Keywords: Multi-objective Valued Constraint Satisfaction Problems, Tractable Class, Directional Substitutable Valuation Functions, Decomposition Scheme for General MOVCSP.

Abstract: To better model several artificial intelligence and combinatorial problems, classical Constraint Satisfaction Problems (CSP) have been extended by considering soft constraints in addition to crisp ones. This gave rise to a Valued Constraint Satisfaction Problems (VCSP). Several real-world artificial intelligence and combinatorial problems require more than one single objective function. In order to present a more appropriate formulation for these real-world problems, a generalization of the VCSP framework called Multi-Objective Valued Constraint Satisfaction Problems (MOVCSP) has been proposed. This paper addresses combinatorial optimization problems that can be expressed as MOVCSP. Despite the NP-hardness of general MOVCSP, we can present tractable versions by forcing the allowable valuation functions to have specific mathematical properties. This is the case for MOVCSP whose dual is a binary MOVCSP with crisp binary valuation functions only and with a weak form of Neighbourhood Substitutable Valuation Functions called Directional Substitutable Valuation Functions.

1 INTRODUCTION

Constraint Satisfaction Problems (CSP) provide a general and convenient framework to model and solve numerous combinatorial problems including temporal reasoning (van Beek and Manchak, 1996), computer vision (Schlesinger, 2007). In the standard CSP framework, the constraints are defined by crisp relations, which specify the consistent combinations of values. With these relations, one can force some pairs of intervals to overlap, and any plan that does not meet this requirement is considered as inconsistent even though the intervals are very close.

However, one may need to express various degrees of consistency in order to reflect the specificity of the problem at hand. The valued constraint satisfaction problems (VCSPs) approach (Schiex et al., 1995) is intended to model such situations. A VCSP consists of a set of variables taking values in discrete sets called domains. A valued constraint is defined through the use of a valuation function. The role of a valuation function is to associate a degree of desirability to each combination of values. The problem is to find an assignment of values to variables from their respective domains with an optimal cost.

The computational complexity of finding the optimal solution to a VCSP has been largely studied in many works and several classes of tractable VCSPs, that is, VCSPs that are solvable in polynomial time, have been identified and solved. Tractability is obtained by limiting the set of allowed valuation functions and or by detecting some desirable properties exhibited by the problem structure (Cohen et al., 2008a; Cohen et al., 2008b; Greco and Scarcello, 2011; Cohen et al., 2012; Cooper and Zivny, 2011; Cooper and Zivny, 2012; Helaoui and Naanaa, 2013; Helaoui et al., 2013; Cooper et al., 2016; Carbonnel and Cooper, 2016).

However, in real-world situations like the discrete time/cost trade-off problem (Vanhoucke, 2005; Debels and Vanhoucke, 2007; Tavana et al., 2014), one may need to express multiple objectives to optimize in order to reflect the specificity of the problem at hand (Greco and Scarcello, 2013).

Incorporating conflicting objective functions divide the solution set into dominated and non-dominated solutions. With reference to Pareto, Non Dominated Solutions (NDS) are solutions where we cannot improve further the attainability of one objective without degrading the attainability of another, which means that a compromise should be found.
Table 1: The Π project.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Predecessors</th>
<th>choice 1</th>
<th>choice 2</th>
<th>choice 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA</td>
<td>–</td>
<td>(15,10)</td>
<td>(9,25)</td>
<td>(3,50)</td>
</tr>
<tr>
<td>TB</td>
<td>TA</td>
<td>(15,10)</td>
<td>(12,30)</td>
<td>(6,90)</td>
</tr>
<tr>
<td>TC</td>
<td>TA</td>
<td>(15,10)</td>
<td>(9,35)</td>
<td>(6,25)</td>
</tr>
<tr>
<td>TD</td>
<td>TA</td>
<td>(30,20)</td>
<td>(24,50)</td>
<td>(21,80)</td>
</tr>
<tr>
<td>TF</td>
<td>TB,TC</td>
<td>(15,10)</td>
<td>(9,30)</td>
<td>(3,60)</td>
</tr>
<tr>
<td></td>
<td>TD,TF</td>
<td>(15,10)</td>
<td>(12,58)</td>
<td>(6,250)</td>
</tr>
</tbody>
</table>

Example 1. Discrete Time Cost Trade Off Problem (De et al., 1997).

