Generalising Submodularity and Horn Clauses: Tractable Optimization Problems Defined by Tournament Pair Multimorphisms

David A. Cohen,
Dept. of Computer Science, Royal Holloway,
University of London, UK
dave@cs.rhul.ac.uk

Martin C. Cooper,
IRIT, University of Toulouse III, 31062 Toulouse, France
cooper@irit.fr

Peter G. Jeavons,
Computing Laboratory, University of Oxford, UK
Peter.Jeavons@comlab.ox.ac.uk

Abstract

The *submodular function minimization problem* (SFM) is a fundamental problem in combinatorial optimization and several fully combinatorial polynomial-time algorithms have recently been discovered to solve this problem. The most general versions of these algorithms are able to minimize any submodular function whose domain is a set of tuples over any totally-ordered finite set and whose range includes both finite and infinite values.

In this paper we demonstrate that this general form of SFM is just one example of a much larger class of tractable discrete optimization problems defined by valued constraints. These tractable problems are characterized by the fact that their valued constraints have an algebraic property which we call a *tournament pair multimorphism*. This larger tractable class also includes the problem of satisfying a set of Horn clauses (HORN-SAT), as well as various extensions of this problem to larger finite domains.

Keywords: discrete optimization, constraint satisfaction problem, valued constraint satisfaction, tractability, submodularity, tournament opera-

tion, majority operation, modular decomposition.

1 Introduction

In this paper we study a generic discrete optimization problem known as the **valued constraint satisfaction problem** (VCSP) [49]. This problem generalises the standard constraint satisfaction problem [22] by allowing different costs to be associated with different solutions. It provides a very general framework which includes many standard combinatorial optimisation problems as special cases, including MAX-SAT [19], MAX-CSP [11], MIN-ONES SAT [19], and MIN-COST HOMOMORPHISM [29].

The complexity of the VCSP depends on the types of valued constraints which are allowed. For certain types of valued constraints an optimal solution can be obtained in polynomial time; such constraints are called *tractable* valued constraints.

In the special case where each variable has just 2 possible values, a complete characterization has been obtained of all tractable classes of valued constraints with positive real-valued or infinite costs [8, 12]. This result extends the earlier characterizations of the tractable classes for the SAT [48] and MAX-SAT [19] problems.

Over larger sets of possible values a complete characterization of the tractable cases is not yet known, but a number of examples have been identified. Two important classes of tractable valued constraints are **submodular functions** (see Example 3.7) and **Horn clauses** (see Example 2.4). In this paper we show that these two examples are members of a large family of tractable valued constraint classes which can be treated in a uniform way. To obtain this generalisation, we introduce a class of operations known as **tournament operations**, and show that any set of valued constraints associated with an arbitrary pair of tournament operations defines a tractable optimization problem.

The paper is organised as follows. In Section 2 we define the standard constraint satisfaction problem, and in Section 3 we extend this definition to the more general framework of the valued constraint satisfaction problem and define the notion of a multimorphism. In Section 4 we consider multimorphisms defined by special kinds of operations known as tournament operations. In Section 5 we consider the set of all feasible assignments to a valued constraint satisfaction problem, and the set of all optimal assignments, and show that in certain cases these sets can be efficiently represented. In Section 6 we begin a more detailed examination of tournament operations

by considering decompositions of the associated tournament graphs, and in Section 7 we examine the structure of valued constraints which have a tournament pair multimorphism. Using these results we show in Section 8 that all such valued constraints give rise to tractable optimisation problems, and then in Section 9 we give some examples. Finally, in Section 10 we suggest some directions for future research.

2 Constraints and polymorphisms

In this section we present the terminology and notation used to describe the standard constraint satisfaction problem (CSP) and discuss the techniques which have been used to identify tractable cases. In Section 3 we extend these ideas to the *valued* constraint satisfaction problem.

Definition 2.1 *An instance of the* ***constraint satisfaction problem,*** CSP*, is a tuple* $\mathcal{P} = \langle V, D, C \rangle$ *where:*

- V *is a finite set of* ***variables****;*
- D *is a finite set of possible* ***values****;*
- C *is a set of* ***constraints****. Each element of* C *is a pair* $c = \langle \sigma, R \rangle$ *where* σ *is a tuple of variables called the* ***scope*** *of* c*, and* R *is a relation over* D *of arity* $|\sigma|$ *called the* ***constraint relation*** *of* c*.*

Definition 2.2 *For any* CSP *instance* $\mathcal{P} = \langle V, D, C \rangle$*, an* ***assignment*** *for* $\mathcal{P}$ *is a mapping* $s : V \to D$*.*

A ***solution*** *to* $\mathcal{P}$ *is an assignment* s *which satisfies all of the constraints. That is, for each* $\langle \sigma, R \rangle \in C$*, where* $\sigma = \langle v_1, v_2, \ldots, v_r \rangle$*, the tuple* $\langle s(v_1), s(v_2), \ldots, s(v_r) \rangle \in R$*.*

Example 2.3 The standard problem of colouring the vertices of a graph G with k colours so that adjacent vertices are assigned different colours can be viewed as a special case of the CSP, where the constraint relation of each constraint is the binary disequality relation, $R_{\neq}$, given by

$$R_{\neq} = \{\langle a, b \rangle \in D^2 \mid a \neq b\}.$$

For any given graph $\langle V, E \rangle$, we have the corresponding CSP instance $\langle V, D, C \rangle$, where $D = \{1, 2, \ldots, k\}$ and $C = \{\langle \langle v_i, v_j \rangle, R_{\neq} \rangle \mid \{v_i, v_j\} \in E\}$.

This problem is well-known to be NP-complete when $k \geq 3$. □

Example 2.4 The propositional satisfiability problem for Horn clauses, HORN-SAT, can be viewed as a special case of the CSP, where the constraint relations are relations over a 2-element set which are specified by Horn clauses. Such relations describe the possible satisfying assignments for a particular Horn clause; for example, the relation

$$R_{\neg x \vee \neg y \vee z} = \{\langle 0,0,0\rangle, \langle 0,0,1\rangle, \langle 0,1,0\rangle, \langle 0,1,1\rangle, \langle 1,0,0\rangle, \langle 1,0,1\rangle, \langle 1,1,1\rangle\}$$

describes the satisfying assignments for the Horn clause $\neg x \vee \neg y \vee z$, where the value 0 corresponds to *false* and the value 1 corresponds to *true*.

The problem of satisfying any set of Horn clauses can be solved in linear time [23]. □

If Γ is a set of relations over some fixed set D, we will write CSP(Γ) to denote the class of all CSP instances where the constraint relations of all constraints lie in Γ.

For certain sets of relations Γ the problem CSP(Γ) is NP-complete. (For example, the set $\{R_{\neq}\}$, where $R_{\neq}$ is the disequality relation over some set D with $|D| \geq 3$, as defined in Example 2.3.) For other sets of relations Γ the problem CSP(Γ) can be solved in polynomial time. (For example, the set of all relations specified by Horn clauses, as defined in Example 2.4.)

A finite set of relations Γ will be called **tractable** if there exists a polynomial-time algorithm to solve CSP(Γ). An infinite set of relations Γ will be called tractable if all finite subsets of Γ are tractable.

Many new tractable sets of relations have been identified by investigating certain invariance properties of relations, known as polymorphisms [7, 25, 37].

Definition 2.5 *A function $f : D^m \to D$ is a **polymorphism** of a relation $R \subseteq D^r$ if for all $\langle a_{11}, \ldots, a_{1r}\rangle, \ldots, \langle a_{m1}, \ldots, a_{mr}\rangle \in R$, we also have*

$$\langle f(a_{11}, \ldots, a_{m1}), \ldots, f(a_{1r}, \ldots, a_{mr})\rangle \in R.$$

If a relation R has a polymorphism f, then we will say that R is **preserved by** f.

Example 2.6 The relations over the 2-element domain $\{0,1\}$ which are specified by Horn clauses are precisely the relations having the polymorphism $\min : \{0,1\}^2 \to \{0,1\}$, which returns the minimum of its 2 arguments.

For example, if we take any 2 tuples from the relation $R_{\neg x \vee \neg y \vee z}$ defined in Example 2.4, (such as $\langle 0,1,1\rangle$ and $\langle 1,0,1\rangle$), and apply the operation

min co-ordinatewise, then we obtain a new tuple, $(\langle 0,0,1\rangle)$, which is also a member of this relation.

A binary operation, min, which returns the minimum of its two arguments, can be defined on any finite totally-ordered set D of arbitrary size. Hence, for each such D there is an obvious generalisation to the set, $\Gamma_{\min}$, consisting of all relations over D which are preserved by the operation min. It has been shown [38] that $\mathrm{CSP}(\Gamma_{\min})$ is tractable for all finite sets D. □

Many other tractable sets of relations have been identified, or extended, thanks to the study of polymorphisms [3, 4, 6, 7, 37]. In fact, it is known that the existence of a non-trivial polymorphism of a set of relations Γ is a necessary condition for tractability of $\mathrm{CSP}(\Gamma)$ [35].

Definition 2.7 *A **majority operation** is a function $f : D^3 \to D$ satisfying*

$$\forall x, y \in D, \quad f(x,x,y) = f(x,y,x) = f(y,x,x) = x$$

It has been shown that having a majority operation as a polymorphism is a sufficient condition for tractability of a set of relations [25, 36, 37]. However, it is the following more specific property of relations preserved by a majority operation which is of more interest to us in this paper.

Definition 2.8 *The **projection** of a relation R of arity r onto a pair of positions i and j, which we denote by $\Pi_{ij}R$, is the binary relation containing all pairs that can be extended to elements of R. That is,*

$$\Pi_{ij}R \stackrel{\text{def}}{=} \{\langle x_i, x_j\rangle \mid \exists \langle x_1, \ldots, x_r\rangle \in R\}.$$

*A relation R of arity r is said to be **decomposable into its binary projections** if*

$$R = \{\langle x_1, \ldots, x_r\rangle \in D^r \mid \forall i,j \in \{1,\ldots,r\},\ \langle x_i, x_j\rangle \in \Pi_{ij}R\}.$$

Lemma 2.9 ([36]) *Any relation which is preserved by a majority operation is decomposable into its binary projections.*

Finally, we will occasionally make use of the following standard definitions from the field of constraint satisfaction [22].

Definition 2.10 *A partial assignment to a subset W of the variables of a* CSP *instance is **consistent** if it satisfies all the constraints whose scopes are contained in W.*

Definition 2.11 *A* CSP *instance is* ***k-consistent*** *if, for every subset W of $k-1$ variables and any other variable $v \notin W$, every consistent partial assignment to W can be extended to a consistent partial assignment to $W \cup \{v\}$.*

Definition 2.12 *A* CSP *instance is* ***strong k-consistent*** *if it is j-consistent for all $j \leq k$.*

3 Valued constraints and multimorphisms

In the constraint satisfaction problem, the aim is simply to find an assignment to the variables which satisfies all of the constraints. In other words, standard constraint satisfaction problems deal with *feasibility* rather than *optimization*. To provide a more general framework, the notion of an all-or-nothing constraint relation can be extended to the notion of a **cost function** which assigns a specified cost to each possible assignment. We use $\overline{\mathbb{R}}_+$ to denote $\{u \in \mathbb{R} : u \geq 0\} \cup \{\infty\}$.

Definition 3.1 *For any set D, an order-r* ***cost function*** *on D is a function $\phi : D^r \to \overline{\mathbb{R}}_+$ which assigns a* ***cost*** *$\phi(a_1, \ldots, a_r)$ to each combination of values $a_1, \ldots, a_r \in D^r$.*

Definition 3.2 *A cost function ϕ is said to be* ***crisp*** *if $\phi(x_1, \ldots, x_r) \in \{0, \infty\}$ for all choices of $\langle x_1, \ldots, x_r \rangle$.*

A constraint relation can be modelled by a crisp cost function which assigns a cost of 0 to permitted assignments and a cost of ∞ to disallowed assignments.

