The limits of SDP relaxations for general-valued CSPs

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

It has been shown that for a general-valued constraint language Γ the following statements are equivalent: (1) any instance of VCSP(Γ) can be solved to *optimality* using a *constant level* of the Sherali-Adams LP hierarchy; (2) any instance of VCSP(Γ) can be solved to optimality using the *third level* of the Sherali-Adams LP hierarchy; (3) the support of Γ satisfies the "*bounded width condition*", i.e., it contains weak near-unanimity operations of all arities.

We show that if the support of Γ violates the bounded width condition then not only is VCSP(Γ) not solved by a constant level of the Sherali-Adams LP hierarchy but it requires linear levels of the Lasserre SDP hierarchy (also known as the sum-of-squares SDP hierarchy). For Γ corresponding to linear equations in an Abelian group, this result follows from existing work on inapproximability of Max-CSPs. By a breakthrough result of Lee, Raghavendra, and Steurer [STOC'15], our result implies that for any Γ whose support violates the bounded width condition no SDP relaxation of polynomial-size solves VCSP(Γ).

We establish our result by proving that various reductions preserve exact solvability by the Lasserre SDP hierarchy (up to a constant factor in the level of the hierarchy). Our results hold for general-valued constraint languages, i.e., sets of functions on a fixed finite domain that take on rational or infinite values, and thus also hold in notable special cases of $\{0, \infty\}$-valued languages (CSPs), $\{0, 1\}$-valued languages (Min-CSPs/Max-CSPs), and $\mathbb{Q}$-valued languages (finite-valued CSPs).

CCS Concepts: • **Theory of computation** → **Problems, reductions and completeness**;

Additional Key Words and Phrases: discrete optimisation, valued constraint satisfaction problems, convex relaxations, SDP, Lasserre hierarchy, sum of squares

ACM Reference Format:
Johan Thapper and Stanislav Živný. 2018. The limits of SDP relaxations for general-valued CSPs. *ACM Trans. Comput. Theory* 9, 4, Article 39 (March 2018), 24 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

1.1 CSPs and exact solvability

Constraint satisfaction problems (CSPs) constitute a broad class of computational problems that involve assigning labels to variables subject to constraints to be satisfied and/or optimised, as nicely explained in a survey by Hell and Nešetřil [28]. One line of research focuses on CSPs parametrised by a set of (possibly weighted) relations known as a constraint

Authors' addresses: Johan Thapper, Université Paris-Est, Marne-la-Vallée, Paris, France, thapper@u-penm.fr; Stanislav Živný, University of Oxford, Oxford, UK, standa.zivny@cs.ox.ac.uk.

1942-3454/2018/3-ART39 $15.00
https://doi.org/0000001.0000001

language [29]. In their influential paper, Feder and Vardi conjectured that for decision CSPs every constraint language gives rise to a class of problems that belongs to P or is NP-complete [20]. The dichotomy conjecture of Feder and Vardi has been verified in several important special cases by Schaefer [47], Hell and Nešetřil [27], Bulatov [8, 11], and Barto, Kozik, and Niven [6] mostly using the so-called algebraic approach [4, 10]. Remarkably, the dichotomy conjecture has recently been solved independently by Bulatov [9] and Zhuk [58], respectively.

Using concepts from the extensions of the algebraic approach to optimisation problems [17], the exact solvability of purely optimisation CSPs, known as finite-valued CSPs, has been established by the authors [50] (these include Min/Max-CSPs as a special case). Putting together decision and optimisation problems in one framework, the exact complexity of so-called general-valued CSPs has been established, modulo the (now proved) classification of decision CSPs, by the works of Kozik and Ochremiak [35] and Kolmogorov, Krokhin, and Rolínek [31]. A result that proved useful when classifying both finite-valued and general-valued CSPs is an algebraic characterisation of the power of the basic linear programming relaxation for decision CSPs [36] and general-valued CSPs [32].

1.2 Approximation

Convex relaxations, such as linear programming (LP) and semidefinite programming (SDP), have long been powerful tools for designing efficient exact and approximation algorithms [55, 56]. In particular, for many combinatorial problems, the introduction of semidefinite programming relaxations allowed for a new structural and computational perspective [1, 23, 30]. The Lasserre SDP hierarchy [39] is a sequence of semidefinite relaxations for certain 0-1 polynomial programs, each one more constrained than the previous one. The kth level of the Lasserre SDP hierarchy requires any set of k variables of the relaxation, which live in a finite-dimensional real vector space, to be consistent in a very strong sense. The kth level of the hierarchy can be solved in time $L \cdot n^{O(k)}$, where n is the number of variables and L is the length of a binary encoding of the input. If an integer program has n variables then the nth level of the Lasserre SDP hierarchy is tight, i.e., the only feasible solutions are convex combinations of integral solutions. The Lasserre SDP hierarchy is similar in spirit to the Lovász-Schrijver SDP hierarchy [43] and the Sherali-Adams LP hierarchy [49], but the Lasserre SDP hierarchy is stronger [40].

An important line of research, going back to a seminal work of Yannakakis [57], focuses on proving lower bounds on the size of LP formulations. Chan, Lee, Raghavendra, and Steurer [14] showed that Sherali-Adams LP relaxations are universal for Max-CSPs in the sense that for every polynomial-size LP relaxation of a Max-CSP instance I there is a constant level of the Sherali-Adams LP hierarchy of I that achieves the same approximation guarantees. This result has been improved to subexponential-size LP relaxations by Kothari, Meka, and Raghavednra [33]. Moreover, Ghosh and Tulsiani [22] have shown that in fact the basic LP relaxation enjoys the same universality property (among super-constant levels of the Sherali-Adams LP hierarchy). For related work on the integrality gaps for the Sherali-Adams LP and Lovász-Schrijver SDP hierarchies, we refer the reader to [15, 16, 48] and the references therein.

Recent years have seen some remarkable progress on lower bounds for the Lasserre SDP hierarchy. Schoenebeck showed that certain problems require linear levels of the Lasserre SDP hierarchy [48]. In particular, Schoenebeck showed, among other things, that cn levels, for some constant $0 < c < 1$, of the Lasserre SDP hierarchy cannot prove that certain Max-CSPs (corresponding to equations on the Boolean domain) are unsatisfiable [48]. Tulsiani extended this work to Max-CSPs corresponding to equations over Abelian groups of prime orders [53]. Finally, Chan extended this to Max-CSPs corresponding to equations over Abelian groups of arbitrary size [13]. In a recent breakthrough, Lee, Raghavendra, and Steurer [42] showed that the Lasserre SDP relaxations are universal for Max-CSPs in the sense that for every polynomial-size SDP relaxation of a Max-CSP instance I there is a constant level of the Lasserre SDP hierarchy of I that achieves the same approximation guarantees. One of the many ingredients of the proof in [42] is to view the Lasserre SDP hierarchy as the Sum-of-Squares algorithm [38], which relates to proof complexity [45]. (In fact, Schoenebeck's above-mentioned result had independently been obtained by Grigoriev [24] using this view.)

1.3 Bounded width condition

We now informally describe the bounded width condition (BWC). A set of operations on a fixed finite domain satisfies the BWC if it contains "weak near-unanimity" operations of all possible arities. An operation is called a weak near-unanimity operation if it is symmetric when all the arguments but one are the same. (A formal definition is given in Section 3.1.) An example of a ternary weak-near unanimity operation is a majority operation, which satisfies $f(x,x,y) = f(x,y,x) = f(y,x,x) = x$ for all x and y. Polymorphisms [10], which are at the heart of the algebraic approach to CSPs, are operations that combine satisfying assignments to a CSP instance and produce a new satisfying assignment. We say that a CSP instance I satisfies the BWC if the set of all polymorphisms of I satisfies the BWC.

In an important series of papers by Maróti and McKenzie [44], Larose and Zádori [37], Barto and Kozik [4], and Bulatov [12], it was established that the BWC captures precisely the decision CSPs that are solved by Datalog, a natural and well-studied local propagation algorithm [20].

1.4 Contributions

In our previous work [51] (which we refer the reader to for more information and background), we studied the power of the Sherali-Adams LP hierarchy for exact solvability of general-valued CSPs. In particular, we have shown in [51] that general-valued CSPs that are solved exactly by a constant level of the Sherali-Adams LP hierarchy are precisely those general-valued CSPs that satisfy the BWC. In more detail, fractional polymorphisms of a general-valued CSP instance I are probability distributions over polymorphisms of I that in a sense preserve the weighted relations of I. For a constraint language Γ, we denote by $\mathrm{supp}(\Gamma)$ the set of operations that appear in the support of some fractional polymorphism of Γ. (Formal definitions are given in Section 2.) The following theorem is the main result of [51].

Theorem 1.1 ([51, Theorem 3.3]). *Let Γ be a general-valued constraint language of finite size. The following are equivalent:*

(i) $\mathrm{VCSP}(\Gamma)$ *is solved by a* constant *level of the Sherali-Adams LP hierarchy.*
(ii) $\mathrm{VCSP}(\Gamma)$ *is solved by the* third *level of the Sherali Adams LP hierarchy.*
(iii) $\mathrm{supp}(\Gamma)$ *satisfies the BWC.*

In this follow-up work, we study the power of the Lasserre SDP hierarchy for exact solvability of general-valued CSPs. As our main contribution (stated as Theorem 3.5), we show that general-valued CSPs that are not solved by a constant level of the Sherali-Adams LP hierarchy require linear levels of the Lasserre SDP hierarchy. As a direct corollary, the results of Lee, Raghavendra, and Steurer [42] imply that such general-valued CSPs are not solved by *any* polynomial-size SDP relaxation.

In order to prove our result, we will strengthen the proof of the implication $(i) \implies (iii)$ of Theorem 1.1. The idea is to show that if $\text{supp}(\Gamma)$ violates the BWC, then Γ can *simulate* linear equations in some Abelian group. It suffices to show that linear equations require linear levels of the Lasserre SDP hierarchy and that the simulation preserves exact solvability by the Lasserre SDP hierarchy (up to a constant factor in the level of the hierarchy). As discussed before, the former is actually known (in a stronger sense of inapproximability of linear equations) [13, 24, 48, 53] and will be discussed in Section 3.4. Our contribution is proving the latter. While the simulation involves only local replacements via gadgets, it needs to be done with care. In particular, we emphasise that the simulation involves steps, such as going to the core and interpretations, which are commonly used in the algebraic approach to CSPs but not in the literature on convex relaxations and approximability of CSPs [53]. Indeed, the algebraic approach to CSPs gives the right tools for the intuitive (but non-trivial to capture formally) meaning of "simulating equations".

1.5 Related work

In our main result, Theorem 3.5, the BWC is required to hold, as in Theorem 1.1, for the support of the fractional polymorphisms [17] of the general-valued CSPs. This is a natural requirement since polymorphisms do not capture the complexity of general-valued CSPs but the fractional polymorphisms do so [17, 31].

The BWC was also shown [5, 18] to capture precisely the Max-CSPs that can be robustly approximated, as conjectured by Guruswami and Zhou [25]. This work is similar to ours but different. In particular, Dalmau and Krokhin showed [18] that various reductions preserve robust approximability of equations, and thus showing that Max-CSPs not satisfying the BWC cannot be robustly approximated, assuming P≠NP and relying on Håstad's inapproximability results for linear equations [26]. (Barto and Kozik [5] then showed that Max-CSPs satisfying the BWC can be robustly approximated.) However, note that linear equations *can* be solved exactly using Gaussian elimination and thus this result is not applicable in our setting. Our result, on the other hand, shows that various reductions preserve exact solvability of equations by a *particular* algorithm (the Lasserre SDP hierarchy) independently of P vs. NP. Moreover, the pp-definitions and pp-interpretations used in [5, 18] were required to be equality-free. We prove that our reductions are well-behaved without this assumption.

Our main result is incomparable with the results obtained by Schoenebeck [48], Tulsiani [53], and Chan [13] in the context of (in)approximability. On the one hand, our results capture exact solvability rather than approximability. On the other hand, we give a stronger result as our result applies to general-valued CSPs rather than only to Max-CSPs or finite-valued CSPs. General-valued CSPs are more expressive than their special cases Max-CSPs and finite-valued CSPs since general-valued CSPs also include decision CSPs as a special case and thus can use "hard" or "strict" constraints. The results on Max-CSPs [13, 48, 53] were extended by (problem-specific) reductions to some problems (such as Vertex Cover) which are not captured by Max-CSPs but are captured by general-valued CSPs. Our results are not problem specific and apply to *all* general-valued CSPs. In particular, we give a *complete* characterisation of which *general-valued* CSPs are solved exactly by the Lasserre SDP hierarchy.

