Sherali-Adams relaxations for valued CSPs

Johan Thapper, Stanislav Zivny

Sherali-Adams relaxations for valued CSPs*

Johan Thapper[1] and Stanislav Živný[2]

[1] Université Paris-Est, Marne-la-Vallée, France
thapper@u-pem.fr
[2] Department of Computer Science, University of Oxford, UK
standa.zivny@cs.ox.ac.uk

Abstract. We consider Sherali-Adams linear programming relaxations for solving valued constraint satisfaction problems to optimality. The utility of linear programming relaxations in this context have previously been demonstrated using the lowest possible level of this hierarchy under the name of the basic linear programming relaxation (BLP). It has been shown that valued constraint languages containing only finite-valued weighted relations are tractable if, and only if, the integrality gap of the BLP is 1. In this paper, we demonstrate that almost all of the known tractable languages with arbitrary weighted relations have an integrality gap 1 for the Sherali-Adams relaxation with parameters $(2,3)$. The result is closely connected to the notion of bounded relational width for the ordinary constraint satisfaction problem and its recent characterisation.

1 Introduction

The constraint satisfaction problem provides a common framework for many theoretical and practical problems in computer science. An instance of the *constraint satisfaction problem* (CSP) consists of a collection of variables that must be assigned labels from a given domain subject to specified constraints. The CSP is NP-complete in general, but tractable fragments can be studied by, following Feder and Vardi [13], restricting the constraint relations allowed in the instances to a fixed, finite set, called the constraint language. The most successful approach to classifying the language-restricted CSP is the so-called algebraic approach [5, 3].

An important type of algorithms for CSPs are *consistency methods*. A constraint language is of *bounded relational width* if any CSP instance over this language can be solved by establishing (k, ℓ)-minimality for some fixed integers $1 \leq k \leq \ell$ [1]. The power of consistency methods for constraint languages has recently been fully characterised [21, 3] and it has been shown that any constraint language that is of bounded relational width is of relational width at most $(2, 3)$[1].

The CSP deals with only feasibility issues: Is there a solution satisfying certain constraints? In this work we are interested in problems that capture both

* The authors were supported by London Mathematical Society grant 41355. Stanislav Živný was supported by a Royal Society University Research Fellowship.

feasibility and optimisation issues: What is the best solution satisfying certain constraints? Problems of this form can be cast as valued constraint satisfaction problems [16].

An instance of the *valued constraint satisfaction problem* (VCSP) is given by a collection of variables that is assigned labels from a given domain with the goal to *minimise* an objective function given by a sum of weighted relations, each depending on some subset of the variables [8]. The weighted relations can take on finite rational values and positive infinity. The CSP corresponds to the special case of the VCSP when the codomain of all weighted relations is $\{0, \infty\}$.

Like the CSP, the VCSP is NP-hard in general and thus we are interested in the restrictions which give rise to tractable classes of problems. We restrict the *valued constraint language*; that is, all weighted relations in a given instance must belong to a fixed set of weighted relations on the domain. Languages that give rise to classes of problems solvable in polynomial time are called *tractable*, and languages that give rise to classes of problem that are NP-hard are called *intractable*. The computational complexity of Boolean (on a 2-element domain) valued constraint languages [8] and conservative (containing all $\{0, 1\}$-valued unary weighted relations) valued constraint languages [18] have been completely classified with respect to exact solvability.

Every VCSP problem has a natural linear programming (LP) relaxation, proposed independently by a number of authors, e.g. [6], and referred to as the *basic* LP relaxation (BLP) of the VCSP. It is the first level in the Sheralli-Adams hierarchy [24], which provides successively tighter LP relaxations of an integer LP. The BLP has been considered in the context of CSPs for robust approximability [20, 10] and constant-factor approximation [12, 9]. Higher levels of Sheral-Adams hierarchy have been considered for (in)approximability of CSPs [11, 30] but we are not aware of any results related to exact solvability of (valued) CSPs. Semidefinite programming relaxations have also been considered in the context of CSPs for approximability [23] and robust approximability [2].

Consistency methods, and in particular strong 3-consistency has played an important role as a preprocessing step in establishing tractability of valued constraint languages. Cohen et al. proved the tractability of valued constraint languages improved by a symmetric tournament pair (STP) multimorphism via strong 3-consistency preprocessing, and an involved reduction to submodular function minimisation [7]. They also showed that the tractability of any valued constraint language improved by a tournament pair multimorphism via a preprocessing using results on constraint languages invariant under a 2-semilattice polymorphism, which relies on $(3, 3)$-minimality, and then reducing to the STP case. The only tractable conservative valued constraint languages are those admitting a pair of fractional polymorphisms called STP and MJN [18]; again, the tractability of such languages is proved via a 3-consistency preprocessing reducing to the STP case. It is natural to ask whether this nested use of consistency methods are necessary.

Contributions In [26, 17], the authors showed that the BLP of the VCSP can be used to solve the problem for many valued constraint languages. In [27], it was then shown that for VCSPs with weighted relations taking only finite values, the BLP precisely characterises the tractable (finite-)valued constraint languages; i.e., if BLP fails to solve any instance of some valued constraint language of this type, then this language is NP-hard.

