Algebraic Properties of Valued Constraint Satisfaction Problem

Marcin Kozik[1] and Joanna Ochremiak[2,⋆]

[1] Jagiellonian University
[2] University of Warsaw

Abstract. The paper presents an algebraic framework for optimization problems expressible as Valued Constraint Satisfaction Problems. Our results generalize the algebraic framework for the decision version (CSPs) provided by Bulatov et al. [SICOMP 2005].
We introduce the notions of weighted algebras and varieties, and use the Galois connection due to Cohen et al. [SICOMP 2013] to link VCSP languages to weighted algebras. We show that the difficulty of VCSP depends only on the weighted variety generated by the associated weighted algebra.
Paralleling the results for CSPs we exhibit a reduction to cores and rigid cores which allows us to focus on idempotent weighted varieties. Further, we propose an analogue of the Algebraic CSP Dichotomy Conjecture; prove the hardness direction and verify that it agrees with known results for VCSPs on two-element sets [Cohen et al. 2006], finite-valued VCSPs [Thapper and Živný 2013], and conservative VCSPs [Kolmogorov and Živný 2013].

1 Introduction

An instance of the Constraint Satisfaction Problem (CSP) consists of variables (to be evaluated in a domain) and constraints restricting the evaluations. The aim is to find an evaluation satisfying all the constraints or satisfying the maximal possible number of constraints or approximating the maximal possible number of satisfied constraints etc. depending on the version of the problem. Further one can divide constraint satisfaction problems with respect to the size of the domain, the allowed constraints or the shape of the instances.

A particularly interesting version of the CSP was proposed in a seminal paper of Feder and Vardi [11]. In this version the CSP is defined by a *language* which consists of relations over a finite set. An instance of such a CSP is allowed if all the constraint relations are from this set. The goal is to determine whether an instance has a solution satisfying all the constraints.

⋆ The first author was supported by the Polish National Science Centre (NCN) grant 2011/01/B/ST6/01006; the second author was supported by the Polish National Science Centre (NCN) grant 2012/07/B/ST6/01497.

Each language clearly defines a problem in NP; the whole family of problems is interesting for another reason: it is robust enough to include some well studied computational problems, e.g. 2-colorability, 3-SAT, solving systems of linear equations over $\mathbb{Z}_p$, and still is conjectured [11] not to contain problems of intermediate complexity. This conjecture (which holds for languages on two-element sets by the result of Schaefer [18]) is known as the Constraint Satisfaction Dichotomy Conjecture of Feder and Vardi. Confirming this conjecture would establish CSPs as one of the largest natural subclasses of NP without problems of intermediate complexity.

The conjecture always attracted a lot of attention, but the first results, even very interesting ones, were usually very specialized (e.g. [12]). A major breakthrough appeared with a series of papers establishing *the algebraic approach to CSP* [3, 7, 14]. This deep connection with an independently developed branch of mathematics introduced a new viewpoint and provided tools necessary to tackle wide classes of CSP languages at once. At the heart of this approach lies a Galois connection between languages and clones of operations called *polymorphisms* (which completely determine the complexity of the language).

Results obtained using these new methods include a full complexity classifications for CSPs on three-element sets [5] and those containing all unary relations [4, 6]. Moreover, the algebraic approach to CSP allowed to propose a boundary between the tractable and NP-complete problems: this conjecture is known as the Algebraic Dichotomy Conjecture. Unfortunately, despite many efforts (e.g. [5]), both conjectures remain open.

The Valued Constraint Satisfaction Problem (VCSP) further extends the approach proposed by Feder and Vardi. The role of constraints is played by *cost functions* describing the price of choosing particular values for variables as a part of the solution. This generalization allows to construct languages modeling standard optimization problems, for example MAX-CUT. Moreover, by allowing ∞ as a cost of a tuple, a VCSP language can additionally model every problem that CSP can model, as well as hybrid problems like MIN-VERTEX-COVER. This makes the extended framework even more general (compare the survey [15]).

A number of classes of VCSPs have been thoroughly investigated. The underlying structure suggested capturing the properties of languages of cost functions using an amalgamation of algebraic and numerical techniques [10, 20]. The first approach which provides a Galois correspondence (mirroring the Galois correspondence for CSPs) was proposed by Cohen et al. [9]. A weighted clone defined in this paper fully captures the complexity of a VCSP language.

The present paper builds on that correspondence imitating the line of research for CSPs [7]. It is organized in the following way: Section 2 contains preliminaries and basic definitions. In Section 3 we present a reduction to cores and rigid cores. Section 4 introduces a concept of a weighted algebra and a weighted variety, and shows that those notions are well behaved in the context of the Galois connection for VCSP. Reductions developed in Section 3 together with definitions from Section 4 allow us to focus on idempotent varieties. Section 5 states a conjecture postulating (for idempotent varieties) the division between the tractable and NP-hard cases of VCSP. The conjecture is clearly a strengthening of the Algebraic Dichotomy Conjecture [7]. Section 5 contains additionally the proof of the hardness direction of the conjecture as well as the reasoning showing that the conjecture agrees with complexity classifications for VCSPs on two-element sets [10], with finite-valued cost functions [20], and with conservative cost functions [16].

