2015 IEEE 56th Annual Symposium on Foundations of Computer Science

The Complexity of General-Valued CSPs

Vladimir Kolmogorov*, Andrei Krokhin† and Michal Rolínek‡
*IST Austria, Klosterneuburg, Austria. Email: vnk@ist.ac.at
†Durham University, UK. Email: andrei.krokhin@durham.ac.uk
‡IST Austria, Klosterneuburg, Austria. Email: michal.rolinek@ist.ac.at

Abstract

An instance of the Valued Constraint Satisfaction Problem (VCSP) is given by a finite set of variables, a finite domain of labels, and a sum of functions, each function depending on a subset of the variables. Each function can take finite values specifying costs of assignments of labels to its variables or the infinite value, which indicates an infeasible assignment. The goal is to find an assignment of labels to the variables that minimizes the sum.

We study, assuming that P $\neq$ NP, how the complexity of this very general problem depends on the set of functions allowed in the instances, the so-called constraint language. The case when all allowed functions take values in $\{0, \infty\}$ corresponds to ordinary CSPs, where one deals only with the feasibility issue and there is no optimization. This case is the subject of the Algebraic CSP Dichotomy Conjecture predicting for which constraint languages CSPs are tractable (i.e. solvable in polynomial time) and for which NP-hard. The case when all allowed functions take only finite values corresponds to finite-valued CSP, where the feasibility aspect is trivial and one deals only with the optimization issue. The complexity of finite-valued CSPs was fully classified by Thapper and Živný.

An algebraic necessary condition for tractability of a general-valued CSP with a fixed constraint language was recently given by Kozik and Ochremiak. As our main result, we prove that if a constraint language satisfies this algebraic necessary condition, and the feasibility CSP (i.e. the problem of deciding whether a given instance has a feasible solution) corresponding to the VCSP with this language is tractable, then the VCSP is tractable. The algorithm is a simple combination of the assumed algorithm for the feasibility CSP and the standard LP relaxation. As a corollary, we obtain that a dichotomy for ordinary CSPs would imply a dichotomy for general-valued CSPs.

Keywords

Valued constraint satisfaction problem; complexity; dichotomy; fractional polymorphism.

I. Introduction

Computational problems from many different areas involve finding an assignment of labels to a set of variables, where that assignment must satisfy some specified feasibility conditions and/or optimize some specified objective function. In many such problems, the feasibility conditions are local and also the objective function can be represented as a sum of functions, each of which depends on some subset of the variables. Examples include: Gibbs energy minimization, Markov Random Fields (MRF), Conditional Random Fields (CRF), Min-Sum Problems, Minimum Cost Homomorphism, Constraint Optimization Problems (COP) and Valued Constraint Satisfaction Problems (VCSP) [1], [2], [3], [4], [5], [6].

The constraint satisfaction problem provides a common framework for many theoretical and practical problems in computer science [7], [5]. An instance of the *constraint satisfaction problem* (CSP) consists of a collection of variables that must be assigned labels from a given domain subject to specified constraints [8]. The CSP is equivalent to the problem of evaluating conjunctive queries on databases [9], and to the homomorphism problem for relational structures [10]. The CSP deals only with the feasibility issue: can all constraints be satisfied simultaneously?

There are several natural optimization versions of the CSP: Max CSP (or Min CSP) where the goal is to find the assignment maximizing the number of satisfied constraints (or minimizing the number of unsatisfied constraints) [11], [7], [12], [13], problems like Max-Ones and Min-Hom where the constraints must be satisfied and some additional function of the assignment is to be optimized [7], [14], [15], and, the most general version,

0272-5428/15 $31.00 © 2015 IEEE
DOI 10.1109/FOCS.2015.80

IEEE computer society

valued CSP or VCSP (also known as soft CSP), where each combination of values for variables in a constraint has a cost and the goal is to minimize the aggregate cost [16], [17], [18], [19]. Thus, an instance of the VCSP amounts to minimizing a sum of functions, each depending on a subset of variables. By using infinite costs to indicate infeasible combinations, VCSP can model both feasibility and optimization aspects and so considerably generalises all the problems mentioned above [16], [17], [3]. There is much activity and very strong results concerning various aspects of approximability of (V)CSPs (see e.g. [20], [21], [22], [7], [23], [24], [25], [26] for a small sample), but in this paper we focus on solving VCSPs to optimality.

We assume throughout the paper that P $\neq$ NP. Since all the above problems are NP-hard in full generality, a major line of research in CSP tries to identify the tractable cases of such problems (see books/surveys [27], [7], [28], [3]), the primary motivation being the general picture rather than specific applications. The two main ingredients of a constraint are (a) variables to which it is applied and (b) relations/functions specifying the allowed combinations of values or the costs for all combinations. Therefore, the main types of restrictions on CSP are (a) *structural* where the hypergraph formed by sets of variables appearing in individual constraints is restricted [29], [30], and (b) *language-based* where the constraint language, i.e. the set of relations/functions that can appear in constraints, is fixed (see, e.g. [31], [27], [7], [10], [19]). The ultimate sort of results in these directions are *dichotomy* results, pioneered by [32], which characterise the tractable restrictions and show that the rest are as hard as the corresponding general problem (which cannot generally be taken for granted). The language-based direction is considerably more active than the structural one, there are many partial language-based dichotomy results, e.g. [33], [34], [17], [7], [12], [13], [35], [15], but many central questions are still open. In this paper, we study VCSPs with a fixed constraint language on a finite domain, and all further discussion concerns only such CSPs and VCSPs.

Related Work. The CSP Dichotomy Conjecture, stating that each CSP is either tractable or NP-hard, was first formulated by Feder and Vardi [10]. The universal-algebraic approach to this problem was discovered in [31], [36], [37], and the precise boundary between the tractable cases and NP-hard cases was conjectured in algebraic terms in [31], in what is now known as the Algebraic CSP Dichotomy Conjecture (see Conjecture 16). The hardness part was proved in [31], and it is the tractability part that is the essence of the conjecture. This conjecture is still open in full generality and is the object of much investigation, e.g. [38], [39], [40], [41], [42], [31], [34], [27], [43]. It is known to hold for domains with at most 3 elements [33], [32], for smooth digraphs [42], and for the case when all unary relations are available [41], [34]. The main two polynomial-time algorithms used for CSPs are based one on local consistency ("bounded width") and the other on compact representation of solution sets ("few subpowers"), and their applicability (in pure form) is fully characterized in [38], [40] and [43], respectively.

At the opposite (to CSP) end of the VCSP spectrum are the finite-valued CSPs, in which functions do not take infinite values. In such VCSPs, the feasibility aspect is trivial, and one has to deal only with the optimization issue. One polynomial-time algorithm that solves tractable finite-valued CSPs is based on the so-called basic linear programming (BLP) relaxation, and its applicability (also for the general-valued case) was fully characterized in [18] (see Theorem 17). The complexity of finite-valued CSPs was completely classified in [19], where it is shown that all finite-valued CSPs not solvable by BLP are NP-hard.

For general-valued CSPs, full classifications are known for the Boolean case (i.e., when the domain is two-element) [17] and also for the case when all 0-1-valued unary cost functions are available [35]. The algebraic approach to the CSP was extended to VCSPs in [16], [44], [17], [45], and was also key to much progress. An algebraic necessary condition for a VCSP to be tractable was recently proved by Kozik and Ochremiak in [45], where this condition was also conjectured to be sufficient (see Theorem 14 and Conjecture 15 below). This conjecture can be called the Algebraic VCSP Dichotomy Conjecture, and it is a generalization of the corresponding conjecture for CSP. A large family of VCSPs satisfying the necessary condition from [45] has recently been shown tractable via a low-level Sherali-Adams hierarchy relaxation [46].

Our proof uses the technique of "lifting a language" introduced in [47].

