# An Algebraic Theory of Complexity for Valued Constraints: Establishing a Galois Connection*

David A. Cohen[1], Páidí Creed[1], Peter G. Jeavons[2], and Stanislav Živný[3]

[1] Department of Computer Science, Royal Holloway, University of London, UK
[2] Department of Computer Science, University of Oxford, UK
[3] University College, University of Oxford, UK
{d.cohen,paidi}@rhul.ac.uk {peter.jeavons,standa.zivny}@cs.ox.ac.uk

**Abstract.** The complexity of any optimisation problem depends critically on the form of the objective function. Valued constraint satisfaction problems are discrete optimisation problems where the function to be minimised is given as a sum of cost functions defined on specified subsets of variables. These cost functions are chosen from some fixed set of available cost functions, known as a valued constraint language. We show in this paper that when the costs are non-negative rational numbers or infinite, then the complexity of a valued constraint problem is determined by certain algebraic properties of this valued constraint language, which we call *weighted polymorphisms*. We define a Galois connection between valued constraint languages and sets of weighted polymorphisms and show how the closed sets of this Galois connection can be characterised. These results provide a new approach in the search for tractable valued constraint languages.

## 1 Introduction

Classical constraint satisfaction is concerned with the feasibility of satisfying a collection of constraints. The extension of this framework to include optimisation is now also being investigated and a theory of so-called *soft constraints* is being developed. Several alternative mathematical frameworks for soft constraints have been proposed in the literature, including the very general frameworks of 'semi-ring based constraints' and 'valued constraints' [6]. For simplicity, we shall adopt the valued constraint framework here as it is sufficiently powerful to model a wide range of optimisation problems [17]. In this framework, every tuple of values allowed by a constraint has an associated *cost*, and the goal is to find an assignment with minimal total cost. The general constraint satisfaction problem (CSP) is NP-hard, and so is unlikely to have a polynomial-time algorithm. However, there has been much success in finding tractable fragments of the CSP by restricting the types of relation allowed in the constraints. A set of allowed relations has been called a *constraint language* [26]. For some constraint languages the associated constraint satisfaction problems with constraints chosen

* This research was supported by EPSRC grant EP/F01161X/1. Stanislav Živný is supported by a Junior Research Fellowship at University College, Oxford.

from that language are solvable in polynomial-time, whilst for other constraint languages this class of problems is NP-hard [27, 26, 23]; these are referred to as *tractable languages* and *NP-hard languages*, respectively. Dichotomy theorems, which classify each possible constraint language as either tractable or NP-hard, have been established for languages over 2-element domains [32], 3-element domains [10], for conservative languages [13, 4], and maximal languages [11, 9].

The general *valued* constraint satisfaction problem (VCSP) is also NP-hard, but again we can try to identify tractable fragments by restricting the types of allowed *cost functions* that can be used to define the valued constraints. A set of allowed cost functions has been called a *valued constraint language* [17]. Much less is known about the complexity of the optimisation problems associated with different valued constraint languages, although some results have been obtained for certain special cases. In particular, a complete characterisation of complexity has been obtained for valued constraint languages over a 2-element domain with real-valued or infinite costs [17]. This result generalises a number of earlier results for particular optimisation problems such as Max-Sat [20] and Min-Ones [21]. One class of tractable cost functions that has been extensively studied is the class of *submodular* cost functions [21, 17, 28, 22, 29, 34].

In the classical CSP framework it has been shown that the complexity of any constraint language over any finite domain is determined by certain algebraic properties known as *polymorphisms* [27, 26]. This result has reduced the problem of the identification of tractable constraint languages to that of the identification of suitable sets of polymorphisms. In other words, it has been shown to be enough to study just those constraint languages which are characterised by having a given set of polymorphisms. Using the algebraic approach, considerable progress has now been made towards a complete characterisation of the complexity of constraint languages over finite domains of arbitrary size [23, 12, 3, 1, 2, 5].

In the VCSP framework it has been shown that a more general algebraic property known as a *multimorphism* can be used to analyse the complexity of certain valued constraint languages [14, 17]. Multimorphisms have been used to show that there are precisely eight maximal tractable valued constraint languages over a 2-element domain with real-valued or infinite costs, and each of these is characterised by having a particular form of multimorphism [17]. Furthermore, it was shown that many known maximal tractable valued constraint languages over larger finite domains are precisely characterised by a single multimorphism and that key NP-hard examples have (essentially) no multimorphisms [17, 16].

Cohen et al. [15] later generalised the notion of a multimorphism slightly, to that of a *fractional polymorphism*. They showed that fractional polymorphisms, together with the polymorphisms of the underlying feasibility relations, characterise the complexity of any valued constraint language with non-negative rational or infinite costs over any finite domain [15].

*Contributions* In this paper, we extend the results of [15] by introducing a new algebraic construct which we call a *weighted polymorphism*. We are able to show, using the ideas of [15], that the weighted polymorphisms of a valued constraint language are sufficient on their own to determine the complexity of that language. In addition, we are now able to define a Galois connection between valued constraint languages and sets of weighted polymorphisms, and characterise the closed sets on both sides.