Let Π be a project defined as follows:
Π is comprised of 6 tasks: TA, TB, TC, TD, TE, and TF. The predecessors of each task are defined by column “Predecessor” of Table 1.
The various options of the execution times and the relative costs of each task are given in columns 3, 4, and 5. For instance, Task TA could be executed in 15 time units with cost 10 or in 9 time units with cost 25 or even in 3 time unit but the cost rise to 50. Solving the problem amounts to finding, for each task, one choice such that both global costs and global makespan are optimized and the precedence constraints are satisfied.

The multi-objectives valued constraint satisfaction problems (MOVCP) presented in (Ali et al., 2019) is intended to model such situations. A Multi-objectives VCSP consists in a VCSP where the goal is to find an assignment of values to variables, from their respective domains, with an optimal multi-objectives valuations. Solving a problem with several multi-objectives functions is commonly referred to as multi-objective problem. The goal is to compute the best set of compromise solutions called Pareto borders.

Furthermore, interchangeability and substitutability are two techniques that have been initially introduced for CSP (Freuder, 1991). In (Lecoutre et al., 2012), Neighbourhood Substitutability has been extended to VCSP. A decomposition directional substitutability algorithm that applies when the studied problem does not satisfy the conditions of interchangeability or substitutability has been proposed in (Naanaa, 2008; Naanaa et al., 2009) respectively for CSP and CSOP: a VCSP with Crisp binary Constraint.

In this paper, we present tractable versions of MOVCP by forcing the allowable valuation functions to have specific mathematical properties. This is the case for MOVSCP whose dual is a binary crisp MOVCP with crisp binary valuation functions only and with Directional Substitutable Valuation Functions. We denote this MOVCP class by $L(MODS)$. We also take advantage of the discovered tractable class to conceive a decomposition scheme for general MOVCP.

The paper is organized as follows: the next Section introduces MOVCP. In Section 3 we study Soft Directional Substitutable MOVCP. We conclude in Section 4.

2 MULTI-OBJECTIVE VCSP

In a MOVCP, and as for a VCSP (Schiex et al., 1995), for each objective $j = 1, 2, \ldots, k$, we assume a set $E_j$ of possible valuations which is a totally ordered with a minimal element $\perp_j$ and a maximal element $\top_j$. In addition, we need $k$ monotone operators $\oplus_j, j = 1, \ldots, k$. These components can be gathered in $k$ valuation structures each of which can be specified as follows:

Definition 1. A valuation structure $S_j$ is the triple $S_j = (E_j, \oplus_j, \preceq_j)$, where
- $E_j$ is a set of valuations for the objective function $j$;
- $\preceq_j$ is a total order on $E_j$;
- $\oplus_j$ is commutative, associative and monotone binary operator.

Once the valuation structure $S$ is specified, the multi-objective valued constraint satisfaction problem (MOVCP) can be defined as follows:

Definition 2. A multi-objective valued constraint satisfaction problem denoted (MOVCP) is defined by the tuple $(X, D, C, S)$ such as:
- $X$ is a finite set of variables;
- $D$ is a finite set of value domain, such that $D_x \in D$ denotes the domain of $x \in X$.
- $S = (S_1, \ldots, S_k)$, where each $S_j$ is a valuation structure of objective $j$;
- $C$ is a set of valued constraints. Each constraint is an ordered pair $(\sigma, \Phi)$, where $\sigma \subseteq X$ is the scope of the constraint and $\Phi$ is a $k$-functions vector $\langle \phi_1, \ldots, \phi_k \rangle$, where each function $\phi_j$ is from $\Pi_{e \in \sigma} D_e$ to $E_j$.

The arity of a multi-objective valued constraint is the size of its scope. The arity of a MOVCP is the maximum over the arities of all its constraints.

To simplify the notation, and if there is no confusion, we will denote each $\oplus_j$ by $\oplus$, each $\preceq_j$ by $\preceq$, and each $\phi_j$ by $\phi$.

