Tractable triangles*

Martin C. Cooper[1] and Stanislav Živný[2]

[1] IRIT, University of Toulouse III, 31062 Toulouse, France
[2] University College, University of Oxford, UK
cooper@irit.fr standa.zivny@cs.ox.ac.uk

Abstract. We study the computational complexity of binary valued constraint satisfaction problems (VCSP) given by allowing only certain types of costs in every triangle of variable-value assignments to three distinct variables. We show that for several computational problems, including CSP, Max-CSP, finite-valued VCSP, and general-valued VCSP, the only non-trivial tractable classes are the well known maximum matching problem and the recently discovered joint-winner property [9].

1 Introduction

1.1 Background

An instance of the constraint satisfaction problem (CSP) consists of a collection of variables which must be assigned values subject to specified constraints. Each CSP instance has an underlying undirected graph, known as its *constraint network*, whose vertices are the variables of the instance, and two vertices are adjacent if corresponding variables are related by some constraint. Such a graph is also known as the *structure* of the instance.

An important line of research on the CSP is to identify all tractable cases which are recognisable in polynomial time. Most of this work has been focused on one of the two general approaches: either identifying forms of constraint which are sufficiently restrictive to ensure tractability no matter how they are combined [3,16], or else identifying structural properties of constraint networks which ensure tractability no matter what forms of constraint are imposed [13].

The first approach has led to identifying certain algebraic properties known as polymorphisms [20] which are necessary for a set of constraint types to ensure tractability. A set of constraint types with this property is called a tractable *constraint language*. The second approach has been used to characterise all tractable cases of bounded-arity CSPs (such as binary CSPs): the *only* class of structures which ensures tractability (subject to certain complexity theory assumptions) are structures of *bounded tree-width* [19].

In practice, constraint satisfaction problems usually do not possess a sufficiently restricted structure or use a sufficiently restricted constraint language to

* Martin Cooper is supported by ANR Projects ANR-10-BLAN-0210 and ANR-10-BLAN-0214. Stanislav Živný is supported by a Junior Research Fellowship at University College, Oxford.

fall into any of these tractable classes. Nevertheless, they may still have properties which ensure they can be solved efficiently, but these properties concern both the structure and the form of the constraints. Such properties have sometimes been called *hybrid* reasons for tractability [12,7,6,8].

Since in practice many constraint satisfaction problems are over-constrained, and hence have no solution, *soft* constraint satisfaction problems have been studied [12]. In an instance of the soft CSP, every constraint is associated with a function (rather than a relation as in the CSP) which represents preferences among different partial assignments, and the goal is to find the best assignment. Several very general soft CSP frameworks have been proposed in the literature [29,2]. In this paper we focus on one of the very general frameworks, the *valued* constraint satisfaction problem (VCSP) [29].

Similarly to the CSP, an important line of research on the VCSP is to identify tractable cases which are recognisable in polynomial time. Is is well known that structural reasons for tractability generalise to the VCSP [1,12]. In the case of language restrictions, only a few conditions are known to guarantee tractability of a given set of valued constraints [5,4,21,22].

1.2 Contributions

In this paper, we study hybrid tractability of binary VCSPs for various valuation structures that correspond to the CSP, CSP with soft unary constraints, Max-CSP, finite-valued VCSP and general-valued VCSP.

We focus on classes of instances defined by allowed combinations of binary costs in every assignment to 3 different variables (called a *triangle*). Our motivation for this investigation is that one such restriction, the so-called joint-winner property has recently been shown to define a tractable class with several practical applications [9].

The JWP (joint-winner property) states that for any triangle of variable-value assignments $\{\langle v_i, a\rangle, \langle v_j, b\rangle, \langle v_k, c\rangle\}$, no one of the binary costs $c_{ij}(a,b)$, $c_{jk}(b,c)$, $c_{ik}(a,c)$ is strictly less than the other two. This holds, for example, if there is a (soft) not-equal constraint between each pair of variables (v_i, v_j), (v_j, v_k), (v_i, v_k), by transitivity of equality. In [9] we gave several applications of the JWP in CSPs and VCSPs. For example, the class of CSP instances satisfying the JWP generalises the AllDifferent constraint with arbitrary unary constraints, since its binary constraints are equivalent to allowing at most one assignment from each of a set of disjoint sets of (variable,value) assignments. We also showed how to code a set of non-overlapping SoftAllDifferent constraints with either graph- or variable-based costs as a VCSP satisfying the JWP. As another example, a job-shop scheduling problem in which the aim is to minimise the sum, over all jobs, of their time until completion can be coded as a VCSP satisfying the JWP [9]. The JWP has also been generalised to VCSPs in which the objective function is the sum of hierarchically nested arbitrary convex cost functions [10]: applications include soft hierarchical global cardinality constraints, useful in rostering problems.

For finite valuation structures (corresponding to the CSP and Max-CSP), there are only finitely many possibilities of multi-sets of binary costs in a triangle. For example, in Max-CSP there are only four possible multi-sets of costs, namely $\{0, 0, 0\}$, $\{0, 0, 1\}$, $\{0, 1, 1\}$ and $\{1, 1, 1\}$. However, for infinite valuation structures (corresponding to the finite-valued CSP and general-valued VCSP) there are infinitely many combinations. Obviously, we cannot consider them all, and hence we consider an equivalence relation based on the total order on the valuation structure. There are 4 equivalence classes, thus giving 4 types of combinations of the three binary costs α, β, γ given by $\alpha = \beta = \gamma$, $\alpha = \beta < \gamma$, $\alpha = \beta > \gamma$, $\alpha < \beta < \gamma$.

For all valuation structures we consider, we prove a dichotomy theorem, thus identifying all tractable cases with respect to the equivalence relation on the combinations of costs. It turns out that there are only two non-trivial tractable cases: the well-known maximum weighted matching problem [15], and the recently discovered joint-winner property [9].

The study of the tractability of classes of instances defined by properties on triangles of costs can be seen as a first step on the long road towards the characterisation of tractable classes of VCSPs based on so-called hybrid properties which are not captured by restrictions on the language of cost functions or the structure of the constraint graph. The intractability results in this paper provide initial guidelines for such a research program.