The Galois connection we establish here can be applied to the search for tractable valued constraint languages in a very similar way to the application of polymorphisms to the search for tractable constraint languages in the classical CSP. First, we need only consider valued constraint languages characterised by weighted polymorphisms. This greatly simplifies the search for a characterisation of all tractable valued constraint languages. Second, any tractable valued constraint language with finite rational or infinite costs must have a non-trivial weighted polymorphism. Hence the results of this paper provide a powerful new set of tools in the search for a polynomial-time/NP-hard dichotomy for finite-domain optimisation problems defined by valued constraints. In the conclusion section we will mention recent results obtained using the Galois connection established in this paper.

Despite the fact that the proof of the main result uses similar techniques to [15], namely linear programming and Farkas Lemma, the main contribution of this paper is significantly different: [15] has shown that fractional polymorphisms capture the complexity of valued constraint languages. Here, we prove the same for weighted polymorphisms, but also establish a 1-to-1 correspondence between valued constraint languages and particular sets of weighted polymorphisms, which we call weighted clones. This is crucial for using weighted polymorphisms in searching for new tractable valued constraint languages. Our results show that a linear program can be set up not only to answer the question of whether a given cost function is expressible over a valued constraint language, but also for the question of whether a given weighted operation belongs to a weighted clone. (We do not elaborate on this application in much detail, but it follows straightforwardly from the proofs of the main results.)

The structure of the paper is as follows. In Section 2 we describe the Valued Constraint Satisfaction Problem and define the notion of expressibility. In Sections 3 and 4 we introduce weighted relational clones (valued constraint languages closed under a certain notion of expressibility) and weighted clones respectively, and state the main result: weighted relational clones are in 1-to-1 correspondence with weighted clones. In Section 5 we give a proof of the main new theorem establishing the Galois connection. Finally, in Section 6, we mention some recent results based on the results of this paper.

## 2 Valued Constraint Satisfaction Problems

We shall denote by $\mathbb{Q}_+$ the set of all non-negative rational numbers.[4] We define $\overline{\mathbb{Q}}_+ = \mathbb{Q}_+ \cup \{\infty\}$ with the standard addition operation extended so that for all $a \in \mathbb{Q}_+$, $a + \infty = \infty$ and $a\infty = \infty$. Members of $\overline{\mathbb{Q}}_+$ are called *costs*.

[4] To avoid computational problems, we work with rational numbers rather than real numbers. We could work with the algebraic reals, but the rationals are sufficiently general to encode many standard optimisation problems; see, for example [17].

A function $\phi$ from $D^r$ to $\overline{\mathbb{Q}}_+$ will be called a *cost function* on $D$ of *arity* $r$.

**Definition 1.** *An instance of the* **valued constraint satisfaction problem**, *(***VCSP***), is a triple $\mathcal{P} = \langle V, D, C\rangle$ where: $V$ is a finite set of* **variables***; $D$ is a set of possible* **values***; $C$ is a multi-set of* **constraints***. Each element of $C$ is a pair $c = \langle \sigma, \phi\rangle$ where $\sigma$ is a tuple of variables called the* **scope** *of $c$, and $\phi$ is a $|\sigma|$-ary cost function on $D$ taking values in $\overline{\mathbb{Q}}_+$. An* **assignment** *for $\mathcal{P}$ is a mapping $s : V \to D$. The* **cost** *of an assignment $s$, denoted $Cost_{\mathcal{P}}(s)$, is given by the sum of the costs for the restrictions of $s$ onto each constraint scope, that is,*

$$Cost_{\mathcal{P}}(s) \stackrel{\text{def}}{=} \sum_{\langle\langle v_1,v_2,\ldots,v_m\rangle,\phi\rangle\in C} \phi(s(v_1), s(v_2), \ldots, s(v_m)).$$

*A* **solution** *to $\mathcal{P}$ is an assignment with minimal cost, and the question is to find a solution.*

A **valued constraint language** is any set $\Gamma$ of cost functions from some fixed set $D$. We define VCSP($\Gamma$) to be the set of all VCSP instances in which all cost functions belong to $\Gamma$. A valued constraint language $\Gamma$ is called **tractable** if, for every finite subset $\Gamma_f \subseteq \Gamma$, there exists an algorithm solving any instance $\mathcal{P} \in \mathrm{VCSP}(\Gamma_f)$ in polynomial time. Conversely, $\Gamma$ is called **NP-hard** if there is some finite subset $\Gamma_f \subseteq \Gamma$ for which VCSP($\Gamma_f$) is NP-hard.

We now define a closure operator on cost functions, which adds to a set of cost functions all other cost functions which can be *expressed* using that set, in the sense defined below.