Our Contribution. We completely classify the complexity of VCSPs with a fixed constraint language modulo the complexity of CSPs (see Theorem 21). Clearly, for a VCSP to be tractable, it is necessary that the correspond-

ing feasibility CSP is tractable. We prove that any VCSP satisfying this necessary condition and the necessary condition of Kozik and Ochremiak is tractable. The polynomial-time algorithm that solves such VCSP is a simple combination of the (assumed) polynomial-time algorithm for the feasibility CSP and BLP (see Theorem 22). Thus, our dichotomy theorem generalizes the dichotomy for finite-valued CSPs from [19], and, with the help of the CSP tractability result from [40], it also implies the tractability of VCSPs shown tractable in [46].

Our result says that any dichotomy for CSP (not necessarily the one predicted by the Algebraic CSP Dichotomy Conjecture) will imply a dichotomy for VCSP. However, if the Algebraic CSP Dichotomy Conjecture holds then the necessary algebraic condition of Kozik and Ochremiak guarantees tractability of the feasibility CSP (see [45]), implying that this algebraic condition alone is necessary and sufficient for tractability of a VCSP, and also that all the intractable VCSPs are NP-hard. In particular, the Algebraic CSP Dichotomy Conjecture implies the Algebraic VCSP Dichotomy Conjecture.

II. Preliminaries

A. Valued Constraint Satisfaction Problems

Throughout the paper, let D be a fixed finite set and let $\overline{\mathbb{Q}} = \mathbb{Q} \cup \{\infty\}$ denote the set of rational numbers with (positive) infinity.

Definition 1: We denote the set of all functions $f : D^n \to \overline{\mathbb{Q}}$ by $\mathcal{F}_D^{(n)}$ and let $\mathcal{F}_D = \bigcup_{n \geq 1} \mathcal{F}_D^{(n)}$. We will often call the functions in $\mathcal{F}_D$ *cost functions* over D. For every cost function $f \in \mathcal{F}_D^{(n)}$, let $\operatorname{dom} f = \{x \mid f(x) < \infty\}$. Note that $\operatorname{dom} f$ can be considered both as an n-ary relation and as a n-ary function such that $\operatorname{dom} f(x) = 0$ if and only if $f(x)$ is finite.

We will call the set D the *domain*, elements of D *labels* (for variables), and say that the cost functions in $\mathcal{F}_D$ take *values*. Note that in some papers on VCSP, e.g. [16], [46], cost functions are called weighted relations.

Definition 2: An instance of the *valued constraint satisfaction problem* (VCSP) is a function from D^V to $\overline{\mathbb{Q}}$ given by

$$f_{\mathcal{I}}(x) = \sum_{t \in T} f_t(x_{v(t,1)}, \ldots, x_{v(t,n_t)}), \tag{1}$$

where V is a finite set of variables, T is a finite set of constraints, each constraint is specified by a cost function f_t of arity n_t and indices $v(t,k)$, $k = 1, \ldots, n_t$. The goal is to find an *assignment* (or *labeling*) $x \in D^V$ that minimizes $f_{\mathcal{I}}$. The value of an optimal assignment is denoted by $\operatorname{Opt}(\mathcal{I})$.

Definition 3: Any set $\Gamma \subseteq \mathcal{F}_D$ is called a *valued constraint language* over D, or simply a *language*. We will denote by $\operatorname{VCSP}(\Gamma)$ the class of all VCSP instances in which the constraint functions f_t are all contained in Γ.

This framework subsumes many other frameworks studied earlier and captures many specific well-known problems, including k-SAT, GRAPH k-COLOURING, MAX CUT, MIN VERTEX COVER and others (see [3]). Note that if every function in Γ takes values in $\{0, \infty\}$ (such functions are often called *crisp*) then $\operatorname{VCSP}(\Gamma)$ is a pure feasibility problem, commonly known as $\operatorname{CSP}(\Gamma)$.

The main goal of our line of research is to classify the complexity of problems $\operatorname{VCSP}(\Gamma)$. Problems $\operatorname{CSP}(\Gamma)$ and $\operatorname{VCSP}(\Gamma)$ are called tractable if, for each finite $\Gamma' \subseteq \Gamma$, $\operatorname{VCSP}(\Gamma')$ is tractable. Also, $\operatorname{VCSP}(\Gamma)$ is called NP-hard if, for some finite $\Gamma' \subseteq \Gamma$, $\operatorname{VCSP}(\Gamma')$ is NP-hard. One advantage of defining tractability in terms of finite subsets is that the tractability of a valued constraint language is independent of whether the cost functions are represented explicitly (say, via full tables of values, or via tables for the finite-valued parts) or implicitly (via oracles). Following [31], we say that $\operatorname{VCSP}(\Gamma)$ is *globally tractable* there is a polynomial-time algorithm solving $\operatorname{VCSP}(\Gamma)$, assuming all functions in instances are given by full tables of values. For CSPs, there is no example of $\operatorname{CSP}(\Gamma)$ that is tractable, but not globally tractable, and it is conjectured in [31] that no such $\operatorname{CSP}(\Gamma)$ exists.

B. Polymorphisms, Expressibility, Cores

Let $\mathcal{O}_D^{(m)}$ denote the set of all operations $g : D^m \to D$ and let $\mathcal{O}_D = \cup_{m \geq 1} \mathcal{O}_D^{(m)}$. When D is clear from the context, we will sometimes write simply $\mathcal{O}^{(m)}$ and $\mathcal{O}$.

Any language Γ defined on D can be associated with a set of operations on D, known as the polymorphisms of Γ, which allow one to combine (often in a useful way) several feasible assignments into a new one.

Definition 4: An operation $g \in \mathcal{O}_D^{(m)}$ is a *polymorphism* of a cost function $f \in \mathcal{F}_D$ if, for any $x^1, x^2, \ldots, x^m \in \operatorname{dom} f$, we have that $g(x^1, x^2, \ldots, x^m) \in \operatorname{dom} f$ where g is applied component-wise.

For any valued constraint language Γ over a set D, we denote by $\operatorname{Pol}(\Gamma)$ the set of all operations on D which are polymorphisms of every $f \in \Gamma$.

Example 5: Let $f \in \mathcal{F}_{\{0,1\}}^{(n)}$ be such that $f(1, \ldots, 1, 0) = \infty$ and $f(a_1, \ldots, a_n) = 0$ otherwise. It corresponds to the Horn clause $(x_1 \vee \ldots \vee x_{n-1} \vee \overline{x_n})$. Then it is well known and easy to see that the binary operation $\min \in \mathcal{O}_{\{0,1\}}$ is a polymorphism of f.

Clearly, if g is a polymorphism of a cost function f, then g is also a polymorphism of $\operatorname{dom} f$. For $\{0, \infty\}$-valued functions, which naturally correspond to relations, the notion of a polymorphism defined above coincides with the standard notion of a polymorphism for relations. Note that the projections (aka dictators), i.e. operations of the form $e_n^i(x_1, \ldots, x_n) = x_i$, are polymorphisms of all valued constraint languages. Polymorphisms play the key role important role in the algebraic approach to the CSP, but, for VCSPs, more general constructs are necessary, which we now define.

Definition 6: An m-ary *fractional operation* ω on D is a probability distribution on $\mathcal{O}_D^{(m)}$. The support of ω is defined as $\operatorname{supp}(\omega) = \{g \in \mathcal{O}_D^{(m)} \mid \omega(g) > 0\}$.

Definition 7: A m-ary fractional operation ω on D is said to be a *fractional polymorphism* of a cost function $f \in \mathcal{F}_D$ if, for any $x^1, x^2, \ldots, x^m \in \operatorname{dom} f$, we have

$$\sum_{g \in \operatorname{supp}(\omega)} \omega(g) f(g(x^1, \ldots, x^m)) \leq \frac{1}{m}(f(x^1) + \ldots + f(x^m)). \tag{2}$$

For a constraint language Γ, $\operatorname{fPol}(\Gamma)$ will denote the set of all fractional operations that are fractional polymorphisms of each function in Γ. Also, let $\operatorname{fPol}^+(\Gamma) = \{g \in \mathcal{O}_D \mid g \in \operatorname{supp}(\omega), \omega \in \operatorname{fPol}(\Gamma)\}$.

