Linear Programming and the Complexity of Finite-valued CSPs

Johan Thapper1 Stanislav Živný2
1 Université Paris-Sud 11, France
2 University of Warwick, United Kingdom
thapper@lri.fr s.zivny@warwick.ac.uk

Abstract

We consider the complexity classification project for the valued constraint satisfaction problem (VCSP) restricted to rational-valued cost functions. A year ago, classifications were known for Boolean domains and for the case when all unary cost functions are allowed. In the time that has passed since then, there has been significant progress in the area. In this paper, we describe the full complexity classification for finite-valued CSPs. Our result is a dichotomy. It states that the problem of minimising VCSP instances that use only rational-valued cost functions from a fixed set is either tractable or NP-hard, depending on the set. In the tractable cases, all instances are solved to optimality by establishing optimal soft arc consistency (OSAC), or even by the weaker basic linear programming relaxation (BLP). In the hard cases, it is possible to express a cost function that models the Max-Cut problem. Our result implies the full complexity classification for the Max-CSP which until these results seemed to be entirely out of reach.

1 Introduction

The valued constraint satisfaction problem, or VCSP for short, was introduced by Schiex et al. [13] as a unifying framework for studying constraint programming with soft constraints. We study the special case where the valuation structure is \( \mathbb{Q} \) with addition as the aggregation operator, and each constraint comes with an additional weight. This special case is also called the finite-valued case, to distinguish it from the case where positive infinity is allowed as a value. From now on, whenever we talk about the VCSP, we will be referring to the finite-valued case.

Definition 1 An instance \( I \) of the valued constraint satisfaction problem is given by a set \( V = \{x_1, \ldots, x_n\} \) of variables, a finite domain \( D \), and an objective function \( f_I(x_1, \ldots, x_n) = \sum_{i=1}^q w_i \cdot f_i(\bar{x}_i) \) where, for every \( 1 \leq i \leq q \), \( f_i : D^{\text{var}(f_i)} \to \mathbb{Q} \) is a cost function, \( \bar{x}_i \in \text{var}(f_i) \) is a tuple of variables, and \( w_i \in \mathbb{Q}_{\geq 0} \) is a weight. A solution to \( I \) is a function \( \sigma : V \to D \), its measure given by \( \sum_{i=1}^q w_i \cdot f_i(\sigma(\bar{x}_i)) \), where \( \sigma \) is applied componentwise. The goal is to find a solution of minimum measure. This measure is denoted \( \text{Opt}(I) \).

A set \( \Gamma \) of cost functions is called a valued constraint language, or simply a language. VCSP(\( \Gamma \)) is the class of instances in which all functions are from \( \Gamma \).

The systematic study of the complexity of VCSP(\( \Gamma \)) was initiated by Cohen et al. [3]. The goal is to determine, for a given \( \Gamma \), whether VCSP(\( \Gamma \)) is solvable in polynomial-time (tractable) or NP-hard. Languages on two-element domains [3], three-element domains [7], and conservative languages [11] (i.e., languages containing all unary functions) have been completely classified with respect to exact solvability. In the special case of \( \{0,1\} \)-valued languages, which correspond to Max-CSPs, languages on two-element domains [5], three-element domains [8], four-element domains [9], and conservative languages [6] have been classified.

Our contribution is a classification of all tractable valued languages as those admitting a binary idempotent and symmetric fractional polymorphism. We also show that this is a polynomial-time checkable condition. Our main result can be stated as follows, where BLP is a linear programming relaxation related to optimal soft arc consistency (see Section 3):

Theorem 1.1 Let \( \Gamma \) be a finite-valued language defined on a domain \( D \). VCSP(\( \Gamma \)) is tractable if BLP solves VCSP(\( \Gamma \)). Otherwise, VCSP(\( \Gamma \)) is NP-hard.

*The results presented in this paper have previously been published in [14] and [15].
Overview

In Section 2, we introduce the basic technical notions necessary for presenting our results. In Section 3, we give a characterisation of the power of a linear programming relaxation, and in Section 4, we establish our main dichotomy result. We finish with some open problems in Section 5.

2 Preliminaries

We define three important notions: the expressive power of a valued language, fractional polymorphisms, and the notion of a core valued language.