Definition 3.3 *An instance of the* ***valued constraint satisfaction problem,*** VCSP*, is a tuple $\mathcal{P} = \langle V, D, C \rangle$ where:*

- *V is a finite set of* ***variables****;*
- *D is a finite set of possible* ***values****;*
- *C is a set of* ***valued constraints****. Each element of C is a pair $c = \langle \sigma, \phi \rangle$ where σ is a tuple of variables called the* ***scope*** *of c, and ϕ is a mapping from $D^{|\sigma|}$ to $\overline{\mathbb{R}}_+$, called the* ***cost function*** *of c.*

In the original, more general, definition of the Valued Constraint Satisfaction Problem [49], costs were allowed to lie in any positive totally-ordered monoid S. Under the additional assumptions of discreteness and the existence of a partial inverse operation, it has been shown [16] that such a structure S can be decomposed into independent positive totally-ordered monoids, each of which is isomorphic to a subset of $\overline{\mathbb{R}}_+$ with the operator being either standard addition, $+$, or bounded addition, $+_k$, where $a +_k b = \min\{k, a + b\}$. The latter case is of some interest, because it can be used to model the process of branch and bound search (k being the cost of the best solution found so far) [42]. However, for the purposes of this paper we shall restrict attention to the standard case studied in Mathematical Programming where all costs lie in $\overline{\mathbb{R}}_+$ and are combined using standard addition.

Definition 3.4 *For any* VCSP *instance* $\mathcal{P} = \langle V, D, C\rangle$*, 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.*

Example 3.5 We can encode the search for a minimum cut in a weighted directed graph G as a VCSP instance $\mathcal{P}$ with a variable for each node of G, domain $\{0, 1\}$, and a valued constraint $\langle\langle i, j\rangle, \chi^{w_{ij}}\rangle$ for each directed edge $\langle i, j\rangle$ of weight w_{ij} in G, where

$$\chi^w(x, y) \;=\; \begin{cases} w & \text{if } (x, y) = (0, 1) \\ 0 & \text{otherwise} \end{cases}$$

If we impose unary constraints on the source and target nodes to ensure that they take the values $\{0\}$ and $\{1\}$, respectively, then any minimum cut in G corresponds to the set of directed edges $\langle i, j\rangle$ whose corresponding variables are labeled $(0, 1)$ in some solution to $\mathcal{P}$. □

If Γ is a set of cost functions $\phi : D^r \to \overline{\mathbb{R}}_+$, for some fixed set D, we will write VCSP(Γ) to denote the class of all VCSP instances where the cost functions of all the valued constraints lie in Γ. A finite set of cost functions Γ will be called **tractable** if there exists a polynomial-time algorithm to solve VCSP(Γ). An infinite set of cost functions Γ will be called tractable if all finite subsets of Γ are tractable.

To analyse the complexity of problems of the form VCSP(Γ) for different choices of Γ we shall make use of a generalization of the notion of polymorphism which is known as a **multimorphism** [12].

Definition 3.6 ([12]) *A list of functions, $\langle f_1, \ldots, f_m \rangle$, where each f_i is a function from D^m to D, is a **multimorphism** of a cost function $\phi : D^r \to \overline{\mathbb{R}}_+$ if, for all $\langle a_{11}, \ldots, a_{1r} \rangle$, ..., $\langle a_{m1}, \ldots, a_{mr} \rangle \in D^r$, we have*

$$\sum_{i=1}^{m} \phi(f_i(a_{11}, \ldots, a_{m1}), \ldots, f_i(a_{1r}, \ldots, a_{mr})) \quad \leq \quad \sum_{i=1}^{m} \phi(a_{i1}, \ldots, a_{ir}) \qquad (1)$$

Note that if $\langle f_1, \ldots, f_m \rangle$ is a multimorphism of a cost function ϕ, then the average cost of a set of m assignments is lowered, or *improved* by applying the functions $f_1, \ldots, f_m$ co-ordinatewise. This observation explains the following choice of notation.

Notation: The set of all cost functions $\phi : D^r \to \overline{\mathbb{R}}_+$ which have $\langle f_1, \ldots, f_m \rangle$ as a multimorphism will be denoted $\mathrm{Imp}(f_1, \ldots, f_m)$.

Example 3.7 A cost function ϕ has the multimorphism $\langle \min, \max \rangle$ if and only if ϕ satisfies the **submodularity** condition

$$\forall \overline{x}, \overline{y} \in D^r \quad \phi(\overline{x} \vee \overline{y}) + \phi(\overline{x} \wedge \overline{y}) \;\leq\; \phi(\overline{x}) + \phi(\overline{y})$$

where $\vee$ and $\wedge$ represent co-ordinatewise maximum and minimum operations respectively.

Submodularity [26, 51] is usually defined over totally-ordered domains, but this definition can be extended to the case in which the domain D has an arbitrary lattice structure, in which case $\vee$ and $\wedge$ represent co-ordinatewise join and meet operations respectively.

Submodular function minimization (SFM) [26, 51] is a tractable discrete optimization problem which has applications in such diverse areas as statistical physics [1] and the design of electrical networks [44]. Well-known examples of submodular functions are the cut function of a graph [20] (see Example 3.5) or of a hypergraph [27], and the rank function of a matroid.

The ellipsoid algorithm provides a polynomial-time algorithm for SFM in theory, but is not efficient in practice [30]. Recently, several more efficient polynomial-time algorithms have been published to solve SFM [34, 50, 32, 33]. The fact that these algorithms can be applied to minimize a submodular function defined on a distributive lattice [32] (also known as a ring family

[50]) has been used to show that they can be applied to submodular functions which may take on both finite and infinite values over totally-ordered finite domains of arbitrary size [12]. The complexity of the fastest known algorithm for SFM is $O((n^4\gamma + n^5)\min\{\log M, n^2 \log n\})$, where n is the number of variables, γ is the time to calculate the objective function ψ and M is the maximum absolute value of ψ [33]. □

Example 3.8 The minimization of **bisubmodular** functions is studied in [28]. Bisubmodular functions are integer-valued functions on $\{0,1,2\}^r$, and can be characterized [9, 12] as those functions having the binary multimorphism $\langle \min_0, \max_0 \rangle$, where the functions $\min_0, \max_0 : \{0,1,2\}^2 \to \{0,1,2\}$ are defined as follows:

$$\min_0(a,b) = \begin{cases} \min(a,b) & \text{if } \{a,b\} \neq \{1,2\} \\ 0 & \text{otherwise} \end{cases}$$

$$\max_0(a,b) = \begin{cases} \max(a,b) & \text{if } \{a,b\} \neq \{1,2\} \\ 0 & \text{otherwise} \end{cases}$$

An example of a bisubmodular function is the rank function of a delta-matroid [28].

An integer-valued bisubmodular function ψ can be minimized in $O(n^5\gamma \log M)$ time where γ is the time to calculate the objective function ψ, M is the maximum value of the function ψ and n is the number of variables [28]. □

Example 3.9 The set of crisp cost functions over some fixed finite totally-ordered domain D which all have the multimorphism $\langle \min, \min \rangle$ corresponds to the tractable set of relations $\Gamma_{\min}$ defined in Example 2.6 which generalise the Horn clause satisfiability problem.

We can generalise this class further by dropping the requirement for the cost functions to be crisp. This gives a larger tractable class of cost functions which also allow arbitrary monotone finite-valued cost functions on the same variables [9, 12]. □

We have previously shown [8] that a set of cost functions over a Boolean domain is tractable if it has a non-trivial multimorphism and NP-complete otherwise. Over non-Boolean domains, the situation is more complex, but it is known that the complexity of any set of cost functions over any finite domain is characterized by certain algebraic properties which can be seen as generalised multimorphisms [13].

4 Tournament operations

In this section we focus on the properties of a particular kind of operation, which we call a *tournament operation*[1].

Definition 4.1 *A **tournament operation** is a binary operation $f : D^2 \to D$ with the following properties:*

- *f is **conservative**, that is $f(x, y) \in \{x, y\}$, for all $x, y \in D$.*
- *f is **commutative**, that is $f(x, y) = f(y, x)$, for all $x, y \in D$.*

*The **dual** of a tournament operation f is the unique tournament operation g satisfying $x \neq y \Rightarrow g(x, y) \neq f(x, y)$, for all $x, y \in D$.*

Note that, by definition, a tournament operation is necessarily **idempotent**, that is, $f(x, x) = x$, for all $x \in D$.

Definition 4.2 *A **tournament pair** is a pair $\langle f, g\rangle$, where f and g are both tournament operations. A tournament pair $\langle f, g\rangle$ is called **symmetric** if g is the dual of f.*

Examples of tournament pairs are the multimorphism $\langle \min, \min\rangle$ (Example 3.9) and the multimorphism $\langle \min, \max\rangle$ which defines submodularity (Example 3.7). It should be noted, however, that the multimorphism $\langle \min_0, \max_0\rangle$ which defines bisubmodularity (Example 3.8) is not a tournament pair since $\min_0$ and $\max_0$ are not conservative.

We will show in Section 8 that any set of cost functions with a tournament pair as a multimorphism is tractable. We first establish a partial converse of this result: any tractable set of cost functions containing all unary cost functions which is characterised by a binary multimorphism, must be characterised by a tournament pair multimorphism (assuming that $P \neq NP$).

Proposition 4.3 *For any binary operations f, g, if $\mathrm{Imp}(f, g)$ contains all unary cost functions, then either $\langle f, g\rangle$ is a symmetric tournament pair or $\mathrm{VCSP}(\mathrm{Imp}(f, g))$ is NP-hard.*

Proof: Since $\mathrm{Imp}(f, g)$ contains all unary cost functions, it is an easy consequence of Definition 3.6 that

$$\forall x, y \in D \quad \{f(x, y), g(x, y)\} \;=\; \{x, y\} \tag{2}$$

[1]The reason for this choice of terminology will be made clear in Section 6, where we explain the connection between tournament operations and directed graphs.

It follows that $\langle f, g \rangle$ is a symmetric tournament pair if f is commutative.

Consider now the case in which f is not commutative, that is, $f(a, b) \neq f(b, a)$ for some $a, b \in D$. Define the binary cost function $\phi_{XOR} : D^2 \to \overline{\mathbb{R}}_+$ as follows.

$$\phi_{XOR}(x, y) \quad = \quad \begin{cases} 1 & \text{if } x, y \in \{a, b\} \text{ and } x = y \\ 0 & \text{if } x, y \in \{a, b\} \text{ and } x \neq y \\ \infty & \text{otherwise} \end{cases}$$

Using Equation 2 it is easily verified that $\phi_{XOR} \in \mathrm{Imp}(f, g)$. However, $\mathrm{VCSP}(\{\phi_{XOR}\})$ can be shown to be NP-hard by a polynomial-time reduction from the MAX-2-SAT problem restricted to the XOR predicate, which is known to be NP-hard [18, 19]. Hence in this case $\mathrm{VCSP}(\mathrm{Imp}(f, g))$ is NP-hard. ∎

If we relax the conditions so that we require only *crisp* unary functions to be included, then we can still show that any tractable set of cost functions characterised by a binary multimorphism must have a tournament pair as a multimorphism.

Proposition 4.4 *For any binary operations f, g, if $\mathrm{Imp}(f, g)$ contains all crisp unary cost functions, then either $\mathrm{Imp}(f, g) \subseteq \mathrm{Imp}(f', g')$ for some tournament pair $\langle f', g' \rangle$ or $\mathrm{VCSP}(\mathrm{Imp}(f, g))$ is NP-hard.*

Proof: If $\mathrm{Imp}(f, g)$ contains all crisp unary cost functions, then it is straightforward to verify that the functions f, g must be conservative, and hence idempotent.

For any $a, b \in D$, denote the restrictions of f, g on $\{a, b\}$ by f_{ab}, g_{ab}. (In other words, f_{ab} is the function $f|_{\{a,b\} \times \{a,b\}} : \{a, b\}^2 \to \{a, b\}$). Since f_{ab}, g_{ab} are idempotent, this leaves just four possibilities for each of the functions f_{ab}, g_{ab}. Out of these four, two are commutative and the other two are *projections* (i.e., one of the functions $p_1, p_2 : \{a, b\}^2 \to \{a, b\}$ such that for all u, v, $p_1(u, v) = u$ and $p_2(u, v) = v$). If both f_{ab} and g_{ab} are projections, then $\mathrm{Imp}(f, g)$ contains *all* crisp cost functions $\phi : D^r \to \{0, \infty\}$ such that $\phi(\overline{x}) = \infty$ if $\overline{x} \notin \{a, b\}^r$, so $\mathrm{Imp}(f, g)$ is NP-hard, by a polynomial-time reduction from SAT.