2 Preliminaries

2.1 The Valued Constraint Satisfaction Problem

Throughout the paper, let $\overline{\mathbb{Q}} = \mathbb{Q} \cup \{\infty\}$. We assume that $x + \infty = \infty$ and $y \cdot \infty = \infty$ for $y \geq 0$. An r-ary *relation* on a set D is a subset of D^r, a *cost function* on D of arity r is a function from D^r to $\overline{\mathbb{Q}}$. We denote by Φ_D the set of all cost functions on D. A cost function which takes only finite values is called *finite-valued*. A $\{0, \infty\}$-valued cost function is called *crisp* and can be viewed as a relation.

Definition 1. *An* instance of the valued constraint satisfaction problem (VCSP) *is a triple* $\mathcal{I} = (V, D, \mathcal{C})$ *with* V *a finite set of* variables*,* D *a finite* domain *and* $\mathcal{C}$ *a finite multi-set of* constraints. *Each constraint is a pair* $C = (\sigma, \varrho)$ *with* σ *a tuple of variables of length* r *and* ϱ *a cost function on* D *of arity* r*.*

An assignment *for* $\mathcal{I}$ *is a mapping* $s\colon V \to D$*. The* cost *of an assignment* s *is given by* $Cost_{\mathcal{I}}(s) = \sum_{(\sigma,\varrho)\in\mathcal{C}} \varrho(s(\sigma))$ *(where* s *is applied component-wise). To solve* $\mathcal{I}$ *is to find an assignment with a minimal cost, called an* optimal *assignment.*

Any set $\Gamma \subseteq \Phi_D$ is called a *valued constraint language* over D, or simply a *language*. If all cost functions from Γ are $\{0, \infty\}$-valued or finite-valued, we call it a *crisp* or *finite-valued* language, respectively.

By VCSP(Γ) we denote the class of all VSCP instances in which all cost functions in all constraints belong to Γ. VCSP(Γ_{crisp}), where Γ_{crisp} is the language consisting of all crisp cost functions on some fixed set D, is equivalent to the classical CSP. For an instance $\mathcal{I} \in$ VCSP(Γ) we denote by $\mathrm{Opt}_\Gamma(\mathcal{I})$ the cost of an optimal assignment. We say that a language Γ is *tractable* if, for every finite subset $\Gamma' \subseteq \Gamma$, there exists an algorithm solving any instance $\mathcal{I} \in$ VCSP(Γ') in polynomial time, and we say that Γ is *NP-hard* if VCSP(Γ') is NP-hard for some finite $\Gamma' \subseteq \Gamma$.

Weighted Relational Clones. We follow the exposition of [9] and define a closure operator on valued constraint languages that preserves tractability.

Definition 2. *A cost function* ϱ *is* expressible *over a valued constraint language* $\Gamma \subseteq \Phi_D$ *if there exists an instance* $\mathcal{I}_\varrho \in$ VCSP(Γ) *and a list* $(v_1, \ldots, v_r)$ *of variables of* $\mathcal{I}_\varrho$*, such that*

$$\varrho(x_1, \ldots, x_r) = \min_{\{s\colon V\to D \ \mid \ s(v_i)=x_i\}} Cost_{\mathcal{I}_\varrho}(s).$$

Note that the list of variables $(v_1, \ldots, v_r)$ in the definition above might contain repeated entries. Hence, it is possible that there are no assignments s such that $s(v_i) = x_i$ for all i. We define the minimum over the empty set to be ∞.

Definition 3. *A set $\Gamma \subseteq \Phi_D$ is a* weighted relational clone *if it is closed under expressibility, scaling by non-negative rational constants, and addition of rational constants. We define* $\mathrm{wRelClo}(\Gamma)$ *to be the smallest weighted relational clone containing Γ.*

If $\varrho(x_1, \ldots, x_r) = \varrho_1(y_1, \ldots, y_s) + \varrho_2(z_1, \ldots, z_t)$ for some fixed choice of arguments $y_1, \ldots, y_s, z_1, \ldots, z_t$ from amongst $x_1, \ldots, x_r$ then the cost function ϱ is said to be obtained by *addition* from the cost functions ϱ_1 and ϱ_2. It is easy to see that a weighted relational clone is closed under addition, and minimisation over arbitrary arguments.

The following result shows that we can restrict our attention to languages which are weighted relational clones.

Theorem 4 (Cohen et al. [9]). *A valued constraint language Γ is tractable if and only if* $\mathrm{wRelClo}(\Gamma)$ *is tractable, and it is NP-hard if and only if* $\mathrm{wRelClo}(\Gamma)$ *is NP-hard.*

Weighted polymorphisms. A k-ary *operation* on D is a function $f\colon D^k \to D$. We denote by $\mathcal{O}_D$ the set of all finitary operations on D and by $\mathcal{O}_D^{(k)}$ the set of all k-ary operations on D. The k-ary *projections*, defined for all $i \in \{1, \ldots, k\}$, are the operations $\pi_i^{(k)}$ such that $\pi_i^{(k)}(x_1, \ldots, x_k) = x_i$. Let $f \in \mathcal{O}_D^{(k)}$ and $g_1, \ldots, g_k \in \mathcal{O}_D^{(l)}$. The l-ary operation $f[g_1, \ldots, g_k]$ defined by $f[g_1, \ldots, g_k](x_1, \ldots, x_l) = f(g_1(x_1, \ldots, x_l), \ldots, g_k(x_1, \ldots, x_l))$ is called the *superposition* of f and $g_1, \ldots, g_k$.