2.1 Expressive Power

The basic notion for the study of the complexity of constraint satisfaction problems is the expressive power and the related reduction.

Definition 2 For a valued language \( \Gamma \), we let \( \langle \Gamma \rangle \) be the set of all functions \( f(x_1, \ldots, x_m) \) such that for some instance \( I \in \text{VCSP}(\Gamma) \) with objective function \( f_1(x_1, \ldots, x_n), m \leq n, \) we have \( f(x_1, \ldots, x_m) = \min_{x_{m+1}, \ldots, x_n} f_1(x_1, \ldots, x_m, x_{m+1}, \ldots, x_n) \). We say that \( \Gamma \) expresses \( f \) and we call \( \langle \Gamma \rangle \) the expressive power of \( \Gamma \).

In other words, the expressive power \( \langle \Gamma \rangle \) is the closure of \( \Gamma \) under addition, multiplication by nonnegative constants, and minimisation over extra variables.

Theorem 2.1 ([3, 2]) Let \( \Gamma \) and \( \Gamma' \) be valued languages with \( \Gamma' \subseteq \langle \Gamma \rangle \) and \( \Gamma \) finite. Then, \( \text{VCSP}(\Gamma') \) polynomial-time reduces to \( \text{VCSP}(\Gamma) \).

In particular, if \( \Gamma \) expresses some “NP-hard cost function” \( h \), then, by Theorem 2.1, \( \text{VCSP}(\{h\}) \) reduces to \( \text{VCSP}(\Gamma) \), and hence \( \Gamma \) is NP-hard.

2.2 Fractional Polymorphisms

Let \( \Gamma \) be a valued language defined on \( D \). An \( m \)-ary operation on \( D \) is a function \( g : D^m \rightarrow D \). The set of all \( m \)-ary operations on \( D \) is denoted by \( \mathcal{O}_D^{(m)} \). An \( m \)-ary fractional operation is a function \( \omega : \mathcal{O}_D^{(m)} \rightarrow \mathbb{Q}_{\geq 0} \). A (binary) fractional operation \( \omega \) is called a (binary) fractional polymorphism [2] of \( \Gamma \) if \( \sum_{g \in \mathcal{O}_D^{(m)}} \omega(g) = 1 \) and for every function \( f \in \Gamma \) and tuples \( \bar{x}, \bar{y} \in D^{\text{ar}(f)} \), it holds that

\[
\sum_{g \in \mathcal{O}_D^{(2)}} \omega(g)g(f(\bar{x}, \bar{y})) \leq \frac{1}{2} \left( f(\bar{x}) + f(\bar{y}) \right),
\]

where \( g(\bar{x}, \bar{y}) = (g(x_1, y_1), \ldots, g(x_{\text{ar}(f)}, y_{\text{ar}(f)}) \rangle \). General \( m \)-ary fractional polymorphisms are defined analogously.

An operation \( g \) is idempotent if \( g(x, \ldots, x) = x \). A binary operation \( g \) is symmetric if \( g(x, y) = g(y, x) \). A fractional operation is called idempotent (symmetric) if all operations in \( \{ g \mid \omega(g) > 0 \} \) are idempotent (symmetric).

Example 1 Let \( < \) be a total order on \( D \). A function \( f : D^n \rightarrow \mathbb{Q} \) is called submodular (w.r.t. \( < \)) if, for every \( \bar{x}, \bar{y} \in D^n \),

\[
f(\min_{<} (\bar{x}, \bar{y})) + f(\max_{<} (\bar{x}, \bar{y})) \leq f(\bar{x}) + f(\bar{y}).
\]

Note that this is equivalent to saying that the fractional operation that assigns \( \frac{1}{2} \) to each of the binary operations \( \text{min} \) and \( \text{max} \) is an idempotent and symmetric fractional polymorphism of the language \( \{ f \} \).

2.3 Valued Cores

The concept of a core of a constraint language is fundamental in the theoretical study of CSPs. Intuitively, a non-core is a language that can be squashed to an equivalent language over a strictly smaller domain. An analogous concept for valued constraint languages was introduced in [7]. Here, we give a different, but equivalent, definition.