The intuition behind the notion of fractional polymorphism is that it allows one to combine several feasible assignments into new feasible assignments so that the expected value of a new assignment (non-strictly) improves the average value of the original assignments.

Example 8: If ω is a binary fractional operation on $D = \{0, 1\}$ such that $\omega(\min) = \omega(\max) = 1/2$, then it is well-known and easy to check that the finite-valued functions with fractional polymorphism ω are the submodular functions. Moreover, functions with this fractional polymorphism that are not necessarily finite-valued precisely correspond to submodular functions defined on a ring family.

More examples of fractional polymorphisms can be found in [3], [18], [19].

We remark that, in some papers (e.g., in [16], [3]), fractional polymorphisms (and closely related objects called weighted polymorphisms) are defined as rational-valued functions, which is sufficient for analysing the complexity of VCSPs with finite constraint languages. However, real-valued fractional polymorphisms are necessary to analyse infinite constraint languages [48], [45], [19].

The key observation in the algebraic approach to (V)CSP is that neither the complexity nor the algebraic properties of a language Γ change when functions "expressible" from Γ in a certain way are added to it.

Definition 9: For a constraint language Γ, let $\langle\Gamma\rangle$ denote the set of all functions $f(x_1, \ldots, x_k)$ such that, for some instance $\mathcal{I}$ of $\operatorname{VCSP}(\Gamma)$ with objective function $f_{\mathcal{I}}(x_1, \ldots, x_k, x_{k+1}, \ldots, x_n)$, we have

$$f(x_1, \ldots, x_k) = \min_{x_{k+1}, \ldots, x_n} f_{\mathcal{I}}(x_1, \ldots, x_k, x_{k+1}, \ldots, x_n).$$

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

Lemma 10 ([44], [17]): Let $f \in \langle\Gamma\rangle$. Then

1) if $\omega \in \operatorname{fPol}(\Gamma)$ then ω is a fractional polymorphism of f and of $\operatorname{dom} f$;
2) $\operatorname{VCSP}(\Gamma)$ is tractable if and only if $\operatorname{VCSP}(\Gamma \cup \{f, \operatorname{dom} f\})$ is tractable;
3) $\operatorname{VCSP}(\Gamma)$ is NP-hard if and only if $\operatorname{VCSP}(\Gamma \cup \{f, \operatorname{dom} f\})$ is NP-hard.

The dichotomy problem for VCSPs can be reduced to a class of constraint languages called rigid cores, defined below. Apart from reducing the cases that need to be considered, this reduction enabled the use of much more powerful results from universal algebra than what can be done without this restriction (see, e.g. [45]).

For a subset $D' \subseteq D$, let $u_{D'}$ be the function defined as follows: $u_{D'}(d) = 0$ if $d \in D'$ and $u_{D'}(d) = \infty$ otherwise. We write u_d for $u_{\{d\}}$. Let $\mathcal{C}_D = \{\{u_d\} \mid d \in D\}$.

Lemma 11 ([45]): For any valued constraint language Γ' on a finite set D', there is a subset $D \subseteq D'$ and a valued constraint language Γ on D such that $\mathcal{C}_D \subseteq \Gamma$ and the problems $\mathrm{VCSP}(\Gamma')$ and $\mathrm{VCSP}(\Gamma)$ are polynomial-time equivalent.

This language Γ is called the *rigid core* of Γ', and it can be obtained from Γ' as follows. Let g' be a unary operation on D' with minimum $|g'(D')|$ among all unary operations $g' \in \mathrm{fPol}^+(\Gamma')$. Then D is set to be $g'(D')$ and Γ is set to be $\{f|_D : f \in \Gamma'\} \cup \mathcal{C}_D$. Thus, the intuition behind moving to the rigid core is that (a) one removes labels from the domain that can always be (uniformly) replaced in any solution to an instance without increasing its value, and (b) one allows constraints of the form u_d that can be used to fix labels for variables, leading to applicability of more powerful algebraic results.

C. Cyclic and symmetric operations

Several types of operations play a special role in the algebraic approach to (V)CSP.

Definition 12: An operation $g \in \mathcal{O}_D^{(m)}$, $m \geq 2$, is called

- *idempotent* if $g(x, \dots, x) = x$ for all $x \in D$;
- *Taylor* if, for each $1 \leq i \leq m$, it satisfies an identity of the form $g(\triangle_1, \triangle_2, \dots, \triangle_m) = g(\square_1, \square_2, \dots, \square_m)$ where all $\triangle_j, \square_j$ are in $\{x, y\}$ and $\triangle_i \neq \square_i$.
- *cyclic* if $g(x_1, x_2, \dots, x_m) = g(x_2, \dots, x_m, x_1)$ for all $x_1, \dots, x_m \in D$;
- *symmetric* if $g(x_1, x_2, \dots, x_m) = g(x_{\pi(1)}, x_{\pi(2)}, \dots, x_{\pi(m)})$ for all $x_1, \dots, x_m \in D$, and any permutation π on $[m]$.

A fractional operation ω is said to be idempotent/cyclic/symmetric if all operations in $\mathrm{supp}(\omega)$ have the corresponding property.

It is well known and easy to see that all polymorphisms and fractional polymorphisms of a rigid core are idempotent.

The following lemma is contained in the proof of Theorem 50 in [45].

Lemma 13: Let Γ be a rigid core on a set D. Then the following are equivalent:

1) $\mathrm{fPol}^+(\Gamma)$ contains a Taylor operation of arity at least 2;
2) Γ has a cyclic fractional polymorphism of (some) arity at least 2;
3) Γ has a cyclic fractional polymorphism of every prime arity $p > |D|$.

The following theorem is Corollary 51 from [45].

Theorem 14 ([45]): Let Γ be a valued constraint language that is a rigid core. If $\mathrm{fPol}^+(\Gamma)$ does not contain a Taylor operation then $\mathrm{VCSP}(\Gamma)$ is NP-hard.

Kozik and Ochremiak state a conjecture (which they attribute to L. Barto) that the above theorem describes all NP-hard valued constraint languages, and all other languages are tractable. Using Lemma 13, we restate the original conjecture via cyclic fractional polymorphisms.

Conjecture 15 ([45]): Let Γ be a valued constraint language that is a rigid core. If Γ has a cyclic fractional polymorphism of arity at least 2, then $\mathrm{VCSP}(\Gamma)$ is tractable.

Note that, for a finite core Γ (but with fixed D), the above condition can be checked in polynomial time. Indeed, if $p > |D|$ is some fixed prime number, then it is sufficient to check for a cyclic fractional polymorphism of arity p. Such polymorphisms, by definition, are solutions to a system of linear inequalities. Since the number of cyclic operations of arity p on D is constant, the system will have size polynomial in Γ and its feasibility can be decided by linear programming.

For the case when (possibly infinite) Γ consists of $\{0, \infty\}$-valued functions, $\mathrm{VCSP}(\Gamma)$ is actually a CSP. For such Γ, any probability distribution on polymorphisms (of the same arity) is a fractional polymorphism.

Then a theorem and a conjecture (the latter now known as the Algebraic CSP Dichotomy Conjecture) equivalent to Theorem 14 and Conjecture 15 were given in [31]. One of several equivalent forms of the Algebraic CSP Dichotomy Conjecture is as follows.

Conjecture 16 ([31], [39]): Let Γ be a valued constraint language that is a rigid core and that consists of $\{0,\infty\}$-valued functions. If Γ has a cyclic polymorphism of arity at least 2, then $\mathrm{VCSP}(\Gamma)$ is tractable. Otherwise, $\mathrm{VCSP}(\Gamma)$ is NP-hard.

In view of this, it is natural to call Conjecture 15 the *Algebraic VCSP Dichotomy Conjecture*.

D. Basic LP relaxation

Symmetric operations are known to be closely related to LP-based algorithms for CSP-related problems. One algorithm in particular has been known to solve many VCSPs to optimality. This algorithm is based on the so-called *basic LP relaxation*, or BLP, defined as follows.