Our results generalise some of the results of Dawar and Wang [19] and Atserias and Ochremiak [3]. In particular, using definability in counting logics, Dawar and Wang have established our main result in the special case of $\mathbb{Q}$-valued languages, i.e., for finite-valued CSPs [19]. Moreover, using tools from proof complexity, Atserias and Ochremiak have established (among other things) our main result in the special case of $\{0, \infty\}$-valued languages, i.e., for (decision) CSPs [3].

2 PRELIMINARIES

2.1 General-valued CSPs

We first describe the framework of general-valued constraint satisfaction problems (VCSPs). Let $\overline{\mathbb{Q}} = \mathbb{Q} \cup \{\infty\}$ denote the set of rational numbers extended with positive infinity. Throughout the paper, let D be a fixed finite set of size at least two, also called a *domain*; we call the elements of D *labels*. We denote by $[n]$ the set $\{1, \dots, n\}$.

Definition 2.1. An r-ary *weighted relation* over D is a mapping $\phi : D^r \to \overline{\mathbb{Q}}$. We write $\operatorname{ar}(\phi) = r$ for the arity of ϕ.

A weighted relation $\phi\colon D^r \to \{0, \infty\}$ can be seen as the (ordinary) relation $\{\mathbf{x} \in D^r \mid \phi(\mathbf{x}) = 0\}$. We will use both viewpoints interchangeably.

For any r-ary weighted relation ϕ, we denote by $\operatorname{Feas}(\phi) = \{\mathbf{x} \in D^r \mid \phi(\mathbf{x}) < \infty\}$ the underlying r-ary *feasibility relation*, and by $\operatorname{Opt}(\phi) = \{\mathbf{x} \in \operatorname{Feas}(\phi) \mid \forall \mathbf{y} \in D^r : \phi(\mathbf{x}) \le \phi(\mathbf{y})\}$ the r-ary *optimality relation*, which contains the tuples on which ϕ is minimised.

Definition 2.2. Let $V = \{x_1, \dots, x_n\}$ be a set of variables. A *valued constraint* over V is an expression of the form $\phi(\mathbf{x})$ where ϕ is a weighted relation and $\mathbf{x} \in V^{\operatorname{ar}(\phi)}$. The tuple $\mathbf{x}$ is called the *scope* of the constraint.

Definition 2.3. An instance I of the *valued constraint satisfaction problem* (VCSP) is specified by a finite set $V = \{x_1, \dots, x_n\}$ of variables, a finite set D of labels, and an *objective function* ϕ_I expressed as follows:

$$\phi_I(x_1, \dots, x_n) = \sum_{i=1}^{q} \phi_i(\mathbf{x}_i),$$

where each $\phi_i(\mathbf{x}_i)$, $1 \le i \le q$, is a valued constraint. Each constraint may appear multiple times in I. An *assignment* to I is a map $\sigma\colon V \to D$. The goal is to find an assignment that minimises the objective function.

For a VCSP instance I, we write $\operatorname{Val}_{\mathrm{VCSP}}(I, \sigma)$ for $\phi_I(\sigma(x_1), \dots, \sigma(x_n))$, and $\operatorname{Opt}_{\mathrm{VCSP}}(I)$ for the minimum of $\operatorname{Val}_{\mathrm{VCSP}}(I, \sigma)$ over all assignments σ.

An assignment σ with $\operatorname{Val}_{\mathrm{VCSP}}(I, \sigma) < \infty$ is called *satisfying*. An assignment σ with $\operatorname{Val}_{\mathrm{VCSP}}(I, \sigma) = \operatorname{Opt}_{\mathrm{VCSP}}(I)$ is called *optimal*.

A VCSP instance I is called *satisfiable* if there is a satisfying assignment to I. Constraint satisfaction problems (CSPs) are a special case of VCSPs with (unweighted) relations with the goal to determine the existence of a satisfying assignment.

A *general-valued constraint language* (or just a *constraint language* for short) over D is a set of weighted relations over D. As is common in the (V)CSP literature, we will focus on constraint languages of *finite* size. We denote by $\operatorname{VCSP}(\Gamma)$ the class of all VCSP instances in which the weighted relations are all contained in Γ. A constraint language Γ is called *crisp* if Γ contains only (unweighted) relations. For a crisp language Γ, $\operatorname{VCSP}(\Gamma)$ is equivalent to the well-studied (decision) $\operatorname{CSP}(\Gamma)$ [28]. We remark that for $\{0, 1\}$-valued constraint languages, $\operatorname{VCSP}(\Gamma)$ is also known as Min-CSP(Γ) or Max-CSP(Γ) (since for exact solvability these are equivalent).

For a constraint language Γ, let $\operatorname{ar}(\Gamma)$ denote $\max\{\operatorname{ar}(\phi) \mid \phi \in \Gamma\}$.

Example 2.4. Let $D = \{0, 1\}$. We define several weighted relations.

- $\phi_{\mathsf{cut}}(x, y) = 1$ if $x + y = 0 \pmod 2$ and $\phi_{\mathsf{cut}}(x, y) = 0$ otherwise.
- $\phi_{\mathsf{mc}}(x, y) = 1$ if $x + y = 1 \pmod 2$ and $\phi_{\mathsf{mc}}(x, y) = 0$ otherwise.
- For $a \in D$, $c_a(x) = 0$ if $x = a$ and $c_a(x) = \infty$ otherwise.
- For $a \in D$, $R_a(x, y, z) = 0$ if $x + y + z = a \pmod 2$ and $R_a(x, y, z) = \infty$ otherwise.

Let $\Gamma_{\mathsf{cut}} = \{\phi_{\mathsf{cut}}, c_0, c_1\}$, $\Gamma_{\mathsf{mc}} = \{\phi_{\mathsf{mc}}\}$, and $\Gamma_{\mathsf{eq}} = \{R_0, R_1\}$. Then, VCSP($\Gamma_{\mathsf{cut}}$) corresponds to the (s, t)-Min-Cut problem, VCSP(Γ_{mc}) corresponds to the Min-UnCut problem, and finally VCSP(Γ_{eq}) corresponds to the feasibility problem for systems of linear questions in three variables over $\mathbb{Z}_2$.

2.2 Fractional polymorphisms

We next define fractional polymorphisms, which are algebraic properties known to capture the computational complexity of the underlying class of VCSPs.

Given an r-tuple $\mathbf{x} \in D^r$, we denote its ith entry by $\mathbf{x}[i]$ for $1 \le i \le r$. A mapping $f \colon D^m \to D$ is called an m-ary *operation* on D; f is *idempotent* if $f(x, \ldots, x) = x$. We apply an m-ary operation f to m r-tuples $\mathbf{x}_1, \ldots, \mathbf{x}_m \in D^r$ coordinatewise, that is,

$$f(\mathbf{x}_1, \ldots, \mathbf{x}_m) = (f(\mathbf{x}_1[1], \ldots, \mathbf{x}_m[1]), \ldots, f(\mathbf{x}_1[r], \ldots, \mathbf{x}_m[r])).$$

Definition 2.5. Let ϕ be a weighted relation on D and let f be an m-ary operation on D. We call f a *polymorphism of* ϕ if, for any $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \text{Feas}(\phi)$, we have that $f(\mathbf{x}_1, \ldots, \mathbf{x}_m) \in \text{Feas}(\phi)$.

For a constraint language Γ, we denote by $\text{Pol}(\Gamma)$ the set of all operations which are polymorphisms of all $\phi \in \Gamma$. We write $\text{Pol}(\phi)$ for $\text{Pol}(\{\phi\})$.

The intuition behind polymorphisms is that if $\text{Pol}(\Gamma)$ contains only "trivial" operations (such as projections, cf. Example 2.8) then checking for a satisfiable solution to an instance of VCSP(Γ) is NP-hard, whereas if $\text{Pol}(\Gamma)$ contains a "non-trivial" operation then this can be done in polynomial time. This intuition was formalised in the algebraic dichotomy conjecture [10] recently proved in [9, 58].

The following notions are known to capture the complexity of general-valued constraint languages [17, 35] and will also be important in this paper. A probability distribution ω over the set of m-ary operations on D is called an m-ary *fractional operation*. For a fractional operation ω, "$f \sim \omega$" means that f is a random operation (of the same arity as ω) drawn according to the distribution ω. We define $\text{supp}(\omega)$ to be the set of operations assigned positive probability by ω. We denote by avg the average operator; i.e., $\text{avg}\{a_1, \ldots, a_m\} = (1/m) \sum_{i=1}^{m} a_i$.

Definition 2.6. Let ϕ be a weighted relation on D and let ω be an m-ary fractional operation on D. We call ω a *fractional polymorphism of* ϕ if $\text{supp}(\omega) \subseteq \text{Pol}(\phi)$ and for any $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \text{Feas}(\phi)$, we have

$$\mathop{\mathbb{E}}_{f \sim \omega} [\phi(f(\mathbf{x}_1, \ldots, \mathbf{x}_m))] \;\le\; \text{avg}\{\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_m)\}.$$

For a general-valued constraint language Γ, we denote by $\text{fPol}(\Gamma)$ the set of all fractional operations which are fractional polymorphisms of all weighted relations $\phi \in \Gamma$. We write $\text{fPol}(\phi)$ for $\text{fPol}(\{\phi\})$.

In case of fractional polymorphisms, the important operations are those that are assigned positive probability.

Definition 2.7. Let Γ be a general-valued constraint language on D. We define

$$\operatorname{supp}(\Gamma) \;=\; \bigcup_{\omega\in \operatorname{fPol}(\Gamma)} \operatorname{supp}(\omega).$$

The intuition behind fractional polymorphisms is that if $\operatorname{supp}(\Gamma)$ contains only "trivial" operations then finding an optimal solution to an instance of VCSP(Γ) is NP-hard, whereas if $\operatorname{supp}(\Gamma)$ contains a "non-trivial" operation then this can be done in polynomial time. This intuition was formalised in [17, 35] and proved in [31]. We now give some examples.

Example 2.8. Let $D = \{0,1\}$ and recall the constraint languages Γ_{cut}, Γ_{mc}, and Γ_{eq} defined in Example 2.4.

Consider the two binary operations min and max on D that return the smaller and the larger of its two arguments, respectively. The constraint language Γ_{cut} admits ω_{sub} as a fractional polymorphism, where $\omega_{\mathsf{sub}}(\min) = \omega_{\mathsf{sub}}(\max) = \frac{1}{2}$. In fact, the set of all weighted relations that admit ω_{sub} as a fractional polymorphism is precisely the class of *submodular* functions. Note that both min and max are binary commutative operations. By [32, Corollary 6], the fact that $\operatorname{supp}(\Gamma_{\mathsf{cut}})$ contains a binary commutative operation implies that VCSP(Γ_{cut}) is solved by the first level of the Sherali-Adams LP hierarchy.

Since VCSP(Γ_{mc}) is essentially the problem Min-UnCut, it is NP-hard. This fact can also be deduced from looking at the binary fractional polymorphisms of Γ_{mc}. For $i \in \{1,2\}$, we denote by π_i the binary operation that returns its ith argument (these are known as projections). Also, for $i \in \{1,2\}$, we denote by π'_i the binary operation defined by $\pi'_i(0,0) = 1$, $\pi'_i(1,1) = 0$, and $\pi'_i(x,y) = \pi_i(x,y)$ for $x \neq y$. For any $0 \leq p \leq \frac{1}{2}$, the binary fractional operation ω_p defined by $\omega_p(\pi_1) = \omega_p(\pi_2) = p$ and $\omega_p(\pi'_1) = \omega_p(\pi'_2) = \frac{1}{2} - p$ is a fractional polymorphism of Γ_{mc}. It is not hard to show that all fractional polymorphisms of Γ_{mc} are of this form, and hence there is no binary commutative operation in $\operatorname{supp}(\Gamma_{\mathsf{mc}})$. It then follows from [32] that VCSP(Γ_{mc}) is not solved by the first level of the Sherali-Adams LP hierarchy, and by the results in [50] that VCSP(Γ_{mc}) is NP-hard.

Finally, let m denote the ternary operation defined by $m(x,y,z) = x+y+z \pmod 2$. The constraint language Γ_{eq} admits m as a polymorphism and thus any instance of VCSP(Γ_{eq}) can be solved in polynomial time [29]. However, Pol(Γ_{eq}) does not contain any weak near-unanimity operation of arity 3 (defined in Section 3.1). It therefore follows from Thereom 3.5 of this paper that VCSP(Γ_{eq}) requires linear levels of the Lasserre SDP hierarchy.

2.3 Expressibility, interpretability, and simulation

In this section we formally define the various types of gadget constructions needed to establish our main result. We also introduce the important notion of cores.

