The Combined Basic LP and Affine IP Relaxation for Promise VCSPs on Infinite Domains

Caterina Viola

Department of Computer Science, University of Oxford, UK
caterina.viola@cs.ox.ac.uk

Stanislav Živný

Department of Computer Science, University of Oxford, UK
standa.zivny@cs.ox.ac.uk

──── **Abstract** ────

Convex relaxations have been instrumental in solvability of constraint satisfaction problems (CSPs), as well as in the three different generalisations of CSPs: valued CSPs, infinite-domain CSPs, and most recently promise CSPs. In this work, we extend an existing tractability result to the three generalisations of CSPs combined: We give a sufficient condition for the combined basic linear programming and affine integer programming relaxation for exact solvability of promise valued CSPs over infinite-domains. This extends a result of Brakensiek and Guruswami [SODA'20] for promise (non-valued) CSPs (on finite domains).

1 Introduction

Constraint satisfaction.   Constraint satisfaction problems (CSPs) are a wide class of computational decision problems. An instance of a CSP is defined by finitely many relations (constraints) that must hold among finitely many given variables; the computational task is to decide whether it is possible to find an assignment of labels from a fixed set (the domain) to the variables so that all the constraints are satisfied. Many problems in computer science (e.g., from artificial intelligence, scheduling, computational linguistic, computational biology and verification) can be modelled as CSPs by choosing an appropriate set of constraints. However, there are many other problems in which some of the constraints may be violated at a cost or in which there are satisfying assignments which are preferable to others. These situations are captured by valued constraint satisfaction problems.

Valued constraint satisfaction.   An instance of a valued constraint satisfaction problem (VCSP) is defined by finitely many cost functions (valued constraints) depending on finitely many given variables and a (rational) threshold; the computational task is to decide whether

it is possible to find an assignment of labels from the domain to the variables so that the value of the sum of the cost functions is at most the given threshold. In VCSP instances, cost functions can take on rational or infinite values. VCSPs not only capture optimisation problems but are also a generalisation of CSPs: the non-feasibility of an assignment is modelled by allowing the cost functions to evaluate to $+\infty$. A CSP can thus be seen as a VCSP in which the cost functions take values in $\{0, +\infty\}$.

Finite domains. In the case in which the domain (i.e., the fixed set of possible labels for the variables) is a *finite* set the computational complexity of both CSPs and VCSPs have been completely classified. Moreover, in both frameworks a dichotomy theorem holds: every CSP and VCSP is either in P or is NP-complete, depending on some algebraic condition of the underlying set of allowed relations and cost functions, respectively. A dichotomy theorem for CSPs was conjectured by Feder and Vardi [27]. The attempt to prove the conjecture motivated the introduction of the so-called universal algebraic approach [20] for CSPs, which was later extended to VCSPs in [39], where an analogue of the complexity dichotomy was conjectured for VCSPs. The dichotomy conjectures for finite-domain CSPs and VCSPs inspired an intensive line of research. A complexity classification of finite-domain VCSPs for sets of cost functions taking finite (rational-only) values was established in [45]. A complexity classification of VCSPs was consequently established in [35], assuming a dichotomy for CSPs, which was proved independently in [19] and [51].

Infinite domains. Although most research on CSPs and VCSPs in the past two decades focused on finite-domain problems, the literature is full of problems (studied independently of CSPs and VCSPs) that can be modelled as CSPs or VCSPs *only* if infinite domains are allowed. For instance, solvability of linear Diophantine equations [24, 33] and the model-checking problem for Kozen's modal μ-calculus [38] are examples of problems that can be modelled as infinite-domain CSPs. Linear Programming, Linear Least Square Regression [14], and Minimum Correlation Clustering [2] are examples of problems that can be modelled as infinite-domain VCSPs. The classes of infinite-domain CSPs and infinite-domain VCSPs are huge! In fact, *every* computational problem over a finite alphabet is Turing-equivalent to an infinite-domain CSP [6]. Therefore, only by focussing on special classes of infinite-domain CSPs (and VCSPs) is it possible to obtain general complexity results. There is a rich literature on the computational complexity of special classes of infinite-domain CSPs, e.g., [9, 8, 13, 32, 7, 11, 12, 5].

Promise constraint satisfaction. Both infinite-domains CSPs and VCSPs are extensions of the original (finite-domain) CSPs. Promise constraint satisfaction problems (PCSPs) are a third, recently introduced extension of CSPs [15, 21, 3, 28]. Informally, in a PCSP the goal is to find an approximately good solution to a problem under the assumption (the promise) that the problem has a solution. The difference between CSPs and PCSPs is that in a PCSP instance each constraint comes with two relations (not necessarily on the same domain), a "strict" and a "weak" relation. The computational task is then to distinguish between being able to satisfy all the strict constraints versus not being able to satisfy all the weak constraints. A CSP can be seen as a PCSP in which the strict and weak constraints coincide. Perhaps the most well-known example of a PCSP is the approximate graph colouring problem, in which the task is to distinguish k-colourable graphs from graphs that are not c-colourable, for some $c > k$. (For $c = k$, we get the standard k-colouring problem.) Kazda recently introduced the framework of promise VCSPs on finite domains [34], where he generalised some of the algebraic reductions from (finite-domain) promise CSPs to (finite-domain) promise VCSPs. As far as we are aware, the only other related work on (finite-domain) promise VCSPs is [1].

Convex relaxations.   One of the most effective ways to design a polynomial-time algorithm for solving combinatorial and optimisation problems is to employ convex relaxations. The idea of convex relaxations is to transform the original problem to an integer program which is then relaxed to a polynomial-time solvable convex program [14], e.g. a linear program. In the context of CSPs, convex relaxations have been studied for robust solvability [40, 26, 4, 25]. Convex relaxations have been also successfully applied to the study of the three extensions of CSPs. For VCSPs, characterisations of the applicability of the basic linear programming relaxation [36], constant levels of the Sherali-Adams linear programming hierarchy [46], and a polynomial-size semidefinite programming relaxation [47] have been provided for exact solvability. In the PCSP framework, the polynomial-time tractability via a specific convex relaxation has been characterised for the basic linear programming relaxation [21], affine integer programming relaxation [21], and their combination [16, 17, 18]. For infinite-domain VCSPs, a sufficient condition has been identified for the solvability via a combination of the basic linear programming relaxation and an efficient sampling algorithm (that is, polynomial-time many-one reduction to a finite-domain VCSP) [10, 49].

Contributions.   We initiate the study of convex relaxations for the three generalisations of CSPs *combined*; that is, convex relaxations for promise valued constraint satisfaction problems on infinite-domains. We focus on the combined basic linear programming (BLP) and affine integer programming (AIP) relaxation introduced by Brakensiek and Guruswami [17]. This relaxation is stronger than both the BLP and AIP relaxations individually in the sense that if a class of promise VCSPs is solved by, say, the BLP relaxation then it is also solved by the combined relaxation (and the same holds true for the AIP relaxation). The power of the combined relaxation for (finite-domain) promise CSPs was established in [18]. Rather surprisingly, the combined relaxation gives an algorithm that solves all tractable (non-promise) CSPs on Boolean domains, identified in Schaefer's work [43], thus giving a unified algorithm.