Let $\mathbb{M}_n = \{\mu \geq 0 \mid \sum_{x\in D^n} \mu(x) = 1\}$ be the set of probability distributions over labelings in D^n. We also denote $\Delta = \mathbb{M}_1$; thus, Δ is the standard $(|D|-1)$-dimensional simplex. The corners of Δ can be identified with elements in D. For a distribution $\mu \in \mathbb{M}_n$ and a variable $v \in \{1,\dots,n\}$, let $\mu_{[v]} \in \Delta$ be the marginal probability of distribution μ for v:

$$\mu_{[v]}(a) \;=\; \sum_{x\in D^n : x_v = a} \mu(x) \qquad \forall a \in D.$$

Given a VCSP instance $\mathcal{I}$ in the form (1), we define the value $\mathrm{BLP}(\mathcal{I})$ as follows:

$$\begin{aligned}
\mathrm{BLP}(\mathcal{I}) \;&=\; \min \sum_{t\in T} \sum_{x\in \mathrm{dom}\, f_t} \mu_t(x) f_t(x) \qquad (3)\\
\text{s.t.}\quad (\mu_t)_{[k]} \;&=\; \alpha_{v(t,k)} && \forall t \in T, k \in \{1,\dots,n_t\}\\
\mu_t \;&\in\; \mathbb{M}_{n_t} && \forall t \in T\\
\mu_t(x) \;&=\; 0 && \forall t \in T, x \notin \mathrm{dom}\, f_t\\
\alpha_v \;&\in\; \Delta && \forall v \in V
\end{aligned}$$

If there are no feasible solutions then $\mathrm{BLP}(\mathcal{I}) = \infty$. The objective function and all constraints in this system are linear, therefore this is a linear program. Its size is polynomial in the size of $\mathcal{I}$, so $\mathrm{BLP}(\mathcal{I})$ can be found in time polynomial in $|\mathcal{I}|$.

We say that BLP *solves* $\mathcal{I}$ if $\mathrm{BLP}(\mathcal{I}) = \min_{x\in D^n} f_{\mathcal{I}}(x)$, and BLP solves $\mathrm{VCSP}(\Gamma)$ if it solves all instances $\mathcal{I}$ of $\mathrm{VCSP}(\Gamma)$. If BLP solves $\mathrm{VCSP}(\Gamma)$ and Γ is a rigid core, then the optimal solution for every instance can be found by using the standard self-reducibility method. In this method, one goes through the variables in some order, finding $d \in D$ for the current variable v such that instances $\mathcal{I}$ and $\mathcal{I} + u_d(v)$ have the same optimal value (which can be checked by BLP), updating $\mathcal{I} := \mathcal{I} + u_d(v)$, and moving to the next variable. At the end, the instance will have a unique feasible assignment whose value is the optimum of the original instance. Note that in this case $\mathrm{VCSP}(\Gamma)$ is globally tractable.

Theorem 17 ([18]): BLP solves $\mathrm{VCSP}(\Gamma)$ if and only if, for every $m > 1$, Γ has a symmetric fractional polymorphism of arity m.

Theorem 18 ([18], [19]): Let Γ be a rigid core constraint language that is finite-valued. If Γ has a symmetric fractional polymorphism of arity 2 then BLP solves $\mathrm{VCSP}(\Gamma)$, and so $\mathrm{VCSP}(\Gamma)$ is tractable. Otherwise, $\mathrm{VCSP}(\Gamma)$ is NP-hard.

III. Main Result

Definition 19: Let $\mathcal{I}$ be a VCSP instance over variables V with domain D. The *feasibility instance*, $\mathrm{Feas}(\mathcal{I})$, associated to $\mathcal{I}$ is a CSP instance obtained from $\mathcal{I}$ by replacing each constraint function f_t with $\mathrm{dom}\, f_t$.

For a language Γ, let $\mathrm{Feas}(\Gamma) = \{\mathrm{dom}\, f \mid f \in \Gamma\}$. Then the instances of the problem $\mathrm{CSP}(\mathrm{Feas}(\Gamma))$ are the instances $\mathrm{Feas}(\mathcal{I})$ where $\mathcal{I}$ runs through all instances of $\mathrm{VCSP}(\Gamma)$.

Definition 20: Let $\mathcal{I}$ be a VCSP instance over variables V with domain D. For each variable $v \in V$, let $D_v = \{d \in D \mid d = \sigma(v) \text{ for some feasible solution } \sigma \text{ for } \mathcal{I}\}$. Then $(1, \infty)$-*minimal instance* $\bar{\mathcal{I}}$ associated with $\mathcal{I}$ is the VCSP instance obtained from $\mathcal{I}$ by adding, for each $v \in V$, the constraint $u_{D_v}(x_v)$.

Note that if Γ is a rigid core and the problem $\mathrm{CSP}(\mathrm{Feas}(\Gamma))$ is tractable, then, for any instance $\mathcal{I}$ of $\mathrm{VCSP}(\Gamma)$, one can construct the associated $(1, \infty)$-minimal instance in polynomial time. Indeed, to find out whether a given $d \in D$ is in D_v, one only needs to decide whether the CSP instance obtained from $\mathrm{Feas}(\mathcal{I})$ by adding the constraint $u_d(x_v)$ is satisfiable. Since Γ is a rigid core, the latter instance is also an instance of $\mathrm{CSP}(\mathrm{Feas}(\Gamma))$.

If Γ is a rigid core then, for $\mathrm{VCSP}(\Gamma)$ to be tractable, Γ must satisfy the assumption of Conjecture 15, and also, clearly, the feasibility part of the problem, $\mathrm{CSP}(\mathrm{Feas}(\Gamma))$, must be tractable. Our main result shows that if these necessary conditions are satisfied then $\mathrm{VCSP}(\Gamma)$ is indeed tractable.

Theorem 21: Let Γ be a valued constraint language over domain D that is a rigid core. If the following conditions hold then $\mathrm{VCSP}(\Gamma)$ is tractable:

1) Γ has a cyclic fractional polymorphism of arity at least 2, and
2) $\mathrm{CSP}(\mathrm{Feas}(\Gamma))$ is tractable.

Otherwise, $\mathrm{VCSP}(\Gamma)$ is not tractable.

In Theorem 21, the intractability part for (absence of) the first condition follows from Theorem 14, and it is obvious for the second condition. The tractability part follows from Theorem 22 below.

Theorem 22: Let Γ be an arbitrary language that has a cyclic fractional polymorphism of arity at least 2. If $\mathcal{I}$ is an instance of $\mathrm{VCSP}(\Gamma)$ and $\bar{\mathcal{I}}$ is its associated $(1, \infty)$-minimal instance, then $\mathrm{Opt}(\mathcal{I}) = \mathrm{BLP}(\bar{\mathcal{I}})$.

Indeed, if Γ is a rigid core satisfying conditions (1) and (2) from Theorem 21 and $\mathcal{I}$ is an instance of $\mathrm{VCSP}(\Gamma)$ then the equality $\mathrm{Opt}(\mathcal{I}) = \mathrm{BLP}(\bar{\mathcal{I}})$ means that we can efficiently find the optimum value for $\mathcal{I}$ by constructing $\bar{\mathcal{I}}$ (which we can do efficiently because Γ is a rigid core and $\mathrm{CSP}(\mathrm{Feas}(\Gamma))$ is tractable) and then applying BLP to $\bar{\mathcal{I}}$. Then we can find an optimal assignment for $\mathcal{I}$ by self-reduction (see discussion before Theorem 17).

Recall the notion of global tractability from Section II-A. The algorithm that we just described gives the following.

Corollary 23: Let Γ be a valued constraint language over domain D that is a rigid core. If

1) Γ has a cyclic fractional polymorphism of arity at least 2, and
2) $\mathrm{CSP}(\mathrm{Feas}(\Gamma))$ is globally tractable,

then $\mathrm{VCSP}(\Gamma)$ is globally tractable.

Let us now discuss how Theorem 21 implies some earlier results.

As we explained in Section I, if the Algebraic CSP Dichotomy Conjecture holds, then condition (2) in Theorem 21 can be omitted and all intractable VCSPs are NP-hard. Since this conjecture holds when $|D| \leq 3$ [33], [32] or when D is arbitrary finite, but Γ contains all unary crisp functions [41], [34], we get the following corollaries.