Definition 2.9. We say that an m-ary weighted relation ϕ is *expressible* over a general-valued constraint language Γ if there exists an instance I of VCSP(Γ) with variables $x_1,\ldots,x_m,v_1,\ldots,v_p$ such that

$$\phi(x_1,\ldots,x_m) = \min_{v_1,\ldots,v_p} \phi_I(x_1,\ldots,x_m,v_1,\ldots,v_p).$$

For a fixed set D, let $\phi^D_=$ denote the binary equality relation $\{(x,x) \mid x\in D\}$. We denote by $\langle\Gamma\rangle$ the set of weighted relations obtained by taking the closure of $\Gamma\cup\{\phi^D_=\}$, where D is the domain of Γ, under expressibility, the Feas and Opt operations, scaling by nonnegative rational constants, and addition of rational constants.

Definition 2.10. Let Γ and Δ be general-valued constraint languages on domain D and D', respectively. We say that Δ *has an interpretation in* Γ with parameters (d, S, h) if there exists a $d \in \mathbb{N}$, a set $S \subseteq D^d$, and a surjective map $h : S \to D'$ such that $\langle \Gamma \rangle$ contains the following weighted relations:

- $\phi_S : D^d \to \overline{\mathbb{Q}}$ defined by $\phi_S(\mathbf{x}) = 0$ if $\mathbf{x} \in S$ and $\phi_S(\mathbf{x}) = \infty$ otherwise;
- $h^{-1}(\phi_=^{D'})$; and
- $h^{-1}(\phi_i)$, for every weighted relation $\phi_i \in \Delta$,

where $h^{-1}(\phi_i)$, for an m-ary weighted relation ϕ_i, is the dm-ary weighted relation on D defined by $h^{-1}(\phi_i)(\mathbf{x}_1, \ldots, \mathbf{x}_m) = \phi_i(h(\mathbf{x}_1), \ldots, h(\mathbf{x}_m))$, for all $\mathbf{x}_1, \ldots, \mathbf{x}_m \in S$.

It follows from Definition 2.10 that interpretations compose.

Remark 1. A weighted relation being expressible over $\Gamma \cup \{\phi_=^D\}$ is the analogue of a relation being definable by a *primitive positive (pp)* formula (using existential quantification and conjunction) over a relational structure with equality. Indeed, when Γ is crisp, the two notions coincide. Also, for a crisp Γ the notion of an interpretation coincides with the notion of a *pp-interpretation* for relational structures [7].

For a subset of the domain $S \subseteq D$, we define the restriction of a language Γ on S as follows.

Definition 2.11. Let Γ be a general-valued constraint language with domain D and let $S \subseteq D$. The *sub-language* $\Gamma[S]$ *of* Γ *induced by* S is the constraint language defined on domain S and containing the restriction of every weighted relation $\phi \in \Gamma$ onto S.

Appropriate notions of cores have played an important role in the complexity classification of CSPs [9, 10, 58] and VCSPs [31, 35]. We define a core based on the unary operations in the support of a language, as is done in [31, 51].

Definition 2.12. A general-valued constraint language Γ is *a core* if all unary operations in $\mathrm{supp}(\Gamma)$ are bijections. A general-valued constraint language Γ' is a *core of* Γ if Γ' is a core and $\Gamma' = \Gamma[f(D)]$ for some unary $f \in \mathrm{supp}(\Gamma)$.

We can now give a formal definition of the notion of *simulation* used in the statement of our main result, Theorem 3.5. Recall from Example 2.4 that c_a denotes the constant unary relation containing the label a. Let $\mathcal{C}_D = \{c_a \mid a \in D\}$ be the set of all constant unary relations on the set D.

Definition 2.13. Let Γ' be a core of a general-valued constraint language Γ on domain $D' \subseteq D$. We say that Γ *can simulate* a general-valued constraint language Δ if Δ has an interpretation in $\Gamma' \cup \mathcal{C}_{D'}$.

We note that simulation is known to preserve polynomial-time solvability [10, 17, 35, 50]. We will show later, in Theorem 3.11, that simulation additionally preserves exact solvability in the Lasserre SDP hierarchy, defined in Section 3.2, up to a constant factor in the level of the hierarchy.

3 LOWER BOUNDS ON LP AND SDP RELAXATIONS

Every VCSP instance has a natural LP relaxation known as the *basic LP relaxation* (BLP). The power of BLP for exact solvability of CSP(Γ), where Γ is a crisp constraint language, has been characterised (in terms of the polymorphisms of Γ) in [36]. The power of BLP for exact solvability of VCSP(Γ), where Γ is a general-valued constraint language, has been characterised (in terms of the fractional polymorphisms of Γ) in [32].

The Sherali-Adams LP hierarchy [49] gives a systematic way of strengthening the BLP relaxation. BLP being the first level, the kth level of the Sherali-Adams LP hierarchy adds to the BLP linear constraints satisfied by the integral solutions and involving at most k variables. One can think of the variables of the kth level as probability distributions over assignments to at most k variables of the original instance.

The Lasserre SDP hierarchy [39] is a significant strengthening of the Sherali-Adams LP hierarchy: real-valued variables are replaced by vectors from a finite-dimensional real vector space. Intuitively, the norms of these vectors again induce probability distributions over assignments to at most k variables of the original instance (for the kth level of the Lasserre SDP hierarchy). Since these distributions have to come from inner products of vectors, this is a tighter relaxation. In particular, it is known that the kth level of the Lasserre SDP hierarchy is at least as tight as the kth level of the Sherali-Adams LP hierarchy [40].

It is well known that for a problem with n variables, the nth levels of both of these two hierarchies are exact, i.e., the solutions to the nth levels are precisely the convex combinations of the integral solutions. However, it is not clear how to solve the nth levels in polynomial time. In general, taking an n-variable instance of VCSP(Γ), the kth level of both hierarchies can be solved in time $L \cdot n^{O(k)}$, where L is the length of a binary encoding of the input. In particular, this is polynomial for a fixed k.

In this section, we will define the Sherali-Adams LP and the Lasserre SDP hierarchies and state known and new results regarding their power and limitations for exact solvability of general-valued CSPs.

3.1 Sherali-Adams LP Hierarchy

Let I be an instance of the VCSP with $\phi_I(x_1, \ldots, x_n) = \sum_{i=1}^{q} \phi_i(\mathbf{x}_i)$, $X_i \subseteq V = \{x_1, \ldots, x_n\}$ and $\phi_i \colon D^{\mathrm{ar}(\phi_i)} \to \overline{\mathbb{Q}}$. We will use the notational convention to denote by X_i the *set* of variables occurring in the scope $\mathbf{x}_i$.

A *null constraint* on a set $X \subseteq V$ is a constraint with a weighted relation identical to 0. It is sometimes convenient to add null constraints to a VCSP instance as placeholders, to ensure that they have scopes where required, even if these relations may not necessarily be members of the corresponding constraint language Γ. In order to obtain an equivalent instance that is formally in VCSP(Γ), the null constraints can simply be dropped, as they are always satisfied and do not influence the value of the objective function.

Let k be an integer. The kth level of the Sherali-Adams LP hierarchy [49], henceforth called the SA(k)-relaxation of I, is given by the following linear program. Ensure that for every non-empty $X \subseteq V$ with $|X| \le k$ there is some constraint $\phi_i(\mathbf{x}_i)$ with $X_i = X$, possibly by adding null constraints. The variables of the SA(k)-relaxation, given in Figure 1, are $\lambda_i(\sigma)$ for every $i \in [q]$ and assignment $\sigma \colon X_i \to D$. We slightly abuse notation by writing $\sigma \in \mathrm{Feas}(\phi_i)$ for $\sigma \colon X_i \to D$ such that $\sigma(\mathbf{x}_i) \in \mathrm{Feas}(\phi_i)$.

We write $\mathrm{Opt}_{\mathrm{LP}}(I, k)$ for the optimal value of an LP-solution to the SA(k)-relaxation of I.

Definition 3.1. Let Γ be a general-valued constraint language. We say that VCSP(Γ) is *solved by the kth level of the Sherali-Adams LP hierarchy* if for every instance I of VCSP(Γ) we have $\mathrm{Opt}_{\mathrm{VCSP}}(I) = \mathrm{Opt}_{\mathrm{LP}}(I, k)$.

We now describe the main result from [51], which captures the power of Sherali-Adams LP relaxations for exact optimisation of VCSPs.

$$\text{minimise} \sum_{i=1}^{q} \sum_{\sigma \in \text{Feas}(\phi_i)} \lambda_i(\sigma)\phi_i(\sigma(\mathbf{x}_i))$$

subject to

$$\lambda_i(\sigma) \geq 0 \qquad \forall i \in [q], \sigma\colon X_i \to D \tag{S1}$$

$$\lambda_i(\sigma) = 0 \qquad \forall i \in [q], \sigma\colon X_i \to D, \sigma(\mathbf{x}_i) \notin \text{Feas}(\phi_i) \tag{S2}$$

$$\sum_{\sigma\colon X_i \to D} \lambda_i(\sigma) = 1 \qquad \forall i \in [q] \tag{S3}$$

$$\sum_{\substack{\sigma\colon X_i \to D \\ \sigma|_{X_j} = \tau}} \lambda_i(\sigma) = \lambda_j(\tau) \qquad \forall i, j \in [q] : X_j \subseteq X_i, |X_j| \leq k, \tau\colon X_j \to D \tag{S4}$$

Fig. 1. The kth level of the Sherali-Adams LP hierarchy, SA(k).

An m-ary idempotent operation $f\colon D^m \to D$ is called a *weak near-unanimity* (WNU) operation if, for all $x, y \in D$,

$$f(y, x, x, \dots, x) = f(x, y, x, x, \dots, x) = \cdots = f(x, x, \dots, x, y). \tag{WNU}$$

Definition 3.2. A set of operations satisfies the *bounded width condition (BWC)* if it contains a (not necessarily idempotent) m-ary operation satisfying the identities (WNU), for every $m \geq 3$.

Recall from Section 1 Theorem 1.1, which characterises the power of constant levels of the Sherali-Adams LP hierarchy for exact solvability of VCSPs in terms of the BWC.

Remark 2.

(i) While it is not clear from the definition that condition (iii) of Theorem 1.1 is decidable, it is known to be equivalent to a decidable condition. Briefly, let Γ' be a core of Γ defined on $D' \subseteq D$. By [51, Lemma 3.7], Γ satisfies the BWC if and only if $\Gamma' \cup \mathcal{C}_{D'}$ satisfies the BWC. By [34, Theorem 2.8], $\Gamma' \cup \mathcal{C}_{D'}$ satisfies the BWC if and only there are a ternary WNU f and a 4-ary WNU g in $\text{supp}(\Gamma' \cup \mathcal{C}_{D'})$ satisfying $f(y, x, x) = g(y, x, x, x)$ for all $x, y \in D'$. Finally, checking for the existence of such operations can be done using a linear program.

(ii) It is possible to obtain a solution to an instance I of VCSP(Γ) from the optimal value of the SA(3)-relaxation of I [51, Section 3.6].

(iii) Theorem 1.1 says that if $\text{supp}(\Gamma)$ violates the BWC then VCSP(Γ) requires more than a constant level of the Sherali-Adams LP hierarchy for exact solvability. The proof in [51] actually shows that in this case $\Omega(\sqrt{n})$ levels are required for exact solvability of n-variable instances of VCSP(Γ).