In this paper, we show that a higher-level Sherali-Adams linear programming relaxation [24] suffices to solve most of the previously known tractable valued constraint languages with arbitrary weighted relations, and in particular, all known valued constraint languages that involve some optimisation (and thus do not reduce to constraint languages containing only relations) except for valued constraint languages of generalised weak tournament pair type [29]; such languages are known to be tractable [29] but we do not know whether they are tractable by our linear programming relaxation.

Our main result, Theorem 4, shows that if the support clone of a valued constraint language Γ of finite size contains weak near-unanimity operations of all but finitely many arities, then Γ is tractable via the Sherali-Adams relaxation with parameters $(2, 3)$. This tractability condition is precisely the bounded relational width condition for constraint languages of finite size containing all constants [21, 3], and our proof fundamentally relies on the results of Barto and Kozik [3] and Barto [1].

It is folklore that the kth level of Sherali-Adams hierarchy establishes k-consistency for CSPs. We demonstrate that one linear programming relaxation is powerful enough to establish consistency as well as solving an optimisation problem in one go without the need of nested applications of consistency methods. For example, valued constraint languages having a tournament pair multimorphism were previously known to be tractable using ingenious application of various consistency techniques, advanced analysis of constraint networks using modular decompositions, and submodular function minimisation [7]. Here, we show that an even less restrictive condition (having a binary conservative commutative operation in some fractional polymorphism) ensures that the Sherali-Adams relaxation solves all instances to optimum.

Finally, we also give a short proof of the dichotomy theorem for conservative valued constraint languages [18], which previously needed lengthy arguments (although we still rely on Takhanov [25] for a part of the proof).

2 Preliminaries

Valued CSPs Throughout the paper, let D be a fixed finite set of size at least two. We call D the *domain*, the elements of D *labels* and say that weighted relations take *values*. Let $\overline{\mathbb{Q}} = \mathbb{Q} \cup \{\infty\}$ denote the set of rational numbers with (positive) infinity.

Definition 1. *An m-ary* relation *over D is any mapping $\phi : D^m \to \{c, \infty\}$ for some $c \in \mathbb{Q}$. We denote by $\mathbf{R}_D$ the set of all relations on D.*

Definition 2. *An m-ary* weighted relation *over D is any mapping $\phi : D^m \to \overline{\mathbb{Q}}$. We write $ar(\phi) = m$ for the arity of ϕ. We denote by $\mathbf{\Phi}_D$ the set of all weighted relations on D.*

For any m-ary weighted relation $\phi \in \mathbf{\Phi}_D$, we denote by $\mathrm{Feas}(\phi) = \{\mathbf{x} \in D^m \,|\, \phi(\mathbf{x}) < \infty\} \in \mathbf{R}_D$ the underlying m-ary *feasibility relation*, and by $\mathrm{Opt}(\phi) = \{\mathbf{x} \in \mathrm{Feas}(\phi) \mid \forall \mathbf{y} \in D^m : \phi(\mathbf{x}) \leq \phi(\mathbf{y})\} \in \mathbf{R}_D$ the m-ary *optimality relation*, which contains the tuples on which ϕ is minimised. A weighted relation $\phi : D^m \to \overline{\mathbb{Q}}$ is called *finite-valued* if $\mathrm{Feas}(\phi) = D^m$.

Definition 3. *Let $V = \{x_1, \ldots, x_n\}$ be a set of variables. A* valued constraint *over V is an expression of the form $\phi(\mathbf{x})$ where $\phi \in \mathbf{\Phi}_D$ and $\mathbf{x} \in V^{ar(\phi)}$. The number m is called the* arity *of the constraint, the weighted relation ϕ is called the* constraint weighted relation*, and the tuple $\mathbf{x}$ the* scope *of the constraint.*

Definition 4. *An instance of the* valued constraint satisfaction problem*, VCSP, is specified by a finite set $V = \{x_1, \ldots, x_n\}$ of variables, a finite set D of labels, and an* objective function *I expressed as follows: $I(x_1, \ldots, x_n) = \sum_{i=1}^{q} \phi_i(\mathbf{x}_i)$, where each $\phi_i(\mathbf{x}_i)$, $1 \leq i \leq q$, is a valued constraint over V. Each constraint can appear multiple times in I. The goal is to find an* assignment *(or* solution*) of labels to the variables minimising I.*

A solution is called *feasible* (or *satisfying*) if it is of finite value. A VCSP instance I is called *satisfiable* if there is a feasible solution to I. CSPs are a special case of VCSPs with (unweighted) relations with the goal to determine the existence of a feasible solution.

Example 1. In the Min-UnCut problem the goal is to find a partition of the vertices of a given graph into two parts so that the number of edges inside the two partitions is minimised. For a graph (V, E) with $V = \{x_1, \ldots, x_n\}$, this NP-hard problem can be expressed as the VCSP instance $I(x_1, \ldots, x_n) = \sum_{(i,j) \in E} \phi_{\mathsf{xor}}(x_i, x_j)$ over the Boolean domain $D = \{0, 1\}$, where $\phi_{\mathsf{xor}} : \{0,1\}^2 \to \overline{\mathbb{Q}}$ is defined by $\phi_{\mathsf{xor}}(x, y) = 1$ if $x = y$ and $\phi_{\mathsf{xor}}(x, y) = 0$ if $x \neq y$.