By extending the argument from [17], we establish a sufficient algebraic condition on the combined relaxation for the solvability of promise VCSPs in which the domain of the "weak cost functions" is possibly infinite (Theorem 4). The proof of this result draws on ideas introduced in [17] but requires a non-trivial amount of technical machinery to make it work in the infinite-domain valued setting. While our relaxation is inspired by [17], it is appropriately modified to work in the optimisation setting (of valued (P)CSPs). We remark that the condition we give is known to be necessary already in special cases of our setting, namely for finite-domain non-valued PCSPs [18]. As an application of our main result, we derive an algebraic condition under which an *infinite-domain* promise VCSP admitting an efficient sampling algorithm can be solved in polynomial time using the combined relaxation (Theorem 9). We emphasise that our main results (Theorems 4 and 9) are appreciatively general, and in particular hold for various special cases of our framework; e.g., for finite-domain promise VCSPs and infinite-domain promise CSPs.

Approximability of Max-CSPs.   PCSPs are approximability problems in which we require that all constraints should be satisfied, although only in a weaker sense. Another very natural and well-studied form of relaxation is to try to maximise the number of satisfied constraints. Convex relaxations have played a crucial role in this research direction on approximability of (finite-domain) Max-CSPs, going back to the work of Goemans and Williamson [30], e.g., [42, 48, 41, 23, 22, 37, 29].

Paper outline. This is an extended abstract of our work. Given the page limit, we do not introduce some of the standard notation and do not include all proofs. All details with more explanation and examples can be found in the full version of the paper [50].

2   Preliminaries

Throughout the paper, we denote by x_i the i-th component of a tuple x. We denote by $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{Q}_{\geq 0}$ the set of whole numbers, integer numbers, rational numbers, and nonnegative rational numbers, respectively. For every $m \in N$, we denote by $[m]$ the set $\{1, \ldots, m\} \subset \mathbb{N}$. Finally, for every $k \in \mathbb{Q}$ we use the $\lceil k \rceil$ and $\lfloor k \rfloor$ to denote the minimum natural number that is at least k and the maximum natural number that is at most k, respectively.

Valued Constraint Satisfaction Problems. A *valued structure* Γ (over D) consists of a signature τ consisting of function symbols f, each equipped with an arity $\mathrm{ar}(f)$; a set $D = \mathrm{dom}(\Gamma)$ (the *domain*); and, for each $f \in \tau$, a *cost function*, i.e., a function $f^\Gamma \colon D^{\mathrm{ar}(f)} \to \mathbb{Q} \cup \{+\infty\}$. Here, $+\infty$ is an extra element with the expected properties that for all $c \in \mathbb{Q} \cup \{+\infty\}$, we have $(+\infty) + c = c + (+\infty) = +\infty$ and $c < +\infty$ for every $c \in \mathbb{Q}$. Given a valued structure Γ with signature τ, for every $f \in \tau$ we define $\mathrm{dom}(f) := \{t \in D^{\mathrm{ar}(f)} \mid f^\Gamma(t) < +\infty\}$.

Let Γ be a valued structure with domain D and signature τ. The *valued constraint satisfaction problem* for Γ, denoted by $\mathrm{VCSP}(\Gamma)$, is the following computational problem.

An *instance* of $\mathrm{VCSP}(\Gamma)$ is a triple $I := (V, \phi, u)$ where V is a finite set of variables; ϕ is an expression of the form $\sum_{i=1}^{m} f_i(v_1^i, \ldots, v_{\mathrm{ar}(f_i)}^i)$, where $f_1, \ldots, f_m \in \tau$ and all the v_j^i are variables from V (each summand is called a τ-term); and u is a value from $\mathbb{Q}$. The task is to decide whether there exists an assignment $s \colon V \to D$, whose *cost*, defined as

$$\phi^\Gamma(s(v_1), \ldots, s(v_{|V|})) := \sum_{i=1}^{m} f_i^\Gamma(s(v_1^i), \ldots, s(v_{\mathrm{ar}(f_i)}^i))$$

is finite, and if so, whether there is one whose cost is at most u.

We remark that, given a valued structure Γ over a *finite* signature, the representation of the structure Γ is inessential for computational complexity as Γ is not part of the input.

Fractional Homomorphisms and Fractional Polymorphisms. Let C and D be two sets. A map $g \colon D^m \to C$ is called an m-ary *operation*. For any $m \in \mathbb{N}$, we denote by C^{D^m} the set of all maps $g \colon D^m \to C$.

Let Γ and Δ be valued structures with the same signature τ with domains C and D, respectively. A *fractional homomorphism* [44] from Δ to Γ is a discrete probability measure χ with a non-empty support on C^D such that for every function symbol $\gamma \in \tau$ and tuple $a \in D^{\mathrm{ar}(\gamma)}$, it holds that

$$\mathbb{E}_{h \sim \chi}[\gamma^\Gamma(h(a))] = \sum_{h \in \mathrm{Supp}(\chi)} \chi(h)\gamma^\Gamma(h(a)) \leq \gamma^\Delta(a), \tag{1}$$

where the functions h are applied component-wise. We write $\Delta \to_f \Gamma$ to indicate the existence of a fractional homomorphism from Δ to Γ.

The following proposition, proved for completeness in the full version [50], is adapted from [44], where it was proved in the case of finite-domain valued structure, and appears in [10], where it was stated for valued structures with arbitrary domains and for fractional homomorphisms with finite supports.

▶ **Proposition 1.** *Let Γ and Δ be valued structures over the same signature τ with domains C and D, respectively. Assume $\Delta \rightarrow_f \Gamma$. Let $V = \{v_1, \ldots, v_n\}$ be a set of variables and ϕ a sum of finitely many τ-terms with variables from V. For every $u \in \mathbb{Q}$, if there exists an assignment $s \colon V \to D$ such that $\phi^\Delta(s(v_1), \ldots, s(v_n)) \leq u$, then there exists an assignment $s' \colon V \to C$ such that $\phi^\Gamma(s'(v_1), \ldots, s'(v_n)) \leq u$. In particular, it holds that $\inf_C \phi^\Gamma \leq \inf_D \phi^\Delta$.*

Let Γ be a valued structure with domain C and signature τ. An m-ary *fractional polymorphism* of Γ is a discrete probability measure on C^{C^m} with a non-empty support such that for every $f \in \tau$ and tuples $a^1, \ldots, a^m \in C^{\mathrm{ar}(f)}$ it holds (with g applied componentwise below) that

$$\mathbb{E}_{g \sim \omega}[f^\Gamma(g(a^1, \ldots, a^m))] = \sum_{g \in C^{C^m}} \omega(g) f^\Gamma(g(a^1, \ldots, a^m)) \leq \frac{1}{m} \sum_{i=1}^m f^\Gamma(a^i).$$

Promise VCSPs. Let Γ and Δ be two valued structures over the same signature τ with domains C and D, respectively. We say that (Δ, Γ) is a *promise valued template* if there exists a fractional homomorphism from Δ to Γ. Given a promise valued template (Δ, Γ), the *promise valued constraint satisfaction problem* [34] for (Δ, Γ), denoted by $\mathrm{PVCSP}(\Delta, \Gamma)$, is the following computational problem.

An *instance I* of $\mathrm{PVCSP}(\Delta, \Gamma)$ is a triple $I := (V, \phi, u)$ where V is a finite set of variables; ϕ is an expression of the form $\sum_{i=1}^m f_i(v_1^i, \ldots, v_{\mathrm{ar}(f_i)}^i)$, where $f_1, \ldots, f_m \in \tau$ and all the v_j^i are variables from V; and u is a value from $\mathbb{Q}$.