**Definition 2.** *For any VCSP instance $\mathcal{P} = \langle V, D, C\rangle$, and any list $L = \langle v_1, \ldots, v_r\rangle$ of variables of $\mathcal{P}$, the* **projection** *of $\mathcal{P}$ onto $L$, denoted $\pi_L(\mathcal{P})$, is the $r$-ary cost function defined as follows:*

$$\pi_L(\mathcal{P})(x_1, \ldots, x_r) \stackrel{\text{def}}{=} \min_{\{s:V\to D \ \mid \ \langle s(v_1),\ldots,s(v_r)\rangle=\langle x_1,\ldots,x_r\rangle\}} Cost_{\mathcal{P}}(s)\,.$$

*We say that a cost function $\phi$ is* **expressible** *over a constraint language $\Gamma$ if there exists a VCSP instance $\mathcal{P} \in \mathrm{VCSP}(\Gamma)$ and a list $L$ of variables of $\mathcal{P}$ such that $\pi_L(\mathcal{P}) = \phi$. We define* Express($\Gamma$) *to be the* **expressive power** *of $\Gamma$; that is, the set of all cost functions expressible over $\Gamma$.*

Note that the list of variables $L$ may contain repeated entries, and we define the minimum over an empty set of costs to be $\infty$.

*Example 1.* Let $\mathcal{P}$ be the VCSP instance with a single variable $v$ and no constraints, and let $L = \langle v, v\rangle$. Then, by Definition 2,

$$\pi_L(\mathcal{P})(x, y) = \begin{cases} 0 & \text{if } x = y \\ \infty & \text{otherwise} \end{cases}.$$

Hence for any valued constraint language $\Gamma$, over any set $D$, Express($\Gamma$) contains this binary cost function, which will be called the **equality** cost function.

The next result shows that expressibility preserves tractability.

**Theorem 1 ([15]).** *A valued constraint language $\Gamma$ is tractable if and only if* $\mathrm{Express}(\Gamma)$ *is tractable; similarly, $\Gamma$ is NP-hard if and only if* $\mathrm{Express}(\Gamma)$ *is NP-hard.*

This result shows that, when trying to identify tractable valued constraint languages, it is sufficient to consider only languages of the form $\mathrm{Express}(\Gamma)$. In the following sections, we will show that such languages can be characterised using certain algebraic properties.

## 3 Weighted Relational Clones

**Definition 3.** *We denote by $\mathbf{\Phi}_D$ the set of cost functions on $D$ taking values in $\overline{\mathbb{Q}}_+$ and by $\mathbf{\Phi}_D^{(r)}$ the $r$-ary cost functions in $\mathbf{\Phi}_D$.*

**Definition 4.** *Any cost function $\phi : D^r \to \overline{\mathbb{Q}}_+$ has an associated cost function which takes only the values 0 and $\infty$, known as its* **feasibility relation***, denoted* $\mathrm{Feas}(\phi)$*, which is defined as follows:*

$$\mathrm{Feas}(\phi)(x_1, \dots, x_r) \stackrel{\mathrm{def}}{=} \begin{cases} 0 & \text{if } \phi(x_1, \dots, x_r) < \infty \\ \infty & \text{otherwise} \end{cases} .$$

We now define a closure operator on cost functions with rational costs, which adds to a set of cost functions all other cost functions which can be obtained from that set by a certain affine transformation.

**Definition 5.** *We say $\phi, \phi' \in \mathbf{\Phi}_D$ are* **cost-equivalent***, denoted by $\phi \sim \phi'$, if there exist $\alpha, \beta \in \mathbb{Q}$ with $\alpha > 0$ such that $\phi = \alpha\phi' + \beta$. We denote by $\Gamma_\sim$ the smallest set of cost functions containing $\Gamma$ which is closed under cost-equivalence.*

The next result shows that adding feasibility relations or cost-equivalent cost functions does not increase the complexity of $\Gamma$.

**Theorem 2 ([15]).** *For any valued constraint language $\Gamma$, we have:*

1. *$\Gamma \cup \mathrm{Feas}(\Gamma)$ is tractable if and only if $\Gamma$ is tractable, and $\Gamma \cup \mathrm{Feas}(\Gamma)$ is NP-hard if and only if $\Gamma$ is NP-hard.*
2. *$\Gamma_\sim$ is tractable if and only if $\Gamma$ is tractable, and $\Gamma_\sim$ is NP-hard if and only if $\Gamma$ is NP-hard.*

The algebraic approach to complexity for the classical CSP uses standard algebraic notions of polymorphisms, clones and relational clones [12, 7, 24].

Here we introduce an algebraic theory for valued constraints based on the notions of *weighted polymorphisms*, *weighted clones* and *weighted relational clones*, defined below.

**Definition 6.** *We say a set $\Gamma \subseteq \mathbf{\Phi}_D$ is a* **weighted relational clone** *if it contains the equality cost function and is closed under cost-equivalence and feasibility; rearrangement of arguments; addition of cost functions; and minimisation over arbitrary arguments. For each $\Gamma \subseteq \mathbf{\Phi}_D$ we define* $\mathrm{wRelClone}(\Gamma)$ *to be the smallest weighted relational clone containing $\Gamma$.*

It is a straightforward consequence of Definitions 2 and 6 that, for any valued constraint language $\Gamma \subseteq \mathbf{\Phi}$, the set of cost functions that are cost equivalent to the expressive power of $\Gamma$, together with all associated feasibility relations, is given by the smallest weighted relational clone containing $\Gamma$, as the next result indicates.