A set $C \subseteq \mathcal{O}_D$ is a *clone of operations* (or simply a *clone*) if it contains all projections on D and is closed under superposition. The set of k-ary operations in a clone C is denoted $C^{(k)}$. The smallest possible clone of operations over a fixed set D is the set of all projections on D, which we denote Π_D.

Following [9] we define a k-ary *weighting* of a clone C to be a function $\omega\colon C^{(k)} \to \mathbb{Q}$ such that $\sum_{f \in C^{(k)}} \omega(f) = 0$, and if $\omega(f) < 0$ then f is a projection. The set of operations to which a weighting ω assigns positive weights is called the *support* of ω and denoted $\mathrm{supp}(\omega)$.

A new weighting of the same clone can be obtained by scaling a weighting by a non-negative rational, adding two weightings of the same arity and by the following operation called *superposition*.

Definition 5. *Let ω be a k-ary weighting of a clone C and let $g_1, \ldots, g_k \in C^{(l)}$. A* superposition *of ω and $g_1, \ldots, g_k$ is a function $\omega[g_1, \ldots, g_k]\colon C^{(l)} \to \mathbb{Q}$ defined by*

$$\omega[g_1, \ldots, g_k](f') = \sum_{\{f \in C^{(k)} \;\mid\; f[g_1, \ldots, g_k] = f'\}} \omega(f).$$

The sum of weights that any superposition $\omega[g_1, \ldots, g_k]$ assigns to the operations in $C^{(l)}$ is equal to zero, however, it may happen that a superposition assigns a negative value to an operation that is not a projection. A superposition is said to be *proper* if the result is a valid weighting.

A non-empty set of weightings over a fixed clone C is called a *weighted clone* if it is closed under non-negative scaling, addition of weightings of equal arity and proper superposition with operations from C. For any clone of operations C, the set of all weightings over C and the set of all zero-valued weightings of C are weighted clones.

We say that an r-ary relation R on D is *compatible* with an operation $f\colon D^k \to D$ if, for any list of r-tuples $\mathbf{x_1}, \ldots, \mathbf{x_k} \in R$ we have $f(\mathbf{x_1}, \ldots, \mathbf{x_k}) \in R$ (where f is applied coordinate-wise). Let $\varrho\colon D^r \to \overline{\mathbb{Q}}$ be a cost function. We define $\mathrm{Feas}(\varrho) = \{\mathbf{x} \in D^r \mid \varrho(\mathbf{x}) \text{ is finite}\}$ to be the *feasibility relation* of ϱ. We call an operation $f\colon D^k \to D$ a *polymorphism* of ϱ if the relation $\mathrm{Feas}(\varrho)$ is compatible with it. For a valued constraint language Γ we denote by $\mathrm{Pol}(\Gamma)$ the set of operations which are polymorphisms of all cost functions $\varrho \in \Gamma$. It is easy to verify that $\mathrm{Pol}(\Gamma)$ is a clone. The set of m-ary operations in $\mathrm{Pol}(\Gamma)$ is denoted $\mathrm{Pol}_m(\Gamma)$.

For crisp cost functions (relations) this notion of polymorphism corresponds precisely to the standard notion of polymorphism which has played a crucial role in the complexity analysis for the CSP [3, 14].

Definition 6. *Take ϱ to be a cost function of arity r on D, and let $C \subseteq \mathrm{Pol}(\{\varrho\})$ be a clone of operations. A weighting $\omega\colon C^{(k)} \to \mathbb{Q}$ is called a* weighted polymorphism *of ϱ if, for any list of r-tuples $\mathbf{x_1}, \ldots, \mathbf{x_k} \in \mathrm{Feas}(\varrho)$, we have*

$$\sum_{f \in C^{(k)}} \omega(f) \cdot \varrho(f(\mathbf{x_1}, \ldots, \mathbf{x_k})) \leq 0.$$

For a valued constraint language Γ we denote by $\mathrm{wPol}(\Gamma)$ the set of those weightings of the clone $\mathrm{Pol}(\Gamma)$ that are weighted polymorphisms of all cost functions $\varrho \in \Gamma$. The set of weightings $\mathrm{wPol}(\Gamma)$ is a weighted clone [9].

An operation f is *idempotent* if $f(x, ..., x) = x$. A weighted polymorphism is called *idempotent* if all operations in its support are idempotent. An operation $f \in \mathcal{O}_D^{(k)}$ is *cyclic* if for every $x_1, \ldots, x_k \in D$ we have that $f(x_1, x_2, \ldots, x_k) = f(x_2, \ldots, x_k, x_1)$. A weighted polymorphism is called *cyclic* if its support is non-empty and contains cyclic operations only.

A cost function ϱ is said to be *improved* by a weighting ω if ω is a weighted polymorphism of ϱ. For any set W of weightings over a fixed clone $C \subseteq \mathcal{O}_D$ we denote by $\mathrm{Imp}(W)$ the set of cost functions on D which are improved by all weightings $\omega \in W$.