Paper organisation The rest of this paper is organised as follows. We start, in Section 2, with defining valuation structures, binary valued constraint satisfaction problems and cost types. In Section 3, we present our results on the CSP, followed up with results on the CSP with soft unary constraints. In Section 4, we present our results on the Max-CSP, followed by the results on the finite-valued and general-valued VCSP in Section 5. Finally, we conclude in Section 6.

2 Preliminaries

A *valuation structure*, Ω, is a totally ordered set, with a minimum and a maximum element (denoted 0 and ∞), together with a commutative, associative binary aggregation operator (denoted $\oplus$), such that for all $\alpha, \beta, \gamma \in \Omega$, $\alpha \oplus 0 = \alpha$, and $\alpha \oplus \gamma \geq \beta \oplus \gamma$ whenever $\alpha \geq \beta$. Members of Ω are called *costs*.

An instance of the binary *Valued Constraint Satisfaction Problem* (VCSP) is given by n variables $v_1, \ldots, v_n$ over finite domains $D_1, \ldots, D_n$ of values, unary cost functions $c_i : D_i \to \Omega$, and binary cost functions $c_{ij} : D_i \times D_j \to \Omega$ [29]. (If the domains of all the variables are the same, we denote it by D.) The goal is to find an assignment of values from the domains to the variables which minimises the total cost given by

$$\bigoplus_{i=1}^{n} c_i(v_i) \;\oplus \bigoplus_{1 \leq i < j \leq n} c_{ij}(v_i, v_j) \,.$$

Note that we assume that all binary cost functions c_{ij} exist. The absence of any constraint between variables v_i, v_j is modelled by a cost function c_{ij} which is uniformly zero.

We shall denote by $\mathbb{Q}_+$ the set of all non-negative rational numbers. We define $\overline{\mathbb{Q}}_+ = \mathbb{Q}_+ \cup \{\infty\}$. In this paper, we consider the following valuation structures: $\{0, \infty\}$, $\{0, 1\}$, $\mathbb{Q}_+$ and $\overline{\mathbb{Q}}_+$, where in all cases the aggregation operation is the standard addition operation on rationals, $+$, extended so that $a + \infty = \infty$ for all $a \in \overline{\mathbb{Q}}_+$. These valuation structures correspond to CSP, Max-CSP, finite-valued VCSP and general-valued VCSP, respectively.

Given an infinite valuation structure, such as $\mathbb{Q}_+$ or $\overline{\mathbb{Q}}_+$, there is an infinite number of possible sets of triples of costs. Obviously, we cannot consider all such sets. Therefore, we only consider the cases defined by the total order on Ω. We use curly brackets $\{\}$ for multi-sets. The following table defines possible cost types of 3 costs.

Symbol	**Costs**	**Remark**
$\triangle$	$\{\alpha, \beta, \gamma\}$	$\alpha, \beta, \gamma \in \Omega$, $\alpha \neq \beta \neq \gamma \neq \alpha$
$<$	$\{\alpha, \alpha, \beta\}$	$\alpha, \beta \in \Omega$, $\alpha < \beta$
$>$	$\{\alpha, \alpha, \beta\}$	$\alpha, \beta \in \Omega$, $\alpha > \beta$
$=$	$\{\alpha, \alpha, \alpha\}$	$\alpha \in \Omega$

We use the word *triangle* for any set of assignments $\{\langle v_i, a\rangle, \langle v_j, b\rangle, \langle v_k, c\rangle\}$, where v_i, v_j, v_k are distinct variables and $a \in D_i, b \in D_j, c \in D_k$ are domain values. The multi-set of costs in such a triangle is $\{c_{ij}(a, b), c_{ik}(a, c), c_{jk}(b, c)\}$.

We denote by $\mathfrak{D} = \{\triangle, <, >, =\}$ the set of all possible cost types. Let Ω be a fixed valuation structure. For any set $S \subseteq \mathfrak{D}$, we denote by $\mathcal{A}_\Omega(S)$ ($\mathcal{A}$ for allowed) the set of binary VCSP instances with the valuation structure Ω where for every triangle the multi-set of costs in the triangle is of a type from S.

For instance, if $\Omega = \mathbb{Q}_+$ and $S = \{\triangle\}$, then $\mathcal{A}_\Omega(S)$ is the set of binary finite-valued VCSP instances where for every triangle $\{\langle v_i, a\rangle, \langle v_j, b\rangle, \langle v_k, c\rangle\}$ the multi-set of costs in the triangle $\{c_{ij}(a, b), c_{ik}(a, c), c_{jk}(b, c)\}$ contains exactly three distinct costs.

Our goal is to classify the complexity of $\mathcal{A}_\Omega(S)$ for every $S \subseteq \mathfrak{D}$.

Proposition 1. *Let Ω be an arbitrary valuation structure and $S \subseteq \mathfrak{D}$.*

1. *If $\mathcal{A}_\Omega(S)$ is tractable and $S' \subseteq S$, then $\mathcal{A}_\Omega(S')$ is tractable.*
2. *If $\mathcal{A}_\Omega(S)$ is intractable and $S' \supseteq S$, then $\mathcal{A}_\Omega(S')$ is intractable.*

A triangle $\{\langle v_i, a\rangle, \langle v_j, b\rangle, \langle v_k, c\rangle\}$, where $a \in D_i, b \in D_j, c \in D_k$, satisfies the *joint-winner property* (JWP) if either all three $c_{ij}(a, b)$, $c_{ik}(a, c)$, $c_{jk}(b, c)$ are the same, or two of them are equal and the third one is bigger. A VCSP instance satisfies the joint-winner property if every triangle satisfies the joint-winner property.

Theorem 1 ([9]). *The class of VCSP instances satisfying JWP is tractable.*

In [9], we also showed that the class defined by the joint-winner property is maximal – allowing a single extra triple of costs that violates the joint-winner property renders the class NP-hard.