Corollary 24: Let $|D| \leq 3$ and let Γ be a valued constraint language that is a rigid core on D. If Γ has a cyclic fractional polymorphism then the problem $\mathrm{VCSP}(\Gamma)$ is tractable, otherwise it is NP-hard.

For the case $|D| = 2$, the tractable cases can be characterised by six specific cyclic fractional polymorphisms [17], and it was shown in [45] that the presence of any cyclic fractional polymorphism (when $|D| = 2$) implies the presence of one of those six. Also, Corollary 24 generalizes results from [49], [50] where the dichotomy was shown for the special case when $|D| = 3$ and all non-crisp functions in Γ are unary. The specific conditions for tractability in [49], [50] have not been shown to be directly implied by the presence of a cyclic fractional polymorphism, though.

Corollary 25: Let Γ be a valued constraint language on D that contains all unary crisp functions. If Γ has a cyclic fractional polymorphism then the problem $\mathrm{VCSP}(\Gamma)$ is tractable, otherwise it is NP-hard.

Corollary 25 generalizes a result from [49] where the dichotomy was shown for the special case when Γ includes all unary crisp functions and all non-crisp functions in Γ are unary. Again, the specific condition for tractability in [49] is not known to be directly implied by the presence of a cyclic fractional polymorphism.

It is shown in [45] how Theorem 21 implies the dichotomy results (including specific conditions for tractability) for the finite-valued case from [19] (Theorem 18) and for the case when Γ contains all unary functions taking values in $\{0,1\}$ [35]. The algorithm for the tractable case in [35] is somewhat similar in spirit to our algorithm, and actually inspired the latter.

Let us now explain how Theorem 21 implies the tractability result from [46] (stated below). An idempotent operation $g \in \mathcal{O}_D$ of arity at least 2 with $g(y,x,x\ldots,x,x) = g(x,y,x,\ldots,x,x) = \ldots = g(x,x,x,\ldots,x,y)$ for all $x,y \in D$ is called a *weak near-unanimity* operation. The tractability result result from [46] states that if $\mathrm{fPol}^+(\Gamma)$ contains weak near-unanimity operations of all but finitely many arities, then $\mathrm{VCSP}(\Gamma)$ is tractable (in fact, via a specific algorithm based on Sherali-Adams hierarchy, which does not follow from our results). This condition on $\mathrm{fPol}^+(\Gamma)$ is well known in the algebraic approach to the CSP, it characterizes (when appropriately formulated) CSPs of bounded width [40]. So assume that $\mathrm{fPol}^+(\Gamma)$ satisfies this condition. Since $\mathrm{fPol}^+(\Gamma) \subseteq \mathrm{Pol}(\Gamma)$, the set $\mathrm{Pol}(\Gamma)$ also contains these operations, so $\mathrm{CSP}(\mathrm{Feas}(\Gamma))$ is tractable by [40]. Moreover, by [39], $\mathrm{fPol}^+(\Gamma)$ then also contains a cyclic operation of arity at least 2. Now (the proof of) Theorem 50 of [45] implies that Γ has a cyclic fractional polymorphism of arity at least 2, and then tractability of $\mathrm{VCSP}(\Gamma)$ follows from Theorem 21.

IV. Proof of Theorem 22: Reduction to a block-finite language

We will prove Theorem 22 by constructing, from a given (feasible) instance $\mathcal{I}$, a valued constraint language Γ' on some finite set D' and an instance $\mathcal{I}'$ of $\mathrm{VCSP}(\Gamma')$ such that $\mathrm{Opt}(\mathcal{I}) = \mathrm{Opt}(\bar{\mathcal{I}}) = \mathrm{Opt}(\mathcal{I}') = \mathrm{BLP}(\mathcal{I}') = \mathrm{BLP}(\bar{\mathcal{I}})$. The first two equalities will follow trivially from the construction of Γ' and $\mathcal{I}'$, the last equality holds by Lemma 26 below, while the key equality $\mathrm{Opt}(\mathcal{I}') = \mathrm{BLP}(\mathcal{I}')$ will follow from the fact that BLP solves $\mathrm{VCSP}(\Gamma')$ that we prove, using Theorem 17, in Theorem 29. The construction is inspired by [47], where a similar technique of "lifting" a language was used in a different context.

Let V be the set of variables of instance $\mathcal{I}$, and let

$$f_{\mathcal{I}}(x) \;=\; \sum_{t\in T} f_t(x_{v(t,1)},\ldots,x_{v(t,n_t)}) \qquad \forall x : V \to D \tag{4}$$

be its objective function. For the $(1,\infty)$-minimal instance $\bar{\mathcal{I}}$, the objective function is

$$f_{\bar{\mathcal{I}}}(x) \;=\; \sum_{t\in T} f_t(x_{v(t,1)},\ldots,x_{v(t,n_t)}) + \sum_{v\in V} u_{D_v}(x_v) \qquad \forall x : V \to D \tag{5}$$

Now let $D'_v = \{(v,a) \mid a \in D_v\}$ be a unique copy of D_v. We now define a new language Γ' over domain $D' = \bigcup_{v\in V} D'_v$ as follows:

$$\Gamma' = \bigcup_{t\in T} \left\{ f_t^{\langle v(t,1),\ldots,v(t,n_t)\rangle}, \mathrm{dom}\, f_t^{\langle v(t,1),\ldots,v(t,n_t)\rangle} \right\} \cup \bigcup_{v\in V} \left\{ u_{D'_v} \right\} \cup \{=_{D'}\}$$

where functions $u_{D'_v}$ are as defined above, $=_{D'}$ is the binary $\{0,\infty\}$-valued function corresponding to the equality relation, and, for an n-ary function f over D and variables $v_1,\ldots,v_n \in V$, we define function $f^{\langle v_1,\ldots,v_n\rangle} : (D')^n \to \overline{\mathbb{Q}}$ as follows:

$$f^{\langle v_1,\ldots,v_n\rangle}(x) = \begin{cases} f(\hat{x}) & \text{if } x = ((v_1,\hat{x}_1),\ldots,(v_n,\hat{x}_n)) \\ \infty & \text{otherwise} \end{cases} \qquad \forall x \in (D')^n$$

The above mentioned instance $\mathcal{I}'$ of $\mathrm{VCSP}(\Gamma')$ is obtained from $\bar{\mathcal{I}}$ by replacing each function f_t with $f_t^{\langle v(t,1),\ldots,v(t,n_t)\rangle}$ and replacing each function u_{D_v} with $u_{D'_v}$.

It is straightforward to check that there is a one-to-one correspondence between the sets of feasible solutions to BLP relaxations for $\mathcal{I}'$ and $\bar{\mathcal{I}}$, and that this correspondence also preserves the values of solutions.

Lemma 26: We have $\mathrm{BLP}(\mathcal{I}') = \mathrm{BLP}(\bar{\mathcal{I}})$.

The following fact can also be easily shown.

Lemma 27: If Γ has a cyclic fractional polymorphism of arity $m > 1$ then Γ' has the same property.

Proof: Let ω be a cyclic fractional polymorphism of Γ. Fix an arbitrary element $d' \in D'$. For each operation $g \in \mathrm{supp}(\omega)$, define the operation g' on D' as follows:

$$g'(x_1, \ldots, x_m) = \begin{cases} (v, g(\hat{x}_1, \ldots, \hat{x}_m)) & \text{if } x_1 = (v, \hat{x}_1), \ldots, x_m = (v, \hat{x}_m) \text{ for some } v \in V \\ d' & \text{otherwise} \end{cases}$$

Clearly, each operation g' is cyclic. Consider the fractional operation ω' on D' such that $\omega(g') = \omega(g)$ for all $g \in supp(\omega)$. It is straightforward to check that ω' is a fractional polymorphism of Γ'. ■

V. Block-finite languages

Definition 28: A finite language Γ is called *block-finite* if its domain D can be partitioned into disjoint subsets $\{D_v \mid v \in V\}$ such that

(a) For any $a \in D_v$ with $v \in V$ there exists a polymorphism $g_a \in \mathcal{O}^{(1)}$ of $\mathrm{Feas}(\Gamma)$ such that $g_a(b) = a$ for all $b \in D_v$.