Definition 5. *Any set $\Delta \subseteq \mathbf{R}_D$ is called a* constraint language *over D. Any set $\Gamma \subseteq \mathbf{\Phi}_D$ is called a* valued constraint language *over D. We denote by* $\mathrm{VCSP}(\Gamma)$ *the class of all VCSP instances in which the constraint weighted relations are all contained in Γ. For a constraint language Δ, we denote by* $\mathrm{CSP}(\Delta)$ *the class* $\mathrm{VCSP}(\Delta)$ *to emphasise the fact that there is no optimisation involved.*

Definition 6. *A valued constraint language Γ is called* tractable *if* $\mathrm{VCSP}(\Gamma')$ *can be solved (to optimality) in polynomial time for every finite subset $\Gamma' \subseteq \Gamma$, and Γ is called* intractable *if* $\mathrm{VCSP}(\Gamma')$ *is NP-hard for some finite $\Gamma' \subseteq \Gamma$.*

Operations and Clones We recall some basic terminology from universal algebra. Given an m-tuple $\mathbf{x} \in D^m$, we denote its ith entry by $\mathbf{x}[i]$ for $1 \leq i \leq m$. Any mapping $f : D^k \to D$ is called a k-ary *operation*; f is called *conservative* if $f(x_1, \ldots, x_k) \in \{x_1, \ldots, x_k\}$ and *idempotent* if $f(x, \ldots, x) = x$. We will apply a k-ary operation f to k m-tuples $\mathbf{x}_1, \ldots, \mathbf{x}_k \in D^m$ coordinatewise, that is,

$$f(\mathbf{x}_1, \ldots, \mathbf{x}_k) = (f(\mathbf{x}_1[1], \ldots, \mathbf{x}_k[1]), \ldots, f(\mathbf{x}_1[m], \ldots, \mathbf{x}_k[m])) \,. \tag{1}$$

Definition 7. *Let ϕ be an m-ary weighted relation on D. A k-ary operation f on D is a* polymorphism *of ϕ if, for any $\mathbf{x}_1, \ldots, \mathbf{x}_k \in D^m$ with $\mathbf{x}_i \in \mathrm{Feas}(\phi)$ for all $1 \leq i \leq k$, we have that $f(\mathbf{x}_1, \ldots, \mathbf{x}_k) \in \mathrm{Feas}(\phi)$.*

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

A k-ary *projection* is an operation of the form $\pi_i^{(k)}(x_1, \ldots, x_k) = x_i$ for some $1 \leq i \leq k$. Projections are polymorphisms of all valued constraint languages.

The *composition* of a k-ary operation $f : D^k \to D$ with k ℓ-ary operations $g_i : D^\ell \to D$ for $1 \leq i \leq k$ is the ℓ-ary operation $f[g_1, \ldots, g_k] : D^\ell \to D$ defined by $f[g_1, \ldots, g_k](x_1, \ldots, x_\ell) = f(g_1(x_1, \ldots, x_\ell), \ldots, g_k(x_1, \ldots, x_\ell))$.

We denote by $\mathcal{O}_D$ the set of all finitary operations on D and by $\mathcal{O}_D^{(k)}$ the k-ary operations in $\mathcal{O}_D$. A *clone* of operations, $C \subseteq \mathcal{O}_D$, is a set of operations on D that contains all projections and is closed under composition. It is easy to show that $\mathrm{Pol}(\Gamma)$ is a clone for any valued constraint language Γ.

Definition 8. *A k-ary* fractional operation *ω is a probability distribution over $\mathcal{O}_D^{(k)}$. We define $\mathrm{supp}(\omega) = \{f \in \mathcal{O}_D^{(k)} \mid \omega(f) > 0\}$.*

Definition 9. *Let ϕ be an m-ary weighted relation on D and let ω be a k-ary fractional operation on D. We call ω a* fractional polymorphism *of ϕ (and say that ϕ is* improved by *ω) if $\mathrm{supp}(\omega) \subseteq \mathrm{Pol}(\phi)$ and for any $\mathbf{x}_1, \ldots, \mathbf{x}_k \in D^m$ with $\mathbf{x}_i \in \mathrm{Feas}(\phi)$ for all $1 \leq i \leq k$, we have*

$$\mathop{\mathbb{E}}_{f \sim \omega} [\phi(f(\mathbf{x_1}, \ldots, \mathbf{x_k}))] \;\leq\; \mathrm{avg}\{\phi(\mathbf{x_1}), \ldots, \phi(\mathbf{x_k})\}\,. \tag{2}$$

Definition 10. *For any valued constraint language $\Gamma \subseteq \mathbf{\Phi}_D$, we define $\mathrm{fPol}(\Gamma)$ to be the set of all fractional operations that are fractional polymorphisms of all weighted relations $\phi \in \Gamma$. We write $\mathrm{fPol}(\phi)$ for $\mathrm{fPol}(\{\phi\})$.*

Example 2. A valued constraint language on domain $\{0, 1\}$ is called *submodular* if it has the fractional polymorphism ω defined by $\omega(\min) = \omega(\max) = \frac{1}{2}$, where min and max are the two binary operations that return the smaller and larger of its two arguments respectively with respect to the usual order $0 < 1$.