Theorem 2 ([9]). *Let $\alpha < \beta \leq \gamma$, where $\alpha \in \mathbb{Q}_+$ and $\beta, \gamma \in \overline{\mathbb{Q}}_+$, be a multi-set of (not necessarily distinct) costs that do not satisfy the joint-winner property. The class of instances where the costs in each triangle either satisfy the joint-winner property or are $\{\alpha, \beta, \gamma\}$ is NP-hard*

In this paper we consider a much broader question, whether allowing any arbitrary set S of triples of costs in triangles, where S does not necessarily include all triples allowed by the JWP, defines a tractable class of VCSP instances.

Remark 1. We implicitly allow all unary cost functions. In fact, all our tractability results work with unary cost functions, and our NP-hardness results do no require any unary cost functions.

Remark 2. We consider problems with unbounded domains; that is, the domain sizes are part of the input. However, all our NP-hardness results are obtained for problems with a fixed domain size.[3] In the case of CSPs, we need domains of size 3 to prove NP-hardness, and in all other cases domains of size 2 are sufficient to prove NP-hardness. Since binary CSPs are known to be tractable on Boolean domains, and any VCSP is trivially tractable over domains of size 1, all our NP-hardness results are tight.

Remark 3. Binary finite-valued/general-valued VCSPs have also been studied under the name of pair-wise MinSum or pair-wise Markov Random Field (MRF). Consequently, our results readily apply to these frameworks, and other graphical models equivalent to the VCSP.

3 CSP

In this section, we will focus on the valuation structure $\Omega = \{0, \infty\}$; that is, the Constraint Satisfaction Problem (CSP). It is clear that the $\triangle$ cost type cannot occur. Since there are only 2 possible costs, we split the cost type $=$ into two:

Symbol	**Costs**
0	$\{0, 0, 0\}$
∞	$\{\infty, \infty, \infty\}$

The set of possible cost types is then $\mathfrak{D} = \{<, >, 0, \infty\}$. Indeed, these four cost types correspond precisely to the four possible multi-sets of costs: $\{0, 0, 0\}$, $\{0, 0, \infty\}$, $\{0, \infty, \infty\}$ and $\{\infty, \infty, \infty\}$. The dichotomy presented in this section therefore represents a complete characterisation of the complexity of CSPs defined by placing restrictions on triples of costs in triangles.

[3] In other words, the considered problems are not fixed-parameter tractable [14] in the domain size.

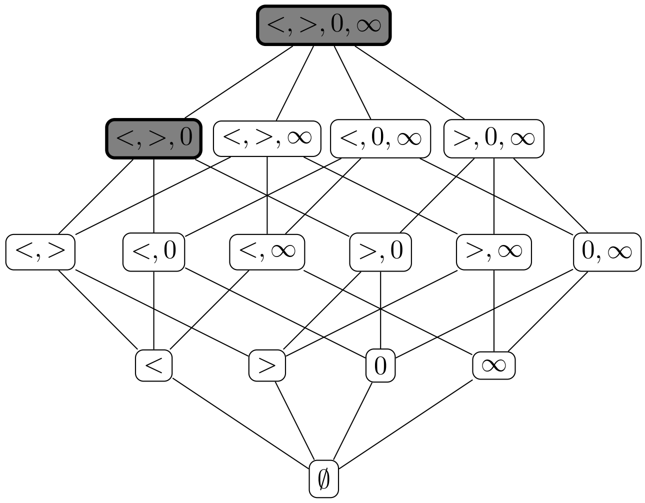

Fig. 1. Complexity of CSPs $\mathcal{A}_{\{0,\infty\}}(S), S \subseteq \{<, >, 0, \infty\}$.

As $\mathcal{A}_{\{0,\infty\}}(\mathfrak{D})$ allows all binary CSPs, $\mathcal{A}_{\{0,\infty\}}(\mathfrak{D})$ is intractable [26] unless the domain is of size at most 2, which is equivalent to 2-SAT, and a well-known tractable class [28].

Proposition 2. *$\mathcal{A}_{\{0,\infty\}}(\mathfrak{D})$ is intractable unless $|D| \leq 2$.*

The joint-winner property for CSPs gives

Corollary 1 (of Theorem 1). *$\mathcal{A}_{\{0,\infty\}}(\{<, 0, \infty\})$ is tractable.*

Proposition 3. *$\mathcal{A}_{\{0,\infty\}}(\{>, 0, \infty\})$ is tractable.*

Proof. Since $<$ is forbidden, if two binary costs in a triangle are zero then the third binary cost must also be zero. In other words, if the assignment $\langle v_1, a_1 \rangle$ is consistent with $\langle v_i, a_i \rangle$ for each $i \in \{2, \dots, n\}$, then for all $i, j \in \{1, \dots, n\}$ such that $i \neq j$, $\langle v_i, a_i \rangle$ is consistent with $\langle v_j, a_j \rangle$. Thus Singleton Arc Consistency, which is a procedure enforcing Arc Consistency for every variable-value pair [27], solves $\mathcal{A}_{\{0,\infty\}}(\{>, 0, \infty\})$. $\square$

Proposition 4. *$\mathcal{A}_{\{0,\infty\}}(\{<, >, \infty\})$ is tractable.*

Proof. This class is trivial: instances with at least three variables have no solution, since the triple of costs $\{0, 0, 0\}$ is not allowed. $\square$

Proposition 5. *$\mathcal{A}_{\{0,\infty\}}(\{<, >, 0\})$ is intractable unless $|D| \leq 2$.*

Proof. It is straightforward to encode the 3-colouring problem as a binary CSP. The result then follows from the fact that 3-colouring is NP-hard for triangle-free graphs, which can be derived from two results from [24]. (Indeed, 3-colouring is NP-hard even for triangle-free graphs of degree at most 4 [25].) The triple of costs $\{\infty, \infty, \infty\}$ cannot occur in the CSP encoding of the colouring of a triangle-free graph. $\square$

Results from this section, together with Proposition 1, complete the complexity classification, as depicted in Figure 1: white nodes represent tractable cases and shaded nodes represent intractable cases.

