, Γ_c is NP-hard.*

It follows readily from Proposition 2.11 that every (generalised) fractional polymorphism of Γ_c for a core constraint language Γ is idempotent.

3. COMPLEXITY CLASSIFICATION

The computational complexity of constraint languages has attracted a lot of attention in the literature. The partial classifications obtained before the results of this paper can be summarised as follows:

— $\{0, 1\}$-valued constraint languages on $|D| = 2$ [Creignou 1995; Creignou et al. 2001].
— $\{0, 1\}$-valued constraint languages on $|D| = 3$ [Jonsson et al. 2006].
— $\{0, 1\}$-valued constraint languages on $|D| = 4$ [Jonsson et al. 2011].

— $\{0,1\}$-valued constraint languages containing all $\{0,1\}$-valued unary functions [Deineko et al. 2008].
— constraint languages on $|D| = 2$ [Cohen et al. 2006a].
— constraint languages on $|D| = 3$ [Huber et al. 2014].
— constraint languages containing $\{0,1\}$-valued unary functions [Kolmogorov and Živný 2013].
— constraint languages containing unary functions and certain special binary functions [Hirai 2015].

In all of these classifications, the hardness reductions essentially came from the condition (MC) and tractable cases were characterised by certain specific binary symmetric fractional polymorphisms including the concepts of submodularity [Jonsson et al. 2006; Deineko et al. 2008; Cohen et al. 2006a], skew bisubmodularity [Huber et al. 2014], 1-defect [Jonsson et al. 2011], and others [Hirai 2015].

3.1. The basic linear programming relaxation

Every VCSP instance has a natural linear programming relaxation, proposed independently by a number of authors [Shlezinger 1976; Koster et al. 1998; Chekuri et al. 2004; Wainwright et al. 2005; Kingsford et al. 2005; Cooper 2008; Cooper et al. 2010; Kun et al. 2012]. This relaxation is referred to as the *basic LP relaxation* (BLP) as it is the first level in the Sherali-Adams hierarchy [Sherali and Adams 1990]. It can be defined as follows.

Let Γ be a constraint language defined on D and let I be a VCSP(Γ) instance given by the set $V = \{x_1, \dots, x_n\}$ of variables and the objective function $f_I(x_1, \dots, x_n) = \sum_{i=1}^{q} w_i \cdot f_i(\bar{x}^i)$ where, for every $1 \leq i \leq q$, $f_i : D^{ar(f_i)} \to \mathbb{Q}$, $\bar{x}^i \in V^{ar(f_i)}$, and $w_i \in \mathbb{Q}_{\geq 0}$ is a weight. For a tuple $\bar{x}$, let $\{\bar{x}\}$ denote the set of elements in $\bar{x}$. The BLP has variables λ_{i,σ_i}, for $1 \leq i \leq q$ and $\sigma_i : \{\bar{x}^i\} \to D$; and variables $\mu_{x,a}$, for $x \in V$ and $a \in D$.

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{q} w_i \sum_{\sigma_i : \{\bar{x}^i\} \to D} f_i(\sigma_i(\bar{x}^i)) \cdot \lambda_{i,\sigma_i} \\
\text{s.t.} \quad & \sum_{\substack{\sigma_i : \{\bar{x}^i\} \to D \\ \sigma_i(x) = a}} \lambda_{i,\sigma_i} = \mu_{x,a} \quad \forall 1 \leq i \leq q, \forall x \in \{\bar{x}^i\}, \forall a \in D \\
& \sum_{a \in D} \mu_{x,a} = 1 \qquad \forall x \in V \\
& 0 \leq \lambda, \mu \leq 1
\end{aligned}
$$

Since Γ is fixed, this relaxation has polynomial size in I. Requiring λ_{i,σ_i} and $\mu_{x,a}$ to be in $\{0,1\}$ provides an integer programming formulation of I with the meaning $\mu_{x,a} = 1$ if, and only if, variable x is assigned value a.

For any VCSP instance I, the BLP gives a lower bound on the measure of an optimal solution to I. Denote this lower bound by BLP(I). We will say that the BLP *solves* VCSP(Γ) if BLP(I) $= \mathrm{Opt}_\Gamma(I)$ for every $I \in$ VCSP(Γ). It can be shown that when the BLP solves VCSP(Γ), then a solution attaining the optimum can also be obtained in polynomial time [Kolmogorov et al. 2015].

A result of the authors characterised the constraint languages for which the BLP relaxation solves VCSP(Γ) in terms of symmetric fractional polymorphisms [Thapper and Živný 2012]. An equivalent simplified condition was subsequently given in [Kolmogorov 2013], see also [Kolmogorov et al. 2015].

THEOREM 3.1 ([THAPPER AND ŽIVNÝ 2012; KOLMOGOROV 2013]). *Let* Γ *be a constraint language. Then BLP solves VCSP*(Γ) *if, and only if,* Γ *admits a binary symmetric fractional polymorphism.*

3.2. Main classification

The main technical contribution of this paper is the following result.

THEOREM 3.2. *Let* D *be an arbitrary finite set and let* Γ *be a constraint language defined on* D*. If* Γ *is a core such that* Γ_c *does not satisfy (MC), then* Γ *admits a binary idempotent and symmetric fractional polymorphism.*

We will also need the following lemma which is proved in Section 5.2.

LEMMA 3.3. *Let* Γ *be a constraint language defined on* D *and let* Γ' *be a core of* Γ*. If* Γ' *admits a binary symmetric fractional polymorphism, then so does* Γ*.*

Theorem 3.2 implies our main result, Theorem 3.4, which shows that having a binary symmetric fractional polymorphism is the *only* reason for tractability, and conversely, that the condition (MC) is the only reason for intractability. This provides a complexity classification of *all* constraint languages defined on *arbitrary* finite domains, thus generalising all previous classifications mentioned above.

THEOREM 3.4 (MAIN). *Let* D *be an arbitrary finite set, let* Γ *be a constraint language defined on* D*, and let* Γ' *be a core of* Γ*.*

— *Either* Γ *has a binary symmetric fractional polymorphism and BLP solves VCSP*(Γ)*;*
— *or (MC) holds for* Γ'_c *and VCSP*(Γ) *is NP-hard.*

PROOF. If Γ'_c satisfies (MC), then VCSP(Γ'_c) is NP-hard by Lemma 2.5. In this case VCSP(Γ) is NP-hard by Proposition 2.11(2) and Lemma 2.7. Otherwise, by Theorem 3.2, Γ'_c and hence Γ' admit a binary symmetric fractional polymorphism. By Lemma 3.3, Γ admits a binary symmetric fractional polymorphism and it follows from Theorem 3.1 that BLP solves VCSP(Γ). $\square$

Theorem 1.1 follows immediately from Theorem 3.4. We remark that the dichotomy classification holds in the special case of $\{0,1\}$-valued constraint languages, that is, for (weighted) *maximum constraint satisfaction problems* (Max-CSPs) [Creignou et al. 2001].[3]

The problem of deciding whether a constraint language Γ is a core and that of deciding whether the tractability condition of Γ is met are discussed in Section 4.

We discuss constraint languages of *infinite* size in Appendix A.

COROLLARY 3.5 (OF THEOREM 3.2). *Let* D *be an arbitrary finite set and let* Γ *be a core constraint language defined on* D*. The following are equivalent:*

(*1*) Γ_c *does not satisfy (MC);*
(*2*) Γ *admits an idempotent and cyclic fractional polymorphism of some arity* $k > 1$*;*
(*3*) Γ *admits an idempotent and symmetric fractional polymorphism of some arity* $k > 1$*;*
(*4*) Γ *admits a binary idempotent and symmetric fractional polymorphism;*
(*5*) *BLP solves VCSP*(Γ)*.*

PROOF. Theorem 3.1 gives $(4) \Leftrightarrow (5)$. The implications $(4) \Rightarrow (3) \Rightarrow (2)$ are trivial. Theorem 3.2 gives the implication $(1) \Rightarrow (4)$. Finally, we will show that $(2) \Rightarrow (1)$. Let

[3]We consider Max-CSPs as Min-CSPs to fit in the VCSP framework; that is, rather than maximising the (weighted) sum of satisfied constraints the goal is to minimise the (weighted) sum of unsatisfied constraints. Note that this kind of construction does not necessarily preserve approximability properties.

ω be a k-ary cyclic fractional polymorphism of Γ. Suppose that Γ_c satisfies (MC). By Lemma 2.4, Γ_c satisfies (MC′); that is, there are distinct $a, b \in D$, a unary cost function $u \in \langle\Gamma_c\rangle$ with $\operatorname{argmin} u = \{a, b\}$, and a binary cost function $f \in \langle\Gamma_c\rangle$ with $f(a, b) = f(b, a) < f(a, a) = f(b, b)$. Consider the tuples $\bar{a}^1 = (a, b)$, $\bar{a}^2 = (b, a)$, and $\bar{a}^i = (a, a)$ for $3 \leq i \leq k$. Note that for every (cyclic) operation $g \in \operatorname{supp}(\omega)$ we have $g(\bar{a}^1, \ldots, \bar{a}^k) = (x_g, x_g)$ for some $x_g \in D$. Using the fact that ω is a fractional polymorphism of u, we first show that $x_g \in \{a, b\}$. Observe that $\sum_g \omega(g)u(g(a_1^1, \ldots, a_1^k)) \leq u^k(a_1^1, \ldots, a_1^k) = \frac{k-1}{k}u(a) + \frac{1}{k}u(b) = u(a) = u(b)$, where the inequality follows from (1). Hence, we must have $x_g = g(a_1^1, \ldots, a_1^k) \in \{a, b\}$ for all $g \in \operatorname{supp}(\omega)$. Furthermore, $\sum_g \omega(g)f(g(\bar{a}^1, \ldots, \bar{a}^k)) = f(a, a) = f(b, b)$, but $f^k(\bar{a}^1, \ldots, \bar{a}^k) = \frac{2}{k}f(a, b) + \frac{k-2}{k}f(a, a) < f(a, a)$. Thus, inequality (1) does not hold for f and ω. Consequently, ω is not a fractional polymorphism of f, which is a contradiction. □

Corollary 3.5 answers Problem 1 from [Huber et al. 2013] that asked about the relationship between the complexity of a constraint language Γ and the existence of various types of fractional polymorphisms of Γ. Note that Corollary 3.5 holds unconditionally. Problem 1 from [Huber et al. 2013] also involved the solvability by the basic SDP relaxation [Raghavendra 2008], which at the time was known to be implied by (2) and imply (1), provided that P $\neq$ NP. Under the same assumption, we conclude that solvability by the basic SDP relaxation is also characterised by any of the equivalent statements of Corollary 3.5.

4. META PROBLEMS

Let Γ be a constraint language defined on D. In this section, we study three *meta problems* relevant to our classification. The first problem is *core recognition*: Given a Γ, is Γ a core? The second problem is *core identification*: Given Γ and Γ', is Γ' a core of Γ? The third problem is *tractability recognition*: Given Γ, is Γ tractable?

We show that all three problems are decidable. The first two problems are co-NP-complete and DP-complete, respectively. On the other hand, if Γ is assumed to be a core, then the tractability of Γ can be decided in polynomial time.

LEMMA 4.1. *Given Γ and $g \in \mathcal{O}_D^{(1)}$, the problem of deciding whether Γ has a unary fractional polymorphism ω with $g \in \operatorname{supp}(\omega)$ is in NP.*

PROOF. By Lemma 2.9, (Γ, g) is a yes-instance if, and only if, the system of linear inequalities in (3) is satisfiable. Since the number of inequalities is polynomial in the size of Γ, this system is satisfiable if, and only if, it has a solution with a polynomial number of non-zero variables. The NP certificate consists of a polynomially large subset of the variables. Writing down the restriction of (3) to this subset and verifying the satisfiability of the resulting system can then be done in polynomial time. □

To every $\{0, 1\}$-valued cost function f on domain D corresponds a relation R defined by $\bar{x} \in R$ if, and only if, $f(\bar{x}) = 0$. A unary operation $g : D \to D$ is said to be an *endomorphism* of R if $\bar{x} \in R$ implies $g(\bar{x}) \in R$.

LEMMA 4.2. *Let f be a $\{0, 1\}$-valued cost function and let R be the corresponding relation. The constraint language $\{f\}$ has a unary fractional polymorphism with support Ψ if, and only if, Ψ is a set of endomorphisms of R.*

PROOF. Let Ψ be a set of endomorphisms of R and let $g \in \Psi$, i.e., $\bar{x} \in R$ implies $g(\bar{x}) \in R$, for all $\bar{x} \in D^{ar(f)}$. Then, $f(\bar{x}) \geq f(g(\bar{x}))$, so $\chi_{\{g\}}$ is a unary fractional polymorphism of $\{f\}$. It follows that $|\Psi|^{-1}\chi_\Psi$ is also a unary fractional polymorphism of $\{f\}$.

For the opposite direction, let ω be a unary fractional polymorphism of $\{f\}$. Then,

$$f(\bar{x}) \geq \sum_{g \in \text{supp}(\omega)} \omega(g) f(g(\bar{x})),$$

for each $\bar{x} \in D^{ar(f)}$. Fix an operation $g \in \text{supp}(\omega)$. If $\bar{x} \in R$, then $f(\bar{x}) = 0$ and so clearly $f(g(\bar{x})) = 0$, i.e., $g(\bar{x}) \in R$. It follows that g is an endomorphism of R. Since $g \in \text{supp}(\omega)$ was chosen arbitrarily, the result follows. □

PROPOSITION 4.3. *Testing whether a given constraint language* Γ *is a core is co-NP-complete.*

PROOF. We show that testing whether a given constraint language Γ is *not* a core is NP-complete. Containment in NP follows from Lemma 4.1 by first guessing a non-injective unary operation g.