For a valued constraint language Γ we define $\mathrm{supp}(\Gamma) \;=\; \bigcup_{\omega \in \mathrm{fPol}(\Gamma)} \mathrm{supp}(\omega)$.

Lemma 1. *For any valued constraint language Γ, $\mathrm{supp}(\Gamma)$ is a clone.*

We note that Lemma 1 has also been observed in [22] and in [14].

A special case of the following lemma has been observed, in the context of Min-Sol problems [29], by Hannes Uppman.[3]

Lemma 2. *Let Γ be a valued constraint language of finite size on a domain D and let $f \in \mathrm{Pol}(\Gamma)$. Then, $f \in \mathrm{supp}(\Gamma)$ if, and only if, $f \in \mathrm{Pol}(\mathrm{Opt}(I))$ for all instances I of VCSP(Γ).*

[3] Private communication.

Cores and Constants Let $\mathcal{C}_D = \{\{(d)\} \mid d \in D\}$ be the set of constant unary relations on D.

Definition 11. *Let Γ be a valued constraint language with domain D and let $S \subseteq D$. The* sub-language $\Gamma[S]$ of Γ induced by S *is the valued constraint language defined on domain S and containing the restriction of every weighted relation $\phi \in \Gamma$ onto S.*

Definition 12. *A valued constraint language Γ is* a core *if all unary operations in* $\text{supp}(\Gamma)$ *are bijections. A valued constraint language Γ' is a* core of Γ *if Γ' is a core and $\Gamma' = \Gamma[f(D)]$ for some $f \in \text{supp}(\omega)$ with ω a unary fractional polymorphism of Γ.*

Lemma 3. *Let Γ be a valued constraint language and Γ' a core of Γ. Then, for all instances I of VCSP(Γ) and I' of VCSP(Γ'), where I' is obtained from I by substituting each function in Γ for its restriction in Γ', the optimum of I and I' coincide.*

Lemma 4 ([22]). *Let Γ be a core valued constraint language. The problems VCSP(Γ) and VCSP($\Gamma \cup \mathcal{C}_D$) are polynomial-time equivalent.*

A special case of Lemma 4 for finite-valued constraint languages was proved by the authors in [27], building on [15], and Lemma 4 can be proved similarly.

3 Sherali-Adams and Valued Relational Width

In this section, we state and prove our main result on the applicability of Sherali-Adams relaxations to VCSPs. First, we define some notions concerning *bounded relational width* which is the basis for our proof.

We write (S, C) for (valued) constraints that involve (unweighted) relations, where S is the scope and C is the constraint relation. For a tuple $\mathbf{x} \in D^S$, we denote by $\pi_{S'}(\mathbf{x})$ its projection onto $S' \subseteq S$. For a constraint (S, C), we define $\pi_{S'}(C) = \{\pi_{S'}(\mathbf{x}) \mid \mathbf{x} \in C\}$.

Let $1 \leq k \leq \ell$ be integers. The following definition is equivalent[4] to the definition of (k, ℓ)-minimality for CSP instances given in [1].

Definition 13. *A CSP-instance $J = (V, D, \{(S_i, C_i)\}_{i=1}^q)$ is said to be (k, ℓ)-*minimal *if:*

- *For every $S \subseteq V$, $|S| \leq \ell$, there exists $1 \leq i \leq q$ such that $S = S_i$.*
- *For every $i, j \in [q]$ such that $|S_j| \leq k$ and $S_j \subseteq S_i$, $C_j = \pi_{S_j}(C_i)$.*

There is a straightforward polynomial-time algorithm for finding an equivalent (k, ℓ)-minimal instance [1]. This leads to the notion of *relational width*:

[4] The two requirements in [1] are: for every $S \subseteq V$ with $|S| \leq \ell$ we have $S \subseteq S_i$ for some $1 \leq i \leq q$; and for every set $W \subseteq V$ with $|W| \leq k$ and every $1 \leq i, j \leq q$ with $W \subseteq S_i$ and $W \subseteq S_j$ we have $\pi_W(C_i) = \pi_W(C_j)$.

Definition 14. *A constraint language Δ has relational width (k, ℓ) if, for every instance $J \in \mathrm{CSP}(\Delta)$, an equivalent (k, ℓ)-minimal instance is non-empty if, and only if, J has a solution.*

A k-ary idempotent operation $f : D^k \to D$ is called a *weak near-unanimity* (WNU) operation if, for all $x, y \in D$, $f(y, x, x, \ldots, x) = f(x, y, x, x, \ldots, x) = f(x, x, \ldots, x, y)$.