3.2 Lasserre SDP Hierarchy

Let I be an instance of the VCSP with $\phi_I(x_1, \dots, x_n) = \sum_{i=1}^{q} \phi_i(\mathbf{x}_i)$, $X_i \subseteq V = \{x_1, \dots, x_n\}$ and $\phi_i\colon D^{\text{ar}(\phi_i)} \to \overline{\mathbb{Q}}$. For $\sigma_i\colon X_i \to D$ and $\sigma_j\colon X_j \to D$, if $\sigma_i|_{X_i \cap X_j} = \sigma_j|_{X_i \cap X_j}$ then we write $\sigma_i \circ \sigma_j\colon (X_i \cup X_j) \to D$ for the assignment defined by $\sigma_i \circ \sigma_j(x) = \sigma_i(x)$ for $x \in X_i$ and $\sigma_i \circ \sigma_j(x) = \sigma_j(x)$ otherwise.

$$
\begin{aligned}
&\text{minimise} \sum_{i=1}^{q} \sum_{\sigma \in \mathrm{Feas}(\phi_i)} ||\boldsymbol{\lambda}_i(\sigma)||^2 \phi_i(\sigma(\mathbf{x}_i)) \\
&\text{subject to} \\
&\quad ||\boldsymbol{\lambda}_0|| = 1 && \text{(L1)} \\
&\quad \langle \boldsymbol{\lambda}_i(\sigma_i), \boldsymbol{\lambda}_j(\sigma_j) \rangle \geq 0 \quad \forall i,j \in [q], \sigma_i \colon X_i \to D, \sigma_j \colon X_j \to D && \text{(L2)} \\
&\quad ||\boldsymbol{\lambda}_i(\sigma)||^2 = 0 \quad \forall i \in [q], \sigma \colon X_i \to D, \sigma(\mathbf{x}_i) \notin \mathrm{Feas}(\phi_i) && \text{(L3)} \\
&\quad \sum_{a \in D} ||\boldsymbol{\lambda}_i(a)||^2 = 1 \quad \forall i \text{ with } |X_i| = 1 && \text{(L4)} \\
&\quad \langle \boldsymbol{\lambda}_i(\sigma_i), \boldsymbol{\lambda}_j(\sigma_j) \rangle = 0 \quad \forall i,j \in [q], \sigma_i \colon X_i \to D, \sigma_j \colon X_j \to D && \text{(L5)} \\
&\qquad\qquad \sigma_i|_{X_i \cap X_j} \neq \sigma_j|_{X_i \cap X_j} \\
&\quad \langle \boldsymbol{\lambda}_i(\sigma_i), \boldsymbol{\lambda}_j(\sigma_j) \rangle = \langle \boldsymbol{\lambda}_{i'}(\sigma_{i'}), \boldsymbol{\lambda}_{j'}(\sigma_{j'}) \rangle \quad \forall i,j,i',j' \in [q], X_i \cup X_j = X_{i'} \cup X_{j'} && \text{(L6)} \\
&\qquad\qquad \sigma_i \colon X_i \to D, \sigma_j \colon X_j \to D, \sigma_{i'} \colon X_{i'} \to D \\
&\qquad\qquad \sigma_{j'} \colon X_{j'} \to D, \sigma_i \circ \sigma_j = \sigma_{i'} \circ \sigma_{j'}
\end{aligned}
$$

Fig. 2. The kth level of the Lasserre SDP hierarchy, $\mathrm{Lasserre}(k)$.

Let k be an integer with $k \geq \max_i(\mathrm{ar}(\phi_i))$.[1] The kth level of the Lasserre SDP hierarchy [38], henceforth called the Lasserre(k)-relaxation of I, is given by the following semidefinite program (we follow the presentation from [53]). Ensure that for every subset (including the empty set) $X \subseteq V$ with $|X| \leq k$ there is some constraint $\phi_i(\mathbf{x}_i)$ with $X_i = X$, possibly by adding null constraints. The vector variables of the Lasserre(k)-relaxation, given in Figure 2, are $\boldsymbol{\lambda}_i(\sigma) \in \mathbb{R}^t$ for every $i \in [q]$ and assignment $\sigma \colon X_i \to D$. Here t is the dimension of the real vector space.[2] We write $\boldsymbol{\lambda}_0$ as a shorthand for $\boldsymbol{\lambda}_i(\emptyset)$ where i is the index for which $X_i = \emptyset$.

For any fixed k and any t polynomial in the size of I, the Lasserre(k)-relaxation of I is of polynomial size in terms of I and can be solved in polynomial time [21].[3] Note that k may not necessarily be constant but it could depend on n, the number of variables of I.

We write $\mathrm{Val}_{\mathrm{SDP}}(I, \boldsymbol{\lambda}, k)$ for the value of the SDP-solution $\boldsymbol{\lambda}$ to the Lasserre(k)-relaxation of I, and $\mathrm{Opt}_{\mathrm{SDP}}(I, k)$ for its optimal value.

[1] It also makes sense to consider relaxations with $k < \max_i(\mathrm{ar}(\phi_i))$, in particular for positive (algorithmic) results, such as the implication $(iii) \Rightarrow (ii)$ in Theorem 1.1. For our main (impossibility) result, we will be interested in k which is linear in the number of variables of I.

[2] Typically, $t = (nd)^{O(k)}$ for an instance with n variables over a domain of size d.

[3] Under technical assumptions which are satisfied by the Lasserre relaxation, SDPs can be solved approximately; for any ϵ there is an algorithm that given an SDP returns vectors for which the objective function is at most ϵ away from the optimum value and the running time is polynomial in the input size and $\log(1/\epsilon)$ [21, 54]. For any language Γ of *finite* size there is $\epsilon = \epsilon(\Gamma)$ such that solving the SDP up to an additive error of ϵ suffices for exact solvability. For instance, take ϵ such that $\epsilon < \min_{\phi \in \Gamma} \min_{\mathbf{x}, \mathbf{y} \in \mathrm{Feas}(\phi), \phi(\mathbf{x}) \neq \phi(\mathbf{y})} |\phi(\mathbf{x}) - \phi(\mathbf{y})|$. Since this paper deals with *impossibility* results these matters are not relevant but we mention it here for completeness.

Definition 3.3. Let Γ be a general-valued constraint language. We say that VCSP(Γ) is *solved by the kth level of the Lasserre SDP hierarchy* if for every instance I of VCSP(Γ) we have $\mathrm{Opt}_{\mathrm{VCSP}}(I) = \mathrm{Opt}_{\mathrm{SDP}}(I, k)$.

We say that an instance I of VCSP(Γ) is a *gap instance for* the kth level of the Lasserre SDP hierarchy if $\mathrm{Opt}_{\mathrm{SDP}}(I, k) < \mathrm{Opt}_{\mathrm{VCSP}}(I)$.

Definition 3.4. Let Γ be a general-valued constraint language. We say that VCSP(Γ) *requires linear levels* of the Lasserre SDP hierarchy if there is a constant $0 < c < 1$ such that for all sufficiently large n there is an n-variable gap instance I_n of VCSP(Γ) for Lasserre($\lfloor cn \rfloor$).

3.3 Main Results

Let $\mathcal{G}$ be an Abelian group over a finite set G and let $r \geq 1$ be an integer. Denote by $E_{\mathcal{G},r}$ the crisp constraint language over domain G with, for every $a \in G$, and $1 \leq m \leq r$, a relation $R_a^m = \{(x_1, \ldots, x_m) \in G^m \mid x_1 + \cdots + x_m = a\}$.

We are now ready to state our main results.

THEOREM 3.5. *Let Γ be a general-valued constraint language of finite size. The following are equivalent:*

(i) VCSP(Γ) *requires linear levels of the Lasserre SDP hierarchy.*
(ii) Γ *can simulate* $E_{\mathcal{G},3}$ *for some non-trivial Abelian group* $\mathcal{G}$*.*
(iii) supp(Γ) *violates the BWC.*

Theorems 1.1 and 3.5 give the following.

COROLLARY 3.6. *Let Γ be a general-valued constraint language of finite size. Then, either* VCSP(Γ) *is solved by the third level of the Sherali-Adams LP relaxation, or* VCSP(Γ) *requires linear levels of the Lasserre SDP relaxation.*

PROOF. Either supp(Γ) satisfies the BWC, in which case VCSP(Γ) is solved by the third level of the Sherali-Adams LP relaxation by Theorem 1.1, or supp(Γ) violates the BWC, in which case VCSP(Γ) requires linear levels of the Lasserre SDP hierarchy by Theorem 3.5. ☐

Recall that a constraint language Γ is called *crisp* if it contains only (unweighted) relations. Our result covers this special case, and thus we get the following corollary, which was independently obtained (using a different proof) in [3].

COROLLARY 3.7. *Let Γ be a crisp constraint language of finite size. Then, either* VCSP(Γ) *is solved by the third level of the Sherali-Adams LP relaxation, or* VCSP(Γ) *requires linear levels of the Lasserre SDP relaxation.*

A constraint language Γ is called *finite-valued* [50] if for every $\phi \in \Gamma$ it holds $\phi(\mathbf{x}) < \infty$ for every $\mathbf{x}$. In this special case, we get the following result, which was independently obtained (using a different proof) in [19].

COROLLARY 3.8. *Let Γ be a finite-valued constraint language of finite size. Then, either* VCSP(Γ) *is solved by the first level of the Sherali-Adams LP relaxation, or* VCSP(Γ) *requires linear levels of the Lasserre SDP relaxation.*

PROOF. Let D be the domain of Γ. If VCSP(Γ) is *not* solved by the first level of the Sherali-Adams LP relaxation, then [50] shows (in different terminology) that Γ can simulate ϕ_{mc} (cf. Example 2.4). Using ϕ_{mc} together with the unary constant relations c_0 and c_1, it is then not difficult to express a ternary weighted relation ϕ such that $\phi(x,y,z)$ minimises on $x+y+z=0 \pmod 2$. Now, $R_0^3 = \mathrm{Opt}(\phi)$ together with c_0 and c_1 can express all remaining relations in $E_{\mathbb{Z}_2,3}$. Overall, we conclude that Γ can simulate $E_{\mathbb{Z}_2,3}$, which proves the claim by Theorem 3.5. □

Lee et al. [41, 42] give some very strong results on approximation-preserving reductions between SDP relaxations. They give a general reduction turning lower bounds on the number of levels of the Lasserre SDP hierarchy needed for approximation to lower bounds on the size of *arbitrary* SDP relaxations. In particular, they show that if linear levels of the Lasserre SDP relaxation are required for some problems then no *polynomial-size SDP relaxation* suffices. We now briefly discuss how their result together with Theorem 3.5 can be used to derive the same consequence for VCSP(Γ) when supp(Γ) violates the BWC.

Lee et al. give in [41, Theorem 6.4] a reduction for turning lower bounds on the number of levels of the Lasserre SDP hierarchy needed for approximate maximisation of Max-CSPs to lower bounds on the size of arbitrary SDP relaxations. In order to apply their theorem in our setting, a number of differences in the setup of this paper and [41] must be addressed. First, [41, Theorem 6.4] is stated only for Boolean domains and proved using [41, Theorem 3.8]. However, a generalisation to arbitrary fixed finite domains follows from [41, Theorem 7.2] [46]. Second, the results in [41, 42] are formulated for the sum-of-squares SDP hierarchy, which is equivalent to the Lasserre SDP hierarchy: the kth level of the sums-of-squares SDP hierarchy is the same as the $(k/2)$th level of the Lasserre SDP hierarchy. Third, while the results in [41, 42] are formulated for constraint languages consisting of a single $\{0,1\}$-valued weighted relation, the proofs give the same result for constraint languages (of finite size) consisting of $[0,1]$-valued weighted relations of different arities [46]. Finally, while the work in [41, 42] deals with maximisation problems, for exact solvability we can equivalently turn to minimisation problems.

3.4 Proof of Theorem 3.5

Let Γ be a general-valued constraint language of finite size. If supp(Γ) violates the BWC then we aim to prove that VCSP(Γ) requires linear levels of the Lasserre SDP hierarchy.

We will follow the approach used in [51] to prove the implication (*i*) $\implies$ (*iii*) of Theorem 1.1. This is based on the idea that if supp(Γ) violates the BWC, then Γ can simulate linear equations in some Abelian group. In order to establish the implications (*iii*) $\implies$ (*ii*) $\implies$ (*i*) of Theorem 3.5, it suffices to show that linear equations require linear levels of the Lasserre SDP hierarchy and that the simulation preserves exact solvability by the Lasserre SDP hierarchy (up to a constant factor in the level of the hierarchy). Our contribution is proving the latter. The former is known [24, 48, 53], as we will now discuss.

Theorem 3.9 ([13]). *Let $\mathcal{G}$ be a finite non-trivial Abelian group. Then,* VCSP($E_{\mathcal{G},3}$) *requires linear levels of the Lasserre SDP hierarchy.*

For Abelian groups of prime orders, Tulsiani showed that there is a constant $0 < c < 1$ such that for every large enough n there is an instance I_n of VCSP($E_{\mathcal{G},3}$) on n variables with $\mathrm{Opt}_{\mathrm{VCSP}}(I_n) = \infty$ and $\mathrm{Opt}_{\mathrm{SDP}}(I_n, \lfloor cn \rfloor) = 0$; i.e., I_n is a gap instance for Lasserre($\lfloor cn \rfloor$) [53, Theorem 4.2].[4] This work was based on the result of Schoenebeck who showed it for Boolean domains [48], thus rediscovering the work of Grigoriev [24]. A generalisation to all Abelian groups was then established by Chan in [13, Appendix D]. Theorem 3.9 states that

[4]We note that [53] uses different terminology from ours: Max-CSP(P) for a k-ary predicate P applied to literals rather than variables.

distinguishing satisfiable instances of VCSP($E_{\mathcal{G},3}$) from instances in which not all constraints are simultaneously satisfiable requires linear levels of the Lasserre SDP hierarchy. We remark that the results in [13, 48, 53] actually prove something much stronger: even distinguishing satisfiable instances from instances in which only a small fraction of the constraints are simultaneously satisfiable requires linear levels of the Lasserre SDP hierarchy.

The following notion of reduction is key in this paper.