Consider now the case in which for each $a, b \in D$ either f_{ab} or g_{ab} is commutative (or both). Define $f', g' : D^2 \to D$ as follows

$$\begin{aligned} f'(a,b) &= \begin{cases} f(a,b) & \text{if } f_{ab} \text{ is commutative} \\ g(a,b) & \text{otherwise} \end{cases} \\ g'(a,b) &= \begin{cases} g(a,b) & \text{if } g_{ab} \text{ is commutative} \\ f(a,b) & \text{otherwise} \end{cases} \end{aligned}$$

Clearly f', g' are tournament operations. It remains to show that $\mathrm{Imp}(f,g) \subseteq \mathrm{Imp}(f',g')$.

Consider an arbitrary cost function $\phi : D^r \to \overline{\mathbb{R}}_+$ in $\mathrm{Imp}(f,g)$. For $\overline{x}, \overline{y} \in D^r$, we use $f(\overline{x}, \overline{y})$ to represent the vector obtained by applying f coordinatewise to $\overline{x}$ and $\overline{y}$. Applying the multimorphism property (Equation 1) twice gives

$$\phi(\overline{x}) + \phi(\overline{y}) \;\geq\; \phi(g(\overline{x},\overline{y})) + \phi(f(\overline{x},\overline{y})) \;\geq\; \phi(p(\overline{x},\overline{y})) + \phi(q(\overline{x},\overline{y}))$$

where $p(\overline{x},\overline{y}) = f(g(\overline{x},\overline{y}), f(\overline{x},\overline{y}))$ and $q(\overline{x},\overline{y}) = g(g(\overline{x},\overline{y}), f(\overline{x},\overline{y}))$. Similarly,

$$\phi(\overline{x}) + \phi(\overline{y}) \;\geq\; \phi(f(\overline{x},\overline{y})) + \phi(g(\overline{x},\overline{y})) \;\geq\; \phi(r(\overline{x},\overline{y})) + \phi(s(\overline{x},\overline{y}))$$

where $r(\overline{x},\overline{y}) = f(f(\overline{x},\overline{y}), g(\overline{x},\overline{y}))$ and $s(\overline{x},\overline{y}) = g(f(\overline{x},\overline{y}), g(\overline{x},\overline{y}))$. By another application of Equation 1,

$$\phi(p(\overline{x},\overline{y})) + \phi(r(\overline{x},\overline{y})) \;\geq\; \phi(f(p(\overline{x},\overline{y}), r(\overline{x},\overline{y}))) + \phi(g(p(\overline{x},\overline{y}), r(\overline{x},\overline{y})))$$

and

$$\phi(s(\overline{x},\overline{y})) + \phi(q(\overline{x},\overline{y})) \;\geq\; \phi(f(s(\overline{x},\overline{y}), q(\overline{x},\overline{y}))) + \phi(g(s(\overline{x},\overline{y}), q(\overline{x},\overline{y})))$$

Now it is tedious but simple (using the fact that f and g are conservative, and checking all 16 possibilities) to show that, for all $x, y \in D$

$$\begin{aligned} f(p(x,y), r(x,y)) &= f'(x,y) \\ g(p(x,y), r(x,y)) &= f'(x,y) \\ f(s(x,y), q(x,y)) &= g'(x,y) \\ g(s(x,y), q(x,y)) &= g'(x,y) \end{aligned}$$

It follows that $2\phi(\overline{x}) + 2\phi(\overline{y}) \geq 2\phi(f'(\overline{x},\overline{y})) + 2\phi(g'(\overline{x},\overline{y}))$, and hence that $\phi \in \mathrm{Imp}(f',g')$. ∎

It is interesting to note that it is possible to have the strict inclusion $\mathrm{Imp}(f,g) \subset \mathrm{Imp}(f',g')$ in the above result, as the next example shows.

Example 4.5 Let $D = \{1, 2, 3\}$ and let $\psi : D^2 \to \{0, \infty\}$ be given by $\psi(\overline{x}) = 0$ if $\overline{x} \in \{\langle 1, 2\rangle, \langle 1, 3\rangle, \langle 2, 2\rangle\}$ and $\psi(\overline{x}) = \infty$ otherwise. Define $f, g : D^2 \to D$ by

$$\begin{aligned} f(a, b) &= \begin{cases} a & \text{if } a, b \in \{1, 2\} \\ \max(a, b) & \text{otherwise} \end{cases} \\ g(a, b) &= \min(a, b) \end{aligned}$$

Then $g' = g$ and f' is given by

$$f'(a, b) = \begin{cases} \min(a, b) & \text{if } a, b \in \{1, 2\} \\ \max(a, b) & \text{otherwise} \end{cases}$$

Now $\psi \in \mathrm{Imp}(f', g')$ but $\psi \notin \mathrm{Imp}(f, g)$ (since $\psi(2, 2) = \psi(1, 3) = 0$ but $\psi(f(\langle 2, 2\rangle, \langle 1, 3\rangle)) = \psi(2, 3) = \infty$). □

5 The set of all feasible/optimal assignments

For any cost function ϕ we define the corresponding sets of feasible assignments and optimal assignments in the following way.

Definition 5.1 *For any cost function $\phi : D^r \to \overline{\mathbb{R}}_+$, the set of **feasible assignments** for ϕ, denoted* $\mathrm{Feas}(\phi)$*, is defined as follows*

$$\mathrm{Feas}(\phi) \stackrel{\text{def}}{=} \{\overline{x} \in D^r : \phi(\overline{x}) < \infty\}$$

*The set of **optimal assignments** for ϕ, denoted* $\mathrm{Opt}(\phi)$*, is defined as follows*

$$\mathrm{Opt}(\phi) \stackrel{\text{def}}{=} \{\overline{x} \in D^r : \forall \overline{y} \in D^r, \phi(\overline{x}) \leq \phi(\overline{y})\}$$

Lemma 5.2 *If $\phi : D^r \to \overline{\mathbb{R}}_+$ has the multimorphism $\langle f_1, \ldots, f_m\rangle$, then the relations* $\mathrm{Opt}(\phi)$ *and* $\mathrm{Feas}(\phi)$ *both have the polymorphism f_i, for $i = 1, 2 \ldots, m$.*

Proof: Consider $\langle a_{11}, \ldots, a_{1r}\rangle, \ldots, \langle a_{m1}, \ldots, a_{mr}\rangle \in \mathrm{Opt}(\phi)$, and let α be the optimal value, i.e. $\alpha = \phi(a_{11}, \ldots, a_{1r})$. It is clear that to satisfy the inequality in Definition 3.6, we must have, for all $i \in \{1, \ldots, m\}$,

$$\phi(f_i(a_{11}, \ldots, a_{m1}), \ldots, f_i(a_{1r}, \ldots, a_{mr})) = \alpha$$

The result for $\mathrm{Opt}(\phi)$ follows immediately. A similar argument gives the result for $\mathrm{Feas}(\phi)$. ∎

Definition 5.3 *If f, g are functions from D^2 to D, then we say that f* absorbs *g if $\forall x, y \in D$,*

$$f(g(x,y),x) = f(g(y,x),x) = f(x,g(x,y)) = f(x,g(y,x)) = x$$

Example 5.4 Let f be a tournament operation and g its dual. If $g(x,y) = x$ then $f(g(x,y),x) = f(x,x) = x$. Conversely, if $g(x,y) = y$ then $f(g(x,y),x) = f(y,x) = x$ (since f and g are dual). Hence f absorbs g. A symmetric arguments shows that g absorbs f. □

Example 5.5 It is easy to verify that the binary functions min and max are mutually absorbing. In fact, by definition, any lattice operations $\wedge$ and $\vee$ are mutually absorbing. □

Example 5.6 Reconsider the operations $\max_0, \min_0$ defined in Example 3.8. It is easy to verify that $\max_0$ absorbs $\min_0$, but $\min_0$ does not absorb $\max_0$ (since $\min_0(\max_0(1,2),1) = 0 \neq 1$). □

Lemma 5.7 *Suppose that $f, g : D^2 \to D$ are idempotent and f absorbs g. If $\phi : D^r \to \overline{\mathbb{R}}_+$ has the multimorphism $\langle f, g\rangle$, then the relations* Feas(ϕ) *and* Opt(ϕ) *are both preserved by a majority operation.*

Proof: By Lemma 5.2, Feas(ϕ) and Opt(ϕ) both have the polymorphisms f and g. Now define the ternary operation $h : D^3 \to D$, for all $x, y, z \in D$, as follows

$$h(x,y,z) \stackrel{\text{def}}{=} f(f(g(x,y),g(x,z)),g(y,z)).$$

It is easy to verify that h is a majority operation (see Definition 2.7), since

$$\begin{array}{rcccccl} h(x,x,z) & = & f(f(x,g(x,z)),g(x,z)) & = & f(x,g(x,z)) & = & x \\ h(x,y,x) & = & f(f(g(x,y),x),g(y,x)) & = & f(x,g(y,x)) & = & x \\ h(x,y,y) & = & f(f(g(x,y),g(x,y)),y) & = & f(g(x,y),y) & = & y \end{array}$$

The set of polymorphisms of any relation is closed under composition [35], so Feas(ϕ) and Opt(ϕ) both have the polymorphism h. ∎

Corollary 5.8 *If $\phi : D^r \to \overline{\mathbb{R}}_+$ has the multimorphism $\langle f, g\rangle$, then* Opt(ϕ) *and* Feas(ϕ) *are preserved by a majority operation in each of the following cases:*

1. *f, g are the meet and join operations of a lattice.*

2. *f, g are the operations* $\max_0, \min_0$ *defined in Example 3.8.*

3. *f is a tournament operation and g is its dual.*

Proof: It is simple to verify that, in each case, f and g are idempotent and f absorbs g, as discussed in Examples 5.4 to 5.6. ∎

When a relation is preserved by a majority operation, and hence decomposable into binary projections, this provides a very compact representation for the relation, by simply listing the binary projections.

Proposition 5.9 *If $f, g : D^2 \to D$ are idempotent binary functions such that, for any cost function $\phi : D^n \to \overline{\mathbb{R}}_+ \in \mathrm{Imp}(f, g)$ the minimum* value *of ϕ can be computed in $O(T(n))$ time, then each binary projection of* $\mathrm{Opt}(\phi)$ *can be computed in $O(|D|^2 T(n))$ time.*

Proof: Consider any $\phi : D^n \to \overline{\mathbb{R}}_+ \in \mathrm{Imp}(f, g)$. We denote by ϕ_{ij}^{ab} the function on $n-2$ arguments obtained by fixing $x_i = a$ and $x_j = b$, that is, we set

$$\phi_{ij}^{ab}(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n) \stackrel{\text{def}}{=} \phi(x_1, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_{j-1}, b, x_{j+1}, \ldots, x_n)$$

Let α_{ij}^{ab} be the minimum value attained by ϕ_{ij}^{ab} on D^{n-2} and let α be the minimum value attained by ϕ on D^n. It follows that

$$\Pi_{ij}[\mathrm{Opt}(\phi)] \quad = \quad \{(a, b) \in D^2 \mid \alpha_{ij}^{ab} = \alpha\}.$$

and thus the binary projections of $\mathrm{Opt}(\phi)$ can be determined by calculating the values of α and α_{ij}^{ab} for all $a, b \in D$.

It follows directly from Definition 3.6 that if ϕ has the multimorphism $\langle f, g \rangle$, where f and g are both idempotent, then ϕ_{ij}^{ab} also has the multimorphism $\langle f, g \rangle$. Hence all of these values can be computed in $O(|D|^2 T(n-2) + T(n))$ time, and hence in $O(|D|^2 T(n))$ time. ∎

Hence, in each of the cases mentioned in Corollary 5.8, a compact representation of all minimizers for a given cost function can be found in polynomial time by using existing polynomial-time algorithms to find the minimal *value*.