Definition 15. *We say that a clone of operations satisfies the* bounded width condition (BWC) *if it contains WNU operations of all but finitely many arities.*

Theorem 1 ([3, 21]). *Let Δ be a constraint language of finite size containing all constant unary relations. Then, Δ has bounded relational width if, and only if,* $\mathrm{Pol}(\Delta)$ *satisfies the BWC.*

Theorem 2 ([1]). *Let Δ be a constraint language. If Δ has bounded relational width, then it has relational width $(2, 3)$.*

Let $I(x_1, \ldots, x_n) = \sum_{i=1}^{q} \phi_i(S_i)$ be an instance of the VCSP, where $S_i \subseteq V = \{x_1, \ldots, x_n\}$ and $\phi_i \colon D^{|S_i|} \to \overline{\mathbb{Q}}$. First, we make sure that every non-empty $S \subseteq V$ with $|S| \leq \ell$ appears in some term $\phi_i(S)$, possibly by adding constant-0 weighted relations. The Sherali-Adams [24] linear programming relaxation with parameters (k, ℓ) is defined as follows. The variables are $\lambda_i(\mathbf{s})$ for every $i \in [q]$ and tuple $\mathbf{s} \in D^{S_i}$.

$$\begin{aligned}
\min \sum_{i=1}^{q} \sum_{\mathbf{s} \in \mathrm{Feas}(\phi_i)} \lambda_i(\mathbf{s}) \phi_i(\mathbf{s}) & \\
\lambda_j(\mathbf{t}) = \sum_{\mathbf{s} \in D^{S_i}, \pi_{S_j}(\mathbf{s}) = \mathbf{t}} \lambda_i(\mathbf{s}) & \qquad \forall i, j \in [q] : S_j \subseteq S_i, |S_j| \leq k, \mathbf{t} \in D^{S_j} \\
\sum_{\mathbf{s} \in D^{S_i}} \lambda_i(\mathbf{s}) = 1 & \qquad \forall i \in [q] \\
\lambda_i(\mathbf{s}) = 0 & \qquad \forall i \in [q], \mathbf{s} \notin \mathrm{Feas}(\phi_i) \\
\lambda_i(\mathbf{s}) \geq 0 & \qquad \forall i \in [q], \mathbf{s} \in D^{S_i}
\end{aligned}$$

The $\mathrm{SA}(k, \ell)$ optimum is always less than or equal to the VCSP optimum, hence the program is a relaxation. In anticipation of our main theorem, we make the following definition.

Definition 16. *A valued constraint language Γ has* valued relational width (k, ℓ) *if, for every instance I of $\mathrm{VCSP}(\Gamma)$, if the SA(k, ℓ)-relaxation of I has a feasible solution, then its optimum coincides with the optimum of I.*

For a feasible solution λ of $\mathrm{SA}(k, \ell)$, let $\mathrm{supp}(\lambda_i) = \{\mathbf{s} \in D^{S_i} \mid \lambda_i(\mathbf{s}) > 0\}$.

Lemma 5. *Let I be an instance of VCSP(Γ). Assume that SA(k, ℓ) for I is feasible. Then, there exists an optimal solution λ^* to SA(k, ℓ) such that, for every i, $\mathrm{supp}(\lambda_i^*)$ is closed under every operation in $\mathrm{supp}(\Gamma)$.*

Theorem 3. *Let Γ be a valued constraint language of finite size containing all constant unary relations. If* $\operatorname{supp}(\Gamma)$ *satisfies the BWC, then Γ has valued relational width* $(2, 3)$.

Proof. Let I be an instance of VCSP(Γ). The dual of the SA(k, ℓ) relaxation can be written in the following form, with variables z_i for $i \in [q]$ and $y_{j,\mathbf{t},i}$ for $i, j \in [q]$ such that $S_j \subseteq S_i$, $|S_j| \leq k$, and $\mathbf{t} \in D^{S_j}$. The dual variables corresponding to $\lambda_i(\mathbf{s}) = 0$ are eliminated together with the dual inequalities for $i, \mathbf{s} \notin \operatorname{Feas}(\phi_i)$.

$$\max \sum_{i=1}^{q} z_i$$

$$z_i \leq \phi_i(\mathbf{s}) + \sum_{j\in[q],S_j\subseteq S_i} y_{j,\pi_{S_j}(\mathbf{s}),i} - \sum_{j\in[q],S_i\subseteq S_j} y_{i,\mathbf{s},j} \quad \forall i \in [q]\,, |S_i| \leq k, \mathbf{s} \in \operatorname{Feas}(\phi_i)$$

$$z_i \leq \phi_i(\mathbf{s}) + \sum_{\substack{j\in[q],S_j\subseteq S_i \\ |S_j|\leq k}} y_{j,\pi_{S_j}(\mathbf{s}),i} \quad \forall i \in [q]\,, |S_i| > k, \mathbf{s} \in \operatorname{Feas}(\phi_i)$$

It is clear that if I has a feasible solution, then so does the SA(k, ℓ) primal. Assume that the SA($2, 3$)-relaxation has a feasible solution. By Lemma 5, there exists an optimal primal solution λ^* such that, for every $i \in [q]$, $\operatorname{supp}(\lambda_i^*)$ is closed under $\operatorname{supp}(\Gamma)$. Let y^*, z^* be an optimal dual solution.