The task is to output YES if there exists an assignment $s \colon V \to D$ with cost

$$\phi^\Delta(s(v_1), \ldots, s(v_{|V|})) := \sum_{i=1}^m f_i^\Delta(s(v_1^i), \ldots, s(v_{\mathrm{ar}(f_i)}^i)) \leq u$$

and output NO if *every* assignment $s' \colon V \to C$ has cost

$$\phi^\Gamma(s'(v_1), \ldots, s'(v_{|V|})) := \sum_{i=1}^m f_i^\Gamma(s'(v_1^i), \ldots, s'(v_{\mathrm{ar}(f_i)}^i)) \not\leq u.$$

Note that every valued structure Γ is fractionally homomorphic to itself and thus $\mathrm{VCSP}(\Gamma)$ is the same as $\mathrm{PVCSP}(\Gamma, \Gamma)$.

Let (Δ, Γ) be a promise valued template. We remark that if the common signature τ is finite then the representation of the template is inessential for the computational complexity of $\mathrm{PVCSP}(\Delta, \Gamma)$ as (Δ, Γ) is not part of the input.

Let $e_i^{(m)} \colon D^m \to D$ denote the m-ary projection on D onto the i-th coordinate. Let $\mathcal{J}_D^{(m)} := \{e_1^{(m)}, \ldots, e_m^{(m)}\}$, i.e., the set of all projections on D.

An m-ary *promise fractional polymorphism*[1] of a promise valued template (Δ, Γ) is a pair $\omega := (\omega_I, \omega_O)$ where ω_O is a discrete probability measure on C^{D^m} with a non-empty support and ω_I is a discrete probability measure with (finite) support $\mathrm{Supp}(\omega_I) = \mathcal{J}_D^{(m)}$ such that for every $f \in \tau$ and tuples $a^1, \ldots, a^m \in D^{\mathrm{ar}(f)}$ it holds that

$$\mathbb{E}_{g \sim \omega_O}[f^\Gamma(g(a^1, \ldots, a^m))] = \sum_{g \in \mathrm{Supp}(\omega)} \omega_O(g) f^\Gamma(g(a^1, \ldots, a^m))$$

$$\leq \sum_{i=1}^m \omega_I(e_i^{(m)}) f^\Delta(a^i) = \mathbb{E}_{e \sim \omega_I}[f^\Delta(e(a^1, \ldots, a^m))]. \quad (2)$$

---

[1] These are called weighted polymorphisms in [34].

Block-Symmetric Maps.   Let S_m be the symmetric group on $\{1, \ldots, m\}$. An m-ary map g is *fully 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 map g is *block-symmetric* if there exists a partition of the coordinates of g into blocks $B_1 \cup \cdots \cup B_k = [m]$ such that g is permutation-invariant within each block B_i. Let $\mathcal{P}_{sym}(g)$ be the set of all partitions into symmetric blocks of g. For $B_1 \cup \cdots \cup B_k \in \mathcal{P}_{sym}(m)$, we define $w(g, B_1 \cup \cdots \cup B_k) := \min_{1 \leq i \leq k} |B_i|$ and we define the *width* of g to be

$$w(g) := \max\{w(g, B_1 \cup \cdots \cup B_k) \mid B_1 \cup \cdots \cup B_k \in \mathcal{P}_{sym}(g)\}.$$

Block-symmetric operation with width 1 are fully symmetric operations.

An m-ary fractional polymorphism ω of a valued structure Γ is *block-symmetric* if there exists a partition of $[m]$ into blocks $B_1 \cup \cdots \cup B_k$ such that every operation in $\text{Supp}(\omega)$ is permutation-invariant within each coordinate block B_i.

Given a promise valued template (Δ, Γ), an m-ary promise fractional polymorphism $\omega = (\omega_I, \omega_O)$ of (Δ, Γ) is *block-symmetric* if

- there exists a partition of the coordinates of g into blocks $B_1 \cup \cdots \cup B_k = [m]$ such that every map in $\text{Supp}(\omega_O)$ is s permutation-invariant within each coordinate block B_i, and
- $\sum_{i \in B_j} \omega_I(e_i^{(m)}) = \frac{|B_j|}{m}$ for every $j \in \{1, \ldots, k\}$.

The proof of the following lemma can be found in the full version [50].

▶ **Lemma 2.** *Let (Δ, Γ) be a promise valued template and let $m \in \mathbb{N}$. If $\omega = (\omega_I, \omega_O)$ is an m-ary block symmetric promise fractional polymorphism of (Δ, Γ), then also $\omega' = (\omega_I', \omega_O)$, where $\omega_I'(e_i^{(m)}) = \frac{1}{m}$ for $1 \leq i \leq m$, is an m-ary block-symmetric promise fractional polymorphism of (Δ, Γ).*

In view of Lemma 2, we will assume without loss of generality that any m-ary block-symmetric promise fractional polymorphism $\omega = (\omega_I, \omega_O)$ is such that ω_I assign $\frac{1}{m}$ to each m-ary projection on the domain of Δ and we will identify ω with ω_O.

The Basic Linear Programming Relaxation.   Every VCSP over a finite domain has a natural linear programming relaxation. Let Δ be a valued structure with finite domain D and signature τ. Let I be an instance of VCSP(Δ) with set of variables $V = \{x_1, \ldots, x_d\}$, objective function $\phi(x_1, \ldots, x_d) = \sum_{j \in J} f_j(x_1^j, \ldots, x_{n_j}^j)$, with $f_j \in \tau$, $x^j = (x_1^j, \ldots, x_{n_j}^j) \in V^{n_j}$, for all $j \in J$ (the set J is finite and indexing the cost functions that are summands of ϕ), and a threshold $u \in \mathbb{Q}$.[2] Define the sets of variables as follows: $W_1 := \{\lambda_j(t) \mid j \in J \text{ and } t \in D^{n_j}\}$, $W_2 := \{\mu_{x_i}(a) \mid x_i \in V \text{ and } a \in D\}$, and $W := W_1 \cup W_2$. Then the *basic linear programming* (BLP) relaxation associated to I (see [44], [36], and references therein) is a linear program with variables W and is defined in Figure 1.

We remark that a solution to the BLP also satisfies the constraints $\sum_{t \in D^{n_j}} \lambda_j(t) = 1$ for all $j \in J$. If there is no feasible solution to the BLP then BLP$(I, \Delta) = +\infty$. For a finite-domain VCSP instance, the corresponding BLP relaxation can be computed in polynomial time.

The Affine Integer Programming Relaxation.   Let Δ be a valued structure with finite domain D and signature τ. Let I be an instance of VCSP(Δ) with set of variables $V = \{x_1, \ldots, x_d\}$, and objective function $\phi(x_1, \ldots, x_d) = \sum_{j \in J} f_j(x_1^j, \ldots, x_{n_j}^j)$, with $f_j \in \tau$, $x^j =$

---

[2] Note that the BLP relaxation does not depend on the threshold u.

$$\mathrm{BLP}(I,\Delta) := \min \sum_{j \in J} \sum_{t \in D^{n_j}} \lambda_j(t) f_j^{\Delta}(t)$$

subject to

$$\sum_{t \in D^{n_j}: t_\ell = a} \lambda_j(t) = \mu_{x_\ell^j}(a) \qquad \text{for all } j \in J, \ell \in \{1, \ldots, n_j\}, a \in D,$$

$$\sum_{a \in D} \mu_{x_i}(a) = 1 \qquad \text{for all } x_i \in V,$$

$$\lambda_j(t) = 0 \qquad \text{for all } j \in J, t \notin \mathrm{dom}(f_j),$$

$$0 \leq \lambda_j(t), \mu_{x_i}(a) \leq 1 \qquad \text{for all } \lambda_j(t) \in W_1, \mu_{x_i}(a) \in W_2.$$

Figure 1 BLP.