**Proposition 1.** *For any $\Gamma \subseteq \mathbf{\Phi}_D$,* $\mathrm{Express}(\Gamma \cup \mathrm{Feas}(\Gamma))_\sim = \mathrm{wRelClone}(\Gamma)$.

Hence, by Theorem 1 and Theorem 2, the search for tractable valued constraint languages taking values in $\overline{\mathbb{Q}}_+$ corresponds to a search for suitable weighted relational clones. As has been done in the crisp case [12], we will now proceed to establish an alternative characterisation for weighted relational clones which facilitates this search.

## 4 Weighted Clones

For any finite set $D$, a function $f : D^k \to D$ is called a $k$-ary *operation* on $D$.

**Definition 7.** *We denote by $\mathbf{O}_D$ the set of all finitary operations on $D$ and by $\mathbf{O}_D^{(k)}$ the $k$-ary operations in $\mathbf{O}_D$.*

**Definition 8.** *The $k$-ary* **projections** *on $D$ are the operations*

$$e_i^{(k)} : D^k \to D\,, \quad (a_1, \dots, a_k) \mapsto a_i\,.$$

**Definition 9.** *We define a $k$-ary* **weighted operation** *on a set $D$ to be a* partial function $\omega : \mathbf{O}_D^{(k)} \to \mathbb{Q}$ *such that $\omega(f) < 0$ only if $f$ is a projection and*

$$\sum_{f \in \mathbf{dom}(\omega)} \omega(f) = 0\,.$$

*The* **domain** *of $\omega$, denoted $\mathbf{dom}(\omega)$, is the subset of $\mathbf{O}_D^{(k)}$ on which $\omega$ is defined. We denote by $ar(\omega) = k$ the arity of $\omega$.*

*We denote by $\mathbf{W}_D$ the finitary weighted operations on $D$ and by $\mathbf{W}_D^{(k)}$ the $k$-ary weighted operations on $D$.*

**Definition 10.** *We say that two $k$-ary weighted operations $\omega, \mu \in \mathbf{W}_D^{(k)}$ are* **weight-equivalent** *if $\mathbf{dom}(\omega) = \mathbf{dom}(\mu)$ and there exists some fixed positive rational $c$, such that $\omega(f) = c\mu(f)$, for all $f \in \mathbf{dom}(\omega)$.*

**Definition 11.** *For any $\omega_1, \omega_2 \in \mathbf{W}_D^{(k)}$, we define the sum of $\omega_1$ and $\omega_2$, denoted $\omega_1 + \omega_2$, to be the $k$-ary weighted operation $\omega$ with $\mathbf{dom}(\omega) = \mathbf{dom}(\omega_1) \cup \mathbf{dom}(\omega_2)$ and*

$$\omega(f) = \begin{cases} \omega_1(f) + \omega_2(f) & f \in \mathbf{dom}(\omega_1) \cap \mathbf{dom}(\omega_2) \\ \omega_1(f) & f \in \mathbf{dom}(\omega_1) \setminus \mathbf{dom}(\omega_2) \\ \omega_2(f) & f \in \mathbf{dom}(\omega_2) \setminus \mathbf{dom}(\omega_1) \end{cases} . \tag{1}$$

**Definition 12.** *Let $f \in \mathbf{O}_D^{(k)}$ and $g_1, \ldots, g_k \in \mathbf{O}_D^{(l)}$. The* superposition *of $f$ and $g_1, \ldots, g_k$ is the $l$-ary operation $f[g_1, \ldots, g_k] : D^l \to D$, $(x_1, \ldots, x_l) \mapsto f(g_1(x_1, \ldots, x_l), \ldots, g_k(x_1 \ldots, x_l))$.*

**Definition 13.** *For any $\omega \in \mathbf{W}_D^{(k)}$ and any $g_1, g_2, \ldots, g_k \in \mathbf{O}_D^{(l)}$, we define the* **translation** *of $\omega$ by $g_1, \ldots, g_k$, denoted $\omega[g_1, \ldots, g_k]$, to be the* partial function *$\omega[g_1, \ldots, g_k]$ from $\mathbf{O}_D^{(l)}$ to $\mathbb{Q}$ defined by*

$$\omega[g_1, \ldots, g_k](f) \stackrel{\text{def}}{=} \sum_{\substack{f' \in \mathbf{dom}(\omega) \\ f = f'[g_1, \ldots, g_k]}} \omega(f') . \tag{2}$$

*The domain of $\omega[g_1, \ldots, g_k]$ is the set of $l$-ary operations $\{f'[g_1, g_2, \ldots, g_k] \mid f' \in \mathbf{dom}(\omega)\}$.*