Theorem 3. *For* $|D| \geq 3$ *a class of binary CSP instances defined as* $\mathcal{A}_{\{0,\infty\}}(S)$*, where* $S \subseteq \{<, >, 0, \infty\}$*, is intractable if and only if* $\{<, >, 0\} \subseteq S$*.*

A simple way to convert classical CSP into an optimisation problem is to allow soft unary constraints. It turns out that the dichotomy given in Theorem 3 remains valid even if soft unary constraints are allowed. We use the notation $\mathcal{A}_{\{0,\infty\}}^{\overline{\mathbb{Q}}_+}(S)$ to represent the set of VCSP instances with binary costs from $\{0, \infty\}$, unary costs from $\overline{\mathbb{Q}}_+$ and whose triples of costs in triangles belong to S. In other words, we now consider VCSPs with crisp binary constraints and soft unary constraints.

Theorem 4. *For* $|D| \geq 3$ *a class of binary CSP instances defined as* $\mathcal{A}_{\{0,\infty\}}^{\overline{\mathbb{Q}}_+}(S)$*, where* $S \subseteq \{<, >, 0, \infty\}$*, is intractable if and only if* $\{<, >, 0\} \subseteq S$*.*

Proof. It suffices to show tractability when S is $\{<, >, \infty\}$, $\{<, 0, \infty\}$ or $\{>, 0, \infty\}$, the three maximal tractable sets in the case of CSP shown in Figure 1, since sets S which are intractable for CSPs clearly remain intractable when soft unary constraints are allowed.

The tractability of $\mathcal{A}_{\{0,\infty\}}^{\overline{\mathbb{Q}}_+}(\{<, 0, \infty\})$ is again a corollary of Theorem 1 since the joint-winner property allows any unary soft constraints.

To solve $\mathcal{A}_{\{0,\infty\}}^{\overline{\mathbb{Q}}_+}(\{>, 0, \infty\})$ in polynomial time, we establish Singleton Arc Consistency in the CSP corresponding to the binary constraints and then loop over all assignments to the first variable. For each assignment a_1 to variable v_1, we can determine the optimal global assignment which is an extension of $\langle v_1, a_1 \rangle$ by simply choosing the assignment a_i for each variable v_i with the least unary cost $c_i(a_i)$ among those assignments $\langle v_i, a_i \rangle$ that are consistent with $\langle v_1, a_1 \rangle$.

As in the proof of Proposition 4, any instance of $\mathcal{A}_{\{0,\infty\}}^{\overline{\mathbb{Q}}_+}(\{<, >, \infty\})$ is tractable, since instances with at least three variables have no solution. $\square$

4 Max-CSP

In this section, we will focus on the valuation structure $\Omega = \{0, 1\}$. It is well known that the VCSP with the valuation structure $\{0, 1\}$ is polynomial-time equivalent to unweighted Max-CSP (no repetition of constraints allowed) [27]. It is clear that the $\triangle$ cost type cannot occur. Since there are only 2 possible costs, we split the cost type $=$ into two:

Symbol	Costs
0	$\{0, 0, 0\}$
1	$\{1, 1, 1\}$

The set of possible cost types is then $\mathfrak{D} = \{<, >, 0, 1\}$. Again, these four costs types correspond precisely to the four possible multi-sets of costs: $\{0, 0, 0\}$, $\{0, 0, 1\}$, $\{0, 1, 1\}$, and $\{1, 1, 1\}$. As for CSP, our dichotomy result for Max-CSP represents a complete characterisation of the complexity of classes of instances defined by placing restrictions on allowed costs in triangles.

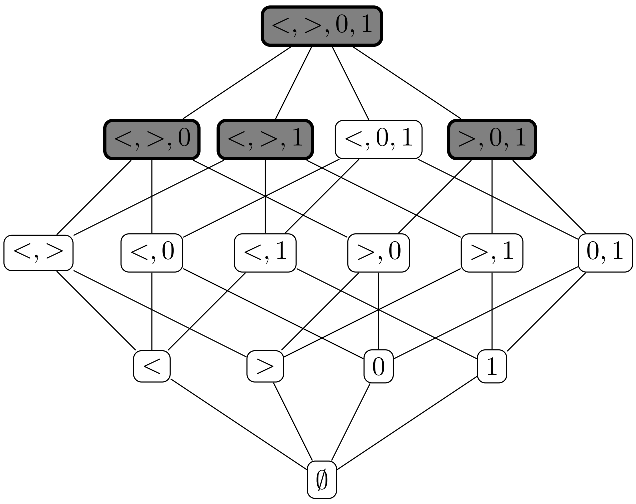

Fig. 2. Complexity of Max-CSPs $\mathcal{A}_{\{0,1\}}(S), S \subseteq \{<, >, 0, 1\}$.

As $\mathcal{A}_{\{0,1\}}(\mathfrak{D})$ allows all binary Max-CSPs, $\mathcal{A}_{\{0,1\}}(\mathfrak{D})$ is intractable [17,26] unless the domain is of size 1.

Proposition 6. *$\mathcal{A}_{\{0,1\}}(\mathfrak{D})$ is intractable unless $|D| \leq 1$.*

The joint-winner property [9] for Max-CSPs gives

Corollary 2 (of Theorem 1). *$\mathcal{A}_{\{0,1\}}(\{<, 0, 1\})$ is tractable.*

Proposition 7. *$\mathcal{A}_{\{0,1\}}(\{<, >\})$ is tractable.*

Proof. We show that $\mathcal{A}_{\{0,1\}}(\{<, >\})$ contains instances on at most 5 variables, thus showing that $\mathcal{A}_{\{0,1\}}(\{<, >\})$ is trivially tractable. Consider an instance of $\mathcal{A}_{\{0,1\}}(\{<, >\})$ on 6 or more variables. Choose 6 arbitrary variables $v_1, \ldots, v_6$ and 6 domain values $d_i \in D_{v_i}$, $1 \leq i \leq 6$. Every cost is either 0 or 1. It is well known [18] and not difficult to show[4] that for every 2-colouring of edges of K_6 (the complete graph on 6 vertices) there is a monochromatic triangle. Therefore, there is a triangle with costs either $\{0,0,0\}$ or $\{1,1,1\}$. But this is a contradiction with the fact that only cost types $<$ (i.e. $\{0,0,1\}$) and $>$ (i.e. $\{1,1,0\}$) are allowed. □

Remark 4. Both $\mathcal{A}_\Omega(\{>\})$ and $\mathcal{A}_\Omega(\{<,>\})$ are tractable over any finite valuation structure Ω due to a similar Ramsey type of argument: given $\Omega = \{0,1,\ldots,K-1\}$, there is $n_0 \in \mathbb{N}$ such that for every graph G on n vertices, where $n \geq n_0$, and every colouring of the edges of G with K colours, there is a monochromatic triangle or an independent set of size 3. Hence there are only finitely many instances, which can be stored in a look-up table. However, once the valuation structure is infinite (e.g. $\mathbb{Q}_+$), both classes become intractable, as shown in the next section.