A graph G is a core if all endomorphisms of its edge relation are injective [Hell and Nešetřil 2004]. It has been shown in [Hell and Nešetřil 1992] that the problem of checking whether a given graph G is not a core is NP-hard, i.e., it is NP-hard to determine whether G has a non-injective endomorphism. By Lemma 4.2, this is the case if, and only if, the cost function f corresponding to the adjacency relation of G has a unary fractional polymorphism with a non-injective operation in its support, i.e., if, and only if, $\{f\}$ is not a core. Therefore, the problem of determining whether Γ is not a core is NP-hard, even if Γ is only allowed to contain a single binary and symmetric $\{0, 1\}$-valued cost function. □

The complexity class DP consists of all decision problems that can be written as the intersection of an NP-problem and a co-NP-problem; equivalently, DP consists of all decision problems that can be written as the difference of two NP-problems [Papadimitriou and Yannakakis 1984]. Next we show that the core identification problem is DP-complete.

PROPOSITION 4.4. *Given two constraint languages* Γ *and* Γ'*, testing whether* Γ' *is a core of* Γ *is DP-complete.*

PROOF. The problem can be described as the intersection between the problem of verifying that Γ' is a core, which is in co-NP by Proposition 4.3, and the problem of verifying that $\Gamma' = \Gamma[g(D)]$ for some g contained in the support of a unary fractional polymorphism of Γ. The latter problem is seen to be in NP by first guessing the operation g, and then using Lemma 4.1. Containment in DP follows.

To show DP-hardness, we will reduce from the following problem: Given two graphs, G and G', with G' a subgraph of G, test whether G' is a core (all endomorphisms of G' are injective) and whether there is a homomorphism from G to G'. This problem has been shown to be DP-hard [Fagin et al. 2005], thus improving a previously known NP-hardness result on the same problem [Chandra and Merlin 1977]. We may in fact assume that G' is an induced subgraph of G since otherwise, it is easy to see that G' cannot be a core of G. Let f and f' be the cost functions corresponding to the adjacency relations of G and G' respectively. Let $\Gamma = \{f\}$ and $\Gamma' = \{f'\}$. By Lemma 4.2, G' is a core if, and only if, every unary fractional polymorphism of Γ' has only injective operations in its support. By Definition 2.6, this is the case if, and only if, Γ' is a core. There is a homomorphism from G to G' if, and only if (since G' is a subgraph of G), there is an endomorphism $g : G \to G$ so that $g(V(G)) = V(G')$. By Lemma 4.2, this is the case if, and only if, there is a unary fractional polymorphism ω of Γ with $g \in \text{supp}(\omega)$ so that $\Gamma' = \Gamma[g(D)]$. Hence, G' is a core of G if, and only if, Γ' is a core of Γ. It follows that the latter problem is DP-hard, even for the specific case when both Γ and Γ' contains a single binary and symmetric $\{0, 1\}$-valued cost function. □

Now we turn our attention to the problem of tractability recognition. Let $X = \{(f, \bar{x}, \bar{y}) \mid f \in \Gamma, \bar{x}, \bar{y} \in D^{ar(f)}\}$. To test whether a finite constraint language Γ is tractable, it suffices, by Theorem 3.2, to test whether it has a binary symmetric fractional polymorphism. This is the case if, and only if, the following system of linear inequalities is satisfiable:

$$\begin{aligned} \sum_{g \in \Omega} \omega(g) f(g(\bar{x}, \bar{y})) \leq f^2(\bar{x}, \bar{y}), \quad & \forall (f, \bar{x}, \bar{y}) \in X, \\ \|\omega\|_1 = 1, & \\ \omega(g) \geq 0, \quad \forall g \in \Omega, & \end{aligned} \tag{5}$$

where Ω is the set of binary operations $g \in \mathcal{O}_D^{(2)}$ on D that are symmetric. It follows that the tractability recognition problem is decidable for any finite Γ. Since the number of variables in the system (5) is exponential in $|D|$, this does not lead to a polynomial-time algorithm. However, when Γ is a core, it turns out that we can solve the system in polynomial time. This reflects a well-known phenomenon for the CSP decision problem, where the problem of deciding whether a constraint language admits various types of polymorphisms is known to have a polynomial-time algorithm only when the language is a core.

For a core Γ, we can restrict Ω to the set of binary operations on D that are symmetric *and idempotent*. The linear programming dual of minimising the objective function 0 subject to (5) (i.e., of determining whether this system is satisfiable) is the problem of maximising δ subject to the following system of inequalities:

$$\begin{aligned} \sum_{f, \bar{x}, \bar{y}} z(f, \bar{x}, \bar{y}) \left(f^2(\bar{x}, \bar{y}) - f(g(\bar{x}, \bar{y})) \right) + \delta \leq 0, \quad & \forall g \in \Omega, \\ z(f, \bar{x}, \bar{y}) \geq 0, \quad & \forall (f, \bar{x}, \bar{y}) \in X. \end{aligned} \tag{6}$$

The solution to (6) that assigns 0 to all variables is always feasible, so the dual optimum is always at least 0. If the dual optimum is 0, then the primal optimum is also 0, so (5) is satisfiable. Otherwise, (6) has a solution of measure greater than 0, so it has solutions of unbounded measure. In this case, (5) is unsatisfiable. The system (6) has a polynomial number of variables, but an exponential number of inequalities.

Assuming that Γ is a core constraint language, we can solve (6) in polynomial time using the *ellipsoid method*. In fact, we can do even better. We can find a dual solution with support on a polynomial number of variables. This means that we can find a binary idempotent and symmetric fractional polymorphism represented by its values on a support of size linear in the size of X and thus in the size of Γ. For a thorough treatment of the ellipsoid algorithm, including Lemma 4.6, we refer to [Grötschel et al. 1988].

Definition 4.5. A *strong separation oracle* for a polyhedron P is given an input $\bar{p} \in \mathbb{Q}^n$ and either returns "$\bar{p} \in P$", or a vector $\bar{a} \in \mathbb{Q}^n$ such that $\bar{a}^\top \bar{x} < \bar{a}^\top \bar{p}$ for all $\bar{x} \in P$.

LEMMA 4.6 (LEMMA 6.5.15 IN [GRÖTSCHEL ET AL. 1988]). *Let $\bar{c} \in \mathbb{Q}^n$ and let $P \subseteq \mathbb{Q}^n$ be a polyhedron defined by $A\bar{x} \leq \bar{b}$, where the encoding sizes of the coefficients of A and $\bar{b}$ are bounded by ϕ. Given a strong separation oracle SEP for P where every output has encoding size at most ϕ, we can, in time polynomial in n, ϕ, and the encoding size of $\bar{c}$, and using a polynomial number of oracle queries to SEP, either*

— *find a basic optimum dual solution with oracle inequalities, or*
— *assert that the dual problem is unbounded or has no solution.*

In Lemma 4.6, a *basic optimum dual solution with oracle inequalities* means a set of inequalities $(\bar{a}^1)^\top \bar{x} \leq \alpha_1, \ldots, (\bar{a}^k)^\top \bar{x} \leq \alpha_k$, valid for P, where $\bar{a}^1, \ldots, \bar{a}^k$ are linearly independent outputs of SEP, and dual variables $\lambda_1, \ldots, \lambda_k \in \mathbb{Q}_{\geq 0}$ such that $\lambda_1 \bar{a}^1 + \cdots + \lambda_k \bar{a}^k = \bar{c}$ and $\lambda_1 \alpha_1 + \cdots + \lambda_k \alpha_k = \max_{\bar{x} \in P} \bar{c}^\top \bar{x}$.

PROPOSITION 4.7. *There is a polynomial-time algorithm that, given a core constraint language* Γ*, either*

— *finds a binary idempotent and symmetric fractional polymorphism* ω *of* Γ*, represented by a subset* $\Omega' \subseteq \Omega$ *with* $\operatorname{supp}(\omega) \subseteq \Omega'$ *together with the restriction of* ω *to* Ω'*, or*
— *asserts that none exists.*

PROOF. Let P denote the polyhedron defined by (6). We will give a polynomial-time algorithm that, given a point $(z, \delta) \in \mathbb{Q}^X \times \mathbb{Q}$ as input, does one of three things:

— answers "unbounded optimum";
— answers "$(z, \delta) \in P$"; or
— returns $\bar{a} \in \mathbb{Q}^X \times \mathbb{Q}$ such that $\bar{a}^\top (x, \delta') < \bar{a}^\top (z, \delta)$ for all $(x, \delta') \in P$.

The algorithm can be seen as a strong separation oracle with an escape clause. We can use it as a strong separation oracle for the polyhedron P, as long as the answer is not "unbounded optimum".

Let $\bar{c}$ be the vector with components $c_{(f,\bar{x},\bar{y})} = 0$ for $(f, \bar{x}, \bar{y}) \in X$ and $c_\delta = 1$. By Lemma 4.6, we can either find a dual solution to (6) given by inequalities returned by the oracle, or we can assert that the dual, (5), has no solution. If the ellipsoid algorithm asserts that the dual has no solution, or if the answer from the separation oracle is ever "unbounded optimum", then we can conclude that (5) is unsatisfiable. Otherwise, an optimum dual solution is described using valid inequalities of the following form:

$$\sum_{(f,\bar{x},\bar{y}) \in X} z(f, \bar{x}, \bar{y})(f^2(\bar{x}, \bar{y}) - f(g(\bar{x}, \bar{y}))) + \delta \leq \alpha_g, \qquad \forall g \in \Omega',$$

$$-z(f, \bar{x}, \bar{y}) \leq \alpha_{(f,\bar{x},\bar{y})}, \qquad \forall (f, \bar{x}, \bar{y}) \in \Upsilon,$$

for some constants $\alpha_g, \alpha_{(f,\bar{x},\bar{y})} \in \mathbb{Q}$ and subsets $\Omega' \subseteq \Omega$ and $\Upsilon \subseteq X$.

The corresponding dual variables are $\omega' : \Omega' \to \mathbb{Q}_{\geq 0}$ and $\upsilon : \Upsilon \to \mathbb{Q}_{\geq 0}$, and they satisfy the following equalities:

$$\sum_{g \in \Omega'} \omega'(g)(f^2(\bar{x}, \bar{y}) - f(g(\bar{x}, \bar{y}))) - \upsilon(f, \bar{x}, \bar{y}) = 0, \qquad \forall (f, \bar{x}, \bar{y}) \in X, \tag{7}$$

$$\sum_{g \in \Omega'} \omega'(g) = 1, \tag{8}$$

where we define $\upsilon(f, \bar{x}, \bar{y}) = 0$ for $(f, \bar{x}, \bar{y}) \in X \setminus \Upsilon$. The dual variables are non-negative, so (7) and (8) imply $f^2(\bar{x}, \bar{y}) \geq \sum_{g \in \Omega'} \omega'(g) f(g(\bar{x}, \bar{y}))$, for all $(f, \bar{x}, \bar{y}) \in X$. Since the inequalities correspond to vectors that are linearly independent, the size of Ω' is bounded by the number of variables of (6), i.e., polynomial in the input size. Clearly, ω' can be extended to a fractional polymorphism of Γ by assigning weight 0 to every operation outside of Ω'.

The separation oracle is given by Algorithm 1. It is based on the observation that in order to verify whether (z, δ) belongs to P, it suffices to find an operation $g \in \Omega$ that minimises $\sum_{f,\bar{x},\bar{y}} z(f, \bar{x}, \bar{y}) f(g(\bar{x}, \bar{y}))$. If (z, δ) satisfies the inequality with respect to this g, then (z, δ) satisfies all inequalities. Otherwise, the vector $\bar{a}$ given by $a_{(f,\bar{x},\bar{y})} = f^2(\bar{x}, \bar{y}) - f(g(\bar{x}, \bar{y}))$ and $a_\delta = 1$ defines a separating hyperplane.

Algorithm 1: Separate(z, δ)

Input: $(z, \delta) \in \mathbb{Q}^X \times \mathbb{Q}$
Output: "unbounded optimum", "$(z, \delta) \in P$", or a separating hyperplane

if $z(f, \bar{x}, \bar{y}) < 0$ *for some* $(f, \bar{x}, \bar{y}) \in X$ **then**
  Let $a_{(f,\bar{x},\bar{y})} := -1$ and set all other components of $\bar{a}$ to 0
  return $\bar{a}$
end

Let $V := \{[x, y] \mid x, y \in D\}$ /* *Construct the VCSP instance I* */
Let $f_I(V) := \sum_{(f,\bar{v})\in X'} z'(f, \bar{v}) f(\bar{v})$
/* *Self reduce using the BLP relaxation* */
Let $g' \colon V \to D \cup \{\bot\}$ be given by $g'(v) = \bot$ for all v
while $\exists v \in V \colon g'(v) = \bot$ **do**
  if $\exists d \in D \colon BLP(I[g' \cup \{v \mapsto d\}]) = BLP(I)$ **then**
    $g' := g' \cup \{v \mapsto d\}$
  else
    return "unbounded optimum"
  end
end
/* *Test whether $(z, \delta) \in P$* */
Let $g \in \Omega$ be the operation $(x, y) \mapsto g'([x, y])$
if $\sum_{(f,\bar{x},\bar{y})\in X} z(f, \bar{x}, \bar{y})(f^2(\bar{x}, \bar{y}) - f(g(\bar{x}, \bar{y}))) + \delta \leq 0$ **then**
  return "$(z, \delta) \in P$"
else
  Let $a_{(f,\bar{x},\bar{y})} := f^2(\bar{x}, \bar{y}) - f(g(\bar{x}, \bar{y}))$, for all $(f, \bar{x}, \bar{y}) \in X$, and $a_\delta := 1$
  return $\bar{a}$
end

Let $[x, y]$ denote the multiset of the elements x and y, and let $V = \{[x, y] \mid x, y \in D\}$. Let $X' = \{(f, \bar{v}) \mid f \in \Gamma, \bar{v} \in V^{ar(f)}\}$. For $(f, \bar{v}) \in X'$, define

$$z'(f, \bar{v}) = \sum_{\substack{\bar{x},\bar{y} \text{ s.t.} \\ v_i=[x_i,y_i]}} z(f, \bar{x}, \bar{y}).$$

The algorithm starts by creating an instance I of VCSP(Γ) over the variables V with $f_I(V) = \sum_{(f,\bar{v})\in X'} z'(f, \bar{v}) f(\bar{v})$. For an operation $g \in \Omega$, define the function $g' : V \to D$ by $[x, y] \mapsto g(x, y)$. Note that this defines a bijection between Ω and the set of all functions from V to D.