$(x_1^j, \ldots, x_{n_j}^j) \in V^{n_j}$, for all $j \in J$ (the set J is finite and indexing the cost functions that are summands of ϕ), and a threshold $u \in \mathbb{Q}$.[3] Define the sets of variables as follows: $R_1 := \{q_j(t) \mid j \in J \text{ and } t \in D^{n_j}\}$, $R_2 := \{r_{x_i}(a) \mid x_i \in V \text{ and } a \in D\}$, and $R := R_1 \cup R_2$. Then the *affine integer programming* (AIP) relaxation associated to I [16, 17] is an integer program with variables R and is defined in Figure 2.

$$\mathrm{AIP}(I,\Delta) := \min \sum_{j \in J} \sum_{t \in D^{n_j}} q_j(t) f_j^{\Delta}(t)$$

subject to

$$\sum_{t \in D^{n_j}: t_\ell = a} q_j(t) = r_{x_\ell^j}(a) \qquad \text{for all } j \in J, \ell \in \{1, \ldots, n_j\}, a \in D,$$

$$\sum_{a \in D} r_{x_i}(a) = 1 \qquad \text{for all } x_i \in V,$$

$$q_j(t) = 0 \qquad \text{for all } j \in J, t \notin \mathrm{dom}(f_j),$$

$$q_j(t), r_{x_i}(a) \in \mathbb{Z} \qquad \text{for all } q_j(t) \in R_1, r_{x_i}(a) \in R_2.$$

Figure 2 AIP.

We remark that a solution to the AIP also satisfies the constraints $\sum_{t \in D^{n_j}} \lambda_q(t) = 1$ for all $j \in J$. If there is no feasible solution to the AIP then $\mathrm{AIP}(I, \Delta) = +\infty$.

For a finite-domain VCSP instance, the corresponding AIP relaxation can be computed in polynomial time. Since the feasibility version of AIP can be solved in polynomial time [33, 16], (the optimisation version of) AIP can be solved in (oracle) polynomial time using an oracle for the feasibility version of the problem (see [31, Theorem 6.4.9]).

---

[3] Note that the AIP relaxation does not depend on the threshold u.

3   The Combined BLP and AIP Relaxation for PVCSPs

Let (Δ, Γ) be a promise valued template such that the domain of Δ is a *finite* set. We may solve $\mathrm{PVCSP}(\Delta, \Gamma)$ by using a *combination* of the BLP relaxation and the AIP relaxation of Δ, as proposed (for finite-domain promise non-valued) CSPs in [17], appropriately modified to the valued setting.

To describe such an algorithm, we need the following definition.

▶ **Definition 3.** *Let Δ be a valued structure with finite domain D and signature τ. Let us consider an instance $I := (V, \phi, u)$ of $\mathrm{VCSP}(\Delta)$ such that $\phi(x_1, \ldots, x_d) = \sum_{j \in J} f_j(x_1^j, \ldots, x_{n_j}^j)$. Assume that $\mathrm{BLP}(I, \Delta) \leq u$. We define $(\lambda^\star, \mu^\star)$ as follows.*

- *If there exists a relative interior point of the rational feasibility polytope of $\mathrm{BLP}(I, \Delta)$ with cost at most u,[4] then $(\lambda^\star, \mu^\star)$ is such a point;*
- *otherwise, $(\lambda^\star, \mu^\star)$ is defined to be a point from the relative interior of the optimal polytope of $\mathrm{BLP}(I, \Delta)$.[5]*

The refinement of $\mathrm{AIP}(I, \Delta)$ *with respect to* $(\lambda^\star, \mu^\star)$ *is the integer program* $\mathrm{AIP}^\star(I, \Delta)$ *obtained by adding to* $\mathrm{AIP}(I, \Delta)$ *the constraints*

$$q_j(t) = 0 \qquad \text{for every } j \in J, t \in D^{n_j} \text{ such that } \lambda_j^\star(t) = 0,$$
$$r_{x_i}(a) = 0 \qquad \text{for every } x_i \in V, a \in D \text{ such that } \mu_{x_i}^\star(a) = 0.$$

■ **Algorithm 1** The combined BLP + AIP Relaxation Algorithm for $\mathrm{PVCSP}(\Delta, \Gamma)$.

Input:
   $I := (V, \phi, u)$, a valid instance of $\mathrm{PVCSP}(\Delta, \Gamma)$
Output:
   YES if there exists an assignment $s\colon V \to \mathrm{dom}(\Delta)$ such that $\phi^\Delta(s(x_1), \ldots, s(x_{|V|})) \leq u$
   NO if there is no assignment $s\colon V \to \mathrm{dom}(\Gamma)$ such that $\phi^\Gamma(s(x_1), \ldots, s(x_{|V|})) \leq u$

$\mathrm{BLP}(I, \Delta)$;
if $\mathrm{BLP}(I, \Delta) \nleq u$ **then**
   output NO;
else
   $(\lambda^\star, \mu^\star)$, as in Definition 3;
   $\mathrm{AIP}^\star(I, \Delta) :=$ refinement of $\mathrm{AIP}(I, \Delta)$ with respect to $(\lambda^\star, \mu^\star)$, as in Definition 3;
   if $\mathrm{AIP}^\star(I, \Delta) \nleq u$ **then**
      output NO;
   else
      output YES;
   end
end

As our main result, we now present a sufficient condition under which Algorithm 1 correctly solves $\mathrm{PVCSP}(\Delta, \Gamma)$.

---

[4]   There is a polynomial-time algorithm [31, 17] that decides the existence of a relative interior point in the rational feasibility polytope of $\mathrm{BLP}(I, \Delta)$ with cost at most u and, in the case it exists, finds it.

[5]   Such a point can be found in polynomial time by applying the algorithm in [31, 17] to the feasibility linear program defined by adding to the constraints defining the feasibility polytope of $\mathrm{BLP}(I, \Delta)$ the additional constraint $\sum_{j \in J} \sum_{t \in D^{n_j}} \lambda_j(t) f_j^\Delta(t) = u$.

▶ **Theorem 4.** *Let* (Δ, Γ) *be a promise valued template such that* Δ *has a* finite domain. *Assume that for all* $L \in \mathbb{N}$ *there exists a block-symmetric promise fractional polymorphisms of* (Δ, Γ) *with arity* $2L + 1$ *having two symmetric blocks of size* $L + 1$ *and* L, *respectively. Then Algorithm 1 correctly solves* $\mathrm{PVCSP}(\Delta, \Gamma)$ *(in polynomial time).*

Note that in Theorem 4 the domain of the valued structure Γ can be finite or (countably) infinite.