Example 5.10 Consider the important special case of submodular function minimisation (SFM) over a Boolean domain, $\{0, 1\}$. In this case the relations

$\Pi_{ij}[\mathrm{Opt}(\phi)]$ are Boolean binary submodular relations. By exhaustion, it is easy to show that any such relation can be represented as a conjunction of 0,1 or 2 of the following relations: $X_i = 0$, $X_i = 1$, $X_j = 0$, $X_j = 1$, $X_i = X_j$, $X_i \leq X_j$, $X_i \geq X_j$. It follows that the set $\mathrm{Opt}(\phi)$ of optimal solutions to an SFM problem over a Boolean domain can be represented by a partial order. Ekin et al. [24] established the same result for $\mathrm{Feas}(\phi)$. □

6 Modular decomposition of tournaments

In this section we introduce a number of ideas from graph theory which will be used in Section 7 to analyse the structure of cost functions with a tournament pair multimorphism.

First we note that there is a one-to-one correspondence between tournament operations f and complete digraphs $G = \langle D, E\rangle$ given by $(x, y) \in E$ iff $x \neq y$ and $f(x, y) = y$, for all $x, y \in D$. Such complete digraphs are usually known as **tournaments**. Hence every tournament operation has an associated tournament, and vice-versa.

If $f : D^2 \to D$ is a tournament operation, then we will write $\langle D, f\rangle$ to represent the corresponding tournament (i.e., the complete digraph on D with an arc from a to b if and only if $f(a, b) = a$). For any $B \subseteq D$, we will write $\langle B, f\rangle$, or simply B, to represent the subtournament $\langle B, f|_{B^2}\rangle$.

Two sets X, Y will be said to **overlap** if they intersect but neither is a subset of the other.

Definition 6.1 ([21]) *Given a tournament $\langle D, f\rangle$, a subset $B \subseteq D$ is called a **module** if for all $c \in D - B$, and all $a, b \in B$, $f(a, c) = a$ if and only if $f(b, c) = b$. A module is **strong** if no other module overlaps it.*

The strong modules of a tournament $\langle D, f\rangle$ can be organized in a tree structure (known as its **modular decomposition**) with root D, a leaf $\{a\}$ for each $a \in D$ and such that at each internal node A the children $A_1, \ldots, A_r$ of A form a partition of A [21].

Definition 6.2 *In the modular decomposition of a tournament $\langle D, f\rangle$, a node A with children $A_1, \ldots, A_r$ is called*

- ***prime** if $\forall I \subset \{1, \ldots, r\}$ such that $1 < |I| < r$, $\bigcup_{i\in I} A_i$ is not a module. (We say that $A_1, \ldots, A_r$ is a **prime partition** of A.)*
- ***linear** if there exists an ordering of $\{1, \ldots, r\}$ such that if $I \subset \{1, \ldots, r\}$ and $1 < |I| < r$, then $\bigcup_{i\in I} A_i$ is a module if and only if the members of*

*I are consecutive in the ordering. (We say that $A_1, \ldots, A_r$ is a **linear partition** of A.)*

All tournaments have a unique modular decomposition in which each node is either prime or linear. In fact, this decomposition can be found in $O(|D|^2)$ (and hence optimal) time [43]. We denote this unique modular decomposition of a tournament $\langle D, f\rangle$ by $\mathrm{MD}(D, f)$. Consider a strong module A of a tournament $\langle D, f\rangle$ such that $|A| > 1$. If the subtournament $\langle A, f\rangle$ is strongly-connected, then the node A is prime, otherwise A is linear and $A_1, \ldots, A_r$ are its strongly-connected components. Any module A which is strongly connected is necessarily strong and hence has a corresponding node in $\mathrm{MD}(D, f)$.

If $A_1, \ldots, A_r$ are the children of A in the modular decomposition $\mathrm{MD}(D, f)$, then for any $i, j \in \{1, \ldots, r\}$ we have that for all $a, b \in A_i$, and all $c, d \in A_j$, $f(a, c) = a \Leftrightarrow f(b, c) = b \Leftrightarrow f(b, d) = b$. Hence f also defines a tournament operation on $\{A_1, \ldots, A_r\}$. We call this the tournament operation **induced** on $\{A_1, \ldots, A_r\}$ by f and we abuse notation by writing $f(A_i, A_j) = A_i$ if $\forall a \in A_i$, $\forall b \in A_j$, $f(a, b) = a$.

A set $A \subseteq D$ is **simple** with respect to the tournament operation $f : D^2 \to D$ if there is no non-trivial congruence class B of elements of A which all behave in the same way with respect to the other elements $A - B$ [6]. In other words, A is simple with respect to f if and only if there is no module B of $\langle A, f\rangle$, with $1 < |B| < |A|$.

The following lemma summarises the discussion above.

Lemma 6.3 *Let $\langle D, f\rangle$ be a tournament, let A be a module of $\langle D, f\rangle$ and suppose that $|A| > 1$ and $\langle A, f\rangle$ is a strongly connected subtournament. Then*

1. *A is a prime node in the modular decomposition $\mathrm{MD}(D, f)$.*
2. *If $A_1, \ldots, A_r$ are the children of A in $\mathrm{MD}(D, f)$, then $\{A_1, \ldots, A_r\}$ is simple and strongly connected with respect to the induced tournament.*

7 Cost functions with tournament pair multimorphisms

In this section we will show that any cost function with a tournament pair as a multimorphism has certain special properties. These results will be used in Section 8 to establish the tractability of $\mathrm{Imp}(f, g)$ for any tournament pair $\langle f, g\rangle$.

A standard technique of constraint satisfaction is to eliminate values from the domains of variables when these values can be shown to be inconsistent. We will adapt this technique to valued constraint satisfaction by defining (partial) cost functions with a reduced domain.

Definition 7.1 *A function* $\phi : D_1 \times \ldots \times D_r \to \overline{\mathbb{R}}_+$ *is* ***domain-reduced*** *if, for each* $i \in \{1, \ldots, r\}$*, and for each* $a \in D_i$ $\exists \overline{x} \in D_1 \times \ldots \times D_r$ *such that* $\overline{x}[i] = a$ *and* $\phi(\overline{x}) < \infty$*.*

In the literature on constraint satisfaction, a VCSP instance where every cost function is domain-reduced is said to be *generalized arc consistent* in its underlying CSP [15, 17].

Given a crisp binary function $\phi : D_1 \times D_2 \to \{0, \infty\}$, the binary relation $\text{Feas}(\phi) = \{\langle d_1, d_2 \rangle \in D_1 \times D_2 \mid \phi(d_1, d_2) = 0\}$ is a bijection of $D_1 \times D_2$ if for each $a \in D_1$ there is a unique $b \in D_2$ such that $\langle a, b \rangle \in \text{Feas}(\phi)$ and for each $b \in D_2$ there is a unique $a \in D_1$ such that $\langle a, b \rangle \in \text{Feas}(\phi)$.

Definition 7.2 *A function* $\phi : D_1 \times D_2 \to \{0, \infty\}$ *is called* ***crisp-bijective*** *if the relation* $\text{Feas}(\phi)$ *is a bijection of* $D_1 \times D_2$*.*

If $\phi : D_1 \times D_2 \to \{0, \infty\}$ is crisp-bijective, then we will abuse notation and write $\phi[a_1]$ (for any $a_1 \in D_1$) to represent the unique $a_2 \in D_2$ such that $\phi(a_1, a_2) = 0$.

Definition 7.3 *A crisp-bijective function* $\phi : D_1 \times D_2 \to \{0, \infty\}$ *is an* ***isomorphism*** *with respect to the tournament operation* $f : D \times D \to D$*, (where* $D_1, D_2 \subseteq D$*) if for all* $a, b \in D_1$*,* $f(a, b) = a \Leftrightarrow f(\phi[a], \phi[b]) = \phi[a]$*.*

Lemma 7.4 *A crisp-bijective function* $\phi : D_1 \times D_2 \to \{0, \infty\}$ *has a symmetric tournament pair* $\langle f, g \rangle$ *as a multimorphism if and only if* ϕ *is an isomorphism with respect to* f *(and hence also* g*).*

Proof: Using $\phi[a]$ to represent the unique $c \in D_2$ such that $\phi(a, c) = 0$, we have that ϕ has the multimorphism $\langle f, g \rangle$ if and only if for all $a, b \in D_1$,

$$0 = \phi(a, \phi[a]) + \phi(b, \phi[b]) \;\geq\; \phi(f(a, b), f(\phi[a], \phi[b])) + \phi(g(a, b), g(\phi[a], \phi[b]))$$

This inequality holds if and only if $f(a, b) = a$ precisely when $f(\phi[a], \phi[b]) = \phi[a]$. ∎

Lemma 7.5 *Let $\phi : D_1 \times D_2 \to \{0, \infty\}$ be a domain-reduced crisp cost function with a symmetric tournament pair $\langle f, g\rangle$ as a multimorphism.*

If D_1 and D_2 are simple and strongly connected with respect to f, then ϕ is either the constant function 0 or crisp-bijective.

Proof: Since D_1, D_2 are simple and strongly connected with respect to f, and the relation $\text{Feas}(\phi) = \{\langle d_1, d_2\rangle \in D_1 \times D_2 \mid \phi(d_1, d_2) = 0\}$ is preserved by f (by Lemma 5.2), Proposition 30 of [6] (or the more general Lemma 3.5 of [5]) tells us that ϕ is either constant 0 or crisp-bijective. ∎

Lemma 7.6 *Let $\phi : D_1 \times D_2 \to \{0, \infty\}$ be a domain-reduced crisp cost function with a symmetric tournament pair $\langle f, g\rangle$ as a multimorphism.*

Let A be a module of $\langle D_1, f\rangle$ and let $B = \{b \in D_2 | \exists a \in D_1, \phi(a, b) = 0\}$. Then B is a module of $\langle D_2, f\rangle$.

Proof: Suppose, for a contradiction, that B is not a module of $\langle D_2, f\rangle$. Then $\exists b, b' \in D_2$, $\exists v \in D_2 - B$ such that $f(v, b) = v$ and $f(v, b') = b'$. Since ϕ is domain-reduced, $\exists u \in D_1$ such that $\phi(u, v) = 0$. Suppose, without loss of generality, that $f(D_1 - A, A) = A$, and let $a \in A$ be such that $\phi(a, b) = 0$. Then $\phi(u, a) = a$ and by the multimorphism property,

$$0 \;=\; \phi(u, v) + \phi(a, b) \;\geq\; \phi(a, v) + \phi(u, b)$$

This implies that $\phi(a, v) = 0$ which contradicts our hypothesis that $v \notin B$. ∎

Definition 7.7 *Let $\phi : D_1 \times D_2 \to \overline{\mathbb{R}}_+$ be a cost function and let $\langle f, g\rangle$ be a symmetric tournament pair on a set D, where $D_1, D_2 \subseteq D$. Let $A_1, \ldots, A_r$ and $B_1, \ldots, B_s$ be prime or linear partitions in the modular decomposition of $\langle D_1, f\rangle$ and $\langle D_2, f\rangle$, respectively. The **induced crisp** cost function $\overline{\phi}$: $\{A_1, \ldots, A_r\} \times \{B_1, \ldots, B_s\} \to \{0, \infty\}$ is defined by $\overline{\phi}(A_i, B_j) = 0$ if and only if $\exists a \in A_i$, $\exists b \in B_j$ such that $\phi(a, b) < \infty$.*

Lemma 7.8 *If $f, g, \phi, \overline{\phi}$ are as in Definition 7.7, then $\phi \in \text{Imp}(f, g)$ implies that $\overline{\phi} \in \text{Imp}(f, g)$.*

Proof: Suppose that $\overline{\phi} \notin \text{Imp}(f, g)$. Then, without loss of generality, there are A_i, A_j and B_k, B_m such that $f(A_i, A_j) = A_i$, $f(B_k, B_m) = B_k$, $\overline{\phi}(A_i, B_m) = \overline{\phi}(A_j, B_k) = 0$ but $\overline{\phi}(A_i, B_k) \neq 0$. But then $\exists a \in A_i, b \in A_j, c \in B_k, d \in B_m$ such that $\phi(a, d) < \infty$, $\phi(b, c) < \infty$ and $\phi(a, c) = \infty$. Since $f(a, b) = a$ and $f(c, d) = c$, we can deduce that $\phi \notin \text{Imp}(f, g)$. ∎

Definition 7.9 *A cost function $\phi : D_1 \times \ldots \times D_r \to \overline{\mathbb{R}}_+$ is **finite** if $\phi(\overline{x}) < \infty$ for all $\overline{x} \in D_1 \times \ldots \times D_r$.*

Lemma 7.10 *Let $\phi : D_1 \times D_2 \to \overline{\mathbb{R}}_+$ be a domain-reduced cost function with a symmetric tournament pair $\langle f, g \rangle$ as a multimorphism.*

If D_1 and D_2 are both strongly connected with respect to f, then either ϕ is finite, or the induced crisp cost function $\overline{\phi}$ on the prime partitions of D_1 and D_2 is crisp-bijective.