Let $\Delta = \{C_i\}_{i=1}^q \cup \{\mathcal{C}_D\}$, where $C_i = \operatorname{supp}(\lambda_i^*)$, and consider the instance $J = (V, D, \{(S_i, C_i)\}_{i=1}^q)$ of CSP(Δ). We make the following observations:

1. By construction of λ^*, $\operatorname{supp}(\Gamma) \subseteq \operatorname{Pol}(\Delta)$, so Δ contains all constant unary relations and satisfies the BWC. By Theorems 1 and 2, the language Δ has relational width $(2, 3)$.
2. The first set of constraints in the primal say that if $i, j \in [q]$, $|S_j| \leq 2$ and $S_j \subseteq S_i$, then $\lambda_j^*(\mathbf{t}) > 0$ (i.e., $\mathbf{t} \in C_j$) iff $\sum_{\mathbf{s}\in D^{S_i},\pi_{S_j}(\mathbf{s})=\mathbf{t}} \lambda_i^*(\mathbf{s}) > 0$ (i.e., $\mathbf{t} \in \pi_{S_j}(C_i)$). In other words, J is $(2, 3)$-minimal.

These two observations imply that J has a satisfying assignment $\sigma \colon V \to D$. By complementary slackness, since $\lambda_i^*(\sigma(S_i)) > 0$ for every $i \in [q]$, we must have equality in the corresponding rows in the dual indexed by i and $\sigma(S_i)$. Hence,

$$\sum_{i=1}^{q} z_i^* \;=\; \sum_{i=1}^{q} \phi_i(\sigma(S_i)) + \Big(\sum_{i=1}^{q} \sum_{\substack{j\in[q],S_j\subseteq S_i \\ |S_j|\leq 2}} y^*_{j,\pi_{S_j}(\sigma(S_i)),i} - \sum_{\substack{i\in[q] \\ |S_i|\leq 2}} \sum_{\substack{j\in[q] \\ S_i\subseteq S_j}} y^*_{i,\sigma(S_i),j}\Big) \quad (3)$$

By noting that $\pi_{S_j}(\sigma(S_i)) = \sigma(S_j)$, we can rewrite the expression in parenthesis on the right-hand side of (3) as:

$$\sum_{\substack{i,j\in[q],S_j\subseteq S_i \\ |S_j|\leq 2}} y^*_{j,\sigma(S_j),i} - \sum_{\substack{i,j\in[q],S_i\subseteq S_j \\ |S_i|\leq 2}} y^*_{i,\sigma(S_i),j} \;=\; 0. \quad (4)$$

Therefore, $\sum_{i=1}^{q} \sum_{\mathbf{s}\in \operatorname{Feas}(\phi_i)} \lambda_i^*(\mathbf{s})\phi_i(\mathbf{s}) = \sum_{i=1}^{q} z_i^* \;=\; \sum_{i=1}^{q} \phi_i(\sigma(S_i))$,where the first equality follows by strong LP-duality, and the second by (3) and (4). Since I was an arbitrary instance of VCSP(Γ), the theorem follows.

4 Generalisations of Known Tractable Languages

In this section, we give some applications of Theorem 3. Firstly, we show that the BWC is preserved by going to a core and the addition of constant unary relations. Hence the BWC guarantees valued relational width $(2, 3)$ also for languages not necessarily containing constant unary relations, as required by Theorem 3.

Lemma 6. *Let Γ be a valued constraint language of finite size on domain D and Γ' a core of Γ on domain $D' \subseteq D$. Then,* $\mathrm{supp}(\Gamma)$ *satisfies the BWC if, and only if,* $\mathrm{supp}(\Gamma' \cup \mathcal{C}_{D'})$ *satisfies the BWC.*

Theorem 4. *Let Γ be a valued constraint language of finite size. If* $\mathrm{supp}(\Gamma)$ *satisfies the BWC, then Γ has valued relational width* $(2, 3)$*.*

Secondly, we show that for any VCSP instance over a language of valued relational width $(2, 3)$ we can not only compute the value of an optimal solution but we can also find an optimal assignment in polynomial time.

Proposition 1. *Let Γ be a valued constraint language of finite size and I an instance of VCSP(Γ). If* $\mathrm{supp}(\Gamma)$ *satisfies the BWC, then an optimal assignment to I can be found in polynomial time.*

Finally, we show that testing for the BWC is a decidable problem.

Proposition 2. *Testing whether a valued constraint language of finite size satisfies the BWC is decidable.*

Tractable Languages Here we give some examples of previously studied valued constraint languages and show that they all have valued relational width $(2, 3)$.

Example 3. Let ω be a ternary fractional operation defined by $\omega(f) = \omega(g) = \omega(h) = \frac{1}{3}$ for some (not necessarily distinct) majority operations f, g, and h. Cohen et al. proved the tractability of any language improved by ω by a reduction to CSPs with a majority polymorphism [8].

Example 4. Let ω be a ternary fractional operation defined by $\omega(f) = \frac{2}{3}$ and $\omega(g) = \frac{1}{3}$, where $f : \{0, 1\}^3 \to \{0, 1\}$ is the Boolean majority operation and $g : \{0, 1\}^3 \to \{0, 1\}$ is the Boolean minority operation. Cohen et al. proved the tractability of any language improved by ω by a simple propagation algorithm [8].

Example 5. Generalising Example 4 from Boolean to arbitrary domains, let ω be a ternary fractional operation such that $\omega(f) = \frac{1}{3}$, $\omega(g) = \frac{1}{3}$, and $\omega(h) = \frac{1}{3}$ for some (not necessarily distinct) conservative majority operations f and g, and a conservative minority operation h; such an ω is called an MJN. Kolmogorov and Živný proved the tractability of any language improved by ω by a 3-consistency algorithm and a reduction, via Example 6, to submodular function minimisation [18].