For every $g \in \Omega$, we have

$$\sum_{(f,\bar{x},\bar{y})\in X} z(f, \bar{x}, \bar{y}) f(g(\bar{x}, \bar{y})) = \sum_{(f,\bar{v})\in X'} \sum_{\substack{\bar{x},\bar{y} \text{ s.t.} \\ v_i=[x_i,y_i]}} z(f, \bar{x}, \bar{y}) f(g(\bar{x}, \bar{y})) = f_I(g'(V)). \quad (9)$$

Instead of optimising the left-hand side of (9) over all $g \in \Omega$, we can optimise $f_I(g'(V))$ over all $g' : V \to D$, i.e., we can try to solve the VCSP(Γ) instance I. Note that, since $\Gamma \subseteq \Gamma_c$ (Section 2.4), I can also be seen as an instance of VCSP(Γ_c). For a (partial) assignment $g' : V \to D \cup \{\bot\}$, we let $I[g']$ denote the VCSP(Γ_c)-instance obtained by adding the constant unary relations $v = g'(v)$ for $v \in V$ such that $g'(v) \neq \bot$.

On lines 1–4, the algorithm checks that all components of z are non-negative. Otherwise, a simple separating hyperplane is returned.

On lines 6–7, the algorithm constructs the instance I.

On lines 9–16, it then tries to solve this instance using the BLP relaxation and self-reduction. This is accomplished by fixing the variables one by one to a value that maintains the BLP optimum (lines 10–12). If this succeeds for all variables, then by (9) and the initial observation, we can determine whether the point is contained in P by verifying a single inequality (line 19).

Otherwise, the instance $I[g']$ of VCSP(Γ_c) has an optimum that is strictly greater than the BLP optimum. By Theorem 3.1, it follows that Γ_c does not have a binary symmetric fractional polymorphism. Since Γ is a core, the same must then be true for Γ. In this case (6) has a non-zero solution, and therefore an unbounded optimum, so the algorithm gives the correct answer on line 14.

Finally, we argue that Algorithm 1 runs in polynomial time. The BLP relaxation of I has size that is polynomial in the size of z and Γ, so the call to BLP($I[g' \cup \{v \mapsto d\}]$) takes polynomial time. The number of calls to BLP is at most $|V| \cdot |D| = \mathcal{O}(|D|^3)$, again polynomial in the size of Γ. □

5. PROOF OF THEOREM 3.2

In this section, we prove Theorem 3.2, which we restate here for the reader's convenience:

THEOREM 3.2. *Let D be an arbitrary finite set and let Γ be a constraint language defined on D. If Γ is a core such that Γ_c does not satisfy (MC), then Γ admits a binary idempotent and symmetric fractional polymorphism.*

5.1. Proof overview

We will need to introduce several important concepts and establish some auxiliary results. First, using Lemma 2.8, we prove, in Section 5.3, the following:

LEMMA 5.1. *Let Δ be an arbitrary constraint language defined on a finite set $D(\Delta)$. If Δ does not satisfy (MC) then Δ has a binary fractional polymorphism ω such that for each $\{a, b\} \subseteq D(\Delta)$, there exists $g \in \operatorname{supp}(\omega)$ with $\{g(a, b), g(b, a)\} \neq \{a, b\}$.*

Let $\mathbf{1}$ be the identity mapping in $\mathcal{O}_D^{(m\to m)}$. For a fractional mapping σ of arity $m \to m$, let

$$\mathcal{V}(\sigma) = \{\mathbf{g}_k \circ \cdots \circ \mathbf{g}_1 \circ \mathbf{1} \mid \mathbf{g}_i \in \operatorname{supp}(\sigma), k \geq 0\}.$$

Let $G = G(\sigma) = (V(G), E(G))$ be the directed graph with

— $V(G) = \mathcal{V}(\sigma)$;
— $E(G) = \{(\mathbf{g}, \mathbf{h} \circ \mathbf{g}) \mid \mathbf{g} \in \mathcal{V}(\sigma), \mathbf{h} \in \operatorname{supp}(\sigma)\}$.

A vertex $\mathbf{g}$ in $V(G)$ is called *recurrent* if, for every other vertex $\mathbf{h} \in V(G)$, there is a path from $\mathbf{h}$ to $\mathbf{g}$ whenever there is a path from $\mathbf{g}$ to $\mathbf{h}$. Let $\mathcal{R}(\sigma)$ denote the set of maximal strongly connected components of recurrent vertices of $V(G)$. Note that $\mathcal{R}(\sigma)$ is a partition of the set of recurrent vertices.

If ρ is a generalised fractional polymorphism of a cost function f, then we say that ρ improves f. The set of all cost functions that are improved by ρ is denoted by $\operatorname{Imp}(\rho)$. The following result is proved in Section 5.5.

THEOREM 5.2. *Let σ be a fractional mapping of arity $m \to m$. There exists a probability distribution w on $\mathcal{R}(\sigma)$ with the following property: if ρ is any fractional mapping of arity $m \to m$ with $\sum_{\mathbf{g}\in C} \rho(\mathbf{g}) = w(C)$ for all $C \in \mathcal{R}(\sigma)$, then $\operatorname{Imp}(\sigma) \subseteq \operatorname{Imp}(\rho)$.*

As the first step in our proof of Theorem 3.2, we apply Lemma 5.1 to Γ_c. By assumption, Γ_c does not satisfy (MC), so we conclude that it has a fractional polymorphism $\hat{\omega}$ with the properties given in the lemma. Furthermore, by Proposition 2.11(1), we know that $\langle\Gamma_c\rangle$ contains a unary function u_a for each $a \in D$ such that $\operatorname{argmin} u_a = \{a\}$. This implies that $\hat{\omega}$ is idempotent. To finish the proof, we will massage $\hat{\omega}$ into a binary symmetric fractional polymorphism using Theorem 5.2.

For a binary operation $g \in \mathcal{O}_D^{(2)}$, define $\bar{g}$ by $\bar{g}(x, y) = g(y, x)$. We denote by $(g, \bar{g}) \in \mathcal{O}_D^{(2\to 2)}$ the mapping defined by $(g, \bar{g})(x, y) = (g(x, y), \bar{g}(x, y))$. Recall that $\chi_{(g,\bar{g})}$ denotes the fractional mapping that that takes the value 1 on the mapping $(g, \bar{g})$ and 0 on all other mappings. Let $\hat{\sigma} = \sum_g \hat{\omega}(g)\chi_{(g,\bar{g})}$. As the second step, we apply Theorem 5.2 to $\hat{\sigma}$. Note that $\Gamma_c \subseteq \operatorname{Imp}(\hat{\sigma})$ and that all $\mathbf{g} \in \mathcal{V}(\hat{\sigma})$ are of the form $\mathbf{g} = (g, \bar{g})$. Let w be the probability distribution in Theorem 5.2 when applied to $\hat{\sigma}$. Fix an arbitrary mapping $\mathbf{g}_C \in C$, for every $C \in \mathcal{R}(\hat{\sigma})$, and let $\hat{\rho} = \sum_C w(C)\chi_{\mathbf{g}_C}$.

A mapping $\mathbf{p} \in \mathcal{O}_D^{(m\to m)}$ is called *permuting* if it acts as a permutation on every tuple in D^m. The following lemma finishes the proof of Theorem 3.2:

LEMMA 5.3 (KEY LEMMA). *For every $f \in \operatorname{Imp}(\hat{\rho})$, $\bar{x}^1, \bar{x}^2 \in D^{ar(f)}$, $\mathbf{g} \in \operatorname{supp}(\hat{\rho})$, and permuting mapping $\mathbf{p} \in \mathcal{O}_D^{(2\to 2)}$, we have $f^2(\mathbf{g}(\bar{x}^1, \bar{x}^2)) = f^2(\mathbf{g} \circ \mathbf{p}(\bar{x}^1, \bar{x}^2))$.*

COROLLARY 5.4. *For every permuting mapping $\mathbf{p} \in \mathcal{O}_D^{(2\to 2)}$, we have $\operatorname{Imp}(\hat{\rho}) \subseteq \operatorname{Imp}(\hat{\rho} \circ \mathbf{p})$, where $\hat{\rho} \circ \mathbf{p} := \sum_{\mathbf{g}\in\operatorname{supp}(\hat{\rho})} \rho(\mathbf{g})\chi_{\mathbf{g}\circ\mathbf{p}}$.*

Let $\mathbf{p} \in \mathcal{O}_D^{(2\to 2)}$ be a mapping that orders its inputs according to some fixed total order on D. By Theorem 5.2 and Corollary 5.4, we have

$$\Gamma \subseteq \Gamma_c \subseteq \operatorname{Imp}(\hat{\sigma}) \subseteq \operatorname{Imp}(\hat{\rho}) \subseteq \operatorname{Imp}(\hat{\rho} \circ \mathbf{p}),$$

so Γ admits $\hat{\rho} \circ \mathbf{p}$. For every $a, b \in D$, $\mathbf{p}(a, b) = \mathbf{p}(b, a)$ so for every $\mathbf{g} \in \operatorname{supp}(\hat{\rho})$, we have $\mathbf{g} \circ \mathbf{p}(a, b) = \mathbf{g} \circ \mathbf{p}(b, a)$. It follows that $\hat{\rho} \circ \mathbf{p}$ is symmetric. Consequently,

$$\sum_{(g_1,g_2)\in\operatorname{supp}(\hat{\rho}\circ\mathbf{p})} \hat{\rho} \circ \mathbf{p}((g_1, g_2))\frac{1}{2}(\chi_{g_1} + \chi_{g_2})$$

is a binary idempotent and symmetric fractional polymorphism of Γ which proves Theorem 3.2.

It remains to prove Lemma 5.3. For this we need two additional results that are stated here and are proved in Sections 5.4 and 5.6.

Definition 5.5. Let $w_a = \sum_{\mathbf{g}:\mathbf{g}(a,b)=(a,a)} \hat{\rho}(\mathbf{g})$ and $w_b = \sum_{\mathbf{g}:\mathbf{g}(a,b)=(b,b)} \hat{\rho}(\mathbf{g})$. We say that $\hat{\rho}$ is *submodular on the pair* $\{a, b\} \subseteq D$ if $w_a = w_b = \frac{1}{2}$.

Let $S = (V(S), E(S))$ be the undirected graph with:

— $V(S) = D$;
— $E(S) = \{\{a, b\} \mid \hat{\rho}$ is submodular on $\{a, b\}\}$.

LEMMA 5.6. *The graph S is connected.*

LEMMA 5.7. *Assume that $\hat{\rho}$ is submodular on $\{a_1, a_2\}$. Let $f \in \operatorname{Imp}(\hat{\rho})$ and $(\bar{y}^1, \bar{y}^2) = \mathbf{g}(\bar{x}^1, \bar{x}^2)$ for some $\mathbf{g} \in \operatorname{supp}(\hat{\rho})$ and $\bar{x}^1, \bar{x}^2 \in D^{ar(f)-1}$. Then $f^2((a_1, \bar{y}^1), (a_2, \bar{y}^2)) = f^2((a_2, \bar{y}^1), (a_1, \bar{y}^2))$.*

PROOF (OF LEMMA 5.3). By construction, $(g, \bar{g})(y, x) = (\bar{g}, g)(x, y)$ for all $(g, \bar{g}) \in \mathcal{V}(\hat{\sigma})$. Therefore, it suffices to show that interchanging the two elements of $\mathbf{g}(\bar{x}^1, \bar{x}^2)$ at any subset of the coordinates does not alter the value of $f^2(\mathbf{g}(\bar{x}^1, \bar{x}^2))$. We show this for the case when only the elements of the first coordinate are interchanged: with $g(\bar{x}^1, \bar{x}^2) = ((a, \bar{y}^1), (b, \bar{y}^2))$, we show that $f^2((a, \bar{y}^1), (b, \bar{y}^2)) = f^2((b, \bar{y}^1), (a, \bar{y}^2))$. The full result follows by applying the same argument to each coordinate. By Lemma 5.6, there exists a path $a = a_0, a_1, \ldots, a_\ell = b$ from a to b in the graph S, and by Lemma 5.7, we have

$$f^2((a_i, \bar{y}^1), (a_{i+1}, \bar{y}^2)) = f^2((a_{i+1}, \bar{y}^1), (a_i, \bar{y}^2)), \tag{10}$$

for all $0 \leq i < \ell$. Summing (10) over $0 \leq i < \ell$, we obtain

$$\sum_{0 \leq i < \ell} f^2((a_i, \bar{y}^1), (a_{i+1}, \bar{y}^2)) = \sum_{0 \leq i < \ell} f^2((a_{i+1}, \bar{y}^1), (a_i, \bar{y}^2)). \tag{11}$$

Finally, by cancelling terms in (11),

$$\frac{1}{2} f((a_0, \bar{y}^1)) + \frac{1}{2} f((a_\ell, \bar{y}^2)) = \frac{1}{2} f((a_\ell, \bar{y}^1)) + \frac{1}{2} f((a_0, \bar{y}^2)),$$

which establishes the result. □

5.2. Proof of Lemma 3.3

Here, we use Theorem 5.2 to prove Lemma 3.3.