The valuation of an assignment $t$ that assigns values to a subset of variables $V \subseteq X$ is obtained by
\[ \Phi(t) = \left( \bigoplus_{(\sigma, \varphi_1) \in C, \sigma \subseteq V} \varphi_1(t \downarrow \sigma), \ldots, \bigoplus_{(\sigma, \varphi_k) \in C, \sigma \subseteq V} \varphi_k(t \downarrow \sigma) \right) \]

Where \( t \downarrow \sigma \) denotes the projection of \( t \) on the variables of \( \sigma \). Hence, an optimal solution of a MOVCSP on \( n \) variables is a \( n \)-tuple \( t \) such that \( \Phi(t) \) is optimal over all possible \( n \)-tuples.

In order to simplify the notation we denote \( v_1, \ldots, v_\sigma \) by \( v_\sigma \).

**Definition 3.** Let \( t_1 \) and \( t_2 \) two solutions.

- We say that solution \( t_1 \) dominates solution \( t_2 \) \( (t_1 \succeq_D t_2) \) if, for each objective \( j \), we have
  \[ \bigoplus_{(\sigma, \varphi) \in C} \varphi_j(t_1 \downarrow \sigma) \leq \bigoplus_{(\sigma, \varphi) \in C} \varphi_j(t_2 \downarrow \sigma) \]
  with at least one objective, we have a strict inequality.

- We say that \( t_1 \) and \( t_2 \) are two Non Dominated Solutions (NDS) if there are two objectives \( j \) and \( j' \) \( \in k \), such that
  \[ \bigoplus_{(\sigma, \varphi) \in C} \varphi_j(t_1 \downarrow \sigma) \leq \bigoplus_{(\sigma, \varphi) \in C} \varphi_j(t_2 \downarrow \sigma) \wedge \]
  \[ \bigoplus_{(\sigma, \varphi) \in C} \varphi_{j'}(t_2 \downarrow \sigma) > \bigoplus_{(\sigma, \varphi) \in C} \varphi_{j'}(t_1 \downarrow \sigma) \]

This allows us to define a partial order between the dominated solutions.

**Lemma 1.** \( \succeq_D \) is transitive.

**Proof Lemma:** According to the Definition 3, we have:

- (i) \( t_1 \succeq_D t_2 \) if and only if for each objective \( j \)
  \[ \bigoplus_{(\sigma, \varphi) \in C} \varphi_j(t_1 \downarrow \sigma) \leq \bigoplus_{(\sigma, \varphi) \in C} \varphi_j(t_2 \downarrow \sigma) \]

- (ii) \( t_2 \succeq_D t_3 \) if and only if for each objective \( j \)
  \[ \bigoplus_{(\sigma, \varphi) \in C} \varphi_j(t_2 \downarrow \sigma) \leq \bigoplus_{(\sigma, \varphi) \in C} \varphi_j(t_3 \downarrow \sigma) \]

- (i) and (ii) imply that for each objective \( j \)
  \[ \bigoplus_{(\sigma, \varphi) \in C} \varphi_j(t_1 \downarrow \sigma) \leq \bigoplus_{(\sigma, \varphi) \in C} \varphi_j(t_3 \downarrow \sigma) \]

Hence \( t_1 \succeq_D t_3 \).

**Example 2.** We will return to the same DTCT project \( \Pi \) presented in Example 1. This project \( \Pi \) can be modelled as a bi-objectives VCSP \( \Pi_1 \) defined such that:

1. \( X \) is a finite set of variables such that each \( x_i \) is a task \( i \);
2. \( \mathcal{D} = \{v_1, v_2, v_3\} \) is a set of finite domains, where \( v_{\text{choice}} \in \mathcal{D} \) denotes the value \( v_{\text{choice}} \) of the variable \( x_i \);
3. \( S = (E, \oplus, \preceq) \) is a fair valuation structure, where \( \oplus \) is the sum and \( E \) is the set of integers.
4. \( C \) is a set of valued constraints. Each unary valued constraint \( C \) is an ordered pair \((\langle x_i \rangle, \Phi(v_{\text{choice}}) = (\Phi_1(v_{\text{choice}}), \Phi_2(v_{\text{choice}})))\).

We get \( \Pi_1 = (X, \mathcal{D}, S, (\langle x_i \rangle, \Phi)) \)

The predecessors\(^2\) of each task are defined in the second column of Table 2. The valuation functions of the bi-objectives VCSP \( \Pi_1 \) are given in columns 3, 4 and 5.