Let \( S \subseteq D \). The sub-language \( \Gamma[S] \) of \( \Gamma \) induced by \( S \) is the valued language defined on domain \( S \) and containing the restriction of every cost function \( f \in \Gamma \) to \( S \).

Definition 3 A valued language \( \Gamma \) is a core if for every unary fractional polymorphism \( \omega \) of \( \Gamma \), all operations \( g \) with \( \omega(g) > 0 \) are injective.

A valued language \( \Gamma' \) is a core of \( \Gamma \) if \( \Gamma' \) is a core and for some unary fractional polymorphism \( \omega \) of \( \Gamma \) and some operation \( h \) with \( \omega(h) > 0 \), we have \( \Gamma' = \Gamma[h(D)] \).

Lemma 2.2 If \( \Gamma' \) is a core of \( \Gamma \), then \( \text{Opt}(I) = \text{Opt}(I') \) for all instances \( I \in \text{VCSP}(\Gamma') \), where \( I' \) is obtained from \( I \) by substituting each function in \( \Gamma \) for its restriction in \( I' \).

For a valued language \( \Gamma \), let \( \Gamma_c \) denote the language containing all functions that can be obtained from a function in \( \Gamma \) by fixing a (possibly empty) subset of its variables to domain values. For a cost function \( f \), let \( \text{argmin}_D(f) = \{ \bar{x} \mid f(\bar{x}) = \min_{y \in D^{\text{ar}(f)}} f(\bar{y}) \} \).

Proposition 2.3 ([7]) Let \( \Gamma \) be a core valued language defined on a finite domain \( D \).

1. For each \( a \in D \), \( \langle \Gamma_c \rangle \) contains a unary function \( u_a \) such that \( \text{argmin}_D u_a = \{ a \} \).
2. \( \Gamma \) is NP-hard if, and only if, \( \Gamma_c \) is NP-hard.
3 The Basic LP Relaxation and OSAC

Let $I \in \text{VCSP}(\Gamma)$ be given as in Definition 1 by a set of variables $V = \{x_1, \ldots, x_n\}$, and an objective function $f(x_1, \ldots, x_n) = \sum_{i=1}^{n} w_i \cdot f_i(\bar{x})$.

For a tuple $\bar{x}$, we denote by $\{\bar{x}\}$ the set of elements in $\bar{x}$. The basic LP relaxation has variables $\lambda_{i,\sigma}$ for every $1 \leq i \leq q$ and $\sigma: \{\bar{x}\} \to D$; and variables $\mu_{x,a}$ for every $x \in V$ and $a \in D$.

$$\min \sum_{i=1}^{q} \left( \sum_{\sigma: \{\bar{x}\} \to D} \lambda_{i,\sigma} \right) \forall i, \bar{x} \in \{\bar{x}\}, a \in D$$

$$= \sum_{\sigma: \{\bar{x}\} \to D} \lambda_{i,\sigma} \forall i, x \in \{\bar{x}\}, a \in D$$

$$\sum_{a \in D} \mu_{x,a} = 1 \forall x \in V$$

$$0 \leq \lambda, \mu \leq 1$$

Theorem 3.1 \([14, 10]\) Let $\Gamma$ be a valued language. Then BLP solves VCSP($\Gamma$) if, and only if, $\Gamma$ has a binary symmetric fractional polymorphism.

LP relaxations such as (1) as well as semidefinite programming (SDP) relaxations have been popular in the study of the approximation properties of VCSPs (sometimes referred to as generalised CSPs) \([12]\). In the constraint optimisation community, a related relaxation has appeared under the name of optimal soft arc consistency (OSAC).

We group the terms of $I$ with respect to their scope. Let $S \subseteq V$. The terms of this scope are of the form $w_i \cdot f_i(\sigma(\bar{x}))$, where $\{\bar{x}\} = S$. For each scope $S$, $x \in S$, and $a \in D$, we have a variable $y_{S,x}(a)$. For each $x \in V$, we have a variable $z_x$.