PROOF (OF LEMMA 3.3). Let ω' be a binary symmetric fractional polymorphism of Γ'. Let $D' \subseteq D$ be the domain of the core Γ', and let μ be a unary fractional polymorphism of Γ with $g \in \text{supp}(\mu)$ such that $\Gamma' = \Gamma[g(D)]$ and thus $D' = g(D)$. Consider the graph $G(\mu)$, and define the fractional operation μ' as follows: for each component $C \in \mathcal{R}(\mu)$, pick any unary operation $h \in C$, note that $g \circ h \in C$, and let $\mu'(g \circ h) = w(C)$. Then, by Theorem 5.2, μ' is a unary fractional polymorphism of Γ with the property that $h'(D) \subseteq g(D) = D'$ for every $h' \in \text{supp}(\mu')$.

Now define the following fractional operation:

$$\omega := \sum_{g' \in \text{supp}(\omega')} \omega'(g') \sum_{h' \in \text{supp}(\mu')} \mu'(h') \chi_{g \circ (h', h')}.$$

Let $f \in \Gamma$ and $\bar{x}^1, \bar{x}^2 \in D^{ar(f)}$. Then,

$$\begin{aligned} f^2(\bar{x}^1, \bar{x}^2) &\leq \sum_{h' \in \text{supp}(\mu')} \mu'(h') f^2(h'(\bar{x}^1), h'(\bar{x}^2)) \\ &\leq \sum_{h' \in \text{supp}(\mu')} \mu'(h') \sum_{g' \in \text{supp}(\omega')} \omega'(g') f^2(g'(h'(\bar{x}^1), h'(\bar{x}^2))) \\ &= \sum_{g \in \text{supp}(\omega)} \omega(g) f^2(g(\bar{x}^1, \bar{x}^2)), \end{aligned}$$

where the first inequality follows since Γ admits μ, and the second inequality follows since Γ' admits ω, and $h'(\bar{x}^1), h'(\bar{x}^2) \in (D')^{ar(f)}$ for every $h' \in \text{supp}(\mu')$. Hence, ω is a binary symmetric fractional polymorphism of Γ, which proves the lemma. □

5.3. Proof of Lemma 5.1

We remark that after the announcement of our work in [Thapper and Živný 2013], the idea in the following proof has been used to prove a generalisation of Lemma 5.1 in [Uppman 2014, Lemma 28], where it has been used to analyse the complexity of certain Min-Cost-Hom problems.

PROOF (OF LEMMA 5.1). Let $\pi_1(x,y) = x$ and $\pi_2(x,y) = y$ be the two binary projections on $D(\Delta)$. Let $\Omega(a,b)$ be the set of operations $g : D(\Delta) \times D(\Delta) \to D(\Delta)$ for which $\{g(a,b), g(b,a)\} \neq \{a,b\}$. Assume that there exist rational values $y(f,\bar{x}) \geq 0$, for $f \in \Delta, \bar{x} \in (D(\Delta) \times D(\Delta))^{ar(f)}$, such that

$$\sum_{f,\bar{x}} y(f,\bar{x}) f(g(\bar{x})) \geq \sum_{f,\bar{x}} y(f,\bar{x}) f(\pi_i(\bar{x})), \quad \forall g \in \mathcal{O}^{(2)}_{D(\Delta)}, i = 1,2, \tag{12}$$

$$\sum_{f,\bar{x}} y(f,\bar{x}) f(g(\bar{x})) > \sum_{f,\bar{x}} y(f,\bar{x}) f(\pi_i(\bar{x})), \quad \forall g \in \Omega(a,b), i = 1,2. \tag{13}$$

Let $V = \{v_{(x,y)} \mid (x,y) \in D(\Delta) \times D(\Delta)\}$ and let $v_1, \ldots, v_n$ be an enumeration of V with $v_1 = v_{(a,b)}$ and $v_2 = v_{(b,a)}$. Define $\iota : V \to D(\Delta) \times D(\Delta)$ by $\iota(v_{(x,y)}) = (x,y)$ and let I be the instance of VCSP(Δ) with variables V and objective function $f_I(v_1, \ldots, v_n) = \sum_{f,\bar{x}} y(f,\bar{x}) f(\iota^{-1}(\bar{x}))$. Define $f(x,y) = \min_{v_3,\ldots,v_n \in D} f_I(x, y, v_3, \ldots, v_n) \in \langle\Delta\rangle$. The equations (12) imply that $\pi_1 \circ \iota$ and $\pi_2 \circ \iota$ are among the optimal solutions to I, and the equations (13) imply that $\pi_1 \circ \iota$ and $\pi_2 \circ \iota$ have strictly smaller measure than any solution $g \in \Omega(a,b)$, so $f(a,b) = f(b,a) < f(x,y)$ for all $\{x,y\} \neq \{a,b\}$.

We conclude that if (MC) cannot be satisfied, then there is no solution to the system (12)+(13). By Lemma 2.8, there is a solution $z_1(g,i), z_2(g,i) \geq 0$ to the following system of equations:

$$\begin{aligned} &\sum_{i=1}^{2} \sum_{g \in \Omega(a,b)} z_1(g,i)(f(g(\bar{x})) - f(\pi_i(\bar{x}))) \\ &+ \sum_{i=1}^{2} \sum_{g \in \mathcal{O}^{(2)}_{D(\Delta)}} z_2(g,i)(f(g(\bar{x})) - f(\pi_i(\bar{x}))) \leq 0, \quad \forall f \in \Delta, \bar{x} \in (D(\Delta) \times D(\Delta))^{ar(f)}, \end{aligned} \tag{14}$$

with $z_1(g,i) \neq 0$ for some $g \in \Omega(a,b)$ and $i \in \{1,2\}$. Define $z_1(g,i) = 0$ for $g \notin \Omega(a,b)$ and let $z(g) = \|z_1 + z_2\|^{-1}(z_1(g,1) + z_1(g,2) + z_2(g,1) + z_2(g,2))$. A solution to (14) then implies a solution to the following system of inequalities:

$$\sum_{g \in \mathcal{O}^{(2)}_{D(\Delta)}} z(g) f(g(\bar{x})) \leq f^2(\pi_1(\bar{x}), \pi_2(\bar{x})), \quad \forall f \in \Delta, \bar{x} \in (D(\Delta) \times D(\Delta))^{ar(f)},$$

with $\|z\|_1 = 1$, $z(g) \geq 0$, and $z(g) > 0$ for some $g \in \Omega(a,b)$. Denote this solution by $z_{a,b}(g)$. Now, if (MC) cannot be satisfied for *any* distinct $a, b \in D(\Delta)$, then we have solutions $z_{a,b}(g)$ for all $a \neq b \in D(\Delta)$. The lemma follows with ω defined by $\omega(g) = (|D(\Delta)|^2 - |D(\Delta)|)^{-1} \sum_{a \neq b} z_{a,b}(g)$. □

5.4. Proof of Lemma 5.6

The aim of this section is to prove that the graph S of submodular pairs is connected. In order to do so, we introduce yet another graph T that records the "definable 2-subsets of D in $\langle\Gamma_c\rangle$". We then show that T is a subgraph of S and that T is connected. Since S and T are defined on the same set of vertices, it then follows that S is connected.

Let $T = (V(T), E(T))$ be the undirected graph with:

— $V(T) = D$;
— $E(T) = \{\{a,b\} \mid$ there exists a unary function $u \in \langle\Gamma_c\rangle$ such that $\operatorname{argmin} u = \{a,b\}\}$.

LEMMA 5.8. $E(T) \subseteq E(S)$.

PROOF. Take an arbitrary edge $\{a,b\} \in E(T)$ and let u_a, u_b, and u_{ab} be unary cost functions in $\langle\Gamma_c\rangle$ such that $\operatorname{argmin} u_a = \{a\}$, $\operatorname{argmin} u_b = \{b\}$, and $\operatorname{argmin} u_{ab} = \{a,b\}$, respectively. Since u_{ab} minimises on $\{a,b\}$ and is improved by both $\hat{\omega}$ and $\hat{\rho}$, we have $g(a,b), g(b,a) \in \{a,b\}$ for every $g \in \operatorname{supp}(\hat{\omega})$ and every $\mathbf{g} = (g,\bar{g}) \in \operatorname{supp}(\hat{\rho})$. By construction of $\hat{\sigma}$, there is a mapping $\mathbf{h} \in \operatorname{supp}(\hat{\sigma})$ for which $\mathbf{h}(a,b) \notin \{(a,b),(b,a)\}$, so by our previous observation, we must have either $\mathbf{h}(a,b) = (a,a)$ or $\mathbf{h}(a,b) = (b,b)$. Suppose that $\mathbf{g}(a,b) \in \{(a,b),(b,a)\}$ for some $\mathbf{g} \in \operatorname{supp}(\hat{\rho})$. Then $\mathbf{h} \circ \mathbf{g}(a,b) = (a,a)$ or (b,b). So $\mathbf{h} \circ \mathbf{g}$ is reachable from $\mathbf{g}$ in G, it is symmetric on $\{a,b\}$, and every $\mathbf{g}'$ reachable from $\mathbf{h} \circ \mathbf{g}$ is symmetric on $\{a,b\}$. Therefore $\mathbf{g}$ cannot be recurrent. But $\operatorname{supp}(\hat{\rho})$ is contained in the set of recurrent states, a contradiction. We conclude that every $\mathbf{g} \in \operatorname{supp}(\hat{\rho})$ is symmetric on $\{a,b\}$ and maps (a,b) to either (a,a) or (b,b).

Let $w_a = \sum_{\mathbf{g}:\mathbf{g}(a,b)=(a,a)} \rho(\mathbf{g})$ and $w_b = \sum_{\mathbf{g}:\mathbf{g}(a,b)=(b,b)} \hat{\rho}(\mathbf{g})$. By the previous argument, we have $w_a + w_b = 1$. By the fractional polymorphism inequality applied to $\hat{\rho}$ and u_a, we have

$$\frac{1}{2}(u_a(a) + u_a(b)) \geq w_a u_a(a) + w_b u_a(b). \tag{15}$$

Since $u_a(a) < u_a(b)$, we have $w_a \geq w_b$. But inequality (15) holds for u_b as well, hence $w_a \leq w_b$, and therefore $w_a = w_b = \frac{1}{2}$. □

LEMMA 5.9. *T is connected.*

To prove this lemma, we will introduce some terminology from the study of hyperplane arrangements which will facilitate our reasoning about the edges of T. For a more thorough treatment of this subject, see [Abramenko and Brown 2008] and [Stanley 2007].

Definition 5.10. Let $\{\bar{v}^i\}_{i\in I}$ be a finite set of vectors in $\mathbb{R}^n$. The set of hyperplanes $\mathcal{A} = \{H_i\}_{i\in I}$, where $H_i = \{\bar{x} \in \mathbb{R}^n \mid \bar{v}^i \cdot \bar{x} = 0\}$, is called a *(linear) hyperplane arrangement*.

To each vector $\bar{x} \in \mathbb{R}^n$, we associate a *sign vector*, $\operatorname{sgn}(\bar{x}) \in \{-1,0,+1\}^I$, where the ith component is given by the sign of $\bar{v}^i \cdot \bar{x}$ for each $i \in I$. For a sign vector $\bar{v} \in \{-1,0,+1\}^I$, a non-empty set $A = \operatorname{sgn}^{-1}(\bar{v}) = \{\bar{x} \in \mathbb{R}^n \mid \operatorname{sgn}(\bar{x}) = \bar{v}\}$ is called a *cell* of $\mathcal{A}$. We denote the defining sign vector, $\bar{v}$ of A, by $\operatorname{sgn}(A)$.

A cell A with $\operatorname{sgn}(A)_i \neq 0$ for all $i \in I$ is called a *chamber*. The chambers are the connected full-dimensional regions of $\mathbb{R}^n \setminus \bigcup_{i\in I} H_i$. A cell P with $\operatorname{sgn}(P)_i = 0$ for exactly one $i \in I$ is called a *panel*. We say that *P is a panel of a chamber A* if the panel P is contained in the topological closure $\operatorname{cl}(A)$ of A. Each panel is a panel of precisely two chambers.

The *chamber graph* of $\mathcal{A}$ is the undirected graph with the chambers of $\mathcal{A}$ as vertices and an edge between two chambers A_1 and A_2 if $\operatorname{sgn}(A_1)$ and $\operatorname{sgn}(A_2)$ differ by a single sign change, or equivalently, if A_1 and A_2 share a common panel. We will use the following properties of the chamber graph that can be found in [Abramenko and Brown 2008, Proposition 1.54].

PROPOSITION 5.11. *The chamber graph of $\mathcal{A}$ is connected and the minimal length of a path between A_1 and A_2 in the chamber graph is equal to the number of positions at which $\operatorname{sgn}(A_1)$ and $\operatorname{sgn}(A_2)$ differ.*

We are now ready to prove Lemma 5.9.