Corollary 1. *Let Γ be a valued constraint language of finite size such that* $\operatorname{supp}(\Gamma)$ *contains a majority operation. Then, Γ has valued relational width* $(2, 3)$.

Example 6. Let ω be a binary fractional operation defined by $\omega(f) = \omega(g) = \frac{1}{2}$, where f and g are conservative and commutative operations and $f(x, y) \neq g(x, y)$ for every x and y; such an ω is called a *symmetric tournament pair* (STP). Cohen et al. proved the tractability of any language improved by ω by a 3-consistency algorithm and an ingenious reduction to submodular function minimisation [7]. Such languages were shown to be the only tractable languages among conservative finite-valued constraint languages [18].

Corollary 2. *Let Γ be a valued constraint language of finite size such that* $\operatorname{supp}(\Gamma)$ *contains two symmetric tournament operations (that is, binary operations f and g that are both conservative and commutative and $f(x, y) \neq g(x, y)$ for every x and y). Then, Γ has valued relational width* $(2, 3)$.

Example 7. Generalising Example 6, let ω be a binary fractional operation defined by $\omega(f) = \omega(g) = \frac{1}{2}$, where f and g are conservative and commutative operations; such an ω is called a *tournament pair*. Cohen et al. proved the tractability of any language improved by ω by a consistency-reduction relying on Bulatov's result [4], which in turn relies on 3-consistency, to the STP case from Example 6 [7].

Corollary 3. *Let Γ be a valued constraint language of finite size such that* $\operatorname{supp}(\Gamma)$ *contains a tournament operation (that is, a binary conservative and commutative operation). Then, Γ has valued relational width* $(2, 3)$.

Example 8. In this example we denote by $\{\{\ldots\}\}$ a multiset. Let ω be a binary fractional operation on D defined by $\omega(f) = \omega(g) = \frac{1}{2}$ and let μ be a ternary fractional operation on D defined by $\mu(h_1) = \mu(h_2) = \mu(h_3) = \frac{1}{3}$. Moreover, assume that $\{\{f(x, y), g(x, y)\}\} = \{\{x, y\}\}$ for every x and y and $\{\{h_1(x, y, z), h_2(x, y, z), h_3(x, y, z)\}\} = \{\{x, y, z\}\}$ for every x, y, and z. Let Γ be a language on D such that for every two-element subset $\{a, b\} \subseteq D$, either $\omega|_{\{a,b\}}$ is an STP or $\mu|_{\{a,b\}}$ is an MJN. Kolmogorov and Živný proved the tractability of Γ by a 3-consistency algorithm and a reduction, via Example 6, to submodular function minimisation [18]. Such languages were shown to be the only tractable languages among conservative valued constraint languages [18].

Corollary 4. *Let Γ be a valued constraint language of finite size with fractional polymorphisms ω and μ as described in Example 8. Then, Γ has valued relational width* $(2, 3)$.

Dichotomy for Conservative Valued Constraint Languages A valued constraint language Γ is called *conservative* if Γ contains all unary $\{0, 1\}$-valued weighted relations. Kolmogorov and Živný gave a dichotomy theorem for such languages, showing that they are either NP-hard, or tractable, cf. Example 8. Here we prove this dichotomy using the SA$(2, 3)$-relaxation as the algorithmic tool.

Lemma 7. *Let Γ be a valued constraint language and I be any instance of VCSP(Γ). Then, VCSP($\Gamma \cup \{\mathrm{Opt}(I)\}$) polynomial-time reduces to VCSP(Γ).*

The following theorem was proved by Takhanov [25] with a reduction, essentially amounting to Lemma 7, added in [18].

Theorem 5 ([18, 25]). *Let Γ be a conservative valued constraint language. If $\mathrm{Pol}(\Gamma)$ does not contain a majority polymorphism, then Γ is NP-hard.*

Theorem 6. *Let Γ be a conservative valued constraint language. Either Γ is NP-hard, or Γ has valued relational width $(2, 3)$.*

Proof. Let F be the set of majority operations in $\mathrm{Pol}(\Gamma)\setminus\mathrm{supp}(\Gamma)$. By Lemma 2, for each $f \in F$, there is an instance I_f of VCSP(Γ) such that $f \notin \mathrm{Pol}(\mathrm{Opt}(I_f))$. Let $\Gamma' = \Gamma \cup \{\mathrm{Opt}(I_f) \mid f \in F\}$. Assume that $\mathrm{Pol}(\Gamma')$ contains a majority polymorphism f. Then, $f \notin F$, so $f \in \mathrm{supp}(\Gamma)$. From Corollary 1, it follows that Γ has valued relational width $(2, 3)$. If $\mathrm{Pol}(\Gamma')$ does not contain a majority polymorphism, then, since Γ is conservative, so is Γ', and hence Γ' is NP-hard by Theorem 5. Therefore, Γ is NP-hard by Lemma 7.

5 Conclusions

We have shown that most previously studied tractable valued constraint languages that are not purely relational fall into the cases covered by Theorem 4. In the full version of this paper, we will prove the converse of Theorem 4, thus giving a precise characterisation of the power of valued relational width $(2, 3)$, as well as some computational complexity consequences.