By the result of Cohen et al. [9] for any finite valued constraint language Γ, we have $\mathrm{Imp}(\mathrm{wPol}(\Gamma)) = \mathrm{wRelClo}(\Gamma)$. This fact, together with Theorem 4, implies that tractable valued constraint languages can be characterized by their weighted polymorphisms.

2.2 Algebras and varieties

An *algebraic signature* is a set of function symbols together with (finite) arities. An *algebra* $\mathbf{A}$ over a fixed signature Σ, has a *universe* A, and a set of *basic operations* that correspond to the symbols in the signature, i.e., if the signature contains a k-ary symbol f then the algebra has a basic operation $f^{\mathbf{A}}$, which is a function $f^{\mathbf{A}}\colon A^k \to A$.

A subset B of the universe of an algebra $\mathbf{A}$ is a *subuniverse* of $\mathbf{A}$ if it is closed under all operations of $\mathbf{A}$. An algebra $\mathbf{B}$ is a *subalgebra* of $\mathbf{A}$ if B is a subuniverse of $\mathbf{A}$ and the operations of $\mathbf{B}$ are restrictions of all the operations of $\mathbf{A}$ to B. Let $(\mathbf{A}_i)_{i\in I}$ be a family of algebras (over the same signature). Their *product* $\Pi_{i\in I}\mathbf{A}_i$ is an algebra with the universe equal to the cartesian product of the A_i's and operations computed coordinate-wise. For two algebras $\mathbf{A}$ and $\mathbf{B}$ (over the same signature), a *homomorphism* from $\mathbf{A}$ to $\mathbf{B}$ is a function $h\colon A \to B$ that preserves all operations. It is easy to see, that an image of an algebra under a homomorphism $h\colon A \to B$ is a subalgebra of $\mathbf{B}$.

Let $\mathcal{K}$ be a class of algebras over a fixed signature Σ. We denote by $\mathrm{S}(\mathcal{K})$ the class of all subalgebras of algebras in $\mathcal{K}$, by $\mathrm{P}(\mathcal{K})$ the class of all products of algebras in $\mathcal{K}$, by $\mathrm{P}_{fin}(\mathcal{K})$ the class of all finite products, and by $\mathrm{H}(\mathcal{K})$ the class of all homomorphic images of algebras in $\mathcal{K}$. If $\mathcal{K} = \{\mathbf{A}\}$ we write $\mathrm{S}(\mathbf{A})$, $\mathrm{P}(\mathbf{A})$, and $\mathrm{H}(\mathbf{A})$ instead of $\mathrm{S}(\{\mathbf{A}\})$, $\mathrm{P}(\{\mathbf{A}\})$, and $\mathrm{H}(\{\mathbf{A}\})$, respectively.

A *variety* $\mathcal{V}(\mathcal{K})$ is the smallest class of algebras closed under all three operations. For an algebra $\mathbf{A}$ the variety $\mathcal{V}(\{\mathbf{A}\})$ (denoted $\mathcal{V}(\mathbf{A})$) is the variety *generated* by $\mathbf{A}$, and $\mathcal{V}_{fin}(\mathbf{A})$ is the class of finite algebras in $\mathcal{V}(\mathbf{A})$. Due to a result of Tarski [19] we know that for any finite algebra $\mathbf{A}$, we have

$$\mathcal{V}(\mathbf{A}) = \mathrm{HSP}(\mathbf{A}) \quad \text{and} \quad \mathcal{V}_{fin}(\mathbf{A}) = \mathrm{HSP}_{fin}(\mathbf{A}).$$

We say that an equivalence relation $\sim$ on A is a *congruence* of $\mathbf{A}$ if it is a subalgebra of $\mathbf{A}^2$. Every congruence $\sim$ of $\mathbf{A}$ determines a *quotient* algebra $\mathbf{A}/{\sim}$.

A *term* t in a signature Σ is a formal expression built from variables and symbols in Σ that syntactically describes the composition of basic operations. For an algebra $\mathbf{A}$ over Σ a *term operation* $t^{\mathbf{A}}$ is an operation obtained by composing the basic operations of $\mathbf{A}$ according to t. Let s and t be a pair of terms in a signature Σ. We say that $\mathbf{A}$ satisfies the *identity* $s \approx t$ if the term operations $s^{\mathbf{A}}$ and $t^{\mathbf{A}}$ are equal. We say that a class of algebras $\mathcal{V}$ over Σ satisfies the identity $s \approx t$ if every algebra in $\mathcal{V}$ does.

It follows from Birkhoff's theorem [2] that the variety $\mathcal{V}(\mathbf{A})$ is the class of algebras that satisfy all the identities satisfied by $\mathbf{A}$. An algebra $\mathbf{A}$ is *finitely generated* if there exists a finite subset F of its domain such that the only subalgebra of $\mathbf{A}$ containing F is $\mathbf{A}$. If $\mathbf{A}$ is finite then $\mathcal{V}(\mathbf{A})$ is *locally finite*, i.e., every finitely generated algebra in $\mathcal{V}(\mathbf{A})$ is finite.