Proof: By Lemma 6.3, both D_1 and D_2 have prime partitions $\{A_1, \ldots, A_r\}$ and $\{B_1, \ldots, B_s\}$ which are simple and strongly connected with respect to the induced tournament. Hence, by Lemma 7.5, if $\overline{\phi}$ is not crisp-bijective, then $\overline{\phi} = 0$. We will show that in this case $\phi(a, b) < \infty$ for any $a \in D_1$, $b \in D_2$.

Suppose that $a \in A_i$ and $b \in B_j$. Since ϕ is domain-reduced, $\exists v \in D_2$ such that $\phi(a, v) < \infty$. Suppose that $v \in B_k$. We claim that $\exists w \in B_j$ such that $\phi(a, w) < \infty$. Assume that $f(B_j, B_k) = B_k$ (the argument for the case $f(B_j, B_k) = B_j$ is symmetric). Since $\{A_1, \ldots, A_r\}$ is strongly connected, $\exists m \in \{1, \ldots, r\}$ such that $f(A_i, A_m) = A_m$. Now, since $\overline{\phi} = 0$, $\exists u \in A_m$, $\exists w \in B_j$ such that $\phi(u, w) < \infty$. Applying the multimorphism property, we obtain

$$\begin{aligned} \infty > \phi(a, v) + \phi(u, w) &\geq \phi(f(a, u), f(v, w)) + \phi(g(a, u), g(v, w)) \\ &= \phi(u, v) + \phi(a, w) \end{aligned}$$

Therefore $\phi(a, w) < \infty$.

Since $\{A_1, \ldots, A_r\}$ and $\{B_1, \ldots, B_s\}$ are strongly connected, $\exists h, l$ such that $f(A_i, A_l) = A_i$ and $f(B_j, B_h) = B_h$. Since $\overline{\phi} = 0$, $\exists c \in A_l$, $\exists d \in B_h$ such that $\phi(c, d) < \infty$. Applying the multimorphism property, we obtain

$$\begin{aligned} \infty > \phi(a, w) + \phi(c, d) &\geq \phi(f(a, c), f(w, d)) + \phi(g(a, c), g(w, d)) \\ &= \phi(a, d) + \phi(c, w) \end{aligned}$$

Therefore $\exists d \in B_h$ such that $\phi(a, d) < \infty$. By a similar argument we can show that $\exists e \in A_m$ such that $\phi(e, b) < \infty$ (where m is such that $f(A_i, A_m) = A_m$). Then applying the multimorphism property gives

$$\begin{aligned} \infty > \phi(a, d) + \phi(e, b) &\geq \phi(f(a, e), f(d, b)) + \phi(g(a, e), g(d, b)) \\ &= \phi(e, d) + \phi(a, b) \end{aligned}$$

Thus $\phi(a, b) < \infty$. ∎

Lemma 7.11 *Let $\phi : D_1 \times D_2 \rightarrow \overline{\mathbb{R}}_+$ be a domain-reduced cost function with a symmetric tournament pair $\langle f, g \rangle$ as a multimorphism.*

If D_1 is strongly connected with respect to f and D_2 is acyclic with respect to f, then ϕ is finite.

Proof: If D_1 is strongly connected, then it must contain a complete cycle with respect to f, i.e., $D_1 = \{a_0, \ldots, a_{r-1}\}$ such that $f(a_i, a_{i+1}) = a_i$ (for $i = 0, \ldots, r-1$) with the addition $i+1$ understood as being modulo r. Note that this complete cycle need not be Hamiltonian, since we allow repeats (i.e. $a_i = a_j$ for some $i \neq j$).

On the other hand, an acyclic tournament defines a total order. Thus $D_2 = \{b_1, \ldots, b_s\}$ such that $f(b_i, b_j) = b_i$ if and only if $i \leq j$ (and where, in this case, $b_i \neq b_j$ if $i \neq j$).

Suppose, for contradiction, that $\phi(a_i, b_j) = \infty$. Since ϕ is domain-reduced, there exists some k' such that $\phi(a_{k'}, b_j)$ is finite. Hence there must be some k such that $\phi(a_k, b_j) < \phi(a_{k+1}, b_j) = \infty$, where the addition $k+1$ is again modulo r. For all $h < j$, we have the following multimorphism inequality:

$$\phi(a_k, b_j) + \phi(a_{k+1}, b_h) \;\geq\; \phi(a_k, b_h) + \phi(a_{k+1}, b_j) \;=\; \infty$$

from which we deduce that $\phi(a_{k+1}, b_h) = \infty$. Since ϕ is domain-reduced, there exists some $m > j$ such that $\phi(a_{k+1}, b_m)$ is finite. For $t \in \{1, \ldots, r\}$, let $m(t)$ be the smallest integer such that $\phi(a_t, b_{m(t)})$ is finite. We have just shown that $m(k+1) > j$. If $m(t) > j$, then for all $h \leq j$, $\phi(a_t, b_h) = \infty$, so

$$\phi(a_t, b_{m(t)}) + \phi(a_{t+1}, b_h) \;\geq\; \phi(a_t, b_h) + \phi(a_{t+1}, b_{m(t)}) \;=\; \infty$$

and hence for all $h \leq j$, $\phi(a_{t+1}, b_h) = \infty$, from which it follows that $m(t+1) > j$. Thus $m(t) > j \;\Rightarrow\; m(t+1) > j$ (where the addition $t+1$ is again modulo r). But then, by induction through the integers $k+1, k+2, \ldots, r-1, 0, 1, \ldots, k$ we can deduce that $m(k) > j$, which contradicts $\phi(a_k, b_j) < \infty$. ∎

We now extend Lemma 7.11 to any domain D_2 which is not strongly connected.

Lemma 7.12 *Let $\phi : D_1 \times D_2 \rightarrow \overline{\mathbb{R}}_+$ be a domain-reduced cost function with a symmetric tournament pair $\langle f, g \rangle$ as a multimorphism.*

If D_1 is strongly connected with respect to f and D_2 is not, then ϕ is finite.

Proof: If $|D_2| = 1$ then ϕ is finite because it is domain-reduced. Otherwise, since D_2 is not strongly connected, it has a partition B_1, B_2 (with $B_1, B_2 \neq \emptyset$) such that $f(B_1, B_2) = B_1$. Let $A_1, \ldots, A_r$ be the prime partition of D_1. Consider the induced crisp cost function $\overline{\phi} : \{A_1, \ldots, A_r\} \times \{B_1, B_2\} \to \{0, \infty\}$. From Lemma 7.11, we know that $\overline{\phi} = 0$. We will show that $\phi(a, b) < \infty$ for any $a \in A_i$, $b \in B_2$. The proof for $b \in B_1$ is entirely similar.

Since ϕ is domain reduced, $\exists u \in A_k$ (for some $k \in \{1, \ldots, r\}$) such that $\phi(u, b) < \infty$. Since D_1 is strongly connected, $\exists j \in \{1, \ldots, r\}$ such that $f(A_j, A_k) = A_k$. Since $\overline{\phi} = 0$, $\exists v \in A_j$, $\exists w \in B_1$ such that $\phi(v, w) < \infty$. Applying the multimorphism property, we obtain

$$\begin{aligned} \infty > \phi(u, b) + \phi(v, w) &\geq \phi(f(u, v), f(b, w)) + \phi(g(u, v), g(b, w)) \\ &= \phi(u, w) + \phi(v, b) \end{aligned}$$

Therefore $\exists v \in A_j$ such that $\phi(v, b) < \infty$. By an easy inductive proof, we can show that

$$\forall h \in \{1, \ldots, r\},\ \exists z \in A_h \text{ such that } \phi(z, b) < \infty \tag{3}$$

since for all h, there is a chain $i_1, \ldots, i_p$ such that $i_1 = h$, $i_p = k$ and $f(A_{i_j}, A_{i_{j+1}}) = A_{i_{j+1}}$ $(j = 1, \ldots, p-1)$.

Now, since ϕ is domain reduced, $\exists c \in D_2$ such that $\phi(a, c) < \infty$. Assume that $f(c, b) = c$ (the proof for the case $f(c, b) = b$ is entirely similar). Since D_1 is strongly connected, $\exists m \in \{1, \ldots, r\}$ such that $f(A_i, A_m) = A_m$. By Equation (3) above, $\exists d \in A_m$ such that $\phi(d, b) < \infty$. Applying the multimorphism property gives

$$\begin{aligned} \infty > \phi(d, b) + \phi(a, c) &\geq \phi(f(d, a), f(b, c)) + \phi(g(d, a), g(b, c)) \\ &= \phi(d, c) + \phi(a, b) \end{aligned}$$

Therefore $\phi(a, b) < \infty$. ∎

We can combine Lemmas 7.10 and 7.12 into the following proposition.

Proposition 7.13 *Let $\phi : D_1 \times D_2 \to \overline{\mathbb{R}}_+$ be a domain-reduced cost function with a symmetric tournament pair $\langle f, g \rangle$ as a multimorphism, where D_1 is strongly connected with respect to f.*

Then either D_2 is strongly connected with respect to f and the induced crisp cost function on the prime partitions of D_1 and D_2 is crisp-bijective, or ϕ is finite.

The final result we shall need shows that in some circumstances cost functions can be expressed as the sum of cost functions with smaller arity. We first extend to cost functions the definition of projection given for relations in Definition 2.8.

Definition 7.14 *Given a cost function $\phi : D_1 \times \ldots \times D_r \to \overline{\mathbb{R}}_+$, and a set of indices $I = \{i_1, \ldots, i_p\}$, the* ***projection*** *of ϕ onto I is the function $\Pi_I(\phi) : D_{i_1} \times \ldots \times D_{i_p} \to \overline{\mathbb{R}}_+$ defined by*

$$\Pi_I \phi(x_1, \ldots, x_p) \stackrel{\text{def}}{=} \min_{\{\overline{z} \in D_1 \times \ldots \times D_r \,|\, \overline{z}[i_j] = x_j \;\; (j=1,\ldots,p)\}} \{\phi(\overline{z})\}$$

For notational convenience, unary and binary projections will be denoted by $\Pi_i(\phi)$ and $\Pi_{ij}(\phi)$ rather than $\Pi_{\{i\}}(\phi)$ and $\Pi_{\{i,j\}}(\phi)$.

Note that if $\langle f_1, \ldots, f_m \rangle$ is a multimorphism of ϕ, then it is also a multimorphism of $\Pi_I(\phi)$ [12].

Lemma 7.15 *Let $\phi : D_1 \times \ldots \times D_r \to \overline{\mathbb{R}}_+$ be a cost function with a symmetric tournament pair $\langle f, g \rangle$ as a multimorphism.*

If D_1 is strongly connected with respect to f and each binary projection $\Pi_{1j}(\phi)$ is finite, for $j = 2, \ldots, r$, then $\phi = \phi_1 + \phi_2$ where $\phi_1 : D_1 \to \overline{\mathbb{R}}_+$ is unary and $\phi_2 : D_2 \times \ldots \times D_r \to \overline{\mathbb{R}}_+$ belongs to $\mathrm{Imp}(f, g)$.

Proof: We prove the result by induction on the arity of ϕ. The result trivially holds if ϕ is unary. Suppose that it holds for cost functions of arity less than r and consider a cost function ϕ of arity $r > 1$.

If D_1 is strongly connected, then it must contain a complete cycle with respect to f, i.e., $D_1 = \{a_0, \ldots, a_{r-1}\}$ such that $f(a_i, a_{i+1}) = a_i$ (for $i = 0, \ldots, r-1$) with the addition $i + 1$ understood as being modulo r.