Definition 3.10. Let Γ and Δ be two general-valued constraint languages of finite size. We write $\Delta \leq_{\mathrm{L}} \Gamma$ if there is a polynomial-time reduction from VCSP(Δ) to VCSP(Γ) with the following property: there is a constant $c \geq 1$ depending only on Γ and Δ such that for any $k \geq 1$, if Lasserre(k) solves VCSP(Γ) then Lasserre(ck) solves VCSP(Δ).

By Definition 3.10, $\leq_{\mathrm{L}}$ reductions compose. Let $\Delta \leq_{\mathrm{L}} \Gamma$. By Definitions 3.4 and 3.10, if VCSP(Δ) requires linear levels of the Lasserre SDP hierarchy then so does VCSP(Γ). An analogous notion of reduction for the Sherali-Adams LP hierarchy, $\leq_{\mathrm{SA}}$, was used in [51].

The following theorem is the main technical contribution of the paper. It shows that a general-valued constraint language can be augmented with various additional weighted relations while preserving exact solvability in the Lasserre SDP hierarchy up to a constant factor in the level of the hierarchy. It is a strengthening of Theorem [51, Theorem 5.5], which showed that the same additional weighted relations preserve exact solvability in the Sherali-Adams LP hierarchy.

THEOREM 3.11. *Let Γ be a general-valued constraint language of finite size on domain D. The following holds:*

(1) If ϕ is expressible in Γ, then $\Gamma \cup \{\phi\} \leq_{\mathrm{L}} \Gamma$.
(2) $\Gamma \cup \{\phi_{=}^{D}\} \leq_{\mathrm{L}} \Gamma$.
(3) If Γ interprets the general-valued constraint language Δ of finite size, then $\Delta \leq_{\mathrm{L}} \Gamma$.
(4) If $\phi \in \Gamma$, then $\Gamma \cup \{\mathrm{Opt}(\phi)\} \leq_{\mathrm{L}} \Gamma$ and $\Gamma \cup \{\mathrm{Feas}(\phi)\} \leq_{\mathrm{L}} \Gamma$.
(5) If Γ' is a core of Γ on domain $D' \subseteq D$, then $\Gamma' \cup \mathcal{C}_{D'} \leq_{\mathrm{L}} \Gamma$.

PROOF. The proof is to a large extent based on a technical lemma, Lemma 4.2, which is stated and proved in Section 4. This lemma shows that, subject to some consistency conditions, a polynomial-time reduction between two constraint languages Δ and Γ that is based on locally replacing valued constraints with weighted relations in Δ by gadgets expressed in Γ can be turned into an $\leq_{\mathrm{L}}$-reduction. The same approach was used in [51, Theorem 5.5] for constructing $\leq_{\mathrm{SA}}$-reductions for (1–3), and (5). In these cases, it therefore essentially suffices to replace the applications of [51, Lemma 6.1] by applications of Lemma 4.2 in the proofs of [51, Lemmas 6.2–6.4, and 6.7].

For case (3), we remark that our definition differs slightly from that of [51] in that we incorporate applications of the operations Opt and Feas as well as scaling by nonnegative rational constants and addition of rational constants in the definition of $\langle\Gamma\rangle$. To accommodate for the operations Opt and Feas in the proof, it suffices to add an application of (4). Furthermore, scaling can be implemented by repeated constraints and the addition of a constant changes the value of the objective function of the VCSP instance by the same constant as the objective function of the SDP relaxation, for all feasible solutions to the corresponding problems.

For case (5), the proof in [51, Lemmas 6.7] also refers to [51, Lemma 5.6] which also hold for $\leq_{\mathrm{L}}$-reductions by Lemma 3.13 below, and cases (1) and (4).

The remaining two reductions in (4) are shown in a more straightforward way for $\leq_{\mathrm{SA}}$-reductions in [51, Lemmas 6.5 and 6.6]. Here, we argue that the proof of [51, Lemmas 6.5] goes through for $\leq_{\mathrm{L}}$-reductions as well, which shows that $\Gamma \cup \{\mathrm{Opt}(\phi)\} \leq_{\mathrm{L}} \Gamma$. We omit the analogous argument for the reduction $\Gamma \cup \{\mathrm{Feas}(\phi)\} \leq_{\mathrm{L}} \Gamma$. In the proof of [51, Lemmas 6.5], an instance I of VCSP($\Gamma \cup \{\mathrm{Opt}(\phi)\}$) is transformed into an instance J of VCSP(Γ) by replacing all occurrences of $\mathrm{Opt}(\phi)$ by multiple copies of ϕ. It is then shown that if I is a gap instance for the SA(k)-relaxation, and λ is an optimal solution to this relaxation, then λ is also a solution to the SA(k)-relaxation of J. Moreover, λ attains a better value than $\mathrm{Opt}_{\mathrm{VCSP}}(J)$, hence J is also a gap instance. This argument goes through also if we take I to be a gap instance for the Lasserre(k)-relaxation, and $\boldsymbol{\lambda}$ an optimal solution to this relaxation. The exact same solution $\boldsymbol{\lambda}$ then also shows that J is a gap instance for the Lasserre(k)-relaxation. □

In order to finish the proof of Theorem 3.5, we need a few additional results. The following result follows, as described in the proof of [51, Theorem 5.4], from [2, 34].

THEOREM 3.12 ([51, THEOREM 5.4]). *Let Δ be a crisp constraint language of finite size that contains all constant unary relations. If* $\mathrm{Pol}(\Delta)$ *violates the BWC, then there exists a finite non-trivial Abelian group $\mathcal{G}$ such that Δ interprets $E_{\mathcal{G},r}$, for every $r \geq 1$.*

The following two lemmas, together with cases (1) and (4) of Theorem 3.11, extend [51, Lemma 5.6 and Lemma 5.7] from $\leq_{\mathrm{SA}}$-reductions to $\leq_{\mathrm{L}}$-reductions.

LEMMA 3.13. *Let Γ be a general-valued constraint language over domain D and let F be a set of operations over D. If* $\mathrm{supp}(\Gamma) \cap F = \emptyset$*, then there exists a crisp constraint language* $\Delta \subseteq \langle\Gamma\rangle$ *such that* $\mathrm{Pol}(\Delta) \cap F = \emptyset$*. Moreover, if Γ and F are finite then so is Δ.*

PROOF. By [51, Lemma 2.9], for each $f \in F \cap \mathrm{Pol}(\Gamma)$, there is an instance I_f of VCSP(Γ) such that $f \notin \mathrm{Pol}(\mathrm{Opt}(\phi_{I_f}))$. Let $\Delta = \{\mathrm{Opt}(\phi_{I_f}) \mid f \in F\} \cup \{\mathrm{Feas}(\phi) \mid \phi \in \Gamma\} \subseteq \langle\Gamma\rangle$. For $f \in F \cap \mathrm{Pol}(\Gamma)$, we have $f \notin \mathrm{Pol}(\mathrm{Opt}(\phi_{I_f})) \supseteq \mathrm{Pol}(\Delta)$. For $f \in F \setminus \mathrm{Pol}(\Gamma)$, we have $f \notin \mathrm{Pol}(\phi)$, for some $\phi \in \Gamma$, so $f \notin \mathrm{Pol}(\Delta)$. It follows that $\mathrm{Pol}(\Delta) \cap F = \emptyset$. □

LEMMA 3.14. *Let Γ be a general-valued constraint language of finite size. If* $\mathrm{supp}(\Gamma)$ *violates the BWC, then there is a crisp constraint language* $\Delta \subseteq \langle\Gamma\rangle$ *of finite size such that* $\mathrm{Pol}(\Delta)$ *violates the BWC.*

PROOF. Since $\mathrm{supp}(\Gamma)$ violates the BWC, there exists an $m \geq 3$ such that $\mathrm{supp}(\Gamma)$ does not contain any m-ary WNU. Let F be the (finite) set of all m-ary WNUs. The result follows by applying Lemma 3.13 to Γ and F. □

We are now ready to prove Theorem 3.5.

PROOF OF THEOREM 3.5. Theorem 1.1 gives the implication (*i*) $\Longrightarrow$ (*iii*) by contraposition: if $\mathrm{supp}(\Gamma)$ satisfies the BWC then, by Theorem 1.1, VCSP(Γ) is solved by any constant level k of the Sherali-Adams LP hierarchy with $k \geq 3$, and thus also by the kth level of the Lasserre SDP hierarchy for $k \geq \mathrm{ar}(\Gamma)$.

Now, suppose that $\mathrm{supp}(\Gamma)$ violates the BWC. Let Γ' be a core of Γ on a domain $D' \subseteq D$ and let $\Gamma_c = \Gamma' \cup \mathcal{C}_{D'}$. By [51, Lemma 3.7], $\mathrm{supp}(\Gamma_c)$ also violates the BWC. By Lemma 3.14, there exists a finite crisp constraint language Δ such that Δ has an interpretation in Γ_c and $\mathrm{Pol}(\Delta)$ violates the BWC. Since $\mathcal{C}_D \subseteq \Gamma_c$, we may assume, without loss of generality, that $\mathcal{C}_D \subseteq \Delta$. By Theorem 3.12, there exists a finite non-trivial Abelian group $\mathcal{G}$ and an interpretation of $E_{\mathcal{G},3}$ in Δ. Since interpretations compose, $E_{\mathcal{G},3}$ has an interpretation in Γ_c. Therefore, Γ can simulate $E_{\mathcal{G},3}$ which gives the implication (*iii*) $\Longrightarrow$ (*ii*).

Finally, by Theorem 3.9, VCSP($E_{\mathcal{G},3}$) requires linear levels of the Lasserre SDP hierarchy. By Theorem 3.11(3) and (5), we have $E_{\mathcal{G},3} \leq_{\mathrm{L}} \Gamma_c \leq_{\mathrm{L}} \Gamma$. Consequently, VCSP($\Gamma$) requires linear levels of the Lasserre SDP hierarchy as well. This gives the implication (ii) $\Longrightarrow$ (i). □

4 AN $\leq_{\mathrm{L}}$-REDUCTION SCHEME

In this section, we will prove Lemma 4.2, which is the key technique used to establish cases (1)–(3) and (5) of Theorem 3.11. It is an analogue of [51, Lemma 6.1], which does the same for the $\leq_{\mathrm{SA}}$-reductions, and the proof is closely modelled on that of [51, Lemma 6.1].

The following observation will be used throughout this section: since the set of vectors $\{\boldsymbol{\lambda}_i(\tau) \mid \tau \in D^{X_i}\}$ for a feasible solution $\boldsymbol{\lambda}$ is orthogonal by (L5), it follows that we have $\| \sum_{\tau \in T} \boldsymbol{\lambda}_i(\tau)\|^2 = \sum_{\tau \in T} \langle \boldsymbol{\lambda}_i(\tau), \boldsymbol{\lambda}_i(\tau)\rangle$ for any subset $T \subseteq D^{X_i}$.

We will also use the following lemma which can be seen as an additional set of constraints on the Lasserre(k)-relaxation but which follows directly from the others.

Lemma 4.1. *Every feasible solution $\boldsymbol{\lambda}$ to the* Lasserre(k)*-relaxation satisfies, in addition to (L1)–(L6), the following:*

$$\sum_{\tau:\ \tau|_{X_j}=\sigma} \boldsymbol{\lambda}_i(\tau) = \boldsymbol{\lambda}_j(\sigma) \qquad \forall i,j \in [q], X_j \subseteq X_i, |X_i| \leq k, \sigma\colon X_j \to D. \quad \text{(L7)}$$

Proof. Consider the norm of the vector $\sum_{\tau:\ \tau|_{X_j}=\sigma} \boldsymbol{\lambda}_i(\tau) - \boldsymbol{\lambda}_j(\sigma)$.

$$\begin{aligned}
&\| \sum_{\tau:\ \tau|_{X_j}=\sigma} \boldsymbol{\lambda}_i(\tau) - \boldsymbol{\lambda}_j(\sigma)\|^2 \\
&= \| \sum_{\tau:\ \tau|_{X_j}=\sigma} \boldsymbol{\lambda}_i(\tau)\|^2 - 2\langle \sum_{\tau:\ X_i \to D} \boldsymbol{\lambda}_i(\tau), \boldsymbol{\lambda}_j(\sigma)\rangle + \|\boldsymbol{\lambda}_j(\sigma)\|^2 \\
&= \| \sum_{\tau:\ \tau|_{X_j}=\sigma} \boldsymbol{\lambda}_i(\tau)\|^2 - 2 \sum_{\tau:\ X_i \to D} \langle \boldsymbol{\lambda}_i(\tau), \boldsymbol{\lambda}_j(\sigma)\rangle + \|\boldsymbol{\lambda}_j(\sigma)\|^2 \\
&= \| \sum_{\tau:\ \tau|_{X_j}=\sigma} \boldsymbol{\lambda}_i(\tau)\|^2 - 2 \sum_{\tau:\ X_i \to D} \langle \boldsymbol{\lambda}_i(\tau), \boldsymbol{\lambda}_i(\tau)\rangle + \|\boldsymbol{\lambda}_j(\sigma)\|^2 \\
&= -\| \sum_{\tau:\ \tau|_{X_j}=\sigma} \boldsymbol{\lambda}_i(\tau)\|^2 + \|\boldsymbol{\lambda}_j(\sigma)\|^2,
\end{aligned}$$

where the next to last equality follows from (L6) since $X_j \subseteq X_i$ and $\sigma = \tau|_{X_j}$. We see that the equality in the lemma is equivalent to:

$$\| \sum_{\tau:\ \tau|_{X_j}=\sigma} \boldsymbol{\lambda}_i(\tau)\|^2 = \|\boldsymbol{\lambda}_j(\sigma)\|^2. \tag{1}$$

We finish the proof by induction on $|X_i \setminus X_j| \geq 1$. There are two base cases:

(i) If $|X_i \setminus X_j| = 1$ and $X_j = \emptyset$, then (1) follows immediately from (L1) and (L4).