*Example 2.* Let $\omega$ be the 4-ary weighted operation on $D$ given by

$$\omega(f) = \begin{cases} -1 & \text{if } f \text{ is a projection} \\ +1 & \text{if } f \in \{\max(x_1, x_2), \min(x_1, x_2), \max(x_3, x_4), \min(x_3, x_4)\} \end{cases} ,$$

and let

$$\langle g_1, g_2, g_3, g_4 \rangle = \left\langle e_1^{(3)}, e_2^{(3)}, e_3^{(3)}, \max(x_1, x_2) \right\rangle .$$

Then, by Definition 13 we have

$$\omega[g_1, g_2, g_3, g_4](f) = \begin{cases} -1 & \text{if } f \text{ is a projection} \\ +1 & \text{if } f \in \{\max(x_1, x_2, x_3), \min(x_1, x_2), \min(x_3, \max(x_1, x_2))\} \\ 0 & \text{if } f = \max(x_1, x_2) \end{cases} .$$

Note that $\omega[g_1, g_2, g_3, g_4]$ satisfies the conditions of Definition 9 and hence is a weighted operation.

*Example 3.* Let $\omega$ be the same as in Example 2 but now consider

$$\langle g_1', g_2', g_3', g_4' \rangle = \left\langle e_1^{(4)}, \max(x_2, x_3), \min(x_2, x_3), e_4^{(4)} \right\rangle .$$

By Definition 13 we have

$$\omega[g_1', g_2', g_3', g_4'](f) = \begin{cases} -1 & \text{if } f \in \{e_1^{(4)}, \max(x_2, x_3), \min(x_2, x_3), e_4^{(4)}\} \\ +1 & \text{if } f \in \left\{ \begin{array}{l} \max(x_1, x_2, x_3), \min(x_1, \max(x_2, x_3)), \\ \max(\min(x_2, x_3), x_4), \min(x_2, x_3, x_4) \end{array} \right\} \end{cases} .$$

Note that $\omega[g_1', g_2', g_3', g_4']$ does not satisfy the conditions of Definition 9 because, for example, $\omega[g_1', g_2', g_3', g_4'](f) < 0$ when $f = \max(x_2, x_3)$, which is not a projection. Hence $\omega[g_1', g_2', g_3', g_4']$ is not a weighted operation.

**Definition 14.** *If the result of a translation is a weighted operation, then that translation will be called a* **proper** *translation.*

*Remark 1.* For any $\omega \in \mathbf{W}_D^{(k)}$, if $g_1, \ldots, g_k$ are *projections*, then it can be shown that the function $\omega[g_1, \ldots, g_k]$ satisfies the conditions of Definition 9, and hence is a weighted operation. This means that a translation by any list of projections is always a proper translation.

We are now ready to define *weighted clones*.

**Definition 15.** *Let $C$ be a clone of operations on $D$. We say a set $W \subseteq \mathbf{W}_D$ is a* **weighted clone** *with* **support** *$C$ if it contains all zero-valued weighted operations whose domains are subsets of $C$ and is closed under weight-equivalence, addition, and proper translation by operations from $C$.*

*For each $W \subseteq \mathbf{W}_D$ we define* $\mathrm{wClone}(W)$ *to be the smallest weighted clone containing $W$.*

*Remark 2.* The support of $\mathrm{wClone}(W)$ is the clone generated by the domains of the elements of $W$. That is, the support of $\mathrm{wClone}(W)$ is $\mathrm{Clone}(\cup_{\omega\in W} \mathbf{dom}(\omega))$.

*Example 4.* For any clone of operations, $C$, there exists a unique weighted clone which consists of all weighted operations assigning weight 0 to each subset of $C$.

**Definition 16.** *Let $\phi \in \mathbf{\Phi}_D^{(r)}$ and let $\omega \in \mathbf{W}_D^{(k)}$. We say that $\omega$ is a* **weighted polymorphism** *of $\phi$ if, for any $x_1, x_2, \ldots, x_k \in D^r$ such that $\phi(x_i) < \infty$ for $i = 1, \ldots, k$, we have*

$$\sum_{f\in\mathbf{dom}(\omega)} \omega(f)\phi(f(x_1, x_2, \ldots, x_k)) \leq 0\,. \tag{3}$$

*If $\omega$ is a weighted polymorphism of $\phi$ we say $\phi$ is* **improved** *by $\omega$.*

Note that, because $a\infty = \infty$ for any value $a$ (including $a = 0$), if inequality (3) holds we must have $\phi(f(x_1, \ldots, x_k)) < \infty$, for all $f \in \mathbf{dom}(\omega)$, i.e., each $f \in \mathbf{dom}(\omega)$ is a polymorphism of $\phi$.

*Example 5.* Consider the class of submodular set functions [31]. These are precisely the cost functions on $\{0, 1\}$ satisfying

$$\phi(\min(x_1, x_2)) + \phi(\max(x_1, x_2)) - \phi(x) - \phi(y) \leq 0\,.$$

In other words, the set of submodular functions are defined as the set of cost functions on $\{0, 1\}$ with the 2-ary weighted polymorphism

$$\omega(f) = \begin{cases} -1 & \text{if } f \in \{e_1^{(2)}, e_2^{(2)}\} \\ +1 & \text{if } f \in \{\min(x_1, x_2), \max(x_1, x_2)\} \end{cases}\,.$$

**Definition 17.** *For any $\Gamma \subseteq \mathbf{\Phi}_D$, we denote by* $\mathrm{wPol}(\Gamma)$ *the set of all finitary weighted operations on $D$ which are weighted polymorphisms of all cost function $\phi \in \Gamma$ and by* $\mathrm{wPol}^{(k)}(\Gamma)$ *the $k$-ary weighted operations in* $\mathrm{wPol}(\Gamma)$.