To prove Theorem 4 we need to use a preliminary lemma and the notion of a *bimultiset-structures*. Let Δ be a valued τ-structure with domain D, let $L \in \mathbb{N}$, and let $B_1 \cup B_2$ any partition of $[2L + 1]$ such that $|B_1| = L + 1$ and $|B_2| = L$. The *bimultiset-structure* $\mathcal{B}_{B_1,B_2}^{2L+1}(\Delta)$ is the valued structure with domain $\left(\binom{D}{L+1} \right) \times \left(\binom{D}{L} \right)$ i.e., the set whose elements (α, β) are pairs of multisets of elements from D of size $L + 1$ and of size L, respectively. For every k-ary function symbol $f \in \tau$, and $(\alpha_1, \beta_1) \ldots, (\alpha_k, \beta_k) \in \left(\binom{D}{L+1} \right) \times \left(\binom{D}{L} \right)$ the function $f^{\mathcal{B}_{B_1,B_2}^{2L+1}(\Delta)}$ is defined as follows

$$f^{\mathcal{B}_{B_1,B_2}^{2L+1}(\Delta)}((\alpha_1, \beta_1) \ldots, (\alpha_k, \beta_k)) := \frac{1}{2L+1} \min_{\substack{t^1, \ldots, t^k \in D^{2L+1}: \\ \{t^\ell\}_{B_1} = \alpha_\ell, \{t^\ell\}_{B_2} = \beta_\ell}} \sum_{i=1}^{2L+1} f^\Delta(t_i^1, \ldots, t_i^k),$$

where $\{t^\ell\}_{B_1}$ denotes the multiset $\{t_i^\ell \mid i \in B_1\}$ and $\{t^\ell\}_{B_2}$ denotes the multiset $\{t_i^\ell \mid i \in B_2\}$.

The proof of the following lemma can be found in the full version [50].

▶ **Lemma 5.** *Let* (Δ, Γ) *be a promise valued template such that* Δ *has a* finite domain. *Let* $L \in \mathbb{N}$ *and assume that* (Δ, Γ) *has a block-symmetric promise fractional polymorphism of arity* $2L + 1$ *with two symmetric blocks* B_1 *and* B_2 *of size* $L + 1$ *and* L, *respectively. Then* $\mathcal{B}_{B_1,B_2}^{2L+1}(\Delta)$ *is fractionally homomorphic to* Γ.

Proof of Theorem 4. Let C be the (possibly infinite) domain of Γ and let D be the finite domain of Δ. Let τ be the common signature of Δ and Γ. Let I be an instance of $\mathrm{PVCSP}(\Delta, \Gamma)$ with variables $V = \{x_1, \ldots, x_n\}$, objective function $\phi(x_1, \ldots, x_n) = \sum_{j \in J} \gamma_j(x^j)$ where J is a finite set of indices, $\gamma_j \in \Gamma$, and $x^j \in V^{\mathrm{ar}(j)}$, and threshold u.

Assume that $\min_D \phi^\Delta \leq u$. Our goal is to show that Algorithm 1 outputs YES. Since $\min_D \phi^\Delta \leq u$ we have $\mathrm{BLP}(I, \Delta) \leq u$ and in particular we have that either $\mathrm{BLP}(I, \Delta) < u$, which by linearity implies the existence of a relative interior point in the feasibility polytope of $\mathrm{BLP}(I, \Delta)$ with value at most u; or $\mathrm{BLP}(I, \Delta) = u = \min_D \phi^\Delta$. In the first case, each coordinate of $(\lambda^\star, \mu^\star)$ is positive if and only if the same coordinate is positive at some point in the feasibility polytope of the BLP. Therefore, the feasibility lattice of $\mathrm{AIP}^\star(I, \Delta)$ includes every possible assignment which is in the support of some feasible solution to $\mathrm{BLP}(I, \Delta)$, including integral solutions and as a consequence $\mathrm{AIP}^\star(I, \Delta) \leq \min_D \phi^\Delta \leq u$. In the second case, each coordinate of $(\lambda^\star, \mu^\star)$ is positive if and only if the same coordinate is positive at some point in the optimal polytope of the BLP. Therefore, the feasibility lattice of $\mathrm{AIP}^\star(I, \Delta)$ includes every possible assignment which is in the support of some optimal solution to $\mathrm{BLP}(I, \Delta)$, including integral solutions and as a consequence $\mathrm{AIP}^\star(I, \Delta) \leq \min_D \phi^\Delta = u$. Thus, in both cases, $\mathrm{BLP}(I, \Delta) \leq u$ and $\mathrm{AIP}^\star(I, \Delta) \leq u$ and hence Algorithm 1 indeed outputs YES, as required.

In the other direction, we want to show (by contrapositive) that if Algorithm 1 outputs YES then $\inf_C \phi^\Gamma \leq u$. Thus, assume that $\mathrm{BLP}(I, \Delta) \leq u$ and $\mathrm{AIP}^\star(I, \Delta) \leq u$. Let $(\lambda^\star, \mu^\star)$ be as in Definition 3 and denote $\mathrm{BLP}^\star(I, \Delta) := \sum_{j \in J} \sum_{t \in D^{n_j}} \lambda_j^\star(t) f_j^\Delta(t)$; observe that $\mathrm{BLP}^\star(I, \Delta) \leq u$ by the definition of $(\lambda^\star, \mu^\star)$. Let $(q^\star, r^\star)$ be a solution to $\mathrm{AIP}^\star(I, \Delta)$ with objective value at most u. Let ℓ be a positive integer such that $\ell \cdot \lambda^\star$, and $\ell \cdot \mu^\star$ are both

integral, and let M be the maximum of the absolute values of the coordinates of both $q^\star$ and $r^\star$. Let us set $L := (M+1)\ell$. From $\mathrm{BLP}^\star(I, \Delta) \le u$ and $\mathrm{AIP}^\star(I, \Delta) \le u$ it immediately follows that

$$\frac{2(M+1)\ell}{2(M+1)\ell+1} \mathrm{BLP}^\star(I, \Delta) + \frac{1}{2(M+1)\ell+1} \mathrm{AIP}^\star(I, \Delta) \le u. \tag{3}$$

We claim that

$$\min_{\left(\binom{D}{L+1}\right) \times \left(\binom{D}{L}\right)} \phi^{\mathcal{B}^{2L+1}_{B_1,B_2}(\Delta)}$$
$$\le \frac{2(M+1)\ell}{2(M+1)\ell+1} \mathrm{BLP}^\star(I, \Delta) + \frac{1}{2(M+1)\ell+1} \mathrm{AIP}^\star(I, \Delta) \tag{4}$$

for all the partitions $B_1 \cup B_2 = [2L+1]$ such that $|B_1| = L+1$ and $|B_2| = L$. To prove the claim, let us define, for every $i \in \{1, \ldots, n\}$ and for every $a \in D$, the following nonnegative integers

$$W_{x_i,B_1}(a) := (M+1)\ell\mu^\star_{x_i}(a) + r^\star_{x_i}(a),$$
$$W_{x_i,B_2}(a) := (M+1)\ell\mu^\star_{x_i}(a).$$

(To check that $W_{x_i,B_1}(a)$ and $W_{x_i,B_2}(a)$ are nonnegative it is enough to observe that if $\mu^\star_{x_i}(a)$ is 0 then, by Definition 3, $r^\star_{x_i}(a)$ is also 0, otherwise $\mu^\star_{x_i}(a)$ is at least $\frac{1}{\ell}$, and the positivity of $W_{x_i,B_1}(a)$, $W_{x_i,B_2}(a)$ follows by the choice of M.) Observe that for every $i \in \{1, \ldots, n\}$ we have that

$$\sum_{a \in D} W_{x_i,B_1}(a) = (M+1)\ell \sum_{a \in D} \mu^\star_{x_i}(a) + \sum_{a \in D} r^\star_{x_i}(a) = (M+1)\ell + 1 = L+1,$$
$$\sum_{a \in D} W_{x_i,B_2}(a) = (M+1)\ell \sum_{a \in D} \mu^\star_{x_i}(a) = (M+1)\ell = L.$$

Let $\nu \colon V \to \left(\binom{D}{L+1}\right) \times \left(\binom{D}{L}\right)$ be the map defined, for every $x_i \in V$, by $\nu(x_i) = (\alpha_i, \beta_i)$, where α_i is the multiset of $\left(\binom{D}{L+1}\right)$ that contains $W_{x_i,B_1}(a)$ many occurrences of a, for every $a \in D$, and β_i is the multiset of $\left(\binom{D}{L}\right)$ that contains $W_{x_i,B_2}(a)$ many occurrences of a, for every $a \in D$.