(ii) If $|X_i \setminus X_j| = 1$ and $X_j \neq \emptyset$, then let $X_r = \{x\} = X_i \setminus X_j$ be a scope on the single variable x, and, for $a \in D$, let σ_a be the assignment $\sigma_a(x) = a$. Now, (1) follows from:

$$\begin{aligned}\|\sum_{\tau\colon \tau|_{X_j}=\sigma} \boldsymbol{\lambda}_i(\tau)\|^2 &= \sum_{a\in D}\langle \boldsymbol{\lambda}_i(\sigma_a\circ\sigma), \boldsymbol{\lambda}_i(\sigma_a\circ\sigma)\rangle \\ &\overset{(L6)}{=} \sum_{a\in D}\langle \boldsymbol{\lambda}_r(\sigma_a), \boldsymbol{\lambda}_j(\sigma)\rangle \\ &= \langle \sum_{a\in D} \boldsymbol{\lambda}_r(\sigma_a), \boldsymbol{\lambda}_j(\sigma)\rangle \\ &\overset{(i)}{=} \langle \boldsymbol{\lambda}_0, \boldsymbol{\lambda}_j(\sigma)\rangle, \\ &\overset{(L6)}{=} \langle \boldsymbol{\lambda}_j(\sigma), \boldsymbol{\lambda}_j(\sigma)\rangle.\end{aligned}$$

Finally, assume that $|X_i \setminus X_j| > 1$ and that $x \in X_i \setminus X_j$. Let r be an index such that $X_r = X_j \cup \{x\}$, and, for $a \in D$, let σ_a be the assignment $\sigma_a(x) = a$. Then,

$$\begin{aligned}\sum_{\tau\colon \tau|_{X_j}=\sigma} \boldsymbol{\lambda}_i(\tau) &= \sum_{a\in D}\sum_{\tau\colon \tau|_{X_r}=\sigma\circ\sigma_a} \boldsymbol{\lambda}_i(\tau) \\ &= \sum_{a\in D} \boldsymbol{\lambda}_r(\sigma\circ\sigma_a) \\ &= \boldsymbol{\lambda}_j(\sigma),\end{aligned}$$

where the last two equalities follow by induction. □

For a solution $\boldsymbol{\lambda}$ to the Lasserre(k)-relaxation of I with the objective function $\sum_{i=1}^{q}\phi(\mathbf{x}_i)$, we denote by $\mathbf{supp}(\boldsymbol{\lambda}_i)$ the positive support of $\boldsymbol{\lambda}_i$, i.e., $\mathbf{supp}(\boldsymbol{\lambda}_i) = \{\sigma\colon X_i \to D \mid ||\boldsymbol{\lambda}_i(\sigma)||^2 > 0\}$.

The following technical lemma is the basis for the reductions in Theorem 3.11.

LEMMA 4.2. *Let Δ and Δ' be general-valued constraint languages of finite size over domains D and D', respectively.*

Let $(I, i) \mapsto J_i$ be a map that to each instance I of VCSP(Δ) *with variables V and objective function $\sum_{i=1}^{q}\phi_i(\mathbf{x}_i)$, and index $i \in [q]$, associates an instance J_i of* VCSP(Δ') *with variables Y_i and objective function ϕ_{J_i}. Let J be the* VCSP(Δ') *instance with variables $V' = \bigcup_{i=1}^{q} Y_i$ and objective function $\sum_{i=1}^{q}\phi_{J_i}$.*

Suppose that the following holds:

(a) For every satisfying and optimal assignment α of J, there exists a satisfying assignment σ^α of I such that

$$\mathrm{Val}_{\mathrm{VCSP}}(I, \sigma^\alpha) \leq \mathrm{Val}_{\mathrm{VCSP}}(J, \alpha).$$

Furthermore, suppose that for any $k \geq \mathrm{ar}(\Delta)$, and any feasible solution $\boldsymbol{\lambda}$ of the Lasserre(k)-*relaxation of I, the following properties hold:*

(b) For $i \in [q]$, and $\sigma\colon X_i \to D$ with positive support in $\boldsymbol{\lambda}$, there exists a satisfying assignment α_i^σ of J_i such that

$$\phi_i(\sigma(\mathbf{x}_i)) \geq \mathrm{Val}_{\mathrm{VCSP}}(J_i, \alpha_i^\sigma);$$

(c) for $i, r \in [q]$, any $X \subseteq V$ with $X_i \cup X_r \subseteq X$, and $\sigma\colon X \to D$ with positive support in $\boldsymbol{\lambda}$,

$$\alpha_i^{\sigma_i}|_{Y_i\cap Y_r} = \alpha_r^{\sigma_r}|_{Y_i\cap Y_r},$$

where $\sigma_i = \sigma|_{X_i}$ and $\sigma_r = \sigma|_{X_r}$.

Then, $I \mapsto J$ is a many-one reduction from $\mathrm{VCSP}(\Delta)$ *to* $\mathrm{VCSP}(\Delta')$ *that certifies* $\Delta \leq_{\mathrm{L}} \Delta'$.

PROOF. First, we show that $\mathrm{Opt}_{\mathrm{VCSP}}(I) = \mathrm{Opt}_{\mathrm{VCSP}}(J)$. From condition (a), if J is satisfiable, then so is I and $\mathrm{Opt}_{\mathrm{VCSP}}(I) \leq \mathrm{Opt}_{\mathrm{VCSP}}(J)$. Conversely, if I is satisfiable, and σ is an optimal assignment to I, then the Lasserre($2k$) solution $\boldsymbol{\lambda}$, where $k \geq \mathrm{ar}(\Delta)$, that assigns a fixed unit vector to $\sigma|_X$ for every $X \subseteq V$ with $|X| \leq 2k$ is feasible. Let $\sigma_i = \sigma|_{X_i}$. By (b), there exist satisfying assignments $\alpha_i^{\sigma_i}$ of J_i, for all $i \in [q]$, such that $\mathrm{Opt}_{\mathrm{VCSP}}(I) \geq \mathrm{Opt}_{\mathrm{SDP}}(I, 2k) \geq \sum_{i \in [q]} \mathrm{Val}_{\mathrm{VCSP}}(J_i, \alpha_i^{\sigma_i})$. Define an assignment $\alpha\colon V' \to D'$ by letting $\alpha(y) = \alpha_i^{\sigma_i}(y)$ for an arbitrary i such that $y \in Y_i$. We claim that $\alpha|_{Y_i} = \alpha_i^{\sigma_i}$, for all $i \in [q]$. From this it follows that α is a satisfying assignment to J such that $\sum_{i \in [q]} \mathrm{Val}_{\mathrm{VCSP}}(J_i, \alpha_i^{\sigma_i}) = \mathrm{Val}_{\mathrm{VCSP}}(J, \alpha) \geq \mathrm{Opt}_{\mathrm{VCSP}}(J)$, and hence that $\mathrm{Opt}_{\mathrm{VCSP}}(I) \geq \mathrm{Opt}_{\mathrm{VCSP}}(J)$. Indeed, let $y \in V'$ and assume that $y \in Y_i$ and $y \in Y_r$. Let $X = X_i \cup X_r$. Then, since $k \geq \mathrm{ar}(\Delta)$ and $||\boldsymbol{\lambda}(\sigma|_X)||^2 > 0$, it follows from (c) that $\alpha_i^{\sigma_i}(y) = \alpha_r^{\sigma_r}(y)$.

Let k' be arbitrary and let $k = \max\{k', \mathrm{ar}(\Delta')\} \cdot \mathrm{ar}(\Delta)$. Assume that I is a gap instance for the Lasserre($2k$)-relaxation of $\mathrm{VCSP}(\Delta)$, and let $\boldsymbol{\lambda}$ be a feasible solution such that $\mathrm{Val}_{\mathrm{SDP}}(I, \boldsymbol{\lambda}, 2k) < \mathrm{Opt}_{\mathrm{VCSP}}(I)$ (where $\mathrm{Opt}_{\mathrm{VCSP}}(I)$ may be ∞, i.e. I may be unsatisfiable). We show that there is a feasible solution $\boldsymbol{\kappa}$ to the Lasserre(k')-relaxation of J such that $\mathrm{Val}_{\mathrm{SDP}}(J, \boldsymbol{\kappa}, k') \leq \mathrm{Val}_{\mathrm{SDP}}(I, \boldsymbol{\lambda}, 2k)$.[5] Then, by condition (a), we have $\mathrm{Opt}_{\mathrm{VCSP}}(I) \leq \mathrm{Opt}_{\mathrm{VCSP}}(J)$. Hence, $\mathrm{Val}_{\mathrm{SDP}}(J, \boldsymbol{\kappa}, k') \leq \mathrm{Val}_{\mathrm{SDP}}(I, \boldsymbol{\lambda}, 2k) < \mathrm{Opt}_{\mathrm{VCSP}}(I) \leq \mathrm{Opt}_{\mathrm{VCSP}}(J)$, so J is a gap instance for the Lasserre(k')-relaxation of $\mathrm{VCSP}(\Delta')$. Since k' was chosen arbitrarily, we have $\Delta \leq_{\mathrm{L}} \Delta'$.

To this end, augment I with null constraints on $X_{q+1}, \ldots, X_{q'}$ so that for every at most $2k$-subset $X \subseteq V$, there exists an $i \in [q']$ such that $X_i = X$. Rewrite the objective function of J as $\sum_{j=1}^{p} \phi'_j(\mathbf{y}'_j)$, $\phi' \in \Delta'$, where, by possibly first adding extra null constraints to J, we will assume that for every at most k'-subset $Y \subseteq V'$, there exists a $j \in [p]$ such that $Y'_j = Y$. Here, Y'_j denotes the set of variables occurring in the tuple $\mathbf{y}'_j$. For each $i \in [q]$, let C_i be the set of indices $j \in [p]$ corresponding to the valued constraints in the instance J_i.

For $m \geq 1$, define $X_{(\leq m)} = \{X \subseteq V \mid X = \bigcup_{i \in S} X_i, S \subseteq [q], |X| \leq m\}$. This is the set of all scopes $X \subseteq V$ of size at most m that can be written as a union of scopes X_j with $j \in [q]$. Note that this set includes some, but not necessarily all, of the scopes X_i, $i \in [q'] \setminus [q]$.

We now extend α_i^σ to all indices $i \in [q'] \setminus [q]$ for which $X_i \in X_{(\leq 2k)}$. For a scope $X \in X_{(\leq 2k)}$, define $Y_X = \bigcup_{j \in [q] : X_j \subseteq X} Y_j$. The idea is that an assignment $\sigma_i\colon X_i \to D$ with $X_i \in X_{(\leq 2k)}$ will be mapped to an assignment $\alpha_i^\sigma\colon Y_{X_i} \to D'$. The assignment α_i^σ will be the union of the assignments α_j^σ over all $j \in [q]$ that satisfy $X_j \subseteq X_i$. For this to be well defined, we need to verify that the assignments α_j^σ are pairwise consistent: Let $\sigma \in \mathbf{supp}(\boldsymbol{\lambda}_i)$, and $r, s \in [q]$ be such that $X_r \cup X_s \subseteq X_i$ and $y \in Y_r \cap Y_s$. Then, by (c), it holds that $\alpha_r^{\sigma_r}(y) = \alpha_s^{\sigma_s}(y)$. Therefore, we can uniquely define $\alpha_i^\sigma\colon Y_{X_i} \to D'$ by letting $\alpha_i^\sigma(y) = \alpha_r^{\sigma_r}(y)$ for any choice of $r \in [q]$ with $X_r \subseteq X_i$ and $y \in Y_r$. This definition is consistent with α_i^σ for $i \in [q]$ in the sense that (c) now holds for all $i, r \in [q']$ such that $X_i, X_r \in X_{(\leq 2k)}$.