References

1. Libor Barto. The collapse of the bounded width hierarchy. *Journal of Logic and Computation*, 2014.
2. Libor Barto and Marcin Kozik. Robust Satisfiability of Constraint Satisfaction Problems. In *Proc. STOC'12*, pages 931–940. ACM, 2012.
3. Libor Barto and Marcin Kozik. Constraint Satisfaction Problems Solvable by Local Consistency Methods. *Journal of the ACM*, 61(1), 2014. Article No. 3.
4. Andrei Bulatov. Combinatorial problems raised from 2-semilattices. *Journal of Algebra*, 298:321–339, 2006.
5. Andrei Bulatov, Andrei Krokhin, and Peter Jeavons. Classifying the Complexity of Constraints using Finite Algebras. *SIAM Journal on Computing*, 34(3):720–742, 2005.
6. Chandra Chekuri, Sanjeev Khanna, Joseph Naor, and Leonid Zosin. A linear programming formulation and approximation algorithms for the metric labeling problem. *SIAM J. on Discrete Mathematics*, 18(3):608–625, 2004.
7. David A. Cohen, Martin C. Cooper, and Peter G. Jeavons. Generalising submodularity and Horn clauses: Tractable optimization problems defined by tournament pair multimorphisms. *Theoretical Computer Science*, 401(1-3):36–51, 2008.

8. David A. Cohen, Martin C. Cooper, Peter G. Jeavons, and Andrei A. Krokhin. The Complexity of Soft Constraint Satisfaction. *Artif. Intell.*, 170:983–1016, 2006.
9. Víctor Dalmau, Andrei Krokhin, and Rajsekar Manokaran. Towards a characterization of constant-factor approximable Min CSPs. In *Proc. SODA'15*, 2015.
10. Víctor Dalmau and Andrei A. Krokhin. Robust Satisfiability for CSPs: Hardness and Algorithmic Results. *ACM ToCT*, 5(4), 2013. Article No. 15.
11. Wenceslas Fernandez de la Vega and Claire Kenyon-Mathieu. Linear programming relaxations of maxcut. In *Proc. SODA'07*, pages 53–61. SIAM, 2007.
12. Alina Ene, Jan Vondrák, and Yi Wu. Local distribution and the symmetry gap: Approximability of multiway partitioning problems. In SODA'13, 306–325, 2013.
13. Tomás Feder and Moshe Y. Vardi. The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.
14. Peter Fulla and Stanislav Živný. A Galois Connection for Valued Constraint Languages of Infinite Size. In *Proc. ICALP'15*, Springer, 2015.
15. Anna Huber, Andrei Krokhin, and Robert Powell. Skew bisubmodularity and valued CSPs. *SIAM Journal on Computing*, 43(3):1064–1084, 2014.
16. Peter Jeavons, Andrei Krokhin, and Stanislav Živný. The complexity of valued constraint satisfaction. *Bulletin of the EATCS*, 113:21–55, 2014.
17. 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.
18. Vladimir Kolmogorov and Stanislav Živný. The complexity of conservative valued CSPs. *Journal of the ACM*, 60(2), 2013. Article No. 10.
19. Marcin Kozik, Andrei Krokhin, Matt Valeriote, and Ross Willard. Characterizations of several Maltsev Conditions. *Algebra Universalis*, 2014. To appear.
20. Gábor Kun, Ryan O'Donnell, Suguru Tamaki, Yuichi Yoshida, and Yuan Zhou. Linear programming, width-1 CSPs, and robust satisfaction. *ITCS'12*, 484–495.
21. Benoit Larose and Lázló Zádori. Bounded width problems and algebras. *Algebra Universalis*, 56:439–466, 2007.
22. Marcin Kozik and Joanna Ochremiak. Algebraic Properties of Valued Constraint Satisfaction Problem. In *Proc. ICALP'15*, Springer, 2015.
23. Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proc. STOC'08*, pages 245–254. ACM, 2008.
24. H. D. Sherali and W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal of Discrete Mathematics*, 3(3):411–430, 1990.
25. Rustem Takhanov. A Dichotomy Theorem for the General Minimum Cost Homomorphism Problem. In *Proc. STACS'10*, pages 657–668, 2010.
26. Johan Thapper and Stanislav Živný. The power of linear programming for valued CSPs. In *Proc. FOCS'12*, pages 669–678. IEEE, 2012.
27. Johan Thapper and Stanislav Živný. The complexity of finite-valued CSPs. In *Proc. STOC'13*, pages 695–704. ACM, 2013. Full version arXiv:1210.2977v3 (February 2015).
28. Johan Thapper and Stanislav Živný. Necessary Conditions on Tractability of Valued Constraint Languages. Technical report, February 2015. arXiv:1502.03482.
29. Hannes Uppman. The Complexity of Three-Element Min-Sol and Conservative Min-Cost-Hom. In *Proc. ICALP'13*, pages 804–815. Springer, 2013.
30. Yuichi Yoshida and Yuan Zhou. Approximation schemes via Sherali-Adams hierarchy for dense constraint satisfaction problems and assignment problems. In *Proc. ITCS'14*, 423–438. ACM, 2014.