**Definition 18.** *For any* $W \subseteq \mathbf{W}_D$*, we denote by* $\mathrm{Imp}(W)$ *the set of all cost functions in* $\mathbf{\Phi}_D$ *that are improved by all weighted operations* $\omega \in W$ *and by* $\mathrm{Imp}^{(r)}(W)$ *the* $r$*-ary cost functions in* $\mathrm{Imp}(W)$*.*

It follows immediately from the definition of a Galois connection [8] that, for any set $D$, the mappings wPol and Imp form a Galois connection between $\mathbf{W}_D$ and $\mathbf{\Phi}_D$. A characterisation of this Galois connection for finite sets $D$ is given by the following two theorems:

**Theorem 3.** *For any finite set D, and any finite* $\Gamma \subseteq \mathbf{\Phi}_D$*,*

$$\mathrm{Imp}(\mathrm{wPol}(\Gamma)) = \mathrm{wRelClone}(\Gamma).$$

**Theorem 4.** *For any finite set D, and any finite* $W \subseteq \mathbf{W}_D$*,*

$$\mathrm{wPol}(\mathrm{Imp}(W)) = \mathrm{wClone}(W).$$

As with any Galois connection [8], this means that there is a one-to-one correspondence between weighted clones and weighted relational clones. Hence, by Proposition 1, Theorem 1, and Theorem 2, the search for tractable valued constraint languages taking values in $\overline{\mathbb{Q}}_+$ corresponds to a search for suitable weighted clones of operations.

## 5 Proof of Theorems 3 and 4

A similar result to Theorem 3 was obtained in [15, Theorem 4] using the related algebraic notion of fractional polymorphism. The proof given in [15] can be adapted in a straightforward way, and we omit the details due to space constraints. We will sketch the proof of Theorem 4. First, we show in Proposition 2 that the weighted polymorphisms of a set of cost functions form a weighted clone. The rest of the theorem then follows from Theorem 5, which states that any weighted operation that improves all cost functions in $\mathrm{Imp}(W)$ is an element of the weighted clone $\mathrm{wClone}(W)$. Due to space constraints we will not include the proof of Theorem 5.

**Proposition 2.** *Let D be a finite set.*

1. *For all* $\Gamma \subset \mathbf{\Phi}_D$*,* $\mathrm{wPol}(\Gamma)$ *is a weighted clone with support* $\mathrm{Pol}(\Gamma)$*.*
2. *For all* $W \subset \mathbf{W}_D$*,* $\mathrm{wClone}(W) \subseteq \mathrm{wPol}(\mathrm{Imp}(W))$*.*

*Proof.* Certainly $\mathrm{wPol}(\Gamma)$ contains all zero-valued weighted operations with domains contained in $\mathrm{Pol}(\Gamma)$, since all of these satisfy the conditions set out in Definition 16. Similarly, $\mathrm{wPol}(\Gamma)$ is closed under addition and weight-equivalence, since both of these operations preserve inequality (3). Hence, to show $\mathrm{wPol}(\Gamma)$ is a weighted clone we only need to show $\mathrm{wPol}(\Gamma)$ is closed under proper translations by members of $\mathrm{Pol}(\Gamma)$.

Let $\omega \in \mathrm{wPol}^{(k)}(\Gamma)$ and suppose $\omega' = \omega[g_1, \ldots, g_k]$ is a proper translation of $\omega$, where $g_1, g_2, \ldots, g_k \in \mathrm{Pol}^{(l)}(\Gamma)$. We will now show that $\omega' \in \mathrm{wPol}^{(l)}(\Gamma)$.

Suppose $\phi$ is an $r$-ary cost function satisfying $\omega \in \mathrm{wPol}(\{\phi\})$, i.e., $\phi$ and $\omega$ satisfy (3) for any $x_1, x_2, \ldots, x_k \in \mathrm{Feas}(\phi)$. Given any $x'_1, x'_2, \ldots, x'_l \in \mathrm{Feas}(\phi)$, set $x_i = g_i(x'_1, x'_2, \ldots, x'_l)$ for $i = 1, 2, \ldots, k$. Then, if we set $f' = f[g_1, \ldots, g_k]$, we have $f'(x'_1, x'_2, \ldots, x'_l) = f(x_1, x_2, \ldots, x_k)$, for any $f \in \mathbf{O}_D^{(k)}$. Hence, by Definition 13, we have

$$\sum_{f' \in \mathbf{dom}(\omega')} \omega'(f')\phi(f'(x'_1, x'_2, \ldots, x'_k) = \sum_{f \in \mathbf{dom}(\omega)} \omega(f)\phi(f(x_1, x_2, \ldots, x_k) \leq 0 \,.$$

For the second part, we observe that $W \subseteq \mathrm{wPol}(\mathrm{Imp}(W))$. Hence, $\mathrm{wClone}(W) \subseteq \mathrm{wClone}(\mathrm{wPol}(\mathrm{Imp}(W))) = \mathrm{wPol}(\mathrm{Imp}(W))$. □

We will make use of the following lemma, which shows that a weighted sum of arbitrary translations of any weighted operations $\omega_1$ and $\omega_2$ can be obtained by translating $\omega_1$ and $\omega_2$ by projection operations, forming a weighted sum, and then translating the result.