Let $j \in [p]$ and define $X_{(\leq m)}(Y'_j) = \{X \in X_{(\leq m)} \mid Y'_j \subseteq Y_X\}$. In particular, if $X_i \in X_{(\leq 2k)}(Y'_j)$, then α_i^σ as defined above can be restricted to an assignment on Y'_j. Next, we show that $X_{(\leq 2k)}(Y'_j)$ is in fact non-empty so that such a scope X_i always exists. Let $n = |V|$. The set $X_{(\leq n)}(Y'_j)$ is non-empty since $\bigcup_{i \in [q]} X_i \in X_{(\leq n)}(Y'_j)$. Arbitrarily pick $X \in X_{(\leq n)}(Y'_j)$. Then, $X = \bigcup_{i \in S} X_i$ for some $S \subseteq [q]$. For each $y \in Y'_j$, let $i(y) \in S$ be an index such that $y \in Y_{i(y)}$ and let $X' = \bigcup_{y \in Y'_j} X_{i(y)}$. Then, $Y'_j \subseteq Y_{X'}$, $X' \subseteq X$, and

[5] We remark here that the vectors in the feasible solution $\boldsymbol{\kappa}$ will live in the same space $\mathbb{R}^t$ as those of $\boldsymbol{\lambda}$. This is not a problem as long as t is chosen sufficiently large enough for both of the relaxations.

$|X'| \leq \max\{k', \mathrm{ar}(\Delta')\} \cdot \mathrm{ar}(\Delta) = k$, so $X' \in X_{(\leq k)}(Y'_j)$. In other words,

$$\text{for every } X \in X_{(\leq n)}(Y'_j), \text{there exists } i \in [q'] \text{ such that } X_i \subseteq X \text{ and } X_i \in X_{(\leq k)}(Y'_j). \quad (2)$$

In particular (2) implies that $X_{(\leq 2k)}(Y'_j) \supseteq X_{(\leq k)}(Y'_j)$ is non-empty for every $j \in [p]$.

For $j \in [p]$, $\alpha\colon Y'_j \to D'$, and $i \in [q']$ such that $X_i \in X_{(\leq 2k)}(Y'_j)$, define

$$\boldsymbol{\mu}^i_j(\alpha) = \sum_{\sigma\colon \alpha^\sigma_i|_{Y'_j} = \alpha} \boldsymbol{\lambda}_i(\sigma). \quad (3)$$

Claim: Definition (3) is independent of the choice of $X_i \in X_{(\leq 2k)}(Y'_j)$. That is,

$$\boldsymbol{\mu}^r_j = \boldsymbol{\mu}^i_j \qquad \forall r, i \in [q'] \text{ such that } X_r, X_i \in X_{(\leq 2k)}(Y'_j). \quad (4)$$

Proof of Claim. First, we prove (4) for $X_r \subseteq X_i$ with $X_r \in X_{(\leq k)}(Y'_j)$ and $X_i \in X_{(\leq 2k)}(Y'_j)$. We have

$$\begin{aligned}
\boldsymbol{\mu}^r_j(\alpha) &\overset{(3)}{=} \sum_{\tau\colon \alpha^\tau_r|_{Y'_j} = \alpha} \boldsymbol{\lambda}_r(\tau) \\
&\overset{(L7)}{=} \sum_{\tau\colon \alpha^\tau_r|_{Y'_j} = \alpha} \; \sum_{\sigma\colon \sigma|_{X_r} = \tau} \boldsymbol{\lambda}_i(\sigma) \\
&= \sum_{\sigma\colon \alpha^{\sigma r}_r|_{Y'_j} = \alpha} \boldsymbol{\lambda}_i(\sigma) \\
&\overset{(c)}{=} \sum_{\sigma\colon \alpha^\sigma_i|_{Y'_j} = \alpha} \boldsymbol{\lambda}_i(\sigma) \\
&\overset{(3)}{=} \boldsymbol{\mu}^i_j(\alpha),
\end{aligned}$$

Next, let $X_r \in X_{(\leq 2k)}(Y'_j)$ and $X_i \in X_{(\leq 2k)}(Y'_j)$ be arbitrary. From (2), it follows that X_r contains a subset $X_s \in X_{(\leq k)}(Y'_j)$ and that X_i contains a subset $X_t \in X_{(\leq k)}(Y'_j)$. Since $|X_s \cup X_t| \leq 2k$, there exists an index $u \in [q']$ such that $X_u = X_s \cup X_t$. The claim (4) now follows by a repeated application of the first case: $\boldsymbol{\mu}^r_j = \boldsymbol{\mu}^s_j = \boldsymbol{\mu}^u_j = \boldsymbol{\mu}^t_j = \boldsymbol{\mu}^i_j$. ∎

By (4), we can pick an arbitrary $X_i \in X_{(\leq 2k)}(Y'_j)$ and uniquely define $\boldsymbol{\kappa}_j = \boldsymbol{\mu}^i_j$.

We now show that this definition of $\boldsymbol{\kappa}$ satisfies the equations (L1)–(L6). Similarly to the definition of $\boldsymbol{\lambda}_0$, we let $\boldsymbol{\kappa}_0$ be a shorthand for $\boldsymbol{\kappa}_j(\emptyset)$, where j is the index for which $Y'_j = \emptyset$.

- The equation (L1) holds as $\boldsymbol{\kappa}_0 = \sum_\sigma \boldsymbol{\lambda}_i(\sigma) = 1$ for an arbitrary i by (L7).
- The equations (L2) holds by the linearity of the inner product.
- The equations (L3) hold trivially if ϕ'_j is a null constraint. Otherwise, $j \in C_i$ for some $i \in [q]$. This implies that $X_i \in X_{(\leq k)}(Y'_j)$, and by (4) we have $\boldsymbol{\kappa}_j = \boldsymbol{\mu}^i_j$. Then, $\alpha \in \mathbf{supp}(\boldsymbol{\kappa}_j)$ implies that there is a $\sigma \in \mathbf{supp}(\boldsymbol{\lambda}_i)$ such that $\alpha^\sigma_i|_{Y'_j} = \alpha$. By condition (b) and equation (L3) for $\boldsymbol{\lambda}_i$, the tuple $\alpha^\sigma_i(\mathbf{y}'_j) \in \mathrm{Feas}(\phi'_j)$, so $\boldsymbol{\kappa}_j$ satisfies (L3).

- We show that the equations (L4) hold for $\boldsymbol{\kappa}$. Let $Y'_j = \{y\}$ be a singleton and let $X_i \in X_{(\leq k)}(Y'_j)$. We have

$$\begin{aligned}
&\sum_{a' \in D'} ||\boldsymbol{\kappa}_j(a')||^2 \\
&\overset{(3)}{=} \sum_{a' \in D'} \langle \sum_{\sigma :\, \alpha_i^\sigma(y)=a'} \boldsymbol{\lambda}_i(\sigma), \sum_{\sigma :\, \alpha_i^\sigma(y)=a'} \boldsymbol{\lambda}_i(\sigma) \rangle \\
&\overset{(L5)}{=} \sum_{a' \in D'} \sum_{\sigma :\, \alpha_i^\sigma(y)=a'} \langle \boldsymbol{\lambda}_i(\sigma), \boldsymbol{\lambda}_i(\sigma) \rangle \\
&= \sum_{\sigma} \langle \boldsymbol{\lambda}_i(\sigma), \boldsymbol{\lambda}_i(\sigma) \rangle \\
&= ||\sum_{\sigma} \boldsymbol{\lambda}_i(\sigma)||^2 \\
&\overset{(L7)}{=} ||\boldsymbol{\lambda}_0||^2 \\
&\overset{(L1)}{=} 1.
\end{aligned}$$

- The equations (L5) hold by linearity of the inner product and by the equations (L5) for $\boldsymbol{\lambda}$.
- Finally, we show that the equations (L6) hold for $\boldsymbol{\kappa}$. Let $r, s \in [p]$, and pick assignments $\alpha_r \colon Y'_r \to D'$, $\alpha_s \colon Y'_s \to D'$. From (2) it follows that there are $X_u \in X_{(\leq k)}(Y'_r)$ and $X_t \in X_{(\leq k)}(Y'_s)$. Then, there is an index $i \in [q']$ such that $X_i = X_u \cup X_t$. It follows that $X_i \in X_{(\leq 2k)}(Y'_r)$ and $X_i \in X_{(\leq 2k)}(Y'_s)$. Therefore,

$$\begin{aligned}
&\langle \boldsymbol{\kappa}_r(\alpha_r), \boldsymbol{\kappa}_s(\alpha_s) \rangle \\
&\overset{(3)}{=} \langle \sum_{\sigma :\, \alpha_i^\sigma|_{Y'_r}=\alpha_r} \boldsymbol{\lambda}(\sigma), \sum_{\sigma' :\, \alpha_i^{\sigma'}|_{Y'_s}=\alpha_s} \boldsymbol{\lambda}(\sigma') \rangle \\
&= \sum_{\sigma :\, \alpha_i^\sigma|_{Y'_r}=\alpha_r} \sum_{\sigma' :\, \alpha_i^{\sigma'}|_{Y'_s}=\alpha_s} \langle \boldsymbol{\lambda}(\sigma), \boldsymbol{\lambda}(\sigma') \rangle \qquad (5) \\
&\overset{(L5)}{=} \sum_{\sigma :\, \alpha_i^\sigma|_{Y'_r \cup Y'_s}=\alpha_r \circ \alpha_s} \langle \boldsymbol{\lambda}(\sigma), \boldsymbol{\lambda}(\sigma) \rangle
\end{aligned}$$

  Now, let $r', s' \in [p]$ be such that $Y'_r \cup Y'_s = Y'_{r'} \cup Y'_{s'}$ and $\alpha_{r'} \colon Y'_{r'} \to D', \alpha_{s'} \colon Y'_{s'} \to D'$ be such that $\alpha_r \circ \alpha_s = \alpha_{r'} \circ \alpha_{s'}$. Then, the right-hand side of (5) is identical for $\langle \boldsymbol{\kappa}_r(\alpha_r), \boldsymbol{\kappa}_s(\alpha_s) \rangle$ and $\langle \boldsymbol{\kappa}_{r'}(\sigma_{r'}), \boldsymbol{\kappa}_{s'}(\sigma_{s'}) \rangle$.

We conclude that $\boldsymbol{\kappa}$ is a feasible solution to the Lasserre(k')-relaxation of J.

Let $i \in [q]$ and note that by (4), for every $j \in C_i$, we have $\boldsymbol{\kappa}_j = \boldsymbol{\mu}^i_j$. Therefore,

$$\begin{aligned}
&\sum_{j\in C_i}\sum_{\alpha\in\mathrm{Feas}(\phi'_j)} ||\boldsymbol{\kappa}_j(\alpha)||^2\phi'_j(\alpha(\mathbf{y}'_j)) \\
&= \sum_{j\in C_i}\sum_{\alpha\in\mathrm{Feas}(\phi'_j)}\sum_{\sigma:\,\alpha_i^\sigma|_{Y'_j}=\alpha} ||\boldsymbol{\lambda}_i(\sigma)||^2\phi'_j(\alpha(\mathbf{y}'_j)) \\
&= \sum_{\sigma:\,\alpha_i^\sigma|_{Y'_j}\in\mathrm{Feas}(\phi'_j)} ||\boldsymbol{\lambda}_i(\sigma)||^2 \sum_{j\in C_i}\phi'_j(\alpha_i^\sigma(\mathbf{y}'_j)) \\
&\le \sum_{\sigma\in\mathbf{supp}(\boldsymbol{\lambda}_i)} ||\boldsymbol{\lambda}_i(\sigma)||^2\phi_i(\sigma),
\end{aligned} \tag{6}$$

where the inequality follows from assumption (b). Summing inequality (6) over $i \in [q]$ shows that $\mathrm{Val}_{\mathrm{SDP}}(J, \boldsymbol{\kappa}, k') \le \mathrm{Val}_{\mathrm{SDP}}(I, \boldsymbol{\lambda}, 2k)$ and the lemma follows. □

ACKNOWLEDGMENTS

We thank the anonymous reviewers of both the conference version [52] and (in particular) the journal version of this paper for their comments.

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 714532). The paper reflects only the authors' views and not the views of the ERC or the European Commission. The European Union is not liable for any use that may be made of the information contained therein. Stanislav Živný was supported by a Royal Society University Research Fellowship.