Establishing optimal soft arc consistency amounts to solving the following linear program \([4]\):

$$\max \sum x \cdot z_x$$

$$\sum_{\sigma: \{\bar{x}\} \to D} w_i \cdot f_i(\sigma(\bar{x})) - \sum_{x \in S} y_{S,x}(\sigma(x)) \geq 0$$

$$\forall \sigma: \{\bar{x}\} \to D$$

$$\sum_{\sigma: \{\bar{x}\} \to D} w_i \cdot f_i(\sigma(\bar{x})) - \sum_{x \in S} y_{S,x}(\sigma(x)) \geq 0$$

$$\forall \sigma: \{\bar{x}\} \to D$$

The condition (MC) can be shown to imply NP-hardness via a reduction from Max-Cut \([3]\). Our main result implies that (MC) is the only source of intractability for valued CSPs.

Theorem 4.1 Let $D$ be an arbitrary finite set and let $\Gamma$ be a core valued language defined on $D$.

- Either $\Gamma$ has a binary idempotent and symmetric fractional polymorphism and BLP solves VCSP($\Gamma$);
- or (MC) holds for $\Gamma$, and VCSP($\Gamma$) is NP-hard.

We say that OSAC solves VCSP($\Gamma$) if the optimum of (2) is equal to Opt($I$) for all $I \in \text{VCSP}(\Gamma)$. As a direct corollary of Theorem 4.1, we conclude that OSAC is unlikely to add any power when it comes to the exact solvability of valued CSPs.

Corollary 4.2 Let $\Gamma$ be a valued language. Assuming that $P \neq NP$, OSAC solves VCSP($\Gamma$) if, and only if, $\Gamma$ has a binary symmetric fractional polymorphism.

5 Discussion and Open Problems

We have completely answered the question of which valued constraint languages on finite domains are solvable exactly in polynomial time. In particular, we have characterised the tractable languages as those that admit a binary idempotent and symmetric fractional polymorphism. Moreover, all tractable languages are solvable by the basic linear programming relaxation (BLP), or equivalently, by establishing optimal soft arc consistency (OSAC).

We end this section by listing a number of interesting open problems related to our results and VCSPs in general. The following meta problems are relevant to our classification. First, can we determine whether $\Gamma$ is a core or not? If not, can we find a core of $\Gamma$. Second, can we determine whether VCSP($\Gamma$) is tractable or NP-hard? Both of these problems are decidable.
Proposition 5.1 Given a finite valued language $\Gamma$, the problem of determining whether $\Gamma$ is a core is decidable, and a core of $\Gamma$ can be found in polynomial time.

Proposition 5.2 Given a finite valued language $\Gamma$, the problem of determining whether $\Gamma$ is tractable or NP-hard is decidable.

These results rely on solving a linear program with as many inequalities as there are (unary or binary) operations on $D$. Hence, when $D$ is a part of the input, these linear programs are exponentially large.

Problem 1 Is it possible to determine, in time polynomial in the representation size of $\Gamma$, (1) whether $\Gamma$ is a core; and (2) whether $\Gamma$ is tractable or NP-hard?

By Corollary 4.2, OSAC solves VCSP($\Gamma$) if and only if BLP does, provided that $P \neq NP$. Clearly, though, the power of OSAC should not depend on a deep complexity theoretical conjecture.

Problem 2 Does BLP solve VCSP($\Gamma$) precisely when OSAC does?

While fractional polymorphisms can be used to characterise tractable constraints and constraint languages, they are somewhat less useful for describing concrete constraints.

Problem 3 Find an alternative characterisation of (some subclass of) constraints admitting a symmetric binary fractional polymorphism.

For example, a function $f : \{0, 1\} \to \{0, 1\}$ is submodular w.r.t. the order $1 < 0$ if and only if it can be represented as a 2-monotone formula: $f(x_1, \ldots, x_k) = (x_{i_1} \land \cdots \land x_{i_k}) \lor (\neg x_{j_1} \land \cdots \land \neg x_{j_k})$ (e.g., [1]).

In practice, as a part of a general branch-and-bound procedure, establishing OSAC or solving the BLP may be prohibitively expensive operations. A weaker consistency notion, called virtual arc consistency (VAC) is introduced in [4]. This consistency can be established using a more efficient algorithm.

Problem 4 Determine a condition on $\Gamma$ that guarantees solvability of VCSP($\Gamma$) using VAC (or some other weaker notion of soft arc consistency).

Références