(b) For any n-ary function $f \in \Gamma$, the relation $\mathrm{dom}\, f$ (viewed as a function $D^n \to \{0, \infty\}$) belongs to Γ. Furthermore, the binary equality relation on D, denoted as $=_D$: $D^2 \to \{0, \infty\}$, also belongs to Γ.

(c) Any n-ary function $f \in \Gamma - \{=_D\}$ satisfies $\mathrm{dom}\, f \subseteq D_{v_1} \times \ldots \times D_{v_n}$ for some $v_1, \ldots, v_n \in V$.

It is easy to see that the language Γ' defined in the previous section is block-finite. It obviously has properties (b) and (c), and it has property (a) because the instance $\mathcal{I}'$ is $(1, \infty)$-minimal. Indeed, if $a = (v, d) \in D'_v$ then, by definition, $\mathcal{I}$ has a feasible solution $\sigma : V \to D$ with $\sigma(v) = d$. Define function g_a as follows: for each $a' = (v', d') \in D'$, set $g_a(a') = (v', \sigma(v'))$. It is easy to check that g_a has the required properties.

From now on, we forget about the original language Γ from the previous section and about the specific language Γ' and work with an arbitrary block-finite language that has a cyclic fractional polymorphism of arity at least 2. For simplicity, we denote our language by Γ. Note that Γ is not necessarily a rigid core, but this property is not required in Theorem 17. By Theorem 17, in order to prove Theorem 22, it remains to show the following.

Theorem 29: Suppose that a block-finite language Γ admits a cyclic fractional polymorphism ν of arity at least 2. Then, for every $m \geq 2$, Γ admits a symmetric fractional polymorphism of arity m.

We now provide a short overview of our techniques; all missing proofs can be found in the full version of the paper [51].

A graph of generalized operations First, we describe a basic tool that will be used for constructing new fractional polymorphisms, namely a graph of generalized operations introduced in [18].

Let $\mathcal{O}^{(m \to m)}$ be the set of mappings $\mathbf{g} : D^m \to D^m$ and let $\mathbb{1} \in \mathcal{O}^{(m \to m)}$ be the identity mapping. Consider a sequence $x \in [D^n]^m$ of m labelings; this means that $x = (x^1, \ldots, x^m)$ where $x^i \in D^n$. For an n-ary function f, we define $f^m(x) = \frac{1}{m}(f(x^1) + \ldots + f(x^m))$. For a mapping $\mathbf{g} = (g_1, \ldots, g_m) \in \mathcal{O}^{(m \to m)}$, we also denote $x^{\mathbf{g}i} = g_i(x)$ for $i \in [m]$ and $\mathbf{g}(x) = (x^{\mathbf{g}1}, \ldots, x^{\mathbf{g}m})$. A probability distribution ρ over $\mathcal{O}^{(m \to m)}$ will be called a *(generalized) fractional polymorphism of* Γ *of arity* $m \to m$ if each function $f \in \Gamma$ satisfies

$$\sum_{\mathbf{g} \in \mathrm{supp}(\rho)} \rho(\mathbf{g}) f^m(\mathbf{g}(x)) \quad \leq \quad f^m(x) \qquad \forall x \in [\mathrm{dom}\, f]^m \tag{6}$$

We will use the following construction in several parts of the proof. Assume that we have some probability distribution ω with a finite support such that (i) each element $s \in \mathrm{supp}(\omega)$ corresponds to an element of $\mathcal{O}^{(m \to m)}$ denoted as $\mathbb{1}^s$, and (ii) this distribution satisfies the following property for each $f \in \Gamma$:

$$\sum_{s \in \mathrm{supp}(\omega)} \omega(s) f^m(\mathbb{1}^s(x)) \quad \leq \quad f^m(x) \qquad \forall x \in [\mathrm{dom}\, f]^m \tag{7}$$

Condition (7) then can be rephrased as saying that vector $\sum_{s \in \mathrm{supp}(\omega)} \omega(s) \chi_{\mathbb{1}^s}$ is a fractional polymorphism of Γ of arity $m \to m$.

For a mapping $\mathbf{g} \in \mathcal{O}^{(m\to m)}$ denote $\mathbf{g}^s = \mathbb{1}^s \circ \mathbf{g}$. (This notation is consistent with the earlier one since $\mathbb{1}^s \circ \mathbb{1} = \mathbb{1}^s$ for any s). We use $\mathbf{g}^{s_1\cdots s_k}$ to denote $(\ldots(\mathbf{g}^{s_1})^{\cdots})^{s_k} = \mathbb{1}^{s_k} \circ \ldots \circ \mathbb{1}^{s_1} \circ \mathbf{g}$. Next, define a directed graph $(\mathbb{G}, E)$ with the nodes $\mathbb{G} = \{\mathbb{1}^{s_1\cdots s_k} \mid s_1, \ldots, s_k \in \mathrm{supp}(\omega), k \geq 0\}$ and the edges $E = \{(\mathbf{g}, \mathbf{g}^s) \mid \mathbf{g} \in \mathbb{G}, s \in \mathrm{supp}(\omega)\}$. This graph can be decomposed into strongly connected components, yielding a directed acyclic graph (DAG) on these components. We define $\mathsf{Sinks}(\mathbb{G}, E)$ to be the set of those strongly connected components $\mathbb{H} \subseteq \mathbb{G}$ of $(\mathbb{G}, E)$ that are sinks of this DAG (i.e. have no outgoing edges). Any DAG has at least one sink, therefore $\mathsf{Sinks}(\mathbb{G}, E)$ is non-empty. We denote $\mathbb{G}^* = \bigcup_{\mathbb{H}\in\mathsf{Sinks}(\mathbb{G},E)} \mathbb{H} \subseteq \mathbb{G}$ and $Range_n(\mathbb{G}^*) = \{\mathbf{g}^*(x) \mid \mathbf{g}^* \in \mathbb{G}^*, x \in [D^n]^m\}$. Also, for a tuple $\hat{x} \in D^m$ we will denote $\mathbb{G}(\hat{x}) = \{\mathbf{g}(\hat{x}) \mid \mathbf{g} \in \mathbb{G}\} \subseteq D^m$.

Main construction For a sequence $x = (x^1, \ldots, x^m) \in D^m$ and a permutation π of $[m]$, we define $x^\pi = (x^{\pi(1)}, \ldots, x^{\pi(m)})$. Similarly, for a mapping $\mathbf{g} = (g_1, \ldots, g_m) \in \mathcal{O}^{(m\to m)}$ define $\mathbf{g}^\pi = (g_{\pi(1)}, \ldots, g_{\pi(m)})$. Let Ω be the set of mappings $\mathbf{g} \in \mathcal{O}^{(m\to m)}$ that satisfy $\mathbf{g}^\pi(x) = \mathbf{g}(x^\pi)$ for any $x \in D^m$ and any permutation π of $[m]$. Equivalently, $g_{\pi(i)}(x) = g_i(x^\pi)$ for any $i \in [m]$.

Let ω be a generalized fractional polymorphism of Γ of arity $m \to m$ satisfying $\mathrm{supp}(\omega) \subseteq \Omega$ *with the largest possible support*. (It can be shown to be well-defined). Let us apply the construction described earlier starting with the chosen distribution ω, where for $\mathbf{g} \in \mathrm{supp}(\omega)$ we define operation $\mathbb{1}^{\mathbf{g}} \in \mathcal{O}^{(m\to m)}$ via $\mathbb{1}^{\mathbf{g}} = \mathbf{g}$. Let the resulting graph be $(\mathbb{G}, E)$. It is not difficult to show that $\mathrm{supp}(\omega) = \mathbb{G}$.

By writing down a linear system that expresses the fact that ω has the largest possible support in Ω, and then applying Farkas lemma we prove the following result.