Let $\overline{y} = \langle y_3, \ldots, y_r \rangle \in D_3 \times \ldots \times D_r$. (If $r = 2$, then $\overline{y} = \langle\rangle$ is just the tuple of length zero.) Consider the cost function $\psi_{\overline{y}} : D_1 \times D_2 \to \overline{\mathbb{R}}_+$ defined by $\psi_{\overline{y}}(u, v) = \phi(u, v, y_3, \ldots, y_r)$. Choose an arbitrary pair $a, b \in D_2$, and assume without loss of generality that $f(a, b) = b$. Since $\phi \in \mathrm{Imp}(f, g)$, the following inequalities follow from the definition of a multimorphism and the duality of f and g.

$$\begin{array}{rcl}
\psi_{\overline{y}}(a_0, a) + \psi_{\overline{y}}(a_1, b) & \leq & \psi_{\overline{y}}(a_0, b) + \psi_{\overline{y}}(a_1, a) \\
\psi_{\overline{y}}(a_1, a) + \psi_{\overline{y}}(a_2, b) & \leq & \psi_{\overline{y}}(a_1, b) + \psi_{\overline{y}}(a_2, a) \\
\vdots & & \\
\psi_{\overline{y}}(a_{r-1}, a) + \psi_{\overline{y}}(a_0, b) & \leq & \psi_{\overline{y}}(a_{r-1}, b) + \psi_{\overline{y}}(a_0, a)
\end{array}$$

Consider first the case in which $\psi_{\overline{y}}$ is finite. By summing the above r inequalities we can see that they are only compatible when there is equality throughout.

Consider now the case in which $\psi_{\overline{y}}$ is not finite. Without loss of generality suppose that $\psi_{\overline{y}}(a_i, a) = \infty$. By the hypothesis that $\Pi_{1j}(\phi)$ is finite for all $j > 1$, and since $\mathrm{Feas}(\phi)$ is decomposable into its binary projections (by Corollary 5.8), we must have $\Pi_{2k}(\phi)(a, y_k) = \infty$ (for some $k \in \{3, \ldots, r\}$) or $\Pi_{jk}(\phi)(y_j, y_k) = \infty$ (for some $j, k \in \{3, \ldots, r\}$). In both cases, there is equality in all r of the above inequalities, as both sides are infinite.

Hence, in all cases, for all $u, v \in D_1$, and all $x, y \in D_2$,

$$\psi_{\overline{y}}(u, x) \;+\; \psi_{\overline{y}}(v, y) \;\;=\;\; \psi_{\overline{y}}(u, y) \;+\; \psi_{\overline{y}}(v, x)$$

Any binary cost function satisfying an identity of this form is called *modular*.

It is known that a binary modular cost function can be expressed as the sum of two unary cost functions [9, 12]. Therefore, $\psi_{\overline{y}}(u, v) = \psi^1_{\overline{y}}(u) + \psi^2_{\overline{y}}(v)$ for some unary functions $\psi^1_{\overline{y}} : D_1 \to \overline{\mathbb{R}}_+$, and $\psi^2_{\overline{y}} : D_2 \to \overline{\mathbb{R}}_+$. It follows that $\phi(u, v, y_3, \ldots, y_r) = \psi_1(u, y_3, \ldots, y_r) + \psi_2(v, y_3, \ldots, y_r)$, where $\psi_1(u, y_3, \ldots, y_r) = \psi^1_{\overline{y}}(u)$ and $\psi_2(v, y_3, \ldots, y_r) = \psi^2_{\overline{y}}(v)$ are $(r-1)$-ary cost functions. Moreover, it is straightforward to verify that we can take ψ_1 to be the function given by $\forall (x_1, x_3, x_4, \ldots, x_r) \in D_1 \times D_3 \times D_4 \times \ldots \times D_r \to \overline{\mathbb{R}}_+$,

$$\psi_1(x_1, x_3, x_4, \ldots, x_r) \;\;=\;\; \min\{\phi(x_1, x_2, \ldots, x_r) : x_2 \in D_2\}$$

(an operation known as projection of cost functions) and $\psi_2 = \phi - \psi_1$. Now, $\psi_1 \in \mathrm{Imp}(f, g)$ since multimorphisms are preserved under projection of cost functions [12]. Thus, by the inductive hypothesis, $\psi_1 = \phi_1 + \psi_3$ where $\phi_1 : D_1 \to \overline{\mathbb{R}}_+$ is unary and $\psi_3 : D_3 \times \ldots \times D_r \to \overline{\mathbb{R}}_+$ belongs to $\mathrm{Imp}(f, g)$. Thus $\phi = \phi_1 + \phi_2$ where $\phi_2 = \psi_3 + \psi_2$. Now $\phi_2 : D_2 \times \ldots \times D_r \to \overline{\mathbb{R}}_+$ belongs to $\mathrm{Imp}(f, g)$ since $\phi - \phi_1 \in \mathrm{Imp}(f, g)$ for all unary functions ϕ_1. ∎

8 Tournament pair multimorphisms give tractability

We will first show that any set of cost functions with a *symmetric* tournament pair multimorphism is tractable by showing that it is possible to construct a reordering of the domains of each of the variables which converts the corresponding VCSP instance to an instance of submodular function minimisation (SFM).

It is known that every tournament $\langle D, f\rangle$ admits a **perfect factorizing permutation**, that is, a linear ordering of D such that all modules are intervals in the ordering [43]. In fact, this total ordering can be obtained by modifying f in the following way.

Definition 8.1 *Let $f : D^2 \to D$ be a tournament operation. A **total ordering derived from** f is a tournament operation $f' : D^2 \to D$ such that*

1. *for all $a, b \in D$, if it is not the case that $a \in A_i$ and $b \in A_j$ $(j \neq i)$ where $A_1, \ldots, A_r$ are the children of a prime node in $\mathrm{MD}(D, f)$, then $f'(a, b) = f(a, b)$.*

2. *for all prime nodes in $\mathrm{MD}(D, f)$ with children $A_1, \ldots, A_r$, the induced tournament $f' : \{A_1, \ldots, A_r\}^2 \to \{A_1, \ldots, A_r\}$ is a total order.*

Theorem 8.2 *If $\langle f, g\rangle$ is a symmetric tournament pair, then $\mathrm{Imp}(f, g)$ is tractable.*

Proof: Let Γ be a finite subset of $\mathrm{Imp}(f, g)$, and let $\mathcal{P} = \langle V, D, C\rangle$ be any instance of $\mathrm{VCSP}(\Gamma)$ and assume that $V = \{v_1, v_2, \ldots, v_n\}$.

The proof proceeds in three stages. We first restrict the domains of the variables of $\mathcal{P}$ in such a way that every cost function is domain-reduced. Second, we construct a total ordering derived from f for each of these restricted domains in polynomial time. Finally, we show that with these total orderings every cost function is submodular, and hence the minimal cost solution can be found in polynomial time.

Stage 1: For the first step consider the CSP instance obtained by replacing each valued constraint $\langle \sigma, \phi\rangle$ in C with the constraint $\langle \sigma, \mathrm{Feas}(\phi)\rangle$. By Corollary 5.8, all of the relations $\mathrm{Feas}(\phi)$ are preserved by a fixed majority operation, and hence by Lemma 2.9 they are decomposable into binary projections. Since Γ is finite, the maximum arity of the cost functions in Γ is bounded by a constant, so we can calculate the binary projections of $\mathrm{Feas}(\phi)$ for each $\langle \sigma, \phi\rangle \in C$ in polynomial time in the size of $\mathcal{P}$. Furthermore each of these binary projections is also preserved by the same majority operation, so we have constructed a CSP instance $\mathcal{P}'$ with binary constraints where each constraint relation is preserved by a fixed majority operation.

We now establish strong 3-consistency in the CSP instance $\mathcal{P}'$ in $O(n^3|D|^3)$ time using standard constraint-processing techniques [31, 14]. The resulting CSP instance $\mathcal{P}''$ has restricted domains $D_1, D_2, \ldots, D_n$ for the variables $v_1, v_2, \ldots, v_n$, respectively, and possibly some new binary constraints.

By Theorem 3.5 of [36], the instance $\mathcal{P}''$ is strong n-consistent, so each value in each restricted domain can be extended to a complete solution. This means that each of the cost functions in $\mathcal{P}$ are domain-reduced when limited to these restricted domains, which completes the first stage of the proof. The extended VCSP instance with all the original valued constraints of $\mathcal{P}$, the restricted domains $D_1, D_2, \ldots, D_n$, and binary crisp cost functions corresponding to all the constraints of $\mathcal{P}''$ will be denoted $\widehat{\mathcal{P}}$. Note that $\text{Cost}_{\widehat{\mathcal{P}}} = \text{Cost}_{\mathcal{P}}$ but we have introduced redundant binary constraints in $\widehat{\mathcal{P}}$ in order to render explicit exactly those assignments to pairs of variables that cannot be extended to a complete solution of finite cost.

Define $\text{Feas}_{ij} : D_i \times D_j \to \{0, \infty\}$ to be the explicit crisp cost function on variables v_i, v_j, i.e. $\text{Feas}_{ij}(a, b) = 0$ iff (a, b) is a consistent assignment to variables (v_i, v_j) in $\mathcal{P}''$.

Stage 2: For the second stage of the proof we need to construct a total ordering f' derived from f. In stage 3 of the proof, we will show that every cost function of $\widehat{\mathcal{P}}$ is an element of $\text{Imp}(f', g')$, where g' is the dual of f'. Since we allow each D_i to have its own individual ordering, we will simplify notation by assuming, in the following, that the sets $D_1, \ldots, D_n$ are disjoint subsets of D. We can then define $f'|_{D_i}$ separately for each $i \in \{1, \ldots, n\}$.

By Definition 8.1, to define $f'|_{D_i}$ we need to choose some total ordering for each of the prime partitions in the modular decomposition of $\langle D_i, f \rangle$. Let A be a prime node in $\text{MD}(D_i, f)$, let $B = \{b \in D_j | \exists a \in A \text{ such that } \text{Feas}_{ij}(a, b) < \infty\}$ and let Feas_{ij}^A denote the restriction of Feas_{ij} to $A \times B$. Now $\text{Feas}_{ij}^A : A \times B \to \overline{\mathbb{R}}_+$ is domain-reduced since each value in A can be extended to a complete solution of $\mathcal{P}''$. Since A is prime, A is strongly connected with respect to f, so by Proposition 7.13, there are two possible cases: either (1) $\text{Feas}_{ij}^A = 0$ or (2) B is strongly connected with respect to f and there is a crisp-bijective binary constraint between the prime partitions of A and B. In the second case, B is a strongly connected module (by Lemma 7.6) and hence, by Lemma 6.3, B is a prime node in $\text{MD}(D_2, f)$. In this case, let $\overline{\text{Feas}_{ij}^A}$ represent the corresponding crisp-bijective induced crisp cost function on the prime partitions of A and B.

We use only these crisp-bijective functions $\overline{\text{Feas}_{ij}^A}$ to define f'. We repeat the following steps until f' has been defined on all the prime partitions in the modular decomposition of each domain. Choose some prime node A in $\text{MD}(D_i, f)$ for some i, such that f' has not yet been defined on the prime partition $\{A_1, \ldots, A_r\}$ of A. Choose an arbitrary ordering of $A_1, \ldots, A_r$. For each j such that $\text{Feas}_{ij}^A \neq 0$: let $\{B_1, \ldots, B_s\}$ be the prime partition of $B = \{b \in D_j | \exists a \in A \text{ such that } \text{Feas}_{ij}(a, b) < \infty\}$; choose the only possible

ordering f' of $\{B_1, \ldots, B_s\}$ such that the crisp-bijective function $\overline{\text{Feas}_{ij}^A}$ is an isomorphism with respect to f'. Because we ensured that $\mathcal{P}''$ was strong n-consistent, we can choose the ordering for one prime partition in one domain D_i arbitrarily, and then propagate this choice to all prime partitions of other domains whose ordering is now determined, without encountering any contradictions.

We repeat this arbitrary choice of ordering and propagation until we have fully defined the total ordering f'. At every step we simply choose an ordering for a prime partition of some domain, examine the binary constraints to neighbouring variables, and propagate as necessary. Since there is no backtracking involved, this process can be completed in polynomial time in the size of $\mathcal{P}$, and this completes the second stage of the proof.