**Lemma 1.** *For any weighted operations $\omega_1 \in \mathbf{W}_D^{(k)}$ and $\omega_2 \in \mathbf{W}_D^{(l)}$ and any $g_1, \ldots, g_k \in \mathbf{O}_D^{(m)}$ and $g'_1, \ldots, g'_l \in \mathbf{O}_D^{(m)}$,*

$$c_1\,\omega_1[g_1, \ldots, g_k] + \ c_2\,\omega_2[g'_1, \ldots, g'_l] = \omega[g_1, \ldots, g_k, g'_1, \ldots, g'_l] \,, \tag{4}$$

*where* $\omega = c_1\,\omega_1[e_1^{(k+l)}, \ldots, e_k^{(k+l)}] + \ c_2\,\omega_2[e_{k+1}^{(k+l)}, \ldots, e_{k+l}^{(k+l)}]$

*Proof.* For any $f \in \mathbf{dom}(\omega)$, the result of applying the right-hand side expression in equation (4) to $f$ is:

$$\sum_{\substack{f' \in \mathbf{dom}(\omega) \\ f = f'[g_1, \ldots, g_k, g'_1, \ldots, g'_l]}} \left( \sum_{\substack{h' \in \mathbf{dom}(\omega_1) \\ f' = h'[e_1^{(k+l)}, \ldots, e_k^{(k+l)}]}} c_1\,\omega_1(h') \quad + \sum_{\substack{h' \in \mathbf{dom}(\omega_2) \\ f' = h'[e_{k+1}^{(k+l)}, \ldots, e_{k+l}^{(k+l)}]}} c_2\,\omega_2(h') \right) .$$

Replacing each $f'$ by the equivalent superposition of $h'$ with projections, we obtain:

$$\sum_{\substack{h' \in \mathbf{dom}(\omega_1) \\ f = h'[g_1, \ldots, g_k]}} c_1\,\omega_1(h') + \sum_{\substack{h' \in \mathbf{dom}(\omega_2) \\ f = h'[g'_1, \ldots, g'_l]}} c_2\,\omega_2(h') \,,$$

which is the result of applying the left-hand-side of Equation 4 to $f$. □

**Theorem 5.** *For all finite $W \subset \mathbf{W}_D$, $\mathrm{wPol}(\mathrm{Imp}(W)) \subseteq \mathrm{wClone}(W)$.*

## 6 Conclusions

We have presented an algebraic theory of valued constraint languages analogous to the theory of clones used to study the complexity of the classical constraint satisfaction problem. We showed that the complexity of any valued constraint language with rational costs is determined by certain algebraic properties of the cost functions allowed in the language: the weighted polymorphisms. Compared to the results in [15], not only have we captured the complexity of valued constraint languages, but we have also established a 1-to-1 connection between valued constraint languages and sets of weighted polymorphisms.

In previous work [27, 26] it has been shown that every tractable crisp constraint language can be characterised by an associated clone of operations. This work initiated the use of algebraic properties in the search for tractable constraint languages, an area that has seen considerable activity in recent years; see, for instance, [11, 13, 12, 10, 28, 22, 3, 1, 2, 5]. The results in this paper show that a similar result holds for the valued constraint satisfaction problem: every tractable valued constraint language is characterised by an associated weighted clone.

We believe that our results here will provide a similar impetus for the investigation of tractable valued constraint satisfaction problems. In fact, we have already commenced investigating minimal weighted clones in order to understand maximal valued constraint languages [19]. Building on Rosenberg's famous classification of minimal clones, we have obtained a similar classification of minimal weighted clones [19]. Furthermore, using the results from this paper, we have proved maximality of several known tractable valued constraint languages, including an alternative proof of the characterisation of all maximal Boolean valued constraint languages from [17]. Details on minimal weighted clones and other applications of our results will be included in the full version of this paper.