3 Core Valued Constraint Languages

For each valued constraint language Γ there is an associated algebra. It has universe D and the set of operations $\mathrm{Pol}(\Gamma)$. If all operations of any given algebra satisfy the identity $f(x, \dots, x) \approx x$ (i.e. are *idempotent*) then we call the algebra *idempotent*. In this section we prove that every valued constraint language which is finite has a computationally equivalent valued constraint language whose associated algebra is idempotent.

Positive Clone. Those polymorphisms of a given language Γ which are assigned a positive weight by some weighted polymorphisms $\omega \in \mathrm{wPol}(\Gamma)$ are of special interest in the rest of the paper. We begin this section by proving that they form a clone.

Let $\mathcal{C}$ be a weighted clone over a set D. The following proposition shows that the set $\bigcup_{\omega \in \mathcal{C}} \mathrm{supp}(\omega)$, together with the set of projections Π_D, is a clone. We call it the *positive clone* of $\mathcal{C}$ and denote by C^+ (if $\mathcal{C}$ is $\mathrm{wPol}(\Gamma)$ then C^+ is denoted by $\mathrm{Pol}^+(\Gamma)$).

Proposition 7. *If $\mathcal{C}$ is a weighted clone then C^+ is a clone.*

Cores. Let Γ be a valued constraint language with a domain D. For $S \subseteq D$ we denote by $\Gamma[S]$ the valued constraint language defined on a domain S and containing the restriction of every cost function $\varrho \in \Gamma$ to S.

By generalizing the arguments for finite-valued languages given in [13, 20], we show that Γ has a computationally equivalent valued constraint language Γ' such that $\mathrm{Pol}_1^+(\Gamma')$ contains only bijective operations. Such a language is called a *core*. Moreover, Γ' can be chosen to be equal to $\Gamma[S]$ for some $S \subseteq D$.

Proposition 8. *For every valued constraint language Γ there exists a core language Γ', such that the valued constraint language Γ is tractable if and only if Γ' is tractable, and it is NP-hard if and only if Γ' is NP-hard.*

For core languages we characterize the set of unary weighted polymorphisms as consisting of all weightings that assign positive weights only to bijective operations preserving all cost functions.

The proposition below witnesses the importance of the positive clone and is used to prove further results in the subsequent sections. Let Γ be a valued constraint language over a domain D which is finite and a core. For each arity m we fix an enumeration of all the elements of D^m. This allows us to treat every m-ary operation $f \in \mathcal{O}_D^{(m)}$ as a $|D^m|$-tuple. We define a $|D^m|$-ary cost function in $\mathrm{wRelClo}(\Gamma)$ that precisely distinguishes the m-ary operations in the positive clone from all the other m-ary polymorphisms.

Proposition 9. *Let Γ be a valued constraint language over a domain D which is finite and a core. For every m there exists a cost function $\varrho\colon \mathcal{O}_D^{(m)} \to \overline{Q}$ in $\mathrm{wRelClo}(\Gamma)$, and a rational number P, such that for every $f \in \mathcal{O}_D^{(m)}$ the following conditions are satisfied:*

1. *$\varrho(f) \geq P$,*
2. *$\varrho(f) < \infty$ if and only if $f \in \mathrm{Pol}(\Gamma)$,*
3. *$\varrho(f) = P$ if and only if $f \in \mathrm{Pol}^+(\Gamma)$.*

Rigid cores. We further reduce the class of languages that we need to consider. Let Γ be a valued constraint language over an n-element domain $D = \{d_1, \ldots, d_n\}$ which is finite and a core. For each $i \in \{1, \ldots, n\}$, let

$$N_i(x) = \begin{cases} 0 & \text{if } x = d_i, \\ \infty & \text{otherwise,} \end{cases}$$

and let Γ_c denote the valued constraint language obtained from Γ by adding all cost functions N_i. Observe that $\mathrm{Pol}(\Gamma_c) = \mathrm{IdPol}(\Gamma)$, where by $\mathrm{IdPol}(\Gamma)$ we denote the set of idempotent polymorphisms of the language Γ. Hence, the only unary polymorphism of Γ_c is the identity, which also means that there is only one unary weighted polymorphism of Γ_c – the zero-valued weighted polymorphism.

A valued constraint language Γ is a *rigid core* if there is exactly one unary polymorphism of Γ, which is the identity. This notion corresponds to the classical notion of a rigid core considered in CSP [7]. The following proposition, together with Proposition 8, implies that for each finite language Γ, there is a computationally equivalent language that is a rigid core.

Proposition 10. *Let Γ be a valued constraint language which is finite and a core. The valued constraint language Γ_c is a rigid core. Moreover, Γ is tractable if and only if Γ_c is tractable, and Γ is NP-hard if and only if Γ_c is NP-hard.*

If Γ is a core language then the positive clone of Γ_c contains precisely the idempotent operations from the positive clone of Γ.

4 Weighted varieties

One of the fundamental results of the algebraic approach to CSP [3, 7, 17] says that the complexity of a crisp language Γ depends only on the variety generated by the algebra $(D, \mathrm{Pol}(\Gamma))$. We generalize this fact to VCSP.