PROOF (OF LEMMA 5.9). For each $a \in D$, we have a unary function $u_a \in \langle\Gamma_c\rangle$ with $\operatorname{argmin} u_a = \{a\}$. For $\bar{x} \in \mathbb{R}^D$, with components x_c, consider the linear combination $f_{\bar{x}}(z) = \sum_{c\in D} x_c u_c(z)$. Note that if $\bar{x}$ is rational and nonnegative, then $f_{\bar{x}} \in \langle\Gamma_c\rangle$. The inequality $f_{\bar{x}}(a) < f_{\bar{x}}(b)$ is equivalent to $\sum_{c\in D} x_c(u_c(a) - u_c(b)) < 0$, i.e., $f_{\bar{x}}$ takes a strictly smaller value on a than on b precisely when the vector $\bar{x}$ is on the negative side of the hyperplane H^{ab} defined by the normal $\bar{v}^{ab}$ with components $v_c^{ab} = u_c(a) - u_c(b)$. Hence, by determining the sign of $\bar{x} \cdot \bar{v}^{ab}$, we can decide whether $f_{\bar{x}}(a) < f_{\bar{x}}(b)$ or $f_{\bar{x}}(a) > f_{\bar{x}}(b)$. If $\bar{x}$ lies *on* the hyperplane, then $f_{\bar{x}}(a) = f_{\bar{x}}(b)$.

For each $a \in D$, let H^a be the hyperplane defined by the *unit vector* $\bar{e}^a$, i.e., $e_a^a = 1$ and $e_c^a = 0$ for $a \neq c$. Fix a strict total order $<_D$ on D. Let $\mathcal{A} = \{H^{ab} \mid a <_D b\} \cup \{H^a \mid a \in D\}$ be a hyperplane arrangement in $\mathbb{R}^D$. Let $\mathcal{C}$ be the set of chambers A that have a positive sign for each H^a, i.e., each $A \in \mathcal{C}$ is contained in the positive (open) orthant of $\mathbb{R}^D$. Since all remaining components of $A \in \mathcal{C}$ are also nonzero, they determine a strict order on the values of the functions $f_{\bar{x}}$, $\bar{x} \in A$. For each $a \in D$, let $U_a = \{A \in \mathcal{C} \mid \forall \bar{x} \in A : \operatorname{argmin} f_{\bar{x}} = \{a\}\}$. Each U_a is non-empty since the vector $\bar{x}$ given by $x_c = \epsilon$ for $c \neq a$ and $x_a = 1$ determines a function minimizing on a when $\epsilon > 0$ is chosen small enough.

Fix $a, b \in D$ and pick any $A_a \in U_a, A_b \in U_b$. Let $A_a = A_0, A_1, \ldots, A_\ell = A_b$ be a minimal-length path from A_a to A_b in the chamber graph. Consider the sign vectors along this path: $\operatorname{sgn}(A_0), \operatorname{sgn}(A_1), \ldots, \operatorname{sgn}(A_\ell)$. By Proposition 5.11 the sign of a fixed component changes at most once along this sequence. In particular, since A_a and A_b both have positive signs for the hyperplanes H^a, it follows that A_i is contained in the positive orthant for every i. Hence, for each i, there is a $a_i \in D$ such that $A_i \in U_{a_i}$. For each i with $a_i \neq a_{i+1}$, the path moves from a chamber where $f_{\bar{x}}$ minimises on a_i to a chamber where it minimises on a_{i+1}. This means that A_i and A_{i+1} share a panel P_i with a sign vector $\operatorname{sgn}(P_i)$ obtained from either $\operatorname{sgn}(A_i)$ or $\operatorname{sgn}(A_{i+1})$ by setting the component corresponding to $H^{a_i a_{i+1}}$ to 0 (assuming $a_i <_D a_{i+1}$). Since all other components of $\operatorname{sgn}(P_i)$ have the same sign as in $\operatorname{sgn}(A_i)$ and $\operatorname{sgn}(A_{i+1})$, we have $f_{\bar{x}}(a_i) = f_{\bar{x}}(a_{i+1}) < f_{\bar{x}}(c)$, for every $\bar{x} \in P_i$ and $c \neq a_i, a_{i+1}$. For a hyperplane arrangement, such as $\mathcal{A}$, that is defined in terms of rational normal vectors, each cell is defined as the solutions to a set of linear equalities and inequalities with rational coefficients. Every cell therefore contains at least one rational vector. In particular, there exists a nonnegative rational vector $\bar{x} \in P_i$ with $\operatorname{argmin} f_{\bar{x}} = \{a_i, a_{i+1}\}$, so $\{a_i, a_{i+1}\} \in E(T)$. This holds for all $0 \leq i < \ell$ with $a_i \neq a_{i+1}$, so we conclude that a subsequence of $a = a_0, a_1, \ldots, a_\ell = b$ is a path in T from a to b. $\square$

5.5. Proof of Theorem 5.2

A (time-homogeneous) finite-state Markov chain M is given by a set of states and conditional probabilities $p(i, j)$ for M to be in state j at time $t + 1$ given that it was in state i at time t. Let $p^{(k)}(i, j)$ denote the probability that M proceeds from state i to state j in exactly k transitions. M is called *irreducible* if, for every pair of states (i, j), there exists $r \geq 1$ with $p^{(r)}(i, j) > 0$. A state i is called *transient* if, for some state j, there is a path (in the graph whose vertices are the states of M and with and edge (i, j) from state i to state j if $p(i, j) > 0$) from i to j but not from j to i. A state that is not transient is called *recurrent*. A state i has *periodicity* r if $r = \gcd\{k \mid p^{(k)}(i, i) > 0\}$. M is called *aperiodic* if all states have periodicity 1. A *stationary distribution* of M is a probability distribution λ on the set of states of M such that $\lambda(i) = \sum_j \lambda(j)p(j, i)$ for all states i. The following is well known.

THEOREM 5.12.
For any finite-state Markov chain M:

(1) *If M is irreducible, then there is a unique stationary distribution λ of M with $\lambda(i) > 0$ for all states i.*
(2) *If M is aperiodic, then for any initial distribution π, there is a stationary distribution λ of M with $\sum_j \pi(j)p^{(k)}(j, i) \to \lambda(i)$ as $k \to \infty$, for all states i.*

(3) *If i is transient, then $p^{(k)}(j,i) \to 0$ as $k \to \infty$, for all states j.*

PROOF. Part (1) follows from [Kemeny and Snell 1976, Theorem 5.1.1 and 5.1.2], where an irreducible chain is called ergodic. (The definition in [Kemeny and Snell 1976] of an ergodic chain differs from the more common one which defines an ergodic chain as an irreducible and aperiodic chain.)

[Kemeny and Snell 1976, Theorem 4.1.4] proves the claim of part (2) for irreducible aperiodic chains. This result can be extended to any aperiodic chain by considering what happens for an initial distribution concentrated on a single state i. Let $\mathcal{R}$ denote the set of maximal strongly connected components in the directed graph that has the recurrent states of M as vertices and an edge from i to j if $p(i,j) > 0$. If i is recurrent, then the restriction of M to the component $C \in \mathcal{R}$ containing i is irreducible, so the chain converges to a stationary distribution on C with the desired properties. Let λ^i be the trivial extension of this distribution to a stationary distribution on M. If instead i is transient, then for each component $C \in \mathcal{R}$, there is some probability that i reaches C. The stationary distribution λ^i is then defined as the unique stationary distribution of each irreducible component, weighted by the probability that i reaches this component. Finally, the full statement of part (2) follows by taking $\lambda = \sum_i \pi(i)\lambda^i$.

Part (3) follows from [Kemeny and Snell 1976, Theorem 3.1.1]. □

Given an $m \to m$ fractional mapping σ, we define a Markov chain $M(\sigma)$ on $G(\sigma)$. Let $w(\mathbf{g},\mathbf{g}') = \sum_{\mathbf{h}\in \mathrm{supp}(\sigma):\mathbf{g}'=\mathbf{h}\circ\mathbf{g}} \sigma(\mathbf{h})$. The transition probabilities are given as follows:

$$p(\mathbf{g},\mathbf{g}') = \begin{cases} \frac{1}{2}w(\mathbf{g},\mathbf{g}') + \frac{1}{2} & \text{if } \mathbf{g} = \mathbf{g}', \text{ and} \\ \frac{1}{2}w(\mathbf{g},\mathbf{g}') & \text{otherwise.} \end{cases}$$

Note that the set of recurrent vertices in $\mathcal{V}(\sigma)$, defined in Section 5.1, is precisely the set of recurrent states of $M(\sigma)$. Let C be a component in $\mathcal{R}(\sigma)$. Define $M(C)$ to be the restriction of $M(\sigma)$ to $C \subseteq \mathcal{V}(\sigma)$. Then, $M(C)$ is also a Markov chain.

LEMMA 5.13. *The Markov chains $M(\sigma)$ and $M(C)$ are aperiodic and each chain $M(C)$ is irreducible.*

PROOF. Aperiodicity follows by construction as $p(\mathbf{g},\mathbf{g}) \geq \frac{1}{2} > 0$ for all $\mathbf{g} \in \mathcal{V}(\sigma)$. Irreducibility follows since each C is a maximal strongly connected component of recurrent states. □

LEMMA 5.14. *Let ρ and λ be probability distributions on $\mathcal{V}(\sigma)$ and assume that $M(\sigma)$ converges to λ when starting in ρ. Then, for every $f \in \mathrm{Imp}(\sigma)$, and $\bar{x}^1,\ldots,\bar{x}^m \in D^{ar(f)}$,*

$$\sum_{\mathbf{g}\in\mathcal{V}(\sigma)} \rho(\mathbf{g}) f^m(\mathbf{g}(\bar{x}^1,\ldots,\bar{x}^m)) \geq \sum_{\mathbf{g}\in\mathcal{V}(\sigma)} \lambda(\mathbf{g}) f^m(\mathbf{g}(\bar{x}^1,\ldots,\bar{x}^m)).$$

PROOF. By k times applying the $m \to m$ fractional polymorphism $\frac{1}{2}(\chi_{\mathbf{1}} + \sigma)$ to the left-hand side, we have

$$\begin{aligned}\sum_{\mathbf{g} \in \mathcal{V}(\sigma)} \rho(\mathbf{g}) f^m(\mathbf{g}(\bar{x}^1, \dots, \bar{x}^m)) &\geq \sum_{\mathbf{g} \in \mathcal{V}(\sigma)} \rho(\mathbf{g}) \frac{1}{2} \Big(f^m(\mathbf{g}(\bar{x}^1, \dots, \bar{x}^m)) \\ &\quad + \sum_{\mathbf{h} \in \mathrm{supp}(\sigma)} \sigma(\mathbf{h}) f^m(\mathbf{h} \circ \mathbf{g}(\bar{x}^1, \dots, \bar{x}^m)) \Big) \\ &= \sum_{\mathbf{g} \in \mathcal{V}(\sigma)} \sum_{\mathbf{g}' \in \mathcal{V}(\sigma)} \rho(\mathbf{g}') p(\mathbf{g}', \mathbf{g}) f^m(\mathbf{g}(\bar{x}^1, \dots, \bar{x}^m)) \\ &\geq \dots \geq \sum_{\mathbf{g} \in \mathcal{V}(\sigma)} \rho^{(k)}(\mathbf{g}) f^m(\mathbf{g}(x^1, \dots, x^m)),\end{aligned}$$

where $\rho^{(k)}(\mathbf{g}) = \sum_{\mathbf{g}' \in \mathcal{V}(\sigma)} \rho(\mathbf{g}') p^{(k)}(\mathbf{g}', \mathbf{g})$. By assumption, $\rho^{(k)}(\mathbf{g}) \to \lambda(\mathbf{g})$ as $k \to \infty$. Since the right-hand side is a linear function in $\rho^{(k)}(\mathbf{g})$, the lemma follows by continuity. □

LEMMA 5.15. *Let $c_1, \dots, c_m \in \mathbb{Q}_{>0}$ and $x_1, \dots, x_m \in \mathbb{Q}$ be such that $\sum_i c_i = 1$, and $x_j \geq \sum_i c_i x_i$ for all j. Then, $x_j = \sum_i c_i x_i$ for all j.*

PROOF. Let $C = \sum_i c_i x_i$. We have $x_j \geq C$ for all j. If $x_j > C$ for some j, then $c_j x_j > c_j C$, so $C = \sum_i c_i x_i > \sum_i c_i C = C$, a contradiction. So, for all j, $x_j \leq C$, and hence $x_j = C$. □

LEMMA 5.16. *Let σ be an $m \to m$ fractional mapping and let $C \in \mathcal{R}(\sigma)$. Then, for all $f \in \mathrm{Imp}(\sigma)$, $\bar{x}^1, \dots, \bar{x}^m \in D^{ar(f)}$, and $\mathbf{g} \in C$,*

$$f^m(\mathbf{g}(\bar{x}^1, \dots, \bar{x}^m)) = \sum_{\mathbf{h} \in C} \lambda(\mathbf{h}) f^m(\mathbf{h}(\bar{x}^1, \dots, \bar{x}^m)),$$

where λ is the unique stationary distribution on $M(C)$.