Proposition 8. *$\mathcal{A}_{\{0,1\}}(\{>,0,1\})$ is intractable unless $|D| \leq 1$.*

Proof. Given an instance of the Max-2SAT problem, we show how to reduce it to a $\{0,1\}$-valued VCSP instance from $\mathcal{A}_{\{0,1\}}(\{>,0,1\})$. The result then follows from the well-known fact that Max-2SAT is NP-hard [17,26]. Recall that an instance of Max-2SAT is given by a set of m clauses of length 2 over n variables $x_1,\ldots,x_n$ and the goal is to find an assignment that maximises the number of clauses that have at least one true literal.

In order to simplify notation, rather than constructing a VCSP instance from $\mathcal{A}_{\{0,1\}}(\{>,0,1\})$ with the goal to minimise the total cost, we construct an instance from $\mathcal{A}_{\{0,1\}}(\{<,0,1\})$ with the goal to maximise the total cost. This implies that the allowed multi-sets of costs in triangles are $\{0,0,1\}$, $\{0,0,0\}$, and $\{1,1,1\}$. Clearly, these two problems are polynomial-time equivalent.

For each variable x_i, we create a large number M of copies x_i^j of x_i with domain $\{0,1\}$, $1 \leq i \leq n$ and $1 \leq j \leq M$. For each variable x_i, the new copies of x_i are pairwise joined by an equality-encouraging cost function h, where $h(x,y) = 1$ if $x = y$ and $h(x,y) = 0$ otherwise. By choosing M very large, we can assume from now on that all copies of x_i will be assigned the same value in all optimal solutions. We can effectively ignore the contribution of these cost functions, which is $K = n\binom{M}{2}$, to the total cost. It is straightforward to check that all triangles involving the new copies of the variables have the allowed costs.

For each clause $(l_1 \vee l_2)$, where l_1 and l_2 are literals, we create a variable z_i with domain $\{l_1,l_2\}$, $1 \leq i \leq m$. For each literal l in the domain of z_k: if l is a positive literal $l = x_i$, we introduce cost function g between z_k and each copy x_i^j of x_i, where $g(l,1) = 1$ and $g(.,.) = 0$ otherwise; if l is a negative literal

[4] Take an arbitrary vertex v in K_6 where every edge is coloured either blue or red. By the pigeonhole principle, v is incident to at least 3 blue or at lest 3 red edges. Without loss of generality, we consider the former case. Let v_1, v_2 and v_3 be the three vertices incident to three blue edges incident to v. If an any of the edges $\{v_1,v_2\}$, $\{v_1,v_3\}$, $\{v_2,v_3\}$ is blue, we have a blue triangle. If all three edges are red, we have a red triangle.

$l = \neg x_i$, we introduce cost function g' between z_k and each copy x_i^j of x_i, where $g'(l, 0) = 1$ and $g'(.,.) = 0$ otherwise.

To make sure that the only multi-sets of costs in all triangles are $\{0, 0, 1\}$, $\{0, 0, 0\}$, and $\{1, 1, 1\}$, we also add cost functions f between the different clause variables z_k and $z_{k'}$ involving the same literal l, where $f(l, l) = 1$ and $f(.,.) = 0$ otherwise. The contribution of all the cost functions between z_k and $z_{k'}$, $1 \leq k \neq k' \leq m$, is less than M and hence of no importance for M very large.

Answering the question of whether the resulting VCSP instance has a solution with a cost $\geq K + pM$ is equivalent to determining whether the original Max-2SAT instance has a solution satisfying at least p clauses. This is because each clause variable z_k can only add a score $\geq M$ if we assign value l to z_k for some literal l which is assigned true. $\square$

Proposition 9. *Both* $\mathcal{A}_{\{0,1\}}(\{<, >, 0\})$ *and* $\mathcal{A}_{\{0,1\}}(\{<, >, 1\})$ *are intractable unless* $|D| \leq 1$.

Proof. We present a reduction from Max-Cut, a well-known NP-hard problem [17], which is NP-hard even on triangle-free graphs [23]. An instance of Max-Cut can easily be modelled as a Boolean $\{0, 1\}$-valued VCSP instance: every vertex of the graph is represented by a variable with the Boolean domain $\{0, 1\}$, and every edge yields cost function f, where $f(x, y) = 1$ if $x = y$ and $f(x, y) = 0$ if $x \neq y$. Observe that since the original graph is triangle-free, there cannot be a triangle with costs $\{1, 1, 1\}$. Therefore, the constructed instance belongs to $\mathcal{A}_{\{0,1\}}(\{<, >, 0\})$.

For the $\mathcal{A}_{\{0,1\}}(\{<, >, 1\})$ case, instead of minimising the total cost, we maximise the total cost for instances from $\mathcal{A}_{\{0,1\}}(\{<, >, 0\})$. Again, we model an instance of the Max-Cut problem using Boolean variables, and every edge yields a cost function g, where $g(x, y) = 0$ if $x = y$ and $g(x, y) = 1$ if $x \neq y$ (where in this case the aim is to maximise the total cost). The constructed instance belongs to $\mathcal{A}_{\{0,1\}}(\{<, >, 0\})$ when the original graph is triangle-free. The result then follows from the fact that Max-Cut is NP-complete on triangle-free graphs [23]. $\square$