A k-ary *weighting* ω of an algebra $\mathbf{A}$ is a function that assigns rational weights to all k-ary term operations of $\mathbf{A}$ in such a way, that the sum of all weights is 0, and if $\omega(f) < 0$ then f is a projection. A (*proper*) *superposition* $\omega[g_1, \ldots, g_k]$ of a weighting ω with a list of l-ary term operations $g_1, \ldots, g_k$ from $\mathbf{A}$ is defined the same way as for clones (see Definition 5). An algebra $\mathbf{A}$ together with a set of weightings closed under non-negative scaling, addition of weightings of equal arity and proper superposition with term operations from $\mathbf{A}$ is called a *weighted algebra*.

For a variety $\mathcal{V}$ over a signature Σ and a term t we denote by $[t]_{\mathcal{V}}$ the equivalence class of t under the relation $\approx_{\mathcal{V}}$ such that $t \approx_{\mathcal{V}} s$ if and only if the variety $\mathcal{V}$ satisfies the identity $t \approx s$ (we skip the subscript, writing $[t]$ instead of $[t]_{\mathcal{V}}$, whenever the variety is clear from the context). Observe that if the variety is locally finite then there are finitely many equivalence classes of terms of a fixed arity [8].

Definition 11. *Let $\mathcal{V}$ be a locally finite variety over a signature Σ. A k-ary* weighting *ω of $\mathcal{V}$ is a function that assigns rational weights to all equivalence classes of k-ary terms over Σ in such a way, that the sum of all weights is 0, and if $\omega([t]) < 0$ then $\mathcal{V}$ satisfies the identity $t(x_1, \ldots, x_k) \approx x_i$ for some $i \in \{1, \ldots, k\}$. The variety $\mathcal{V}$ together with a nonempty set of weightings is called a* weighted variety.

Take any finite algebra $\mathbf{B} \in \mathcal{V}$. A k-ary weighting ω of $\mathcal{V}$ *induces* a weighting $\omega^{\mathbf{B}}$ of $\mathbf{B}$ in a natural way:

$$\omega^{\mathbf{B}}(f) = \sum_{\{[t] \mid t^{\mathbf{B}} = f\}} \omega([t]).$$

If $\omega([t]) < 0$ then the term operation $t^{\mathbf{B}}$ is a projection, and hence the weighting $\omega^{\mathbf{B}}$ is proper. For a weighted variety $\mathcal{V}$, by $\mathbf{B} \in \mathcal{V}$ we mean the algebra $\mathbf{B}$ together with the set of weightings induced by $\mathcal{V}$.

For every weighting ω of a finite weighted algebra $\mathbf{A}$ there is a corresponding weighting ω of the variety $\mathcal{V}(\mathbf{A})$ defined by $\omega([t]) = \omega(t^{\mathbf{A}})$. It follows from Birkhoff's theorem that it is well defined. A weighted variety $\mathcal{V}(\mathbf{A})$ *generated* by a weighted algebra $\mathbf{A}$ is the variety $\mathcal{V}(\mathbf{A})$ together with the set of weightings corresponding to the weightings of $\mathbf{A}$.

We prove that every finite algebra $\mathbf{B} \in \mathcal{V}(\mathbf{A})$ together with the set of weightings induced by $\mathcal{V}(\mathbf{A})$ is a weighted algebra. The only non-trivial part is to show that $\mathbf{B}$ is closed under proper superpositions.

Proposition 12. *For a finite weighted algebra $\boldsymbol{A}$ over a fixed signature Σ and a finite algebra $\boldsymbol{B} \in \mathcal{V}(\boldsymbol{A})$ let $\omega^{\boldsymbol{B}}$ be a k-ary weighting of $\boldsymbol{B}$ induced by the weighted variety $\mathcal{V}(\boldsymbol{A})$. If for some list $f_1^{\boldsymbol{B}}, \ldots, f_k^{\boldsymbol{B}}$ of l-ary term operations from $\boldsymbol{B}$ the composition $\omega^{\boldsymbol{B}}[f_1^{\boldsymbol{B}}, \ldots, f_k^{\boldsymbol{B}}]$ is proper then it is induced by some valid weighting of $\mathcal{V}(\boldsymbol{A})$.*

For a finite weighted algebra $\mathbf{A}$ let $\mathrm{Imp}(\mathbf{A})$ denote the set of those cost functions on A that are improved by all weightings of $\mathbf{A}$. We prove that for each finite weighted algebra $\mathbf{B} \in \mathcal{V}(\mathbf{A})$ the valued constraint language $\mathrm{Imp}(\mathbf{B})$ is not harder then $\mathrm{Imp}(\mathbf{A})$ i.e.:

Lemma 13. *Let $\boldsymbol{A}$ be a finite weighted algebra and let*

$$\boldsymbol{B} \in P_{fin}(\boldsymbol{A}) \text{ or } \boldsymbol{B} \in S(\boldsymbol{A}) \text{ or } \boldsymbol{B} \in H(\boldsymbol{A}) \text{ or finally } \boldsymbol{B} \in \mathcal{V}(\boldsymbol{A})$$

then a VCSP defined by any finite subset of $\mathrm{Imp}(\boldsymbol{B})$ *reduces in polynomial-time to a VCSP for some finite subset of* $\mathrm{Imp}(\boldsymbol{A})$.