Let f_j be a k-ary function symbol appearing as a term of the objective function ϕ. Let us define, for every $t \in D^k$, the following nonnegative integers

$$P_{j,B_1}(t) := (M+1)\ell\lambda^\star_j(t) + q^\star_j(t),$$
$$P_{j,B_2}(t) := (M+1)\ell\lambda^\star_j(t).$$

Observe that

$$\sum_{t \in D^k} P_{j,B_1}(t) = (M+1)\ell \sum_{t \in D^k} \lambda^\star_j(t) + \sum_{t \in D^k} q^\star_j(t) = (M+1)\ell + 1 = L+1,$$
$$\sum_{t \in D^k} P_{j,B_2}(t) = (M+1)\ell \sum_{t \in D^k} \lambda^\star_j(t) = (M+1)\ell = L.$$

We write now

$$\sum_{t \in D^k} P_{j,B_1}(t) f_j^\Delta(t) = \sum_{h=1}^{(M+1)\ell+1} f_j^\Delta(\zeta_1^h, \ldots, \zeta_k^h)$$

where $\zeta^1, \ldots, \zeta^{(M+1)\ell+1}$ are defined to be $(M+1)\ell+1$ elements of D^k such that $P_{j,B_1}(t)$ many of them are equal to t, for every $t \in D^k$; and

$$\sum_{t \in D^k} P_{j,B_2}(t) f_j^\Delta(t) = \sum_{h=1}^{(M+1)\ell} f_j^\Delta(\xi_1^h, \ldots, \xi_k^h)$$

where $\xi^1, \ldots, \xi^{(M+1)\ell}$ are defined to be $(M+1)\ell$ elements of D^k such that $P_{j,B_2}(t)$ many of them are equal to t, for every $t \in D^k$.

We obtain

$$\frac{2(M+1)\ell}{2(M+1)\ell+1}\lambda_j^\star(t)f_j^\Delta(t) + \frac{1}{2(M+1)\ell+1}q_j^\star(t)f_j^\Delta(t)$$

$$=\frac{1}{2(M+1)\ell+1}\left(\sum_{t \in D^k} P_{j,B_1}(t)f_j^\Delta(t) + \sum_{t \in D^k} P_{j,B_2}(t)f_j^\Delta(t)\right)$$

$$=\frac{1}{2L+1}\left(\sum_{h=1}^{L+1} f_j^\Delta(\zeta_1^h, \ldots, \zeta_k^h) + \sum_{h=1}^{L} f_j^\Delta(\xi_1^h, \ldots, \xi_k^h)\right)$$

$$\geq \frac{1}{2L+1} \min_{t^1, \ldots, t^k \in D^m : \{t^\ell\}_{B_1} = \zeta_\ell, \{t^\ell\}_{B_2} = \xi_\ell} \sum_{i=1}^{2L+1} f^\Delta(t_i^1, \ldots, t_i^k)$$

$$=f_j^{\mathcal{B}_{B_1,B_2}^{2L+1}(\Delta)}((\zeta_1, \xi_1), \ldots, (\zeta_k, \xi_k)) = f_j^{\mathcal{B}_{B_1,B_2}^{2L+1}(\Delta)}(\nu(x_1^j), \ldots, \nu(x_k^j)),$$

where the last equality follows because, for every $a \in D$, the number of a's in ζ_h is

$$\sum_{t \in D^k : t_h = a} P_{j,B_1}(t) = (M+1)\ell \sum_{t \in D^k : t_h = a} \lambda_j^\star(t) + \sum_{t \in D^k : t_h = a} q_j^\star(t)$$

$$=(M+1)\ell\mu_{x_h^j}^\star(a) + r_{x_h^j}^\star(a) = W_{x_h^j, B_1}(a),$$

and, for every $a \in D$, the number of a's in ξ_h is

$$\sum_{t \in D^k : t_h = a} P_{j,B_2}(t) = (M+1)\ell \sum_{t \in D^k : t_h = a} \lambda_j^\star(t) = (M+1)\ell\mu_{x_h^j}^\star(a) = W_{x_h^j, B_2}(a).$$

This proves the claim.

From Inequalities (3) and (4) it follows that for all partitions $B_1 \cup B_2 = [2L+1]$ such that $|B_1| = L+1$ and $|B_2| = L$ it holds

$$\min_{\left(\binom{D}{L+1}\right) \times \left(\binom{D}{L}\right)} \phi^{\mathcal{B}_{B_1,B_2}^{2L+1}(\Delta)} \leq u.$$

Moreover, since there exists a block-symmetric promise fractional polymorphisms of (Δ, Γ) of arity $2L+1$ having two symmetric blocks B_1 and B_2 with respective size $L+1$ and L, Lemma 5 implies the existence of a fractional homomorphism from $\mathcal{B}_{B_1,B_2}^{2L+1}(\Delta)$ to Γ. From Proposition 1 it follows that

$$\inf_C \phi^\Gamma \leq \min_{\left(\binom{D}{L+1}\right) \times \left(\binom{D}{L}\right)} \phi^{\mathcal{B}_{B_1,B_2}^{2L+1}(\Delta)} \leq u$$

and this concludes the proof. ◀

It is not difficult to see that the number of symmetric blocks and their size do not play a crucial role in the proof of Theorem 4 (and also in the proof of Lemma 5 in the full version [50]). Moreover, the notion of bimultiset-structure can be straightforwardly generalised to the notion of k-*multiset-structure with blocks of size* $b_1, \ldots, b_k$. In fact, Theorem 4 also holds in the case in which the block-symmetric promise fractional polymorphisms of (Δ, Γ) have an arbitrary number k of blocks each of arbitrary size b_i, given that $k, b_1, \ldots, b_k$ are the same for every arity $2L + 1$. For finite-domain PCSPs, the condition of having a block-symmetric promise polymorphism of arity $2L + 1$ is equivalent to the one of having a block-symmetric promise polymorphism of arity $2L + 1$ with two symmetric blocks of arity $L + 1$ and L, for every $L \in \mathbb{N}$ [18].

4   The Combined Relaxation with Sampling for PVCSPs

We use the notion of a sampling algorithm for a valued structure from [10].