PROOF. For $\mathbf{g} \in C$, let $\chi_{\mathbf{g}}$ be the distribution on $\mathcal{V}$ that assigns probability 1 to $\mathbf{g}$ and 0 to all other mappings in $\mathcal{V}$. By Theorem 5.12(2), $M(\sigma)$ converges to a stationary distribution λ when starting in $\chi_{\mathbf{g}}$. By Lemma 5.14,

$$f^m(\mathbf{g}(\bar{x}^1, \dots, \bar{x}^m)) = \sum_{\mathbf{h} \in \mathcal{V}} \chi_{\mathbf{g}}(\mathbf{h}) f^m(\mathbf{h}(\bar{x}^1, \dots, \bar{x}^m)) \geq \sum_{\mathbf{h} \in \mathcal{V}} \lambda(\mathbf{h}) f^m(\mathbf{h}(\bar{x}^1, \dots, \bar{x}^m)).$$

Note that the chain $M(\sigma)$ stays within the component C when starting in $\chi_{\mathbf{g}}$. Therefore, $\sum_{\mathbf{h} \in \mathcal{V}} \lambda(\mathbf{h}) f^m(\mathbf{h}(\bar{x}^1, \dots, \bar{x}^m)) = \sum_{\mathbf{h} \in C} \lambda(\mathbf{h}) f^m(\mathbf{h}(\bar{x}^1, \dots, \bar{x}^m))$, and $M(C)$ converges to the restriction of λ to C when starting in the restriction of $\chi_{\mathbf{g}}$ to C. Hence, by Theorem 5.12(1), $\lambda(\mathbf{g}) > 0$ for all $\mathbf{g} \in C$. It now follows from Lemma 5.15 with $c_{\mathbf{g}} = \lambda(\mathbf{g})$ and $x_{\mathbf{g}} = f^m(\mathbf{g}(\bar{x}^1, \dots, \bar{x}^m))$, for $\mathbf{g} \in C$, that $f^m(\mathbf{g}(\bar{x}^1, \dots, \bar{x}^m)) = \sum_{\mathbf{h} \in C} \lambda(\mathbf{h}) f^m(\mathbf{h}(\bar{x}^1, \dots, \bar{x}^m))$. □

We are now ready to prove Theorem 5.2 and Lemma 5.7.

PROOF (OF THEOREM 5.2). By Theorem 5.12(2), there exists a stationary distribution λ of $M(\sigma)$ such that $\sum_{\mathbf{g}'} \sigma(\mathbf{g}') p^{(k)}(\mathbf{g}', \mathbf{g}) \to \lambda(\mathbf{g})$ as $k \to \infty$, for all $\mathbf{g} \in \mathcal{V}(\sigma)$. For $C \in \mathcal{R}(\sigma)$, define $w(C) = \sum_{\mathbf{g} \in C} \lambda(\mathbf{g})$. By Theorem 5.12(3), $\lambda(\mathbf{g}) = 0$ for $\mathbf{g} \notin \mathcal{R}(\sigma)$, hence w is a probability distribution on $\mathcal{R}(\sigma)$.

Let ρ be such that $\sum_{\mathbf{g} \in C} \rho(\mathbf{g}) = w(C)$. Arbitrarily pick $f \in \mathrm{Imp}(\sigma)$ and $\bar{x}^1, \dots, \bar{x}^m \in D^{ar(f)}$. Note that, by Lemma 5.14, $f \in \mathrm{Imp}(\lambda)$. Define λ' to be the distribution on C

given by $\lambda'(\mathbf{g}) = \lambda(\mathbf{g})/w(C)$, for $\mathbf{g} \in C$. Then, λ' is a stationary distribution on $M(C)$, and by Theorem 5.12(1), it is unique. Therefore, by Lemma 5.16, we have

$$\begin{aligned}\sum_{\mathbf{g}\in C} \rho(\mathbf{g}) f^m(\mathbf{g}(\bar{x}^1,\ldots,\bar{x}^m)) &= \sum_{\mathbf{g}\in C} \rho(\mathbf{g}) \sum_{\mathbf{h}\in C} \lambda'(\mathbf{h}) f^m(\mathbf{h}(\bar{x}^1,\ldots,\bar{x}^m)) \\ &= w(C) \sum_{\mathbf{h}\in C} \lambda'(\mathbf{h}) f^m(\mathbf{h}(\bar{x}^1,\ldots,\bar{x}^m)) \\ &= \sum_{\mathbf{h}\in C} \lambda(\mathbf{h}) f^m(\mathbf{h}(\bar{x}^1,\ldots,\bar{x}^m)).\end{aligned}$$

As this holds for every $C \in \mathcal{R}(\sigma)$, it follows that $f \in \mathrm{Imp}(\rho)$. □

5.6. Proof of Lemma 5.7

PROOF (OF LEMMA 5.7). Let $C \in \mathcal{R}(\hat{\sigma})$ be the component containing $\mathbf{g}$, and for $i = 1, 2$, let $\Omega_i = \{\mathbf{h} \in \mathrm{supp}(\hat{\rho}) \mid \mathbf{h}(a_1, a_2) = (a_i, a_i)\}$.

$$\begin{aligned}f^2((a_1,\bar{y}^1),(a_2,\bar{y}^2)) &\geq \sum_{\mathbf{h}\in\mathrm{supp}(\hat{\rho})} \hat{\rho}(\mathbf{h}) f^2(\mathbf{h}((a_1,\bar{y}^1),(a_2,\bar{y}^2))) && (16)\\ &= \sum_{\mathbf{h}\in\Omega_1} \hat{\rho}(\mathbf{h}) f^2(\mathbf{h}((a_1,\bar{y}^1),(a_1,\bar{y}^2))) \\ &\quad + \sum_{\mathbf{h}\in\Omega_2} \hat{\rho}(\mathbf{h}) f^2(\mathbf{h}((a_2,\bar{y}^1),(a_2,\bar{y}^2))) && (17)\\ &= \frac{1}{2} f^2((a_1,\bar{y}^1),(a_1,\bar{y}^2)) + \frac{1}{2} f^2((a_2,\bar{y}^1),(a_2,\bar{y}^2)) && (18)\\ &= \frac{1}{2} f^2((a_1,\bar{y}^1),(a_2,\bar{y}^2)) + \frac{1}{2} f^2((a_1,\bar{y}^2),(a_2,\bar{y}^1)), && (19)\end{aligned}$$

where (16) follows by applying $\hat{\rho}$ and (17) follows from $\hat{\rho}$ being idempotent and submodular on $\{a_1, a_2\}$. To obtain (18), note that $\mathbf{h} \circ \mathbf{g} \in C$, so by the first part of Lemma 5.16, $f^2(\mathbf{h} \circ \mathbf{g}((a_i,\bar{x}^1),(a_i,\bar{x}^2))) = f^2(\mathbf{g}((a_i,\bar{x}^1),(a_i,\bar{x}^2))) = f^2((a_i,\bar{y}^1),(a_i,\bar{y}^2))$ for all $\mathbf{h} \in \Omega_i$ and $i = 1, 2$. Finally, (19) follows by rearranging the terms.

This shows the inequality $f^2((a_1,\bar{y}^1),(a_2,\bar{y}^2)) \geq f^2((a_1,\bar{y}^2),(a_2,\bar{y}^1))$. The reverse inequality follows analogously. □

6. SYMMETRIC FRACTIONAL POLYMORPHISMS OF ALL ARITIES

An important step in the proof of Theorem 3.1 is showing that a binary symmetric fractional polymorphism "generates" symmetric fractional polymorphisms of all higher arities. This was proved in [Kolmogorov 2013]. In this section, we demonstrate the power of the Markov chain machinery set up in Section 5.5 by giving an alternative proof of this theorem. The proof idea is the same as that of [Kolmogorov 2013], but the proof is substantially shortened.

THEOREM 6.1 ([KOLMOGOROV 2013]). *Suppose Γ is a constraint language with a symmetric fractional polymorphism of arity 2. Then Γ has symmetric fractional polymorphisms of all arities.*

PROOF. It suffices to prove that if Γ has a symmetric fractional polymorphism of arity $m-1 \geq 2$, then it has one of arity m. Let ω be an $(m-1)$-ary symmetric fractional polymorphism of Γ. For $1 \leq k \leq m$, let $\delta_k \in \mathcal{O}_D^{(m\to m-1)}$ denote the mapping obtained by omitting the kth operation from the identity mapping in $\mathcal{O}_D^{(m\to m)}$. Define

$$\sigma := \sum_{h\in\operatorname{supp}(\omega)} \omega(h)\chi_{(h\circ\delta_1,\dots,h\circ\delta_m)}.$$

Then, σ is a fractional polymorphism of Γ. Let ρ be a fractional polymorphism of Γ of arity $m \to m$ as given by Theorem 5.2 applied to σ, and let $\mathbf{p}$ be any symmetric and permuting mapping of arity $m \to m$. For example, let $\mathbf{p}$ be a mapping that orders its m inputs according to some fixed total order on D. We claim that $\rho' = \rho \circ \mathbf{p}$ is a fractional polymorphism of Γ, from which the theorem follows as ρ' is clearly symmetric.

Let $f \in \Gamma$ and $\bar{x}^1,\dots,\bar{x}^m \in D^{ar(f)}$. It suffices to show that for every $\mathbf{g} \in \operatorname{supp}(\rho)$, $f^m(\mathbf{g}(\bar{x}^1,\dots,\bar{x}^m)) = f^m(\mathbf{g}\circ\mathbf{p}(\bar{x}^1,\dots,\bar{x}^m))$. We do this by showing that for any $1 \leq i \leq ar(f)$ and $1 \leq j_1, j_2 \leq m$, interchanging $x_i^{j_1}$ and $x_i^{j_2}$ does not alter the value of $f^m(\mathbf{g}(\bar{x}^1,\dots,\bar{x}^m))$. The result then follows by repeatedly interchanging such pairs of elements in $(\bar{x}^1,\dots,\bar{x}^m)$ to obtain $\mathbf{p}(\bar{x}^1,\dots,\bar{x}^m)$.

For $1 \leq k \leq m$, let $\pi_k \in \mathcal{O}_D^{(m)}$ denote the projection on the kth component. Since $m \geq 3$, we can pick $k \in \{1,\dots,m\} \setminus \{j_1, j_2\}$. Let $\mathbf{h} \in \operatorname{supp}(\sigma)$ and let τ be a permutation on $\{1,\dots,m\}$ that interchanges j_1 and j_2. By definition of σ,

$$\pi_k \circ \mathbf{h}(x_1,\dots,x_m) = \pi_k \circ \mathbf{h}(x_{\tau(1)},\dots,x_{\tau(m)}), \tag{20}$$

for $x_1,\dots,x_m \in D$. Furthermore, this identity is seen to hold for any $\mathbf{h} = \mathbf{h}_\ell \circ \cdots \circ \mathbf{h}_1 \in \mathcal{V}(\sigma)$ by induction over ℓ.

Let $C \in \mathcal{R}(\sigma)$ be the component containing $\mathbf{g}$ and let λ be the unique stationary distribution on $M(C)$. Then we have

$$\begin{aligned}
\sum_{\mathbf{h}\in C} \lambda(\mathbf{h}) f^{m-1}(\delta_k \circ \mathbf{h}(\bar{x}^1,\dots,\bar{x}^m)) &\geq \sum_{\mathbf{h}\in C} \lambda(\mathbf{h}) \sum_{h\in\operatorname{supp}(\omega)} \omega(h) f(h\circ\delta_k\circ\mathbf{h}(\bar{x}^1,\dots,\bar{x}^m)\\
&= \sum_{\mathbf{h}\in C} \lambda(\mathbf{h}) \sum_{\mathbf{h}'\in\operatorname{supp}(\sigma)} \sigma(\mathbf{h}') f(\pi_k\circ\mathbf{h}'\circ\mathbf{h}(\bar{x}^1,\dots,\bar{x}^m))\\
&= \sum_{\mathbf{h}\in C} \lambda(\mathbf{h}) \cdot 2 \sum_{\mathbf{h}'\in C} p(\mathbf{h},\mathbf{h}') f(\pi_k\circ\mathbf{h}'(\bar{x}^1,\dots,\bar{x}^m))\\
&\quad - \sum_{\mathbf{h}\in C} \lambda(\mathbf{h}) f(\pi_k\circ\mathbf{h}(\bar{x}^1,\dots,\bar{x}^m)),\\
&= \sum_{\mathbf{h}\in C} \lambda(\mathbf{h}) f(\pi_k\circ\mathbf{h}(\bar{x}^1,\dots,\bar{x}^m)),
\end{aligned} \tag{21}$$

where the inequality follows from applying (1) to ω, the first equality follows from the definition of σ, the second equality follows from the definition of the transition probabilities for $M(C)$:

$$\begin{aligned}
\sum_{\mathbf{h}'\in C} p(\mathbf{h},\mathbf{h}') f(\pi_k\circ\mathbf{h}'(\bar{x}^1,\dots,\bar{x}^m)) &= \frac{1}{2} \sum_{\mathbf{h}'\in\operatorname{supp}(\sigma)} \sigma(\mathbf{h}') f(\pi_k\circ\mathbf{h}'\circ\mathbf{h}(\bar{x}^1,\dots,\bar{x}^m))\\
&\quad + \frac{1}{2} f(\pi_k\circ\mathbf{h}(\bar{x}^1,\dots,\bar{x}^m)),
\end{aligned}$$

and the third equality follows by interchanging the order of summation in the first part and then using the fact that λ is the stationary distribution of $M(C)$. By (21) and Lemma 5.15 with $c_k = \frac{1}{m}$ and $x_k = -\sum_{\mathbf{h}\in C} \lambda(\mathbf{h}) f(\pi_k\circ\mathbf{h}(\bar{x}^1,\dots,\bar{x}^m))$, we have $\sum_{\mathbf{h}\in C} \lambda(\mathbf{h}) f^m(\mathbf{h}(\bar{x}^1,\dots,\bar{x}^m)) = \sum_{\mathbf{h}\in C} \lambda(\mathbf{h}) f(\pi_k\circ\mathbf{h}(\bar{x}^1,\dots,\bar{x}^m)$, so by Lemma 5.16, it follows that

$$f^m(\mathbf{g}(\bar{x}^1,\ldots,\bar{x}^m)) = \sum_{\mathbf{h}\in C} \lambda(\mathbf{h}) f(\pi_k \circ \mathbf{h}(\bar{x}^1,\ldots,\bar{x}^m)). \quad (22)$$

By (20), interchanging $x_i^{j_1}$ and $x_i^{j_2}$ does not alter the value of the right-hand side of (22) and hence it does not alter the value of $f^m(\mathbf{g}(\bar{x}^1,\ldots,\bar{x}^m))$. The result follows. □

7. CONCLUSIONS

In this work we have completely answered the question of which finite-valued constraint languages on finite domains are solvable exactly in polynomial time. In particular, we have characterised the tractable constraint languages as those that admit a binary symmetric fractional polymorphism. We have also shown tractability to be a polynomial-time checkable condition, assuming that the constraint language is a core. By previous results, this implies that all tractable constraint languages are solvable by the basic linear programming relaxation. Thus, we have demonstrated that the basic linear programming (BLP) relaxation suffices for exact solvability of finite-valued constraint languages and that, in this context, semidefinite programming relaxations do not add any power.