Therefore the complexity of Γ depends only on the weighted variety generated by the weighted algebra $(D, \mathrm{wPol}(\Gamma))$.

5 Dichotomy conjecture

An operation t of arity k is called a *Taylor operation* of an algebra (or a variety), if t is idempotent and for every $j \leq k$ it satisfies an identity of the form

$$t(\square_1, \square_2, \dots, \square_k) \approx t(\triangle_1, \triangle_2, \dots, \triangle_k),$$

where all $\square_i$s and $\triangle_i$s are substituted with either x or y, but $\square_j$ is x whenever $\triangle_j$ is y. In this section we prove the following theorem:

Theorem 14. *Let Γ be a finite core valued constraint language. If* $\mathrm{Pol}^+(\Gamma)$ *does not have a Taylor operation, then Γ is NP-hard.*

We conjecture[3] that these are the only cases of finite core languages which give rise to NP-hard VCSPs.

Conjecture. *Let Γ be a finite core valued constraint language. If* $\mathrm{Pol}^+(\Gamma)$ *does not have a Taylor operation, then Γ is NP-hard. Otherwise it is tractable.*

For crisp languages $\mathrm{Pol}^+(\Gamma) = \mathrm{Pol}(\Gamma)$. Therefore Theorem 14 generalizes the well-known result of Bulatov, Jeavons and Krokhin [3, 7] concerning crisp core languages. Similarly the above conjecture is a generalization of The Algebraic Dichotomy Conjecture for CSP. Later on we show that it is supported by all known partial results on the complexity of VCSPs.

To prove Theorem 14 we use Proposition 9 and argue that any relation compatible with $\mathrm{Pol}^+(\Gamma)$ can be found as a set of tuples with minimal costs for some cost function improved by $\mathrm{wPol}(\Gamma)$. It is easy to notice that if $\mathrm{Pol}^+(\Gamma)$ does not have a Taylor operation, then such a relation with NP-complete CSP can be constructed.

As the Taylor operation is difficult to work with, in the reminder of the section we use a characterization of Taylor algebras as the algebras possessing a cyclic term. If Γ is a finite core constraint language then $(D, \mathrm{IdPol}^+(\Gamma))$ is a finite idempotent algebra. It follows that $\mathrm{IdPol}^+(\Gamma)$, and hence also $\mathrm{Pol}^+(\Gamma)$, has a Taylor operation if and only if it has an idempotent cyclic operation [1].

5.1 Two-element domain

A complete complexity classification for valued constraint languages over a two-element domain was established in [10]. All tractable languages have been defined via multimorphisms, which are a more restricted form of weighted polymorphisms. A k-ary *multimorphism* of a language Γ, specified as a k-tuple $\langle f_1, \dots, f_k \rangle$ of k-ary operations on D, is a k-ary weighted polymorphism ω of Γ such that for each $i \in \{1, \dots, k\}$, we have that $\omega(\pi_i) = -\frac{1}{k}$, and $\omega(f_i) = \frac{l}{k}$, where l is the number of times the operation f_i appears in the tuple.

[3] The conjecture was suggested in a conversation by Libor Barto, however it might have appeared independently earlier.

An operation $f \in \mathcal{O}_D^{(3)}$ is called a *majority* operation if for every $x, y \in D$ we have that $f(x,x,y) = f(x,y,x) = f(y,x,x) = x$. Similarly, an operation $f \in \mathcal{O}_D^{(3)}$ is called a *minority* operation if for every $x, y \in D$ it satisfies $f(x,x,y) = f(x,y,x) = f(y,x,x) = y$. We show the following proposition:

Proposition 15. *Let Γ be a finite core valued constraint language on $D = \{0,1\}$. Then $\mathrm{Pol}^+(\Gamma)$ has an idempotent cyclic operation if and only if Γ admits at least one of the following six multimorphisms:* $\langle \min, \min \rangle$, $\langle \max, \max \rangle$, $\langle \min, \max \rangle$, $\langle \mathrm{Mjrty}, \mathrm{Mjrty}, \mathrm{Mjrty} \rangle$, $\langle \mathrm{Mnrty}, \mathrm{Mnrty}, \mathrm{Mnrty} \rangle$, $\langle \mathrm{Mjrty}, \mathrm{Mjrty}, \mathrm{Mnrty} \rangle$.

The proposition fully agrees with the classification of VCSP languages on two-element domain in [10].

5.2 Finite-valued languages

Theorem 16 (Thapper and Živný [20]). *Let Γ be a finite-valued constraint language which is a core. If Γ admits an idempotent cyclic weighted polymorphism of some arity $m > 1$, then Γ is tractable. Otherwise it is NP-hard.*

To show that our conjecture agrees with the above complexity classification we prove the following result (which holds for general-valued languages):

Proposition 17. *Let Γ be a core valued constraint language. Then Γ admits an idempotent cyclic weighted polymorphism of some arity $m > 1$ if and only if $\mathrm{Pol}^+(\Gamma)$ contains an idempotent cyclic operation of the same arity.*

5.3 Conservative languages

A valued constraint language Γ over a domain D is called *conservative* if it contains all $\{0,1\}$-valued unary cost functions on D. An operation $f \in \mathcal{O}_D^{(k)}$ is *conservative* if for every $x_1, \ldots, x_k \in D$ we have that $f(x_1, \ldots, x_k) \in \{x_1, \ldots, x_k\}$, and a weighted polymorphism is *conservative* if its support contains conservative operations only.