<table>
<thead>
<tr>
<th>( X )</th>
<th>Predecessors</th>
<th>( \Phi(v_1) )</th>
<th>( \Phi(v_2) )</th>
<th>( \Phi(v_3) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_A )</td>
<td>( x_B )</td>
<td>(15,10)</td>
<td>(9,25)</td>
<td>(3,50)</td>
</tr>
<tr>
<td>( x_A )</td>
<td>( x_B )</td>
<td>(15,10)</td>
<td>(12,30)</td>
<td>(6,90)</td>
</tr>
<tr>
<td>( x_A )</td>
<td>( x_C )</td>
<td>(15,10)</td>
<td>(9,35)</td>
<td>(6,60)</td>
</tr>
<tr>
<td>( x_B )</td>
<td>( x_D )</td>
<td>(30,20)</td>
<td>(24,50)</td>
<td>(21,80)</td>
</tr>
<tr>
<td>( x_B )</td>
<td>( x_E )</td>
<td>(15,10)</td>
<td>(9,30)</td>
<td>(3,60)</td>
</tr>
<tr>
<td>( x_D )</td>
<td>( x_E )</td>
<td>(15,10)</td>
<td>(12,58)</td>
<td>(6,250)</td>
</tr>
</tbody>
</table>

\(^2\)In graph theory, the precedence constraints of \( \Pi_1 \) are satisfied by using RANK algorithm in order to give the variables order of execution.

### 3 Tractable Class for MOVCSPs

If it is more easy to generalize tractability from VCSP to MOVCSP with Soft Neighbourhood Substitutable Valuation Functions only. What about MOVCSP with a weak form of Soft Neighbourhood Substitutable called Soft Directional Substitutable Valuation Functions? In this section, we present tractable class for MOVCSPs that take advantage of Soft Directional Substitutable Valuation Functions.

#### 3.1 The Power of a Binary Crisp MOV CSP

We present the power of a binary MOV CSP with crisp binary valuation functions only motivated by the fact that

1. Any binary MOV CSP with only modular binary functions and any crisp binary functions can be transformed in polynomial time to an equivalent
2. The dual problem of any MOVCSp is a binary MOVCSp with crisp binary valuation functions only.

We define a binary CSOP as a binary VCSP such that binary valuations are only in \{±1, ±\}. A binary MOVCSp is a binary MOVCSp such that binary valuations are only in \{±j, ±\} for each objective \( j \).

Let, for each objective \( j \), \( \phi_j : D \times D' \rightarrow E_j \) be a binary function which is not necessarily modular. In the following, we show that restricting the first argument of \( \phi_j \) to specific subsets of \( D \) yields a family of modular binary functions.

**Definition 4.** Let, for any objective \( j \), \( \phi_j : D \times D' \rightarrow E_j \) be a binary function and let \( a, b \) be in \( D \). We say that \( a \) and \( b \) are modular with regard to all \( \phi_j \), \( a \sim_{a,b} b \), if and only if the restriction of all \( \phi_j \) to \( \{a, b\} \times D' \) is modular.

We note the class of MOVCSp with only modular binary functions and any crisp binary functions \( L_2(M) \). Note that

\[
\Phi \in L_2(M) \iff \forall a, b \in D \land \forall j \in k \ \Phi(a, b) = \Phi(b, a)
\]

(2)

**Lemma 2.** Let \( P \) a binary MOVCSp. If \( \Phi \in L_2(M) \) then it exists a polynomial transformation \( \rho \) such that \( \rho(P) = \) binary MOCSop.

**Proof Lemma:** By applying DECOMPOSE algorithm presented in (Helaoui and Naama, 2013) for each objective \( j \) to any binary MOVCSp with only modular binary functions and any crisp binary functions we get a binary MOCSop. Since DECOMPOSE algorithm run on \( O(ed) \) (where \( e \) is the number of constraints, and \( d \) is the size of the largest value domain) and it must be called for each objective \( j \). Then \( \rho \) can be done on \( O(ked) \). We can conclude that \( \rho \) is a polynomial transformation.

The dual problem of a MOVCSp is a reformulation of the problem that expresses each constraint of the original problem as a variable. The dual problems contain only unary and binary constraints and therefore are binary problems. Therefore, it is possible to apply the known algorithms for such problems.

The dual problem of a MOVCSp is a reformulation of the latter which considers each constraint of the original problem as a variable. The unary constraint associated with such variable specifies unary and binary costs and costs given by the unary and binary constraints of the original problem.