8. ACKNOWLEDGMENTS

The authors wish to thank Hubie Chen for useful comments on an earlier draft of this paper, Vladimir Kolmogorov for pointing out that Theorem 1.1 holds without requiring the constraint language Γ to be a core, and the anonymous referees for their thorough work.

REFERENCES

ABRAMENKO, P. AND BROWN, K. S. 2008. *Buildings – Theory and Applications*. Graduate Texts in Mathematics Series, vol. 248. Springer.

BARTO, L. 2011. The dichotomy for conservative constraint satisfaction problems revisited. In *Proceedings of the 26th IEEE Symposium on Logic in Computer Science (LICS'11)*. IEEE Computer Society, 301–310.

BARTO, L. AND KOZIK, M. 2012. Robust Satisfiability of Constraint Satisfaction Problems. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC'12)*. ACM, 931–940.

BARTO, L. AND KOZIK, M. 2014. Constraint Satisfaction Problems Solvable by Local Consistency Methods. *Journal of the ACM 61,* 1. Article No. 3.

BARTO, L., KOZIK, M., AND NIVEN, T. 2009. The CSP dichotomy holds for digraphs with no sources and no sinks (a positive answer to a conjecture of Bang-Jensen and Hell). *SIAM Journal on Computing 38,* 5, 1782–1802.

BERMAN, J., IDZIAK, P., MARKOVIĆ, P., MCKENZIE, R., VALERIOTE, M., AND WILLARD, R. 2010. Varieties with few subalgebras of powers. *Transactions of the American Mathematical Society 362,* 3, 1445–1473.

BULATOV, A. 2006. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *Journal of the ACM 53,* 1, 66–120.

BULATOV, A., KROKHIN, A., AND JEAVONS, P. 2005. Classifying the Complexity of Constraints using Finite Algebras. *SIAM Journal on Computing 34,* 3, 720–742.

BULATOV, A. A. 2011. Complexity of conservative constraint satisfaction problems. *ACM Transactions on Computational Logic 12,* 4. Article 24.

CHANDRA, A. K. AND MERLIN, P. M. 1977. Optimal implementation of conjunctive queries in relational data bases. In *Proceedings of the 9th Annual ACM Symposium on Theory of Computing (STOC'77)*. ACM, 77–90.

CHEKURI, C., KHANNA, S., NAOR, J., AND ZOSIN, L. 2004. A linear programming formulation and approximation algorithms for the metric labeling problem. *SIAM Journal on Discrete Mathematics 18,* 3, 608–625.

COHEN, D., COOPER, M., JEAVONS, P., AND KROKHIN, A. 2005. Supermodular Functions and the Complexity of MAX-CSP. *Discrete Applied Mathematics 149,* 1-3, 53–72.

COHEN, D. A., COOPER, M. C., CREED, P., JEAVONS, P., AND ŽIVNÝ, S. 2013. An algebraic theory of complexity for discrete optimisation. *SIAM Journal on Computing 42,* 5, 915–1939.

COHEN, D. A., COOPER, M. C., AND JEAVONS, P. G. 2006. An Algebraic Characterisation of Complexity for Valued Constraints. In *Proceedings of the 12th International Conference on Principles and Practice of Constraint Programming (CP'06)*. Lecture Notes in Computer Science Series, vol. 4204. Springer, 107–121.

COHEN, D. A., COOPER, M. C., JEAVONS, P. G., AND KROKHIN, A. A. 2006. The Complexity of Soft Constraint Satisfaction. *Artificial Intelligence 170,* 11, 983–1016.

COOPER, M. C. 2008. Minimization of Locally Defined Submodular Functions by Optimal Soft Arc Consistency. *Constraints 13,* 4, 437–458.

COOPER, M. C., DE GIVRY, S., SÁNCHEZ, M., SCHIEX, T., ZYTNICKI, M., AND WERNER, T. 2010. Soft arc consistency revisited. *Artificial Intelligence 174,* 7–8, 449–478.

COOPER, M. C. AND ŽIVNÝ, S. 2011. Hybrid tractability of valued constraint problems. *Artificial Intelligence 175,* 9-10, 1555–1569.

COOPER, M. C. AND ŽIVNÝ, S. 2012. Tractable triangles and cross-free convexity in discrete optimisation. *Journal of Artificial Intelligence Research 44*, 455–490.

CRAMA, Y. AND HAMMER, P. L. 2011. *Boolean Functions - Theory, Algorithms, and Applications*. Cambridge University Press.

CREED, P. AND ŽIVNÝ, S. 2011. On minimal weighted clones. In *Proceedings of the 17th International Conference on Principles and Practice of Constraint Programming (CP'11)*. Lecture Notes in Computer Science Series, vol. 6876. Springer, 210–224.

CREIGNOU, N. 1995. A dichotomy theorem for maximum generalized satisfiability problems. *Journal of Computer and System Sciences 51,* 3, 511–522.

CREIGNOU, N., KHANNA, S., AND SUDAN, M. 2001. *Complexity Classification of Boolean Constraint Satisfaction Problems*. SIAM Monographs on Discrete Mathematics and Applications Series, vol. 7. SIAM.

DALMAU, V. AND KROKHIN, A. A. 2013. Robust Satisfiability for CSPs: Hardness and Algorithmic Results. *ACM Transactions on Computation Theory 5,* 4. Article No. 15.

DECHTER, R. 2003. *Constraint Processing*. Morgan Kaufmann.

DEINEKO, V., JONSSON, P., KLASSON, M., AND KROKHIN, A. 2008. The approximability of Max CSP with fixed-value constraints. *Journal of the ACM 55,* 4. Article 16.

FAGIN, R., KOLAITIS, P. G., AND POPA, L. 2005. Data exchange: getting to the core. *ACM Transactions on Database Systems 30,* 1, 174–210.

FÄRNQVIST, T. 2012. Constraint optimization problems and bounded tree-width revisited. In *Proceedings of the 9th International Conference on Integration of Artificial Intelligence and Operations Research Techniques in Constraint Programming (CPAIOR'12)*. Lecture Notes in Computer Science Series, vol. 7298. Springer, 163–197.

FEDER, T. AND VARDI, M. Y. 1998. 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.

GAREY, M. R. AND JOHNSON, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman.

GOTTLOB, G., GRECO, G., AND SCARCELLO, F. 2009. Tractable Optimization Problems through Hypergraph-Based Structural Restrictions. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP'09), Part II*. Lecture Notes in Computer Science Series, vol. 5556. Springer, 16–30.

GROHE, M. 2007. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM 54,* 1, 1–24.

GRÖTSCHEL, M., LOVASZ, L., AND SCHRIJVER, A. 1988. *Geometric Algorithms and Combinatorial Optimization*. Algorithms and Combinatorics Series, vol. 2. Springer.

GURUSWAMI, V. AND RAGHAVENDRA, P. 2008. Constraint Satisfaction over a Non-Boolean Domain: Approximation Algorithms and Unique-Games Hardness. In *Proceedings of the 11th International Workshop on Approximation, Randomization and Combinatorial Optimization and the 12th International Workshop on Randomization and Computation (APPROX-RANDOM'08)*. Lecture Notes in Computer Science Series, vol. 5171. Springer, 77–90.

HÅSTAD, J. 2001. Some optimal inapproximability results. *Journal of the ACM 48,* 4, 798–859.

HÅSTAD, J. 2008. Every 2-csp allows nontrivial approximation. *Computational Complexity 17,* 4, 549–566.

HELL, P. AND NEŠETŘIL, J. 1992. The core of a graph. *Discrete Mathematics 109,* 1–3, 117–126.

HELL, P. AND NEŠETŘIL, J. 2004. *Graphs and Homomorphisms*. Oxford University Press.

HELL, P. AND NEŠETŘIL, J. 2008. Colouring, constraint satisfaction, and complexity. *Computer Science Review 2,* 3, 143–163.

HIRAI, H. 2015. Discrete Convexity and Polynomial Solvability in Minimum 0-Extension Problems. *Mathematical Programming, Series A*. to appear.

HUBER, A., KROKHIN, A., AND POWELL, R. 2013. Skew Bisubmodularity and Valued CSPs. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'13)*. SIAM, 1296–1305.

HUBER, A., KROKHIN, A., AND POWELL, R. 2014. Skew bisubmodularity and valued CSPs. *SIAM Journal on Computing 43,* 3, 1064–1084.

IDZIAK, P. M., MARKOVIC, P., MCKENZIE, R., VALERIOTE, M., AND WILLARD, R. 2010. Tractability and learnability arising from algebras with few subpowers. *SIAM Journal on Computing 39,* 7, 3023–3037.

JEAVONS, P. G. 1998. On the Algebraic Structure of Combinatorial Problems. *Theoretical Computer Science 200,* 1-2, 185–204.

JEAVONS, P. G., COHEN, D. A., AND GYSSENS, M. 1997. Closure Properties of Constraints. *Journal of the ACM 44,* 4, 527–548.

JONSSON, P., KLASSON, M., AND KROKHIN, A. 2006. The Approximability of Three-valued MAX CSP. *SIAM Journal on Computing 35,* 6, 1329–1349.

JONSSON, P., KROKHIN, A. A., AND KUIVINEN, F. 2009. Hard constraint satisfaction problems have hard gaps at location 1. *Theoretical Computer Science 410,* 38-40, 3856–3874.

JONSSON, P., KUIVINEN, F., AND NORDH, G. 2008. MAX ONES Generalized to Larger Domains. *SIAM Journal on Computing 38,* 1, 329–365.

JONSSON, P., KUIVINEN, F., AND THAPPER, J. 2011. Min CSP on Four Elements: Moving Beyond Submodularity. In *Proceedings of the 17th International Conference on Principles and Practice of Constraint Programming (CP'11)*. Lecture Notes in Computer Science Series, vol. 6876. Springer, 438–453.

KEMENY, J. G. AND SNELL, J. L. 1976. *Finite Markov chains*. Undergraduate texts in mathematics. Springer, New York. Reprint of the 1960 ed. published by Van Nostrand, Princeton, N.J., in the University series in undergraduate mathematics.

KHANNA, S., SUDAN, M., TREVISAN, L., AND WILLIAMSON, D. 2001. The approximability of constraint satisfaction problems. *SIAM Journal on Computing 30,* 6, 1863–1920.

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

KINGSFORD, C. L., CHAZELLE, B., AND SINGH, M. 2005. Solving and analyzing side-chain positioning problems using linear and integer programming. *Bioinformatics 21,* 7, 1028–1039.

KOLAITIS, P. G. AND VARDI, M. Y. 2000. Conjunctive-Query Containment and Constraint Satisfaction. *Journal of Computer and System Sciences 61,* 2, 302–332.

KOLMOGOROV, V. 2013. The power of linear programming for finite-valued CSPs: a constructive characterization. In *Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP'13)*. Lecture Notes in Computer Science Series, vol. 7965. Springer, 625–636.

KOLMOGOROV, V., THAPPER, J., AND ŽIVNÝ, S. 2015. The power of linear programming for general-valued CSPs. *SIAM Journal on Computing 44,* 1, 1–36.

KOLMOGOROV, V. AND ŽIVNÝ, S. 2013. The complexity of conservative valued CSPs. *Journal of the ACM 60,* 2. Article No. 10.

KOSTER, A., VAN HOESEL, S. P., AND KOLEN, A. W. 1998. The partial constraint satisfaction problem: Facets and lifting theorems. *Operations Research Letters 23,* 3–5, 89–97.

KUN, G., O'DONNELL, R., TAMAKI, S., YOSHIDA, Y., AND ZHOU, Y. 2012. Linear programming, width-1 CSPs, and robust satisfaction. In *Proceedings of the 3rd Innovations in Theoretical Computer Science (ITCS'12)*. ACM, 484–495.

LAURITZEN, S. L. 1996. *Graphical Models*. Oxford University Press.

MARX, D. 2013. Tractable hypergraph properties for constraint satisfaction and conjunctive queries. *Journal of the ACM 60,* 6. Article No. 42.

MONTANARI, U. 1974. Networks of Constraints: Fundamental properties and applications to picture processing. *Information Sciences 7*, 95–132.

PAPADIMITRIOU, C. H. AND YANNAKAKIS, M. 1984. The complexity of facets (and some facets of complexity). *Journal of Computer System Sciences 28,* 2, 244–259.

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

RAGHAVENDRA, P. 2009. Approximating NP-hard problems: Efficient algorithms and their limits. Ph.D. thesis, University of Washington.

SCHAEFER, T. J. 1978. The Complexity of Satisfiability Problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC'78)*. ACM, 216–226.

SCHRIJVER, A. 1986. *Theory of linear and integer programming*. John Wiley & Sons, Inc.

SHERALI, H. D. AND ADAMS, W. P. 1990. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal of Discrete Mathematics 3,* 3, 411–430.