REFERENCES

[1] Sanjeev Arora, Satish Rao, and Umesh V. Vazirani. 2009. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM* 56, 2 (2009). https://doi.org/10.1145/1502793.1502794

[2] Albert Atserias, Andrei Bulatov, and Anuj Dawar. 2009. Affine Systems of Equations and Counting Infinitary Logic. *Theoretical Computer Science* 410 (2009), 1666–1683. Issue 18.

[3] Albert Atserias and Joanna Ochremiak. 2017. Proof complexity meets algebra. In *Proceedings of the 44th International Colloquium on Automata, Languages and Programming (ICALP'17) (LIPIcs)*, Vol. 80. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 110:1–110:14. https://doi.org/10.4230/LIPIcs.ICALP.2017.110

[4] Libor Barto and Marcin Kozik. 2014. Constraint Satisfaction Problems Solvable by Local Consistency Methods. *J. ACM* 61, 1 (2014). https://doi.org/10.1145/2556646 Article No. 3.

[5] Libor Barto and Marcin Kozik. 2016. Robustly Solvable Constraint Satisfaction Problems. *SIAM J. Comput.* 45, 4 (2016), 1646–1669. https://doi.org/10.1137/130915479

[6] Libor Barto, Marcin Kozik, and Todd Niven. 2009. The CSP dichotomy holds for digraphs with no sources and no sinks (a positive answer to a conjecture of Bang-Jensen and Hell). *SIAM J. Comput.* 38, 5 (2009), 1782–1802. https://doi.org/10.1137/070708093

[7] Manuel Bodirsky. 2008. Constraint Satisfaction Problems with Infinite Templates. In *Complexity of Constraints (Lecture Notes in Computer Science)*, Vol. 5250. Springer, 196–228. https://doi.org/10.1007/978-3-540-92800-3_8

[8] Andrei Bulatov. 2006. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *J. ACM* 53, 1 (2006), 66–120. https://doi.org/10.1145/1120582.1120584

[9] Andrei Bulatov. 2017. A dichotomy theorem for nonuniform CSP. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS'17)*. IEEE, 319–330.

[10] Andrei Bulatov, Peter Jeavons, and Andrei Krokhin. 2005. Classifying the Complexity of Constraints using Finite Algebras. *SIAM J. Comput.* 34, 3 (2005), 720–742. https://doi.org/10.1137/S0097539700376676

[11] Andrei A. Bulatov. 2011. Complexity of conservative constraint satisfaction problems. *ACM Transactions on Computational Logic* 12, 4 (2011). https://doi.org/10.1145/1970398.1970400 Article 24.
[12] Andrei A. Bulatov. 2016. Graphs of Relational Structures: Restricted Types. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'16)*. ACM, 642–651. https://doi.org/10.1145/2933575.2933604
[13] Siu On Chan. 2016. Approximation Resistance from Pairwise-Independent Subgroups. *Journal of the ACM* 63, 3 (2016). https://doi.org/10.1145/2873054 Article No. 27.
[14] Siu On Chan, James R. Lee, Prasad Raghavendra, and David Steurer. 2016. Approximate Constraint Satisfaction Requires Large LP Relaxations. *Journal of the ACM* 63, 4 (2016), 34:1–34:22. https://doi.org/10.1145/2811255
[15] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. 2009. Integrality gaps for Sherali-Adams relaxations. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC'09)*. ACM, 283–292. https://doi.org/10.1145/1536414.1536455
[16] Eden Chlamtáč and Madhur Tulsiani. 2012. Convex Relaxations and Integrality Gaps. In *Handbook on Semidefinite, Conic and Polynomial Optimization*, Miguel F. Anjos and Jean B. Lasserre (Eds.). International Series in Operations Research & Management Science, Vol. 166. Springer, 139–169. https://doi.org/10.1007/978-1-4614-0769-0_6
[17] David A. Cohen, Martin C. Cooper, Páidí Creed, Peter Jeavons, and Stanislav Živný. 2013. An algebraic theory of complexity for discrete optimisation. *SIAM J. Comput.* 42, 5 (2013), 915–1939. https://doi.org/10.1137/130906398
[18] Víctor Dalmau and Andrei A. Krokhin. 2013. Robust Satisfiability for CSPs: Hardness and Algorithmic Results. *ACM Transactions on Computation Theory* 5, 4 (2013). https://doi.org/10.1145/2540090 Article No. 15.
[19] Anuj Dawar and Pengming Wang. 2017. Lasserre Lower Bounds and Definability of Semidefinite Programming for CSPs. In *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'17)*. ACM, 1–12.
[20] Tomás Feder and Moshe Y. Vardi. 1998. The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory. *SIAM J. Comput.* 28, 1 (1998), 57–104. https://doi.org/10.1137/S0097539794266766
[21] Bernd Gärtner and Jiří Matoušek. 2012. *Approximation algorithms and semidefinite programming*. Springer Science & Business Media.
[22] Mrinal Kanti Ghosh and Madhur Tulsiani. 2017. From Weak to Strong LP Gaps for all CSPs. In *Proceedings of the 32nd Annual IEEE Conference on Computational Complexity (CCC'17)*. arXiv:1608.00497.
[23] Michel X. Goemans and David P. Williamson. 1995. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *Journal of the ACM* 42, 6 (1995), 1115–1145. https://doi.org/10.1145/227683.227684
[24] Dima Grigoriev. 2001. Linear lower bound on degrees of Positivstellensatz calculus proofs for the parity. *Theoretical Computer Science* 259, 1-2 (2001), 613–622. https://doi.org/10.1016/S0304-3975(00)00157-2
[25] Venkatesan Guruswami and Yuan Zhou. 2012. Tight Bounds on the Approximability of Almost-Satisfiable Horn SAT and Exact Hitting Set. *Theory of Computing* 8, 1 (2012), 239–267. https://doi.org/10.4086/toc.2012.v008a011
[26] Johan Håstad. 2001. Some optimal inapproximability results. *J. ACM* 48, 4 (2001), 798–859. https://doi.org/10.1145/502090.502098
[27] Pavol Hell and Jaroslav Nešetřil. 1990. On the Complexity of H-coloring. *Journal of Combinatorial Theory, Series B* 48, 1 (1990), 92–110. https://doi.org/10.1016/0095-8956(90)90132-J
[28] Pavol Hell and Jaroslav Nešetřil. 2008. Colouring, constraint satisfaction, and complexity. *Computer Science Review* 2, 3 (2008), 143–163. https://doi.org/10.1016/j.cosrev.2008.10.003
[29] Peter Jeavons, David A. Cohen, and Marc Gyssens. 1997. Closure properties of constraints. *J. ACM* 44, 4 (1997), 527–548. https://doi.org/10.1145/263867.263489
[30] David R. Karger, Rajeev Motwani, and Madhu Sudan. 1998. Approximate Graph Coloring by Semidefinite Programming. *Journal of the ACM* 45, 2 (1998), 246–265. https://doi.org/10.1145/274787.274791
[31] Vladimir Kolmogorov, Andrei A. Krokhin, and Michal Rolínek. 2017. The Complexity of General-Valued CSPs. *SIAM J. Comput.* 46, 3 (2017), 1087–1110. https://doi.org/10.1137/16M1091836
[32] Vladimir Kolmogorov, Johan Thapper, and Stanislav Živný. 2015. The power of linear programming for general-valued CSPs. *SIAM J. Comput.* 44, 1 (2015), 1–36. https://doi.org/10.1137/130945648
[33] Pravesh K. Kothari, Raghu Meka, and Prasad Raghavendra. 2017. 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). 590–603. https://doi.org/10.1145/3055399.3055438
[34] Marcin Kozik, Andrei Krokhin, Matt Valeriote, and Ross Willard. 2015. Characterizations of several Maltsev Conditions. *Algebra Universalis* 73, 3-4 (2015), 205–224.
[35] Marcin Kozik and Joanna Ochremiak. 2015. Algebraic Properties of Valued Constraint Satisfaction Problem. In *Proceedings of the 42nd International Colloquium on Automata, Languages and Programming (ICALP'15) (Lecture Notes in Computer Science)*, Vol. 9134. Springer, 846–858. https://doi.org/10.1007/978-3-662-47672-7_69
[36] Gábor Kun, Ryan O'Donnell, Suguru Tamaki, Yuichi Yoshida, and Yuan Zhou. 2012. Linear programming, width-1 CSPs, and robust satisfaction. In *Proceedings of the 3rd Innovations in Theoretical Computer Science (ITCS'12)*. ACM, 484–495. https://doi.org/10.1145/2090236.2090274
[37] Benoit Larose and Lázló Zádori. 2007. Bounded width problems and algebras. *Algebra Universalis* 56 (2007), 439–466. Issue 3-4.
[38] Jean B. Lasserre. 2001. Global Optimization with Polynomials and the Problem of Moments. *SIAM Journal on Optimization* 11, 3 (2001), 796–817. https://doi.org/10.1137/S1052623400366802
[39] Jean B. Lasserre. 2002. An Explicit Equivalent Positive Semidefinite Program for Nonlinear 0-1 Programs. *SIAM Journal on Optimization* 12, 3 (2002), 756–769. https://doi.org/10.1137/S1052623400380079
[40] Monique Laurent. 2003. A Comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre Relaxations for 0-1 Programming. *Mathematics of Operations Research* 28, 3 (2003), 470–496. https://doi.org/10.1287/moor.28.3.470.16391
[41] James R. Lee, Prasad Raghavendra, and David Steurer. 2014. Lower bounds on the size of semidefinite programming relaxations. *CoRR* abs/1411.6317 (November 2014). http://arxiv.org/abs/1411.6317
[42] James R. Lee, Prasad Raghavendra, and David Steurer. 2015. Lower Bounds on the Size of Semidefinite Programming Relaxations. In *Proceedings of the 47th Symposium on Theory of Computing (STOC'15)*. 567–576. https://doi.org/10.1145/2746539.2746599
[43] László Lovász and Alexander Schrijver. 1991. Cones of Matrices and Set-Functions and 0-1 Optimization. *SIAM Journal on Optimization* 1, 2 (1991), 166–190. https://doi.org/10.1137/0801013
[44] Miklós Maróti and Ralph McKenzie. 2008. Existence theorems for weakly symmetric operations. *Algebra Universalis* 59, 3-4 (2008), 463–489. https://doi.org/10.1007/s00012-008-2122-9
[45] Ryan O'Donnell and Yuan Zhou. 2013. Approximability and proof complexity. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'13)*. Society for Industrial and Applied Mathematics, 1537–1556.
[46] Prasad Raghavendra and David Steurer. 2017. Personal communication. (2017).
[47] Thomas J. Schaefer. 1978. The Complexity of Satisfiability Problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC'78)*. ACM, 216–226. https://doi.org/10.1145/800133.804350
[48] Grant Schoenebeck. 2008. Linear Level Lasserre Lower Bounds for Certain k-CSPs. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS'08)*. IEEE Computer Society, 593–602. https://doi.org/10.1109/FOCS.2008.74
[49] H. D. Sherali and W. P. Adams. 1990. A Hierarchy of Relaxations between the Continuous and Convex Hull Representations for Zero-One Programming Problems. *SIAM Journal of Discrete Mathematics* 3, 3 (1990), 411–430.
[50] Johan Thapper and Stanislav Živný. 2016. The complexity of finite-valued CSPs. *J. ACM* 63, 4 (2016). https://doi.org/10.1145/2974019 Article No. 37.
[51] Johan Thapper and Stanislav Živný. 2017. The power of Sherali-Adams relaxations for general-valued CSPs. *SIAM J. Comput.* 46, 4 (2017), 1241–1279. https://doi.org/10.1137/16M1079245
[52] Johan Thapper and Stanislav Živný. 2017. The limits of SDP relaxations for general-valued CSPs. In *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'17)*. ACM, 1–12.
[53] Madhur Tulsiani. 2009. CSP gaps and reductions in the Lasserre hierarchy. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC'09)*. ACM, 303–312. https://doi.org/10.1145/1536414.1536457 Full version available as TR08-104 at ECCC.
[54] Lieven Vandenberghe and Stephen P. Boyd. 1996. Semidefinite Programming. *SIAM Rev.* 38, 1 (1996), 49–95. https://doi.org/10.1137/1038003
[55] Vijay V. Vazirani. 2013. *Approximation algorithms*. Springer Science & Business Media.
[56] David P Williamson and David B Shmoys. 2011. *The design of approximation algorithms*. Cambridge University Press.

[57] Mihalis Yannakakis. 1991. Expressing Combinatorial Optimization Problems by Linear Programs. *J. Comput. System Sci.* 43, 3 (1991), 441–466. https://doi.org/10.1016/0022-0000(91)90024-Y
[58] Dmitriy Zhuk. 2017. The Proof of CSP Dichotomy Conjecture. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS'17)*. IEEE, 331–342.

Received July 2017; revised February 2018; accepted March 2018