▶ **Definition 6.** *Let Γ be a valued structure with domain C and finite signature τ. A* sampling algorithm *for Γ takes as input a positive integer d and computes a finite-domain valued structure Δ fractionally homomorphic to Γ such that, for every finite sum ϕ of τ-terms having at most d distinct variables, $V = \{x_1, \ldots, x_d\}$, and every $u \in \mathbb{Q}$, there exists a solution $s \colon V \to C$ with $\phi^\Gamma(s(x_1), \ldots, s(x_d)) \leq u$ if and only if there exists a solution $h' \colon V \to D$ with $\phi^\Delta(h'(x_1), \ldots, h'(x_d)) \leq u$. A sampling algorithm is called* efficient *if its running time is bounded by a polynomial in d. The finite-domain valued structure computed by a sampling algorithm is called a* sample.

▶ **Example 7.** A valued structure Γ with domain $\mathbb{Q}$ and signature τ is called *piecewise linear homogeneous (PLH)* if, for every $\gamma \in \tau$, the cost function γ^Γ is first-order definable over the structure $\mathfrak{L} = (\mathbb{Q}; \leq, 1, \{c\cdot\}_{c\in\mathbb{Q}})$ where
- $<$ is a relation symbol (i.e., a $\{0, \infty\}$-valued function symbol) of arity 2 and $<^\mathfrak{L}$ is the strict linear order of $\mathbb{Q}$,
- 1 is a constant symbol and $1^\mathfrak{L} := 1 \in \mathbb{Q}$, and
- $c\cdot$ is a unary function symbol for every $c \in \mathbb{Q}$ such that $(c\cdot)^\mathfrak{L}$ is the function $x \mapsto cx$, i.e., the multiplication by c.

If Γ is a PLH valued structure with a finite signature, then it admits an efficient sampling algorithm [10].

Let Γ be a valued structure with a finite signature that admits an efficient sampling algorithm. Observe that for every finite-domain valued structure Δ_d computed by such an efficient sampling algorithm, the pair (Δ_d, Γ) is a promise valued template. The following lemma, proved in the full version [50], follows from the definition of sampling algorithm for a valued structure.

▶ **Lemma 8.** *Let (Γ_1, Γ_2) be a promise valued template with a finite signature. Assume that Γ_1 admits an efficient sampling algorithm. If $\mathrm{PVCSP}(\Delta_d, \Gamma_2)$ is polynomial-time solvable for every finite-domain valued structure Δ_d computed by an efficient sampling algorithm for Γ_1, then $\mathrm{PVCSP}(\Gamma_1, \Gamma_2)$ is polynomial-time solvable.*

Using Theorem 4, it is possible to prove the following result.

▶ **Theorem 9.** *Let (Γ_1, Γ_2) be a promise valued template with a finite signature. Assume that Γ_1 admits an efficient sampling algorithm. Moreover, assume that (Γ_1, Γ_2) has a block-symmetric promise fractional polymorphism of arity $2L+1$ with two symmetric blocks of size $L + 1$ and L, respectively, for all $L \in \mathbb{N}$. Then $\mathrm{PVCSP}(\Gamma_1, \Gamma_2)$ is polynomial-time solvable.*

Note that in Theorem 9 (and in Lemma 8) *both* the respective domains of the valued structures Γ_1 and Γ_2 have arbitrary (countable) cardinality, that is, each of Γ_1 and Γ_2 can have a finite or an infinite domain. In the particular case in which $\Gamma_1 = \Gamma_2$, from Theorem 9 we obtain the following result for infinite-domain (non-promise) VCSPs.

▶ **Corollary 10.** *Let Γ be a valued structure with a finite signature that admits an efficient sampling algorithm. Assume that Γ has a block-symmetric fractional polymorphism of arity $2L + 1$ with two symmetric blocks of size $L + 1$ and L, respectively, for all $L \in \mathbb{N}$. Then VCSP(Γ) is polynomial-time solvable.*

Corollary 10 and the existence of an efficient sampling algorithm for PLH valued structures (see Example 7) imply tractability of certain convex PLH valued structures, as discussed in the full version [50].

References

1 Per Austrin and Johan Håstad. On the usefulness of predicates. *ACM Transactions on Computation Theory*, 5(1):1:1–1:24, 2013. `doi:10.1145/2462896.2462897`.

2 Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004. `doi:10.1023/B:MACH.0000033116.57574.95`.

3 Libor Barto, Jakub Bulín, Andrei Krokhin, and Jakub Opršal. Algebraic approach to promise constraint satisfaction. *CoRR*, 2019. `arXiv:1811.00970v3`.

4 Libor Barto and Marcin Kozik. Robustly solvable constraint satisfaction problems. *SIAM Journal on Computing*, 45(4):1646–1669, 2016. `doi:10.1137/130915479`.

5 Libor Barto and Michael Pinsker. Topology Is Irrelevant (In a Dichotomy Conjecture for Infinite Domain Constraint Satisfaction Problems). *SIAM Journal on Computing*, 49(2):365–393, 2020. `doi:10.1137/18M1216213`.

6 Manuel Bodirsky and Martin Grohe. Non-dichotomies in constraint satisfaction complexity. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 184–196. Springer Verlag, 2008. `doi:10.1007/978-3-540-70583-3_16`.

7 Manuel Bodirsky, Peter Jonsson, and Trung Van Pham. The complexity of phylogeny constraint satisfaction problems. *ACM Transactions on Computational Logic*, 18(3), 2017. `doi:10.1145/3105907`.

8 Manuel Bodirsky, Peter Jonsson, and Timo Von Oertzen. Essential convexity and complexity of semi-algebraic constraints. *Logical Methods in Computer Science*, 8(4), 2012. `doi:10.2168/LMCS-8(4:5)2012`.

9 Manuel Bodirsky and Jan Kára. The complexity of temporal constraint satisfaction problems. *Journal of the ACM*, 57(2), 2010. Article 9. `doi:10.1145/1667053.1667058`.

10 Manuel Bodirsky, Marcello Mamino, and Caterina Viola. Piecewise linear valued CSPs solvable by linear programming relaxation. *CoRR*, 2019. `arXiv:1912.09298`.

11 Manuel Bodirsky, Barnaby Martin, and Antoine Mottet. Discrete temporal constraint satisfaction problems. *Journal of the ACM*, 65(2):9:1–9:41, 2018. `doi:10.1145/3154832`.

12 Manuel Bodirsky, Barnaby Martin, Michael Pinsker, and András Pongrácz. Constraint satisfaction problems for reducts of homogeneous graphs. *SIAM Journal on Computing*, 48(4):1224–1264, 2019. `doi:10.1137/16M1082974`.

13 Manuel Bodirsky and Michael Pinsker. Schaefer's theorem for graphs. *Journal of the ACM*, 62(3):19:1–19:52, 2015. `doi:10.1145/2764899`.

14 Stephen P. Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. URL: `https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf`.

15 Joshua Brakensiek and Venkatesan Guruswami. Promise constraint satisfaction: Structure theory and a symmetric boolean dichotomy. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'18)*, pages 1782–1801. SIAM, 2018. `doi:10.1137/1.9781611975031.117`.

16 Joshua Brakensiek and Venkatesan Guruswami. An Algorithmic Blend of LPs and Ring Equations for Promise CSPs. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'19)*, pages 436–455. SIAM, 2019. `doi:10.1137/1.9781611975482.28`.

17 Joshua Brakensiek and Venkatesan Guruswami. Symmetric polymorphisms and efficient decidability of promise CSPs. In *Proceedings of the 31st ACM-SIAM Symposium on Discrete Algorithms (SODA'20)*, pages 297–304, 2020. `doi:10.1137/1.9781611975994.18`.