Proposition 10. $\mathcal{A}_{\{0,1\}}(\{>, 0\})$ *is tractable.*

Proof. Let I be an instance from $\mathcal{A}_{\{0,1\}}(\{>, 0\})$. The algorithm loops through all possible assignments $\{\langle v_1, a_1 \rangle, \langle v_2, a_2 \rangle\}$ to the first two variables. Suppose that $c_{12}(a_1, a_2) = 1$ (the case $c_{12}(a_1, a_2) = 0$ is similar). Observe that the possible variable-value assignments to other variables $\{\langle v_i, b \rangle \mid 3 \leq i \leq n, b \in D_i\}$ can be uniquely split in two sets L and R such that: (1) for every $\langle v_i, b \rangle \in L$, $c_{1i}(a_1, b) = 1$ and $c_{2i}(a_2, b) = 0$; for every $\langle v_i, b \rangle, \langle v_j, c \rangle \in L$, $c_{ij}(b, c) = 0$; (2) for every $\langle v_i, b \rangle \in R$, $c_{1i}(a_1, b) = 0$ and $c_{2i}(a_2, b) = 1$; for every $\langle v_i, b \rangle, \langle v_j, c \rangle \in R$, $c_{ij}(b, c) = 0$; (3) for every $\langle v_i, b \rangle \in L$ and $\langle v_j, c \rangle \in R$, $c_{ij}(b, c) = 1$. Ignoring unary cost functions for a moment, to find an optimal assignment to the remaining $n-2$ variables, one has to decide how many variables v_i, $3 \leq i \leq n$, will be assigned a value $b \in D_i$ such that $\langle v_i, b \rangle \in L$. The cost of a global assignment involving k variable-value assignments from L is $1 + k + (n - 2 - k) + k(n - 2 - k) = n - 1 + k(n - 2 - k)$. For some variables v_i it could happen that $\langle v_i, b \rangle \in L$ for all $b \in D_i$ or $\langle v_i, c\rangle \in R$ for all $c \in D_i$. If this is the case, then we choose an arbitrary value b for x_i with minimum unary cost $c_i(b)$. This is an optimal choice whatever the assignments to the variables x_j ($j \in \{3, \ldots, i-1, i+1, \ldots, n\}$).

Assuming that all such variables have been eliminated and now taking into account unary cost functions, the function to minimise is given by the objective function (in which we drop the constant term $n-1$):

$$(\sum x_i)(n-2-\sum x_i) \;+\; \sum w_i^L x_i \;+\; \sum w_i^R (1-x_i)$$

(each sum being over $i \in \{3, \ldots, n\}$), where $x_i \in \{0,1\}$ indicates whether v_i is assigned a value from R or L, $w_i^L = \min\{c_i(b) : b \in D_i \wedge \langle v_i, b\rangle \in L\}$, and similarly $w_i^R = \min\{c_i(c) : c \in D_i \wedge \langle v_i, c\rangle \in R\}$. The objective function is thus equal to $k(n-2-k) + \sum w_i^L x_i + \sum w_i^R(1-x_i)$, where, as above, $k = \sum x_i$ is the number of assignments from L. This objective function is minimised either when $k=0$ or when $k = n-2$. This follows from the fact that the contribution of unary cost functions to the objective function is $\sum w_i^L x_i + \sum w_i^R(1-x_i)$ which is at most $n-2$ (since in Max-CSP all unary costs belong to $\{0,1\}$). This is no greater than the value of the quadratic term $k(n-2-k)$ for all values of k in $\{1, \ldots, n-3\}$, i.e. not equal to 0 or $n-2$.

The optimal assignment which involves $k=0$ (respectively $k = n-2$) assignments from L is obtained by simply choosing each value a_i (for $i > 2$) with minimum unary cost among all assignments $\langle v_i, a_i\rangle \in R$ (respectively L).

In the case that $c_{12}(a_1, a_2) = 0$, a similar argument shows that the quadratic term in the objective function is now $2(n-2-k)+k(n-2-k) = (k+2)(n-2-k)$. This is always minimised by setting $k = n-2$ and again the sum of the unary costs is no greater than the value of the quadratic term for other values of $k \neq n-2$. The optimal assignment which involves all $k = n-2$ assignments from L is obtained by simply choosing each value a_i (for $i > 2$) with minimum unary cost among all assignments $\langle v_i, a_i\rangle \in L$. □

Proposition 11. $\mathcal{A}_{\{0,1\}}(\{>, 1\})$ *is tractable.*

Proof. Let I be an instance from $\mathcal{A}_{\{0,1\}}(\{>, 1\})$ without any unary constraints; i.e. all constraints are binary. Observe that every variable-value assignment $\langle v_i, a\rangle$, where $a \in D_i$, is included in zero-cost assignment-pairs involving at most one other variable; i.e. there is at most one variable v_j, such that $c_{ij}(a, b) = 0$ for some $b \in D_j$. In order to minimise the total cost, we have to maximise the number of zero-cost assignment-pairs. In a global assignment, no two zero-cost assignment-pairs can involve the same variable, which means that this can be achieved by a reduction to the maximum matching problem, a problem solvable in polynomial time [15]. We build a graph with vertices given by the variables of I, and there is an edge $\{v_i, v_j\}$ if and only if there is $a \in D_i$ and $b \in D_j$ such that $c_{ij}(a, b) = 0$.

To complete the proof, we show that unary constraints do not make the problem more difficult to solve; it suffices to perform a preprocessing step before the reduction to maximum matching. Let v_i be an arbitrary variable of I. If

$c_i(a) = 1$ for all $a \in D_i$, then we can effectively ignore the unary cost function c_i since it simply adds a cost of 1 to any solution. Otherwise, we show that all $a \in D_i$ such that $c_i(a) = 1$ can be ignored. Take an arbitrary assignment s to all variables such that $s(v_i) = a$, where $c_i(a) = 1$. Now take any $b \in D_i$ such that $c_i(b) = 0$. We claim that assignment s' defined by $s'(v_i) = b$ and $s'(v_j) = s(v_j)$ for every $j \neq i$ does not increase the total cost compared with s. Since the assignment $\langle v_i, a\rangle$ can occur in at most one zero-cost assignment-pair, there are two cases to consider: (1) if there is no $\langle v_j, c\rangle$ with $s(v_j) = c$ such that $c_{ij}(a, c) = 0$, then the claim holds since $c_i(a) = 1$ and $c_i(b) = 0$, so the overall cost can only decrease if we replace a by b; (2) if there is exactly one $j \neq i$ such that $c_{ij}(a, c) = 0$ and $s(v_j) = c$, then again the cost of s' cannot increase because the possible increase of cost by 1 in assigning b to v_i is compensated by the unary cost function c_i. Therefore, before using the reduction to maximal matching, we can remove all $a \in D_i$ such that $c_i(a) = 1$ and keep only those $a \in D_i$ such that $c_i(a) = 0$. ◻

Results from this section, together with Proposition 1, complete the complexity classification, as depicted in Figure 2: white nodes represent tractable cases and shaded nodes represent intractable cases.