Theorem 30: For any $\hat{x} \in D^m$ there exists a function $f \in \langle\Gamma\rangle$ of arity m with $\arg\min f = \mathbb{G}(\hat{x})$.

Now consider a tuple $\hat{x} \in Range_1(\mathbb{G}^*)$ and a function f from Theorem 30 with $\arg\min f = \mathbb{G}(\hat{x})$.

Theorem 31: Assume that one of the following holds:

(a) $m = 2$ and Γ admits a cyclic fractional polymorphism of arity at least 2.

(b) $m \geq 3$ and Γ admits a symmetric fractional polymorphism of arity $m - 1$.

Then for every distinct pair of indices $i, j \in [m]$ there exists $x \in \arg\min f$ with $x_i = x_j$.

It can be easily shown that the last condition implies that $\hat{x}_1 = \ldots = \hat{x}_m$ for all $\hat{x} \in Range_1(\mathbb{G}^*)$; therefore, all operations in $\mathbf{g} = (g_1, \ldots, g_m) \in \mathbb{G}^*$ are symmetric. As shown in [18], Γ admits a generalized fractional polymorphism ρ with $\mathrm{supp}(\rho) \subseteq \mathbb{G}^*$. This gives a symmetric fractional polymorphism of Γ of arity m assuming that (a) or (b) holds. Using induction on m, we then prove Theorem 29.

It remains to prove Theorem 31. Using the fact that Γ is block-finite, we first show that if $x \in \mathbb{G}(\hat{x})$ and $a \in \{x_1, \ldots, x_m\}$ then $(a, \ldots, a) \in \mathrm{dom}\, f$. Then we consider separately cases (a) and (b).

Case $m = 2$: proof of Theorem 31(a) We pick element $a \in A = \{x_1 \mid x \in \mathbb{G}(\hat{x})\}$ that minimizes $f(a, a)$, and then a tuple $(a, b) \in \mathbb{G}(\hat{x}) = \arg\min f$. We show that (b, a) also belongs to $\mathbb{G}(\hat{x})$, and thus $f(b, a) = f(a, b)$. We then apply a cyclic fractional polymorphism of Γ of arity $r \geq 2$ to tuples $(a, b), (b, a), (a, a), \ldots, (a, a)$ (where (a, a) is repeated $r - 2$ times), and after simple manipulations obtain that $f(a, a) \leq f(a, b)$; this gives the claim.

Case $m \geq 3$: proof of Theorem 31(b) Fix distinct $i, j \in [m]$, and define binary function $\bar{f} \in \langle\Gamma\rangle$ via $\bar{f}(a, b) = \min_{x\in D^m : x_i = a, x_j = b} f(x)$. We need to show that $(a, a) \in \arg\min \bar{f}$ for some $a \in D$.

Let $\tilde{\omega}$ be a symmetric fractional polymorphism of Γ of arity $m - 1$. Following the construction in [18], we define graph $(\tilde{\mathbb{G}}, \tilde{E})$ as described earlier in this section, starting with the distribution $\tilde{\omega}$ where for $s \in \mathrm{supp}(\tilde{\omega})$ the mapping $\mathbb{1}^s \in \mathcal{O}^{(m\to m)}$ is defined via $\mathbb{1}^s(x) = (s(x_{-1}), \ldots, s(x_{-m}))$ for $x \in D^m$. (Here $x_{-i} \in D^{m-1}$ is the labeling obtained from x by removing the i-th element). It can be shown that $\tilde{\mathbb{G}} \subseteq \mathbb{G}$. We denote $\tilde{\mathbb{G}}^* = \bigcup_{\mathbb{H}\in\mathsf{Sinks}(\tilde{\mathbb{G}},\tilde{E})} \mathbb{H} \subseteq \tilde{\mathbb{G}}$.

For each $\mathbf{g} \in \tilde{\mathbb{G}}$ and $k \in [m]$ we define labeling $x^{[\mathbf{g}k]} \in D^2$ as follows: set $x = \mathbf{g}(\hat{x})$, and then

- If $k = i$, set $x^{[\mathbf{g}k]} = (x_i, x_j)$. We have $x^{[\mathbf{g}i]} \in \arg\min \bar{f}$ since $x \in \mathbb{G}(\hat{x}) = \arg\min f$.
- If $k = j$, set $x^{[\mathbf{g}k]} = (x_j, x_i)$.
- If $k \neq i$ and $k \neq j$, set $x^{[\mathbf{g}k]} = (x_k, x_k)$. We have $x^{[\mathbf{g}k]} \in \mathrm{dom}\, \bar{f}$ (by block-finiteness of Γ).

We then fix a mapping $\tilde{\mathbf{g}} \in \tilde{\mathbb{G}}^*$, and prove by induction that $\mathbf{g}(\tilde{x}) = (x^{[(\mathbf{g}\circ\tilde{\mathbf{g}})1]}, \ldots, x^{[(\mathbf{g}\circ\tilde{\mathbf{g}})m]})$ for any $\mathbf{g} \in \tilde{\mathbb{G}}$, where $\tilde{x} = (x^{[\tilde{\mathbf{g}}1]}, \ldots, x^{[\tilde{\mathbf{g}}m]}) \in [D^2]^m$. We also show that $\tilde{x} \in Range_2(\tilde{\mathbb{G}}^*) \cap [\mathrm{dom}\, \bar{f}]^m$.

Now pick $k \in [m] - \{i, j\}$. It can be shown that there exists a probability distribution λ over $\tilde{\mathbb{G}}^*$ such that $\sum_{\mathbf{g} \in \tilde{\mathbb{G}}^*} \lambda_{\mathbf{g}} \bar{f}(x^{\mathbf{g}i}) = \sum_{\mathbf{g} \in \tilde{\mathbb{G}}^*} \lambda_{\mathbf{g}} \bar{f}(x^{\mathbf{g}k})$ (for finite-valued Γ's this was proved in [18]). This gives

$$\sum_{\mathbf{g} \in \tilde{\mathbb{G}}^*} \lambda_{\mathbf{g}} \bar{f}(x^{[(\mathbf{g} \circ \tilde{\mathbf{g}})i]}) = \sum_{\mathbf{g} \in \tilde{\mathbb{G}}^*} \lambda_{\mathbf{g}} \bar{f}(x^{[(\mathbf{g} \circ \tilde{\mathbf{g}})k]})$$

Every tuple $x^{[(\mathbf{g} \circ \tilde{\mathbf{g}})i]}$ on the LHS belongs to $\arg\min \bar{f}$. Therefore, every tuple $x^{[(\mathbf{g} \circ \tilde{\mathbf{g}})k]}$ on the RHS corresponding to mapping $\mathbf{g} \in \tilde{\mathbb{G}}^*$ with $\lambda_{\mathbf{g}} > 0$ also belongs to $\arg\min \bar{f}$, and by construction such tuples have the form (a, a) for some $a \in D$. For further details we refer to the full version [51].

ACKNOWLEDGMENT

The first and the third authors are supported by the European Research Council uder the European Union's Seventh Franework Programme (FP7/2007-2013)/ERC grant agreement no 616160.

REFERENCES

[1] Y. Boykov, O. Veksler, and R. Zabih, "Markov random fields with efficient approximations," in *Computer Vision and Pattern Recognition*. IEEE Computer Society, 1998, pp. 648–655.

[2] Y. Crama and P. L. Hammer, *Boolean Functions - Theory, Algorithms, and Applications*. Cambridge University Press, 2011.

[3] P. Jeavons, A. Krokhin, and S. Zivny, "The complexity of valued constraint satisfaction," *Bulletin of the EATCS*, vol. 113, pp. 21–55, 2014.

[4] S. L. Lauritzen, *Graphical Models*. Oxford University Press, 1996.

[5] F. Rossi, P. van Beek, and T. Walsh, Eds., *Handbook of Constraint Programming*. Elsevier, 2006.

[6] M. Wainwright and M. Jordan, "Graphical models, exponential families, and variational inferences," *Foundations and Trends in Machine Learning*, vol. 1, no. 1-2, pp. 1–305, 2008.