18 Joshua Brakensiek, Venkatesan Guruswami, Marcin Wrochna, and Stanislav Živný. The power of the combined basic LP and affine relaxation for promise CSPs. *eccc*, 2020. URL: `http://eccc.hpi-web.de/report/2020/004/`.

19 Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *Proceedings of the 58th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 319–330, 2017. `doi:10.1109/FOCS.2017.37`.

20 Andrei A. Bulatov, Andrei A. Krokhin, and Peter G. Jeavons. Classifying the complexity of constraints using finite algebras. *SIAM Journal on Computing*, 34(3):720–742, 2005. `doi:10.1137/S0097539700376676`.

21 Jakub Bulín, Andrei Krokhin, and Jakub Opršal. Algebraic approach to promise constraint satisfaction. In *Proceedings of the 51st ACM Annual Symposium on Theory of Computing (STOC'19)*, page 602–613. ACM, 2019. `doi:10.1145/3313276.3316300`.

22 Siu On Chan. Approximation resistance from pairwise-independent subgroups. *Journal of the ACM*, 63(3), 2016. Article No. 27. `doi:10.1145/2873054`.

23 Siu On Chan, James R. Lee, Prasad Raghavendra, and David Steurer. Approximate constraint satisfaction requires large LP relaxations. *Journal of the ACM*, 63(4):34:1–34:22, 2016. `doi:10.1145/2811255`.

24 Tsu-Wu J. Chou and George E. Collins. Algorithms for the solution of systems of linear Diophantine equations. *SIAM Journal on Computing*, 11:687–708, 1982. `doi:10.1137/0211057`.

25 Víctor Dalmau, Marcin Kozik, Andrei A. Krokhin, Konstantin Makarychev, Yury Makarychev, and Jakub Opršal. Robust algorithms with polynomial loss for near-unanimity csps. *SIAM Journal on Computing*, 48(6):1763–1795, 2019. `doi:10.1137/18M1163932`.

26 Víctor Dalmau and Andrei A. Krokhin. Robust satisfiability for csps: Hardness and algorithmic results. *ACM Transactions on Computation Theory*, 5(4):15:1–15:25, 2013. `doi:10.1145/2540090`.

27 Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: a study through Datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1999. `doi:10.1137/S0097539794266766`.

28 Miron Ficak, Marcin Kozik, Miroslav Olšák, and Szymon Stankiewicz. Dichotomy for symmetric Boolean PCSPs. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP'19)*, volume 132 of *LIPIcs*, pages 57:1–57:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. `doi:10.4230/LIPIcs.ICALP.2019.57`.

29 Mrinalkanti Ghosh and Madhur Tulsiani. From weak to strong linear programming gaps for all constraint satisfaction problems. *Theory of Computing*, 14(1):1–33, 2018. `doi:10.4086/toc.2018.v014a010`.

30 Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995. `doi:10.1145/227683.227684`.

31 Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2. Springer-Verlag Berlin Heidelberg, 1993. `doi:10.1007/978-3-642-78240-4`.

32 Peter Jonsson and Johan Thapper. Constraint satisfaction and semilinear expansions of addition over the rationals and the reals. *Journal of Computer and System Sciences*, 82(5):912–928, 2016. `doi:10.1016/j.jcss.2016.03.002`.

33 Ravindran Kannan and Achim Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM Journal on Computing*, 8(4):499–507, 1979. `doi:10.1137/0208040`.

34 Alexander Kazda. Minion homomorphisms give reductions between promise valued CSPs, 2020. In preparation.

35 Vladimir Kolmogorov, Andrei A. Krokhin, and Michal Rolínek. The complexity of general-valued csps. *SIAM Journal on Computing*, 46(3):1087–1110, 2017. `doi:10.1137/16M1091836`.

36 Vladimir Kolmogorov, Johan Thapper, and Stanislav Živný. The power of linear programming for general-valued CSPs. *SIAM Journal on Computing*, 44(1):1–36, 2015. `doi:10.1137/130945648`.

37 Pravesh K. Kothari, Raghu Meka, and Prasad Raghavendra. Approximating rectangles by juntas and weakly-exponential lower bounds for LP relaxations of CSPs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC'17)*, pages 590–603, 2017. `doi:10.1145/3055399.3055438`.

38 Dexter Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27:333–354, 1983. `doi:10.1016/0304-3975(82)90125-6`.

39 Marcin Kozik and Joanna Ochremiak. Algebraic properties of valued constraint satisfaction problem. In *Proceedings of the 42nd International Colloquium on Automata, Languages and Programming (ICALP'15)*, volume 9134 of *Lecture Notes in Computer Science*, pages 846–858. Springer, 2015. `doi:10.1007/978-3-662-47672-7_69`.

40 Gábor Kun, Ryan O'Donnell, Suguru Tamaki, Yuichi Yoshida, and Yuan Zhou. Linear programming, width-1 CSPs, and robust satisfaction. In *Proceedings of the 3rd Innovations in Theoretical Computer Science (ITCS'12)*, pages 484–495. ACM, 2012. `doi:10.1145/2090236.2090274`.

41 James R. Lee, Prasad Raghavendra, and David Steurer. Lower bounds on the size of semidefinite programming relaxations. In *Proceedings of the 47th Symposium on Theory of Computing (STOC'15)*, pages 567–576, 2015. `doi:10.1145/2746539.2746599`.

42 Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC'08)*, pages 245–254, 2008. `doi:10.1145/1374376.1374414`.

43 Thomas J. Schaefer. The Complexity of Satisfiability Problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC'78)*, pages 216–226. ACM, 1978. `doi:10.1145/800133.804350`.

44 Johan Thapper and Stanislav Živný. The power of linear programming for valued CSPs. In *Proceedings of the 53rd IEEE Annual Symposium on Foundations of Computer Science (FOCS'12)*, pages 669–678. IEEE Computer Society, 2012. `doi:10.1109/FOCS.2012.25`.

45 Johan Thapper and Stanislav Živný. The complexity of finite-valued CSPs. *Journal of the ACM*, 63(4), 2016. Article No. 37. `doi:10.1145/2974019`.

46 Johan Thapper and Stanislav Živný. The Power of Sherali-Adams Relaxations for General-Valued CSPs. *SIAM Journal on Computing*, 46(4):1241–1279, 2017. `doi:10.1137/16M1079245`.

47 Johan Thapper and Stanislav Živný. The limits of SDP relaxations for general-valued csps. *ACM Transactions on Computation Theory*, 10(3):12:1–12:22, 2018. `doi:10.1145/3201777`.

48 Madhur Tulsiani. CSP gaps and reductions in the Lasserre hierarchy. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC'09)*, pages 303–312. ACM, 2009. `doi:10.1145/1536414.1536457`.

49 Caterina Viola. *Valued Constraint Satisfaction Problems over Infinite Domains*. PhD thesis, Technische Universität Dresden, Dresden, Germany, 2020.

50 Caterina Viola and Stanislav Živný. The combined basic LP and affine IP relaxation for promise VCSPs on infinite domains. *CoRR*, 2020. `arXiv:2007.01779`.

51 Dmitriy Zhuk. A proof of CSP dichotomy conjecture. In *Proceedings of the 58th IEEE Annual Symposium on Foundations of Computer Science (FOCS'17)*, pages 331–342, 2017. `doi:10.1109/FOCS.2017.38`.