Theorem 5. *For $|D| \geq 2$ a class of binary unweighted Max-CSP instances defined as $\mathcal{A}_{\{0,1\}}(S)$, where $S \subseteq \{<, >, 0, 1\}$, is intractable if and only if either $\{<, >, 0\} \subseteq S$, $\{<, >, 1\} \subseteq S$, or $\{>, 0, 1\} \subseteq S$.*

5 VCSP

In this section, we will focus on finite-valued and general-valued VCSP. First, we focus on the valuation structure $\Omega = \mathbb{Q}_+$; that is, the finite-valued VCSP.

The set of possible cost types is $\mathfrak{D} = \{\triangle, <, >, =\}$. As $\mathcal{A}_{\mathbb{Q}_+}(\mathfrak{D})$ allows all finite-valued VCSPs, $\mathcal{A}_{\mathbb{Q}_+}(\mathfrak{D})$ is intractable [5] as it includes the Max-SAT problem for the exclusive or predicate [11].

Proposition 12. *$\mathcal{A}_{\mathbb{Q}_+}(\mathfrak{D})$ is intractable unless $|D| \leq 1$.*

The joint-winner property [9] for finite-valued VCSPs gives

Corollary 3 (of Theorem 1). *$\mathcal{A}_{\mathbb{Q}_+}(\{<, =\})$ is tractable.*

Proposition 13. *$\mathcal{A}_{\mathbb{Q}_+}(\{\triangle\})$ is intractable unless $|D| \leq 1$.*

Proof. We show a reduction from Max-Cut, a well-known NP-hard problem [17]. An instance of Max-Cut can be easily modelled as a Boolean finite-valued VCSP instance: every vertex of the graph is represented by a variable with the Boolean domain $\{0, 1\}$, and every edge yields cost function f, where $f(x, y) = 1$ if $x = y$ and $f(x, y) = 0$ if $x \neq y$. However, the constructed instance does not belong to $\mathcal{A}_{\mathbb{Q}_+}(\{\triangle\})$. Nevertheless, we can amend the VCSP instance by infinitesimal perturbations: all occurrences of the cost 0 are replaced by different numbers that

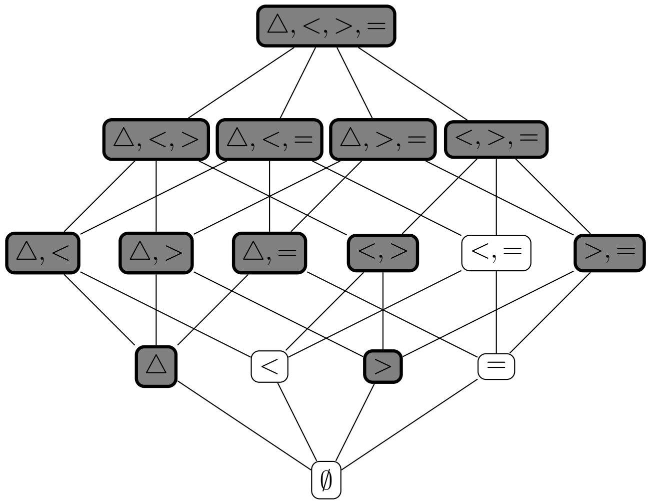

Fig. 3. Complexity of finite-valued VCSPs $\mathcal{A}_{\mathbb{Q}_+}(S), S \subseteq \{\triangle, <, >, =\}$.

are very close to 0, and all occurrences of the cost 1 are replaced by different numbers very close to 1. Now since all the cost are different, clearly the instance belongs to $\mathcal{A}_{\mathbb{Q}_+}(\{\triangle\})$. □

Proposition 14. *$\mathcal{A}_{\mathbb{Q}_+}(\{>\})$ is intractable unless $|D| \leq 1$.*

Proof. We prove this by a perturbation of the construction in the proof of Proposition 8, which shows intractability of $\mathcal{A}_{\mathbb{Q}_+}(\{>, =\})$. In order to simplify the proof, similarly to the proof of Proposition 8, we prove that maximising the total cost in the class $\mathcal{A}_{\mathbb{Q}_+}(\{<\})$ is NP-hard.

In the construction in the proof of Proposition 8 we add $i\epsilon$ to each binary cost $c_{ij}(a, b)$, where $i < j$, if $c_{ij}(a, b)$ was equal to 1. We assume that ϵ is very small ($n\epsilon < 1$). This simply ensures that each triple of costs $\{1, 1, 1\}$ in a triangle of assignments is now perturbed to become $\{1 + i\epsilon, 1 + i\epsilon, 1 + j\epsilon\}$.

In the reduction from Max-2SAT, for each literal l, let C_l be the set of all variable-value assignments corresponding to l (in both the x_i^j and the z_k variables). Recall that all binary costs for pairs of the assignments within C_l were 1 and all binary costs for pairs of the assignments from distinct $C_l, C_{l'}$ were all 0 in the VCSP encoding of the Max-2SAT instance. We place an arbitrary ordering on the literals $l_1 < l_2 < \cdots < l_r$. We then add $i\epsilon$ to each binary cost between two variable-value assignments whenever these assignments correspond to literals l_i, l_j with $i < j$. This simply ensures that each triple of costs $\{0, 0, 0\}$ in a triangle of assignments is now perturbed to become $\{0 + i\epsilon, 0 + i\epsilon, 0 + j\epsilon\}$.

The resulting VCSP instance is in $\mathcal{A}_{\mathbb{Q}_+}(\{>\})$ and correctly codes the original Max-2SAT instance for sufficiently small ϵ. □

Results from this section, together with Proposition 1, complete the complexity classification, as depicted in Figure 3: white nodes represent tractable cases and shaded nodes represent intractable cases.