Stage 3: It remains to show that every cost function ϕ of $\widehat{\mathcal{P}}$ is an element of $\text{Imp}(f', g')$, where g' is the dual of f'. Without loss of generality, assume that $\phi : D_1 \times D_2 \times \cdots \times D_k \to \overline{\mathbb{R}}_+$ and that ϕ is domain-reduced.

We need to show that the multimorphism inequality in Definition 3.6 (Equation 1) holds for the tournament pair $\langle f', g' \rangle$ for arbitrary $\overline{x}, \overline{y} \in D_1 \times \ldots \times D_k$. Since this inequality trivially holds if $\phi(\overline{x})$ or $\phi(\overline{y})$ is infinite, we assume in the following that $\phi(\overline{x})$ and $\phi(\overline{y})$ are both finite.

We know that this multimorphism equality holds for the tournament pair $\langle f, g \rangle$ (because $\phi \in \text{Imp}(f, g)$), so we only need to consider the case where $f'(\overline{x}, \overline{y})$ differs from $f(\overline{x}, \overline{y})$. In other words, when there is some $j \in \{1, \ldots, k\}$ such that $\overline{x}[j]$ and $\overline{y}[j]$ belong to distinct children of some prime node of $\text{MD}(D_j, f)$. Hence, we shall assume in the following, without loss of generality, that $\overline{x}[1] \in A_1^1$ and $\overline{y}[1] \in A_2^1$ where A_1^1, A_2^1 are distinct children of the prime node A^1 in $\text{MD}(D_1, f)$.

Let $A_1^1, \ldots, A_s^1$ be the prime partition of A^1. Without loss of generality, suppose that $\overline{x}[1]$,$\overline{y}[1]$ lie in the distinct parts A_1^1,A_2^1 (respectively) of this prime partition. Now set $a_1^1 = \overline{x}[1]$, $a_2^1 = \overline{y}[1]$ and select an arbitrary element a_i^1 from A_i^1 for each $i = 3, \ldots, s$. Since ϕ is domain-reduced, $\forall i \in \{3, \ldots, s\}$, $\exists \overline{a_i} = \langle a_i^1, \ldots, a_i^k \rangle \in D_1 \times \cdots \times D_k$ such that $\phi(\overline{a_i}) < \infty$. Set $\overline{a_1} = \overline{x}$ and $\overline{a_2} = \overline{y}$. Recall that $\phi(\overline{x})$ and $\phi(\overline{y})$ are also finite.

Let ϕ' denote the cost function ϕ restricted to domains $D_1' \times \cdots \times D_k'$ where $D_j' = \{a_i^j | i = 1, \ldots, s\}$ $(j \in \{1, \ldots, k\})$. Note that ϕ' is domain-reduced, since $\phi'(\langle a_i^1, \ldots, a_i^k \rangle) < \infty$ $(i = 1, \ldots, s)$. Also notice that D_1' is strongly connected, since $\{A_1^1, \ldots, A_s^1\}$ is strongly connected with respect to the induced tournament. Furthermore, the modular decomposition $\text{MD}(D_1', f)$ consists of a single prime node D_1' with prime decomposition $\{a_1^1\}, \ldots, \{a_s^1\}$. Since $\overline{x}, \overline{y} \in D_1' \times \cdots \times D_k'$, it is sufficient to show that

$\phi' \in \mathrm{Imp}(f', g')$.

Now consider the binary projections $\phi'_{1j} = \Pi_{1j}(\phi')$, for $j \in \{1, \ldots, k\}$. Since D'_1 is strongly connected, for each $j = 2, \ldots, k$, it follows from Proposition 7.13 that either (a) ϕ'_{1j} is finite or (b) the induced crisp cost function $\overline{\phi'_{1j}}$ is crisp-bijective. Without loss of generality, suppose that $\overline{\phi'_{1j}}$ is crisp-bijective for $j = 2, \ldots, l$ and ϕ'_{1j} is finite for $j = l+1, \ldots, k$. By Lemma 7.4, D'_j $(j = 1, \ldots, l)$ are all isomorphic to D'_1 (with a single prime node D'_j with decomposition $\{a^j_1\}, \ldots, \{a^j_s\}$ in $\mathrm{MD}(D'_j, f)$).

It follows that ϕ' can be expressed as the sum of the crisp binary crisp-bijective cost functions ϕ'_{1j} $(j = 2, \ldots, l)$ and a cost function $\psi : D'_1 \times D'_{l+1} \times \cdots \times D'_k \to \overline{\mathbb{R}}_+$. Hence it is sufficient to show that (a) $\phi'_{1j} \in \mathrm{Imp}(f', g')$ for $j \in \{2, \ldots, l\}$ and (b) $\psi \in \mathrm{Imp}(f', g')$.

We will first show that $\phi'_{1j} \in \mathrm{Imp}(f', g')$ for $j \in \{2, \ldots, l\}$. We know that A^1 is strongly connected, since it is a prime node of $\mathrm{MD}(D_1, f)$. Consider some $j \in \{2, \ldots, l\}$ and let $A^j = \{b \in D_j | \exists a \in A^1$ such that $\mathrm{Feas}_{1j}(a, b) < \infty\}$. By Lemma 7.6, A^j is a module. Recall that $\mathrm{Feas}^{A^1}_{1j}$ denotes the restriction of Feas_{1j} to $A^1 \times A^j$. Now $\mathrm{Feas}^{A^1}_{1j}$ cannot be finite, since when restricted to domains $D'_1 \times D'_j$ it becomes crisp-bijective. Therefore, by Proposition 7.13, A^j is strongly connected and the induced crisp cost function $\overline{\mathrm{Feas}^{A^1}_{1j}}$ on the prime partitions of A^1 and A^j is crisp-bijective. Furthermore, by Lemma 6.3, A^j is a prime node in $\mathrm{MD}(D_j, f)$. It follows that $\overline{\mathrm{Feas}^{A^1}_{1j}}$ is an isomorphism of f', by the definition of f' in stage 2. Since ϕ'_{1j} is also crisp-bijective, and D'_1 contains exactly one element from each of the parts of the prime partition of A^1, ϕ'_{1j} must also be an isomorphism of f'. Thus, by Lemma 7.4, $\phi'_{1j} \in \mathrm{Imp}(f', g')$.

Next we consider the function ψ. Since each binary projection $\Pi_{1j}(\psi)$ is finite, for $j = l+1, \ldots, k$, we can deduce, by Lemma 7.15, that $\psi = \psi_1 + \psi_2$ where $\psi_1 : D'_1 \to \overline{\mathbb{R}}_+$ is a unary function and $\psi_2 : D'_{l+1} \times \ldots \times D'_k \to \overline{\mathbb{R}}_+$ is a $(k-l)$-ary domain-reduced function in $\mathrm{Imp}(f, g)$.

Now we know that $\psi_1 \in \mathrm{Imp}(f', g')$, since all unary cost functions belong to $\mathrm{Imp}(f', g')$. The cost function ψ_2 satisfies all of the relevant properties of ϕ, but has a lower arity. Hence, by repeating the argument, as necessary, we can continue to reduce the arity until we obtain the result. ∎

Theorem 8.3 *If $\langle f, g \rangle$ is a tournament pair, then $\mathrm{Imp}(f, g)$ is tractable.*

Proof: Consider $\mathcal{P} \in \mathrm{VCSP}(\mathrm{Imp}(f, g))$. Let $\mathrm{Feas}(\mathcal{P})$ denote the CSP instance obtained from $\mathcal{P}$ by replacing each cost function ψ by the constraint

relation Feas(ψ). By Lemma 5.2, the constraint relations of Feas(P) all have the polymorphisms f and g. For each variable v of Feas($\mathcal{P}$), and each value $d \in D$, let Feas($\mathcal{P}$)$_{v=d}$ denote the CSP instance Feas($\mathcal{P}$) with the additional constraint $\langle\langle v\rangle, \{\langle d\rangle\}\rangle$ (i.e., the variable v must be assigned the value d). Each such additional constraint also has the polymorphisms f and g.

Any class of CSP instances where all relations have a conservative commutative polymorphism can be solved in polynomial time [5]. For each variable v and for each value d, if Feas($\mathcal{P}$)$_{v=d}$ has no solution, then we eliminate d from the domain of v. Let $\mathcal{P}'$ denote the resulting VCSP instance. Clearly $\mathcal{P}' \in \mathrm{VCSP}(\mathrm{Imp}(f, g))$.

Suppose that the domain of the ith variable in $\mathcal{P}'$ contains a pair of values a, b such that $f(a, b) = g(a, b) = b$. Let $\overline{x}^a$ (respectively $\overline{x}^b$) represent an optimal solution to $\mathcal{P}'$ such that $\overline{x}[i] = a$ (respectively $\overline{x}[i] = b$). Let ϕ denote $\mathrm{Cost}_{\mathcal{P}'}$, the function obtained by summing all the cost functions of $\mathcal{P}'$. By construction of $\mathcal{P}'$, $\phi(\overline{x}^a) < \infty$ and $\phi(\overline{x}^b) < \infty$. Let $\overline{y} = f(\overline{x}^a, \overline{x}^b)$ and $\overline{z} = g(\overline{x}^a, \overline{x}^b)$, where f and g are applied componentwise. Since ϕ has the multimorphism $\langle f, g\rangle$, we have

$$\phi(\overline{y}) \;+\; \phi(\overline{z}) \;\leq\; \phi(\overline{x}^a) \;+\; \phi(\overline{x}^b)$$

Now $\overline{y}[i] = f(a, b) = b$ and $\overline{z}[i] = g(a, b) = b$. Thus, by definition of $\overline{x}^b$, $\phi(\overline{x}^b) \leq \phi(\overline{y})$ and $\phi(\overline{x}^b) \leq \phi(\overline{z})$. Therefore $2\phi(\overline{x}^b) \leq \phi(\overline{x}^a) + \phi(\overline{x}^b)$. Hence, since $\phi(\overline{x}^b)$ is finite, we have $\phi(\overline{x}^b) \leq \phi(\overline{x}^a)$. It follows that the value a for the ith variable is unnecessary in the search for a single optimal solution to $\mathcal{P}'$.

Therefore, we can eliminate all values a from the domain of a variable v such that there exists b with $f(a, b) = g(a, b) = b$ in the domain of the same variable. On these restricted domains $\langle f, g\rangle$ is a *symmetric* tournament pair, so the result follows from Theorem 8.2. ∎

9 Examples of new tractable classes

In this section we will present examples of novel tractable sets of cost function which are characterised by having a tournament pair multimorphism. Our first example is closely related to the set of submodular functions, but contains cost functions which are not submodular under any permutation of the domain D.

Example 9.1 Let $D = \{1, 2, 3\}$ and consider the tournament operation defined by $f(1, 2) = 1$, $f(2, 3) = 2$, $f(3, 1) = 3$, corresponding to a cyclic

tournament on $D = \{1, 2, 3\}$. Let g be the dual of f (i.e., $g(1, 2) = 2$, $g(2, 3) = 3$, $g(3, 1) = 1$)).

The set $\mathrm{Imp}(f, g)$ contains 3 types of cost functions:

1. all *unary* cost functions $\phi : D \to \overline{\mathbb{R}}_+$.
2. the three binary *cyclic permutations* $\pi_k : D^2 \to \overline{\mathbb{R}}_+$ $(k = 0, 1, 2)$ given by

$$\pi_k(x, y) \quad = \quad \begin{cases} 0 & \text{if } y \equiv x + k \pmod 3 \\ \infty & \text{otherwise} \end{cases}$$

3. cost functions ϕ such that $Feas(\phi) \subseteq D_1 \times \ldots \times D_r$ with $|D_i| \leq 2$, for all $i \in \{1, \ldots, r\}$,

Notice that no re-ordering of the domain can render π_1 submodular. The proof of Theorem 8.2 shows that (after establishing strong 3-consistency) any problem instance $\mathcal{P} \in \mathrm{VCSP}(\mathrm{Imp}(f, g))$ is equivalent to an instance $\mathcal{P}'$ with only these types of cost functions and such that the two subproblems on

1. the set V_3 of variables whose domains are of size 3
2. the set V_2 of variables whose domains are of size 2 or less

form two independent optimization problems. The former is in fact a collection of independent optimization problems on the connected components of the graph whose nodes are the variables V_3 with variables v_i, v_j joined by an edge if and only if there is a cyclic permutation constraint between v_i and v_j in $\mathcal{P}'$. Each of these optimization problems are trivially solvable by exhaustion over at most 3 possible solutions. The associated optimization problem on V_2 can be transformed into a submodular function minimization (SFM) problem by renaming the domain value 1 in domains $D_i = \{1, 3\}$. By renaming 1 as 4, the resulting cost functions on the variables in V_2 are submodular under the usual total order $1 < 2 < 3 < 4$. The tractability of $\mathrm{VCSP}(\mathrm{Imp}(f, g))$ then follows from the tractability of SFM over non-Boolean domains [12] (which is a straightforward generalization of the tractability of SFM over Boolean domains [50, 34, 32]). □

Example 9.2 As an example of a tractable class of valued constraints with a non-symmetric tournament multimorphism, consider $\mathrm{Imp}(f, h)$ where f is given by $f(1, 2) = 1$, $f(2, 3) = 2$, $f(3, 1) = 3$ (as in Example 9.1) and h is the tournament operation given by $h(1, 2) = 2$, $h(2, 3) = 2$, $h(3, 1) = 1$.