SHLEZINGER, M. I. 1976. Syntactic analysis of two-dimensional visual signals in noisy conditions. *Cybernetics and Systems Analysis 12,* 4, 612–628. Translation from Russian.

STANLEY, R. P. 2007. An introduction to hyperplane arrangements. In *Geometric Combinatorics*, E. Miller, V. Reiner, and B. Sturmfels, Eds. IAS/Park City Mathematics Series, vol. 13. American Mathematical Society, Providence, RI, 389–496.

TAKHANOV, R. 2010a. A Dichotomy Theorem for the General Minimum Cost Homomorphism Problem. In *Proceedings of the 27th International Symposium on Theoretical Aspects of Computer Science (STACS'10)*. 657–668.

TAKHANOV, R. 2010b. Extensions of the Minimum Cost Homomorphism Problem. In *Proceedings of the 16th International Computing and Combinatorics Conference (COCOON'10)*. Lecture Notes in Computer Science Series, vol. 6196. Springer, 328–337.

THAPPER, J. AND ŽIVNÝ, S. 2012. The power of linear programming for valued CSPs. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'12)*. IEEE, 669–678.

THAPPER, J. AND ŽIVNÝ, S. 2013. The complexity of finite-valued CSPs. In *Proceedings of the 45th ACM Symposium on the Theory of Computing (STOC'13)*. ACM, 695–704. Full version available at arXiv:1210.2977v3.

UPPMAN, H. 2013. The Complexity of Three-Element Min-Sol and Conservative Min-Cost-Hom. In *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP'13)*. Lecture Notes in Computer Science Series, vol. 7965. Springer, 804–815.

UPPMAN, H. 2014. Computational Complexity of the Extended Minimum Cost Homomorphism Problem on Three-Element Domains. In *Proceedings of the 31st International Symposium on Theoretical Aspects of Computer Science (STACS'14)*. Vol. 25. 651–662.

WAINWRIGHT, M., JAAKKOLA, T., AND WILLSKY, A. 2005. MAP estimation via agreement on trees: message passing and linear programming. *IEEE Transactions on Information Theory 51,* 11, 3697–3717.

WAINWRIGHT, M. J. AND JORDAN, M. I. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning 1,* 1-2, 1–305.

ŽIVNÝ, S. 2012. *The complexity of valued constraint satisfaction problems*. Cognitive Technologies. Springer.

Received May 2014; revised May 2015; accepted November 2015

A. INFINITE CONSTRAINT LANGUAGES

The main result of this article, Theorem 3.4, establishes a dichotomy for finite-valued constraint languages of *finite* size. The finiteness is important when passing from the primal to the dual linear programme, and thus could be considered an artefact of our proof techniques. However, our algorithm, the BLP, only depends on the instance and not in some exponential way on the constraint language. We are therefore able to extend our results to finite-valued constraint languages of *infinite* size; that is, the setting when the cost functions are still represented extensionally.

To state the dichotomy for infinite constraint languages, we need to allow the fractional polymorphisms to take on *real* values. Hence for the rest of this section, an m-ary fractional operation is a function $\omega : \mathcal{O}_D^{(m)} \to \mathbb{R}_{\geq 0}$, $\|\omega\|_1 = 1$. Fractional polymorphisms are defined by inequality (1), using real-valued fractional operations. Note however that the constraint languages, although infinite, still consist of rational-valued cost functions only.

THEOREM A.1. *Let D be an arbitrary finite set, let Γ be a (possibly infinite) constraint language defined on D, and let Γ' be a core of Γ.*

— *Either* Γ *has a binary symmetric real-valued fractional polymorphism and BLP solves VCSP*(Γ)*;*
— *or (MC) holds for* Γ'_c *and VCSP*(Γ) *is NP-hard.*

It follows from [Thapper and Živný 2012; Kolmogorov 2013; Kolmogorov et al. 2015] that for a (possibly infinite) constraint language Γ with a binary symmetric real-valued fractional polymorphism, Γ is not only tractable but also *globally tractable*. Conversely, we need to show that if Γ does not have a binary symmetric fractional polymorphism, then the same holds for some finite subset of Γ. We can then apply Theorem 1.1 to conclude that Γ is NP-hard. This direction is a consequence of the following result, when Ω is taken as the set of symmetric m-ary operations on D. A similar result for countably infinite constraint languages is proved in [Kolmogorov et al. 2015].

LEMMA A.2. *Let* Γ *be a (possibly infinite) constraint language. Let* $\Omega \subseteq \mathcal{O}_D^{(m)}$ *and assume that every finite subset* $\Gamma' \subseteq \Gamma$ *has a fractional polymorphism with support in* Ω*. Then* Γ *has a fractional polymorphism with support in* Ω*.*

PROOF. Note that $|\Omega|$ is finite and let $n = |\Omega|$. Let Y be the set of fractional operations $\omega : \Omega \to \mathbb{R}_{\geq 0}$, $\|\omega\|_1 = 1$. Then Y is a compact set in $\mathbb{R}^n$. Assume, for the sake of contradiction, that Γ does not have a fractional polymorphism with support in Ω. Then, for every $y \in Y$, there is some $f_y \in \Gamma$ and $\bar{x}^1, \dots, \bar{x}^k \in D^{ar(f_y)}$ such that

$$\sum_{g \in \Omega} y(g) f_y(g(\bar{x}^1, \dots, \bar{x}^k)) > f_y^m(\bar{x}^1, \dots, \bar{x}^k).$$

Furthermore, this inequality holds in an open neighbourhood $U_y \subseteq Y$ of y, so $\{U_y\}_{y \in Y}$ is an open cover of Y. Since every open cover of a compact set has a finite subcover, this provides us with a finite subset of Γ that does not have a fractional polymorphism with support in Ω. This is a contradiction, hence Γ must have a fractional polymorphism with support in Ω. □

The proof of Lemma A.2 relies on real-valued fractional polymorphisms, and the obvious question to ask is then whether real values are necessary for Theorem A.1 to hold, or whether it is an artefact of our proof techniques. Perhaps unsurprisingly, we can demonstrate that real-valued fractional polymorphisms are necessary in *some* cases. The following construction is based on a language from [Huber et al. 2014], where it was used for a different result; we will use the same notation as in [Huber et al. 2014].

Let $D = \{-1, 0, 1\}$ and fix the partial order $-1 > 0 < 1$ on D. For $a \in \{-1, 1\}$, define binary operations $\vee_a$ and $\wedge_0$ as follows:

$$1 \vee_a -1 = -1 \vee_a 1 = a \text{ and } x \vee_a y = \max(x, y) \text{ wrt the above order if } \{x, y\} \neq \{-1, 1\};$$

$$1 \wedge_0 -1 = -1 \wedge_0 1 = 0 \text{ and } x \wedge_0 y = \min(x, y) \text{ wrt the above order if } \{x, y\} \neq \{-1, 1\}.$$

Let $\alpha \in (0, 1]$ be an arbitrary real constant, and define the fractional operation ω as follows: $\omega(\wedge_0) = 1/2$, $\omega(\vee_0) = \alpha/2$, and $\omega(\vee_1) = (1 - \alpha)/2$. A cost function is called *α-bisubmodular* if it admits the fractional polymorphism ω.

For an arbitrary rational $\alpha \in (0, 1]$, write $\alpha = p/q$ with $p, q \geq 1$, p and q coprime. Define the unary cost functions $e, u_\alpha, v_\alpha : D \to \mathbb{Q}$ and the binary cost function $f : D^2 \to \mathbb{Q}$ as follows:

	-1	0	1
e	1	0	1
u_α	$p+q$	q	0
v_α	0	p	$p+q$

f	-1	0	1
-1	3	2	1
0	2	0	0
1	1	0	0

Note that u_α and v_α are uniquely defined given α.

PROPOSITION A.3. *Fix an arbitrary irrational value $x \in (0,1)$ and define*

$$\Gamma_x := \{v_\alpha \mid \alpha \in \mathbb{Q} \cap (0,x)\} \cup \{u_\alpha \mid \alpha \in \mathbb{Q} \cap (x,1]\} \cup \{e, f\}.$$

(*1*) *Γ_x is x-bisubmodular and BLP solves VCSP(Γ_x), but*
(*2*) *Γ_x does not admit any rational-valued binary symmetric fractional polymorphism.*

PROOF. We first show part (1). It follows from the definition that unary function u is x-bisubmodular if, and only if,

$$(1+x) \cdot u(0) \leq u(-1) + x \cdot u(1). \tag{23}$$

For the cost function e, condition (23) becomes $(1+x)\cdot 0 \leq 1+x\cdot 1$, so e is x-bisubmodular. For the cost function u_α, since $x < \alpha = p/q$, we have $(1+x)q < p+q$, so (23) holds and u_α is x-bisubmodular. Similarly, one shows that v_α is x-bisubmodular for $x > \alpha$.

It remains to show that f is x-bisubmodular. By an alternative characterisation [Huber et al. 2014, Proposition 2], f is x-bisubmodular if and only if (i) the unary cost functions obtained from f by fixing one argument are x-bisubmodular, and (ii) f is submodular in every orthant; this means that for every $\bar{c} \in \{-1,1\}^2$, the fractional polymorphism inequality (1) holds for x-bisubmodularity for all $\bar{a}^1, \bar{a}^2 \in D^2$ with $\bar{a}^1, \bar{a}^2 \leq \bar{c}$ (here we used the componentwise order on D).

First we verify that the unary cost functions $f(-1,x)$, $f(0,x)$, and $f(1,x)$ are x-bisubmodular. The inequality (23) becomes $(1+x)\cdot 2 \leq 3+x$, $(1+x)\cdot 0 \leq 2 + 0 \cdot x$, and $(1+x)\cdot 0 \leq 1+x$, respectively. Since $x \in (0,1)$, all three inequalities hold, so all three cost functions are x-bisubmodular. By symmetry, $f(x,-1)$, $f(x,0)$, and $f(x,1)$ are also x-bisubmodular.

Next, we verify that f is submodular in every orthant:

— f is constant 0 and hence trivially submodular in the orthant $(1,1)$.
— In the orthant $(-1,-1)$, the only nontrivial case to verify is $\bar{a}^1 = (0,-1)$ and $\bar{a}^2 = (-1,0)$. We have, after multiplying by 2, $f(0,-1) + f(-1,0) = 2 + 2 \geq 1 \cdot f(0,0) + x \cdot f(-1,-1) + (1-x)\cdot f_\alpha(-1,-1) = f(-1,-1) = 3$, which holds true. Hence, f is submodular in the orthant $(-1,-1)$.
— Finally, the two cases $\bar{c} = (1,-1)$ and $\bar{c} = (-1,1)$ are symmetric. In the orthant $(1,-1)$, the only nontrivial case to verify is $\bar{a}^1 = (0,-1)$ and $\bar{a}^2 = (1,0)$. Here, we have $f(0,-1) + f_\alpha(1,0) = 2 + 0 \geq f(0,0) + xf(1,-1) + (1-x)\cdot f(1,-1) = f_\alpha(1,-1) = 1$, which holds true. Hence, f is submodular in the orthants $(1,-1)$ and $(-1,1)$.

We conclude that Γ_x is x-bisubmodular, and hence solved by the BLP relaxation.

We now show part (2). Let ω be an arbitrary binary symmetric fractional polymorphism of Γ_x. For $a \in \{-1,0,1\}$, define $w_a = \sum_{g \in \mathcal{O}_D^{(2)} \mid g(-1,1)=a} \omega(g)$. Clearly, $0 \leq w_a \leq 1$ and $w_{-1} + w_0 + w_1 = 1$. It suffices to show that at least one of the w_a is irrational, $a \in \{-1,0,1\}$, which implies the existence of a binary operation g with $\omega(g) \notin \mathbb{Q}$.

Let $\alpha = p/q$ with $\alpha < x$. Applying the fractional polymorphism inequality (1) to $v_\alpha \in \Gamma_x$, we have $(p+q)/2 = (v_\alpha(-1) + v_\alpha(1))/2 \geq w_0 v_\alpha(0) + w_1 v_\alpha(1) + w_{-1} v_\alpha(-1) = w_0 p + w_1(p+q) + w_{-1}0$, which is equivalent to $w_0 \leq (1+1/\alpha)(1/2 - w_1)$. Since this inequality holds for all rational $\alpha < x$, we have, in the limit as $\alpha \to x$ from below,

$$w_0 \leq (1+1/x)(1/2 - w_1). \tag{24}$$

A similar argument for the cost function $u_\alpha \in \Gamma_x$, for $\alpha > x$, leads to the inequality $w_0 \leq (1+\alpha)(1/2 - w_{-1})$ and, in the limit as $\alpha \to x$ from above,

$$w_0 \leq (1+x)(1/2 - w_{-1}). \tag{25}$$

Add x times the inequality (24) to the inequality (25) to obtain $(1+x)w_0 \leq (1+x)(1-w_1-w_{-1})$. Since $w_{-1}+w_0+w_1=1$, this inequality must hold with equality, and hence the inequalities (24) and (25) can be replaced by the equalities $w_0 = (1+1/x)(1/2-w_1)$ and $w_0 = (1+x)(1/2-w_{-1})$. Since x is irrational, it follows that either $w_0 = 0$ and $w_{-1} = w_1 = 1/2$, or at least one of w_{-1}, w_0, and w_1 is irrational. We demonstrate that the latter holds by showing that $w_{-1} < 1/2$. Applying the fractional polymorphism inequality (1) to $f \in \Gamma_x$, we have $1 = (1+1)/2 = (f(-1,1)+f(1,-1))/2 \geq w_{-1}f(-1,-1)+w_0 f(0,0)+w_1 f(1,1) = w_{-1} \cdot 3$, which gives $w_{-1} \leq 1/3 < 1/2$, and the claim follows. □