Theorem 6. *For $|D| \geq 2$ a class of binary finite-valued VCSP instances defined as $\mathcal{A}_{\mathbb{Q}_+}(S)$, where $S \subseteq \{\triangle, <, >, =\}$, is tractable if and only if $S \subseteq \{<, =\}$.*

We now consider the case of general-valued VCSPs. In other words, we consider the valuation structure $\Omega = \overline{\mathbb{Q}}_+$. Theorem 6 applies to this valuation structure as well. Indeed, the hard cases remain intractable when we allow more triangles (involving infinite costs), and the only tractable case, $\mathcal{A}_{\mathbb{Q}_+}(\{<, =\})$, remains tractable: $\mathcal{A}_{\overline{\mathbb{Q}}_+}(\{<, =\})$ is tractable by Theorem 1.

Theorem 7. *For $|D| \geq 2$ a class of binary general-valued VCSP instances defined as $\mathcal{A}_{\overline{\mathbb{Q}}_+}(S)$, where $S \subseteq \{\triangle, <, >, =\}$, is tractable if and only if $S \subseteq \{<, =\}$.*

6 Conclusions

In the CSP and Max-CSP case, we have obtained a complete dichotomy concerning the tractability of problems defined by placing restrictions on the possible combinations of binary costs in triangles of variable-value assignments. In the case of finite-valued and general-valued VCSP, we have obtained a complete dichotomy with respect to the equivalence classes which naturally follow from the total order on the valuation structure. In particular, we have shown that the joint-winner property is the only tractable class for finite-valued and general-valued VCSPs.

References

1. Bertelé, U., Brioshi, F.: Nonserial dynamic programming. Academic Press (1972)
2. Bistarelli, S., Montanari, U., Rossi, F.: Semiring-based Constraint Satisfaction and Optimisation. Journal of the ACM 44(2), 201–236 (1997)
3. Bulatov, A., Krokhin, A., Jeavons, P.: Classifying the Complexity of Constraints using Finite Algebras. SIAM Journal on Computing 34(3), 720–742 (2005)
4. Cohen, D.A., Cooper, M.C., Jeavons, P.G.: Generalising submodularity and Horn clauses: Tractable optimization problems defined by tournament pair multimorphisms. Theoretical Computer Science 401(1-3), 36–51 (2008)
5. Cohen, D.A., Cooper, M.C., Jeavons, P.G., Krokhin, A.A.: The Complexity of Soft Constraint Satisfaction. Artificial Intelligence 170(11), 983–1016 (2006)
6. Cohen, D., Jeavons, P.: The complexity of constraint languages. In: Rossi, F., van Beek, P., Walsh, T. (eds.) The Handbook of Constraint Programming. Elsevier (2006)
7. Cohen, D.A.: A New Class of Binary CSPs for which Arc-Constistency Is a Decision Procedure. In: Proceedings of the 9th International Conference on Principles and Practice of Constraint Programming (CP'03). Lecture Notes in Computer Science, vol. 2833, pp. 807–811. Springer (2003)

8. Cooper, M.C., Jeavons, P.G., Salamon, A.Z.: Generalizing constraint satisfaction on trees: hybrid tractability and variable elimination. Artificial Intelligence 174(9–10), 570–584 (2010)
9. Cooper, M.C., Živný, S.: Hybrid tractability of valued constraint problems. Artificial Intelligence 175(9-10), 1555–1569 (2011)
10. Cooper, M.C., Živný, S.: Hierarchically nested convex VCSP. In: Proceedings of the 17th International Conference on Principles and Practice of Constraint Programming (CP'11). Lecture Notes in Computer Science, Springer (2011)
11. Creignou, N., Khanna, S., Sudan, M.: Complexity Classification of Boolean Constraint Satisfaction Problems, SIAM Monographs on Discrete Mathematics and Applications, vol. 7. SIAM (2001)
12. Dechter, R.: Constraint Processing. Morgan Kaufmann (2003)
13. Dechter, R., Pearl, J.: Network-based Heuristics for Constraint Satisfaction Problems. Artificial Intelligence 34(1), 1–38 (1988)
14. Downey, R., Fellows, M.: Parametrized Complexity. Springer (1999)
15. Edmonds, J.: Paths, trees, and flowers. Canadian Journal of Mathematics 17, 449–467 (1965)
16. Feder, T., Vardi, M.: 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)
17. Garey, M., Johnson, D.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman (1979)
18. Goodman, A.W.: On Sets of Acquaintances and Strangers at any Party. The American Mathematical Monthly 66(9), 778–783 (1959)
19. Grohe, M.: The complexity of homomorphism and constraint satisfaction problems seen from the other side. Journal of the ACM 54(1), 1–24 (2007)
20. Jeavons, P.: On the Algebraic Structure of Combinatorial Problems. Theoretical Computer Science 200(1-2), 185–204 (1998)
21. Kolmogorov, V., Živný, S.: Generalising tractable VCSPs defined by symmetric tournament pair multimorphisms. Tech. rep. (August 2010)
22. Kolmogorov, V., Živný, S.: The complexity of conservative VCSPs, submitted for publication (2011).
23. Lewis, J.M., Yannakakis, M.: The node-deletion problem for hereditary properties is NP-complete. Journal of Computer System Sciences 20(2), 219–230 (1980)
24. Lovász, L.: Coverings and colorings of hypergraphs. In: Proceedings of the 4th Southeastern Conference on Combinatorics, Graph Theory and Computing. pp. 3–12 (1973)
25. Maffray, F., Preissmann, M.: On the NP-completeness of the k-colorability problem for triangle-free graphs. Discrete Mathematics 162(1-3), 313–317 (1996)
26. Papadimitriou, C.: Computational Complexity. Addison-Wesley (1994)
27. Rossi, F., van Beek, P., Walsh, T. (eds.): The Handbook of Constraint Programming. Elsevier (2006)
28. Schaefer, T.: The Complexity of Satisfiability Problems. In: Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC'78). pp. 216–226 (1978)
29. Schiex, T., Fargier, H., Verfaillie, G.: Valued Constraint Satisfaction Problems: Hard and Easy Problems. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95) (1995)