As shown in the proof of Theorem 8.3, after establishing generalized arc consistency [46] (i.e., eliminating from domains values which have no finite extension in some valued constraint, this operation being repeated until convergence), we can eliminate the value 3 from any domain containing the values 2 and 3. The resulting VCSP instance is an instance of SFM over a collection of 2-valued domains, which again is tractable. □

10 Conclusion

In this paper we have shown that it is possible to unify and extend the tractable problems of Horn clause satisfiability and submodular function minimization via the investigation of tournament multimorphisms.

Over Boolean domains, there remain two other important tractable constraint classes, corresponding to 2-SAT and linear equations [8, 12]. These classes can both be characterised by having *ternary* multimorphisms. Therefore an obvious avenue of future research is the extension of tournament multimorphisms to ternary multimorphisms. Bulatov has already shown [4] that if Γ is a tractable class of *crisp* cost functions containing all unary restrictions, then $\Gamma \subseteq \mathrm{Imp}(f, g, h)$ for some ternary multimorphism $\langle f, g, h \rangle$. It is an open question whether this generalizes to arbitrary (non-crisp) cost functions.

On a more practical level, it is known that SFM over a Boolean domain can be solved in $O(n^3)$ time when the submodular function ϕ is cubic [2], or ϕ is (0,1)-valued [18, 19, 40]. Minimizing a symmetric submodular set function among proper non-empty subsets can also be achieved in $O(n^3)$ time [47, 45]. In the case of non-Boolean domains, a cubic-time algorithm exists for SFM when ϕ is the sum of binary submodular functions [10] or when ϕ is the sum of certain classes of (0,1)-valued functions over a lattice [11, 39, 41]. In each case, the cubic-time complexity is obtained by a reduction to the Min-Cut problem. An obvious avenue of future research is to determine which of these cubic-time classes generalizes to arbitrary tournament pair multimorphisms.

References

[1] Anglès d'Auriac, J-Ch., Igloi, F., Preismann, M. & Sebö, A. "Optimal cooperation and submodularity for computing Potts' partition functions with a large number of statistics", *Journal of Physics A: Math. Gen.* 35 (2002), pp. 6973–6983.

[2] Billionet, A. & Minoux, M. "Maximizing a supermodular pseudo-boolean function: a polynomial algorithm for cubic functions", *Discrete Applied Mathematics* 12 (1985) pp. 1–11.

[3] Bulatov, A.A. "A dichotomy theorem for constraint satisfaction problems on a three-element set", *Journal of the ACM* 53(1) (2006) pp. 66–120.

[4] Bulatov, A.A. "Tractable conservative constraint satisfaction problems", *Proc. 18th IEEE Symposium on Logic in Computer Science (LICS'03)*, (2003) pp. 321–330.

[5] Bulatov, A.A. "Combinatorial problems raised from 2-semilattices", *Journal of Algebra*, (2006) pp. 321–339.

[6] Bulatov, A.A. & Jeavons, P. "Tractable constraints closed under a binary operation", Oxford University Computing Laboratory Technical Report, PRG-TR-12-00, (2000).

[7] Bulatov, A.A., Krokhin, A.A. & Jeavons, P.G. "Classifying the complexity of constraints using finite algebras" *SIAM Journal on Computing*, 34 (2005) pp. 720–742.

[8] Cohen, D., Cooper, M.C. & Jeavons, P. "A complete characterisation of complexity for Boolean constraint optimization problems", *Proc. 10th Int. Conf. on Priciples and Practice of Constraint Programming (CP'04)*, *LNCS* 3258 (2004) pp. 212–226.

[9] Cohen, D., Cooper, M.C., Jeavons, P. & Krokhin, A. " Soft constraints: complexity and multimorphisms", *Proc. CP'03*, *LNCS* 2833 (2003) pp. 244–258.

[10] Cohen, D., Cooper, M.C., Jeavons, P. & Krokhin, A. "A maximal tractable class of soft constraints", *Journal of Artificial Intelligence Research* 22 (2004) pp. 1–22.

[11] Cohen, D., Cooper, M.C., Jeavons, P. & Krokhin, A. "Supermodular functions and the complexity of Max-CSP" *Discrete Applied Mathematics* 149 (2005) pp. 53–72.

[12] Cohen, D., Cooper, M.C., Jeavons, P. & Krokhin, A. "The complexity of soft constraint satisfaction", *Artificial Intelligence* 170(11) (2006) pp. 983–1016.

[13] Cohen, D., Cooper, M.C. & Jeavons, P. "An algebraic characterisation of complexity for valued constraints", *Proc. CP'06*, *LNCS* 4204 (2006) pp. 107–121.

[14] Cooper, M.C. "An optimal k-consistency algorithm", *Artificial Intelligence* 41 (1989) pp. 89–95.

[15] Cooper, M.C. "Reduction operations in fuzzy and valued constraint satisfaction", *Fuzzy Sets and Systems* 134 (2003) pp. 311–342.

[16] Cooper, M.C. "High-order consistency in valued constraint satisfaction", *Constraints* 10 (2005) pp. 283–305.

[17] Cooper, M.C. & Schiex, T. "Arc consistency for soft constraints", *Artificial Intelligence* 154(1-2) (2004) pp. 199–227.

[18] Creignou, N. "A dichotomy theorem for maximum generalised satisfiability problems", *Journal of Computer and Systems Sciences* 51(3), (1995) pp. 511–522.

[19] Creignou, N., Khanna, S. & Sudan, M. *Complexity classification of Boolean constraint satisfaction problems*, volume 7 of *SIAM Monographs on Discrete Mathematics and Applications* (2001).

[20] Cunningham, W.H. "Minimum cuts, modular functions, and matroid polyhedra", *Networks* 15(2) (1985) pp. 205–215.

[21] Dahlhaus, E., Gustedt, J. & McConnell R.M. "Efficient and practical algorithms for sequential modular decomposition", *Journal of Algorithms* 41 (2) (2001) pp. 360–387.

[22] Dechter, R. *Constraint Processing*, Morgan Kaufmann, 2003.

[23] Dowling, W.F. & Gallier, J.H. "Linear-time algorithms for testing the satisfiability of propositional Horn formulae", *Journal of Logic Programming* 1(3) (1984) pp. 267–284.

[24] Ekin, O., Hammer, P.L. & Peled, U.N. "Horn functions and submodular boolean functions", *Theoretical Computer Science* 175 (1997) pp. 257–270.

[25] Feder, T. & Vardi, M.Y. "The computational structure of monotone monadic SNP and constraint satisfaction: a study through Datalog and group theory", *SIAM Journal on Computing* 28(1), (1998) pp. 57–104.

[26] Fujishige, S., *Submodular Functions and Optimisation*, 2nd edn., Annals of Discrete Mathematics 58, Elsevier, 2005.

[27] Fujishige, S. & Patkar, S.B. "Realization of set functions as cut functions of graphs and hypergraphs", *Discrete Mathematics* 226 (2001) pp. 199–210.

[28] Fujishige, S. & Iwata, S. "Bisubmodular Function Minimization", *IPCO 2001*, K. Aardal & B. Gerards (Eds.), *LNCS* 2081, (2001) pp. 160–169.

[29] Gutin, G., Rafiey A. & Yeo, A. "Minimum Cost and List Homomorphisms to Semicomplete Digraphs", *Discrete Applied Mathematics* 154 (2006) pp. 890–897.

[30] Grötschel, M., Lovász, L. & Schrijver, A. "The ellipsoid method and its consequences in combinatorial optimization", *Combinatorica* 1 (1981) pp. 169–198; corrigendum: *Combinatorica* 4 (1984) pp. 291–295.

[31] Han, C.-C. & Lee, C.-H. "Comments on Mohr and Henderson's path consistency algorithm", *Artificial Intelligence* 36 (1988) pp. 125–130.

[32] Iwata, S. "A fully combinatorial algorithm for submodular function minimization", *Journal of Combinatorial Theory, Series B* 84(2), (2002) pp. 203–212.

[33] Iwata, S. "A faster scaling algorithm for minimizing submodular functions", *SIAM J. Comput.* 32(4) (2003) pp. 833-840.

[34] Iwata, S., Fleischer, L. & Fujishige, S. "A combinatorial, strongly polynomial-time algorithm for minimizing submodular functions", *Journal of the ACM* 48(4), (2001), pp. 761–777.

[35] Jeavons, P.G. "On the algebraic structure of combinatorial problems", *Theoretical Computer Science* 200 (1998) pp. 185–204.

[36] Jeavons, P., Cohen, D. & Cooper M.C. "Constraints, consistency and closure", *Artificial Intelligence* 101 (1998) pp. 251–265.

[37] Jeavons, P.G., Cohen D.A. & Gyssens, M. "Closure properties of constraints", *Journal of the ACM* 44 (1997) pp. 527–548.

[38] Jeavons, P.G. & Cooper, M.C. "Tractable constraints on ordered domains", *Artificial Intelligence* 79 (2), (1995) pp. 327–339.

[39] Jonsson, P., Klasson, M. & Krokhin, A. "The approximability of three-valued MAX CSP", *SIAM Journal on Computing*, 35 (6), (2006) pp. 1329–1349.

[40] Khanna, S., Sudan, M., Trevisan, L. & Williamson, D. "The approximability of constraint satisfaction problems", *SIAM Journal on Computing* 30 (6), (2001) pp. 1863–1920.

[41] Krokhin, A. & Larose, B. "Maximum constraint satisfaction on diamonds", *Proceedings CP'05*, *LNCS* 3709 (2005) pp. 388–402.

[42] Larrosa, J. & Schiex, T. "Solving weighted CSP by maintaining arc consistency", *Artificial Intelligence* 159 (2004) pp. 1–26.

[43] McConnell, R.M. & de Montgolfier, F., "Linear-time modular decomposition of directed graphs", *Discrete Applied Mathematics* 145(2) (2005) pp. 198–209.

[44] Narayanan, H, *Submodular Functions and Electrical Networks*, North-Holland, Amsterdam (1997).

[45] Narayanan, H. "A note on the minimization of symmetric and general submodular functions" *Discrete Applied Mathematics* 131(2) (2003) pp. 513-522.

[46] Mohr, R. & Masini, G. "Good old discrete relaxation", *Proc. European Conf. Artificial Intelligence*, Munich (1988) pp. 651–656.

[47] Queyranne, M. "Minimising symmetric submodular functions", *Mathematical Programming* 82 (1-2) (1998) pp. 3-12.

[48] Schaefer, T.J. "The complexity of satisfiability problems" *Proceedings of the 10th ACM Symposium on the Theory of Computing*, *STOC'78* (1978) pp. 216–226.

[49] Schiex, T., Fargier, H. & Verfaillie, G. "Valued constraint satisfaction problems: hard and easy problems", *Proc. of the 14th IJCAI*, Montreal, Canada (1995) pp. 631–637.

[50] Schrijver, A. "A combinatorial algorithm minimizing submodular functions in strongly polynomial time", *Journal of Combinatorial Theory*, Ser. B, 80 (2000) pp. 346–355.

[51] Topkis, D. *Supermodularity and Complementarity*, Princeton University Press (1998).