## References


1. L. Barto, M. Kozik, Constraint Satisfaction Problems of Bounded Width, in: FOCS'09, pp. 461–471.
2. L. Barto, M. Kozik, M. Maróti, T. Niven, CSP dichotomy for special triads, Proceedings of the American Mathematical Society 137 (9) (2009) 2921–2934.
3. L. Barto, M. Kozik, T. Niven, The CSP dichotomy holds for digraphs with no sources and no sinks, SIAM Journal on Computing 38 (5) (2009) 1782–1802.
4. L. Barto, The dichotomy for conservative constraint satisfaction problems revisited, In: LICS'11, 2011.
5. J. Berman, P. Idziak, P. Marković, R. McKenzie, M. Valeriote, R. Willard, Varieties with few subalgebras of powers, Trans. of AMS 362 (3) (2010) 1445–1473.
6. S. Bistarelli, H. Fargier, U. Montanari, F. Rossi, T. Schiex, G. Verfaillie, Semiring-based CSPs and Valued CSPs Constraints 4 (3) (1999) 199–240.
7. V. Bodnarčuk, L. Kalužnin, V. Kotov, B. Romov, Galois theory for Post algebras. I, Cybernetics and Systems Analysis 5 (3) (1969) 243–252.
8. F. Börner, Basics of Galois connections, in: Complexity of Constraints, vol. 5250 of Lecture Notes in Computer Science, Springer, 2008, pp. 38–67.
9. A. Bulatov, A Graph of a Relational Structure and Constraint Satisfaction Problems, in: LICS'04 pp. 448–457.
10. A. Bulatov, A dichotomy theorem for constraint satisfaction problems on a 3-element set, Journal of the ACM 53 (1) (2006) 66–120.

11. A. Bulatov, A. Krokhin, P. Jeavons, The complexity of maximal constraint languages, in: STOC'01, pp. 667–674.
12. A. Bulatov, A. Krokhin, P. Jeavons, Classifying the Complexity of Constraints using Finite Algebras, SIAM Journal on Computing 34 (3) (2005) 720–742.
13. A. A. Bulatov, Tractable Conservative Constraint Satisfaction Problems, in: LICS'03 pp. 321–330.
14. D. Cohen, M. Cooper, P. Jeavons, A. Krokhin, Soft Constraints: Complexity and Multimorphisms, in: CP'03 pp. 244–258.
15. D. A. Cohen, M. C. Cooper, P. G. Jeavons, An Algebraic Characterisation of Complexity for Valued Constraints, in: CP'06 pp. 107–121.
16. D. A. Cohen, M. C. Cooper, P. G. Jeavons, Generalising submodularity and Horn clauses: Tractable optimization problems defined by tournament pair multimorphisms, Theoretical Computer Science 401 (1-3) (2008) 36–51.
17. D. A. Cohen, M. C. Cooper, P. G. Jeavons, A. A. Krokhin, The Complexity of Soft Constraint Satisfaction, Artificial Intelligence 170 (11) (2006) 983–1016.
18. D. A. Cohen, P. G. Jeavons, S. Živný, The expressive power of valued constraints: Hierarchies and collapses, Theoretical Computer Science 409 (1) (2008) 137–153.
19. P. Creed, S. Živný, On minimal weighted clones, In: CP'11.
20. N. Creignou, A dichotomy theorem for maximum generalized satisfiability problems, Journal of Computer and System Sciences 51 (3) (1995) 511–522.
21. N. Creignou, S. Khanna, M. Sudan, Complexity Classification of Boolean Constraint Satisfaction Problems, SIAM, 2001.
22. V. Deineko, P. Jonsson, M. Klasson, A. Krokhin, The approximability of Max CSP with fixed-value constraints, Journal of the ACM 55 (4) (2008).
23. T. Feder, M. Vardi, The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory, SIAM Journal on Computing 28 (1) (1998) 57–104.
24. D. Geiger, Closed systems of functions and predicates, Pacific Journal of Mathematics 27 (1) (1968) 95–100.
25. M. Gyssens, P. Jeavons, D. Cohen, Decomposing Constraint Satisfaction Problems Using Database Techniques, Artificial Intelligence 66 (1) (1994) 57–89.
26. P. Jeavons, On the Algebraic Structure of Combinatorial Problems, Theoretical Computer Science 200 (1-2) (1998) 185–204.
27. P. Jeavons, D. Cohen, M. Gyssens, Closure Properties of Constraints, Journal of the ACM 44 (4) (1997) 527–548.
28. P. Jonsson, M. Klasson, A. Krokhin, The Approximability of Three-valued MAX CSP, SIAM Journal on Computing 35 (6) (2006) 1329–1349.
29. V. Kolmogorov, S. Živný, The complexity of conservative finite-valued CSPs, Tech. rep., arXiv:1008.1555 (August 2010).
30. V. Kolmogorov, S. Živný, Generalising tractable VCSPs defined by symmetric tournament pair multimorphisms, Tech. rep., arXiv:1008.3104 (August 2010).
31. G. Nemhauser, L. Wolsey, Integer and Combinatorial Optimization, John Wiley & Sons, 1988.
32. T. Schaefer, The Complexity of Satisfiability Problems, in: Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC'78), 1978, pp. 216–226.
33. A. Schrijver, Theory of linear and integer programming, John Wiley & Sons, Inc., 1986.
34. S. Živný, D. A. Cohen, P. G. Jeavons, The Expressive Power of Binary Submodular Functions, Discrete Applied Mathematics 157 (15) (2009) 3347–3358.