[7] N. Creignou, S. Khanna, and M. Sudan, *Complexity Classifications of Boolean Constraint Satisfaction Problems*, ser. SIAM Monographs on Discrete Mathematics and Applications, 2001, vol. 7.

[8] U. Montanari, "Networks of Constraints: Fundamental properties and applications to picture processing," *Information Sciences*, vol. 7, pp. 95–132, 1974.

[9] P. Kolaitis and M. Vardi, "Conjunctive-query containment and constraint satisfaction," *Journal of Computer and System Sciences*, vol. 61, pp. 302–332, 2000.

[10] T. Feder and M. Y. Vardi, "The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory," *SIAM Journal on Computing*, vol. 28, no. 1, pp. 57–104, 1998.

[11] D. Cohen, M. Cooper, P. Jeavons, and A. Krokhin, "Supermodular functions and the complexity of Max CSP," *Discrete Applied Mathematics*, vol. 149, no. 1-3, pp. 53–72, 2005.

[12] P. Jonsson, M. Klasson, and A. Krokhin, "The approximability of three-valued Max CSP," *SIAM Journal on Computing*, vol. 35, no. 6, pp. 1329–1349, 2006.

[13] P. Jonsson, F. Kuivinen, and J. Thapper, "Min CSP on four elements: Moving beyond submodularity," in *Proceedings of CP'11*, 2011, pp. 438–453.

[14] P. Jonsson and G. Nordh, "Introduction to the Maximum Solution problem," in *Complexity of Constraints*, ser. LNCS, 2008, vol. 5250, pp. 255–282.

[15] R. Takhanov, "A dichotomy theorem for the general minimum cost homomorphism problem," in *STACS'10*, 2010, pp. 657–668.

[16] D. Cohen, M. Cooper, P. Creed, P. Jeavons, and S. Živný, "An algebraic theory of complexity for discrete optimisation," *SIAM Journal on Computing*, vol. 42, no. 5, pp. 1915–1939, 2013.

[17] D. A. Cohen, M. C. Cooper, P. G. Jeavons, and A. A. Krokhin, "The Complexity of Soft Constraint Satisfaction," *Artificial Intelligence*, vol. 170, no. 11, pp. 983–1016, 2006.

[18] V. Kolmogorov, J. Thapper, and S. Živný, "The power of linear programming for general-valued CSPs," *SIAM Journal on Computing*, vol. 44, no. 1, pp. 1–36, 2015.

[19] J. Thapper and S. Živný, "The complexity of finite-valued CSPs," in *Proceedings of the 45th ACM Symposium on the Theory of Computing (STOC'13)*. ACM, 2013, pp. 695–704.

[20] L. Barto and M. Kozik, "Robust satisfiability of constraint satisfaction problems," in *STOC'12*, 2012, pp. 931–940.

[21] J. Brown-Cohen and P. Raghavendra, "Combinatorial optimization algorithms via polymorphisms," ArXiv:1501.01598, Tech. Rep., 2015.

[22] S. O. Chan, J. R. Lee, P. Raghavendra, and D. Steurer, "Approximate constraint satisfaction requires large LP relaxations," in *FOCS*, 2013, pp. 350–359.

[23] V. Deineko, P. Jonsson, M. Klasson, and A. Krokhin, "The approximability of Max CSP with fixed-value constraints," *Journal of the ACM*, vol. 55, no. 4, p. Article 16, 2008.

[24] A. Ene, J. Vondrák, and Y. Wu, "Local distribution and the symmetry gap: Approximability of multiway partitioning problems," in *SODA*, 2013, pp. 306–325, update at arXiv1503.03905.

[25] J. Håstad, "Every 2-CSP allows nontrivial approximation," *Computational Complexity*, vol. 17, no. 4, pp. 549–566, 2008.

[26] P. Raghavendra, "Optimal algorithms and inapproximability results for every CSP?" in *STOC'08*, 2008, pp. 245–254.

[27] D. Cohen and P. Jeavons, "The complexity of constraint languages," in *Handbook of Constraint Programming*, F. Rossi, P. van Beek, and T. Walsh, Eds. Elsevier, 2006, ch. 8.

[28] N. Creignou, P. Kolaitis, and H. Vollmer, Eds., *Complexity of Constraints*, ser. LNCS. Springer, 2008, vol. 5250.

[29] G. Gottlob, G. Greco, and F. Scarcello, "Tractable optimization problems through hypergraph-based structural restrictions," in *Proceedings of ICALP'09*, 2009, pp. 16–30.

[30] D. Marx, "Tractable hypergraph properties for constraint satisfaction and conjunctive queries," *J. ACM*, vol. 60, no. 6, p. Article 42, 2013.

[31] A. Bulatov, A. Krokhin, and P. Jeavons, "Classifying the Complexity of Constraints using Finite Algebras," *SIAM Journal on Computing*, vol. 34, no. 3, pp. 720–742, 2005.

[32] T. J. Schaefer, "The Complexity of Satisfiability Problems," in *Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC'78)*. ACM, 1978, pp. 216–226.

[33] A. Bulatov, "A dichotomy theorem for constraint satisfaction problems on a 3-element set," *Journal of the ACM*, vol. 53, no. 1, pp. 66–120, 2006.

[34] A. A. Bulatov, "Complexity of conservative constraint satisfaction problems," *ACM Transactions on Computational Logic*, vol. 12, no. 4, 2011, article 24.

[35] V. Kolmogorov and S. Živný, "The complexity of conservative valued CSPs," *Journal of the ACM*, vol. 60, no. 2, p. Article 10, 2013.

[36] P. G. Jeavons, "On the Algebraic Structure of Combinatorial Problems," *Theoretical Computer Science*, vol. 200, no. 1-2, pp. 185–204, 1998.

[37] P. G. Jeavons, D. A. Cohen, and M. Gyssens, "Closure Properties of Constraints," *Journal of the ACM*, vol. 44, no. 4, pp. 527–548, 1997.

[38] L. Barto, "The collapse of the bounded width hierarchy," *Journal of Logic and Computation*, 2014, accepted.

[39] L. Barto and M. Kozik, "Absorbing subalgebras, cyclic terms and the constraint satisfaction problem," *Logical Methods in Computer Science*, vol. 8, no. 1, pp. 1–26, 2012.

[40] ——, "Constraint satisfaction problems solvable by local consistency methods," *Journal of the ACM*, vol. 61, no. 1, p. Article 3, 2014.

[41] L. Barto, "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, 2011, pp. 301–310.

[42] L. Barto, M. Kozik, and T. Niven, "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*, vol. 38, no. 5, pp. 1782–1802, 2009.

[43] P. Idziak, P. Markovic, R. McKenzie, M. Valeriote, and R. Willard, "Tractability and learnability arising from algebras with few subpowers," *SIAM Journal on Computing*, vol. 39, no. 7, pp. 3023–3037, 2010.

[44] D. Cohen, M. Cooper, and P. Jeavons, "An algebraic characterisation of complexity for valued constraints," in *CP'06*, ser. LNCS, vol. 4204, 2006, pp. 107–121.

[45] M. Kozik and J. Ochremiak, "Algebraic properties of valued constraint satisfaction problem," 2015, arXiv1403.0476.

[46] J. Thapper and S. Živný, "Sherali-adams relaxations for valued csps," in *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, 2015, pp. 1058–1069.

[47] V. Kolmogorov, M. Rolínek, and R. Takhanov, "Effectiveness of structural restrictions for hybrid CSPs," 2015, arXiv1504.07067.

[48] P. Fulla and S. Živný, "A galois connection for valued constraint languages of infinite size," in *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, 2015, pp. 517–528.

[49] H. Uppman, "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)*, ser. Lecture Notes in Computer Science, vol. 7965. Springer, 2013, pp. 804–815.

[50] ——, "Computational complexity of the extended minimum cost homomorphism problem on three-element domains," in *31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014)*, 2014, pp. 651–662.

[51] V. Kolmogorov, A. Krokhin, and M. Rolínek, "The complexity of general-valued CSPs," 2015, arXiv1502.07327.