A *Symmetric Tournament Pair* (*STP*) is a conservative binary multimorphism $\langle \sqcap, \sqcup \rangle$, where both operations are commutative, i.e., $\sqcap(x,y) = \sqcap(y,x)$ and $\sqcap(x,y) = \sqcap(y,x)$ for all $x, y \in D$, and moreover $\sqcap(x,y) \neq \sqcup(x,y)$ for all $x \neq y$. A *MJN* is a ternary conservative multimorphism $\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3 \rangle$, such that $\mathrm{Mj}_1, \mathrm{Mj}_2$ are majority operations, and Mn_3 is a minority operation.

Theorem 18 (Kolmogorov and Živný [16]). *Let Γ be a conservative constraint language over a domain D. If Γ admits a conservative binary multimorphism $\langle \sqcap, \sqcup \rangle$ and a conservative ternary multimorphism $\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3 \rangle$, and there is a family M of two-element subsets of D, such that:*

- *for every $\{x,y\} \in M$, $\langle \sqcap, \sqcup \rangle$ restricted to $\{x,y\}$ is an STP,*
- *for every $\{x,y\} \notin M$, $\langle \mathrm{Mj}_1, \mathrm{Mj}_2, \mathrm{Mn}_3 \rangle$ restricted to $\{x,y\}$ is an MJN,*

then Γ is tractable. Otherwise it is NP-hard.

In this case, as well as in the others, it can be shown that the existence of an idempotent cyclic polymorphism in $\mathrm{Pol}^+(\Gamma)$ is equivalent (for conservative Γ) to the tractability conditions from the theorem above.

Acknowledgments

We are grateful to Libor Barto and Jakub Bulin for inspiring discussions on VCSP.

References

1. Libor Barto and Marcin Kozik. Absorbing subalgebras, cyclic terms, and the constraint satisfaction problem. *Logical Methods in Computer Science*, 8(1), 2012.
2. G. Birkhoff. On the structure of abstract algebras. In *Proceedings of the Cambridge Philosophical Society*, volume 31, pages 433–454, 1935.
3. Andrei Bulatov, Peter Jeavons, and Andrei Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM Journal on Computing*, 34:720–742, 2005.
4. Andrei A. Bulatov. Tractable conservative constraint satisfaction problems. In *Proc. of the 18th Symposium on Logic in Computer Science*, page 321, 2003.
5. Andrei A. Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *J. ACM*, 53(1):66–120, 2006.
6. Andrei A. Bulatov. Complexity of conservative constraint satisfaction problems. *ACM Trans. Comput. Logic*, 12(4):24:1–24:66, 2011.
7. Andrei A. Bulatov, Andrei A. Krokhin, and Peter Jeavons. Constraint satisfaction problems and finite algebras. In *Proc. ICALP '00*, pages 272–282, 2000.
8. S. Burris and H.P. Sankappanavar. *A course in universal algebra*. Graduate texts in mathematics. Springer-Verlag, 1981.
9. David A. Cohen, Martin C. Cooper, Páidí Creed, Peter G. Jeavons, and Stanislav Živný. An algebraic theory of complexity for discrete optimization. *SIAM J. Comput.*, 42(5):1915–1939, 2013.
10. David A. Cohen, Martin C. Cooper, Peter G. Jeavons, and Andrei A. Krokhin. The complexity of soft constraint satisfaction. *Artif. Intell.*, 170(11):983–1016, 2006.
11. Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM J. Comput.*, 28(1):57–104, 1999.
12. Pavol Hell and Jaroslav Nešetřil. On the complexity of h-coloring. *Journal of Combinatorial Theory, Series B*, 48(1):92 – 110, 1990.
13. Anna Huber, Andrei Krokhin, and Robert Powell. Skew bisubmodularity and valued CSPs. In *Proc. SODA '13*, pages 1296–1305. SIAM, 2013.
14. Peter Jeavons, David Cohen, and Marc Gyssens. Closure properties of constraints. *J. ACM*, 44(4):527–548, 1997.
15. Peter Jeavons, Andrei Krokhin, and Stanislav Živný. The complexity of valued constraint satisfaction. *Bulletin of the EATCS*, 113:21–55, 2014.
16. Vladimir Kolmogorov and Stanislav Živný. The complexity of conservative valued CSPs. *J. ACM*, 60(2):10:1–10:38, 2013.
17. Benoit Larose and Pascal Tesson. Universal algebra and hardness results for constraint satisfaction problems. *Theor. Comput. Sci.*, 410:1629 – 1647, 2009.
18. Thomas J. Schaefer. The complexity of satisfiability problems. In *Proc. of the 10th ACM Symp. on Theory of Computing*, STOC '78, pages 216–226, 1978.
19. Alfred Tarski. A remark on functionally free algebras. *Annals of Mathematics*, 47(1):163–166, 1946.
20. Johan Thapper and Stanislav Živný. The complexity of finite-valued CSPs. In *Proc. of the 45th ACM Symp. on Theory of Computing*, STOC '13, pages 695–704, 2013.