Binary constraints of the dual problem express the fact that the variables common to two constraints must have the same value.
Table 4: Unary cost valuation functions of the dual problem $P^*$.

<table>
<thead>
<tr>
<th>Values</th>
<th>Unary-</th>
<th>(1,1)</th>
<th>(1,2)</th>
<th>(2,1)</th>
<th>(2,2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Phi$</td>
<td>$\alpha_1, \alpha_2$</td>
<td>$1,1$</td>
<td>$1,2$</td>
<td>$2,1$</td>
<td>$2,2$</td>
</tr>
</tbody>
</table>

- $P^*$ is a bi-objectives VCSP composed of three variables (the constraint of $P_2$): $c_1, c_2$ and $c_3$. (See Figure 2).
- The domain $D_j$ of each variable $c_j$ is formed of four tuple values: $D_j = \{(1,1), (1,2), (2,1), (2,2)\}$.
- Binary constraints in $P_2$ become unary constraints in $P^*$:
  - for each variable $c_j$, the value $(1,1)$ has a unary cost $(\alpha_1, \alpha_2)$, the value $(1,2)$ has an unary cost $(\bot_1, \bot_2)$ and it will be filtered by applying an arc-consistency algorithm, and the value $(2,2)$ has a unary cost $(\beta_1, \beta_2)$.
- Binary constraints in $P^*$ should be added Prohibiting the choice of two values of the same variable in $P_2$. These binary constraints are crisp as they prohibit impossible solutions for the primal problem.

As can be seen in Table 4 and Figure 2, $P^*$ is a bi-objectives CSOP.

![Figure 2: $P^*$ the Dual of the bi-objectives VCSP.](image)

3.2 Directional Substitutability for a Binary MOCSOP

The Directional Substitutability (Naanaa, 2008) is a weak form of Neighbourhood Substitutability (Freuder, 1991). Initially, this concept has been defined for binary CSPs. In this paper, we generalize the concept of directional substitutability to reflect multi-objective unary cost functions involved in binary MOCSOP.

**Definition 6.** The multi objective inconsistency graph of a binary MOCSOP $\mathcal{P}$ is a simple graph $GI(\mathcal{P})$ in which the vertices correspond to the $k$ values of the variables and edges connecting pairs of vertices representing for each objective incompatible values.

Multi Objective Directional substitution for binary MOCSOP, in this paper, is defined using, as a reference, an orientation of the multi objective inconsistency graph of a binary MOCSOP.

**Definition 7.** An orientation $A_j$ of multi objective inconsistency graph of binary MOCSOPs is an assignment for each objective $j$ of a direction to each edge $(\phi_j(a), \phi_j(b))$ of graph resulting to the arc $(\phi_j(a), \phi_j(b))$ or arc $(\phi_j(b), \phi_j(a))$.

The concept of multi-objective directional substitutability is then a binary relation defined, in this paper, as follows:

**Definition 8.** Let $\mathcal{P} = (X, \mathcal{D}, C, S)$ be a binary MOCSOP, $x \in X$ and $a, b \in D_x$. A value $a$ is said Multi Objective Directionally Substitutable (MODS) to $b$ with reference to $A_j$ (notation $a \preceq_{A_j} b$) if for each objective $j = 1, \ldots, k$:

1. for all $(\langle x \rangle, \phi) \in C^1$, we have $\phi_j(a) \preceq_j \phi_j(b)$
2. for all $(\langle x, y \rangle, \phi) \in C^2$ and $a' \in D_y$, we have $\phi_j(a) \preceq_j \phi_j(a')$ and $\phi_j(b) \preceq_j \phi_j(a')$.

With reference to this definition we can define Multi-Objective Directional Substitutable Valued Constraint Satisfaction Problem with Crisp binary Constraints as follows:

**Definition 9.** Let $\mathcal{P}$ a binary MOVCSOP. $\mathcal{P}$ is a binary Multi-Objective Directionally Substitutable Valued Constraint Satisfaction Problem with only Crisp binary Constraints $\mathcal{P} \in L(MODS)$ if and only if for each variable $x$ of $\mathcal{P}$ and for each $a, b \in D_x$: $a$ is MODS to $b$ or $b$ is MODS to $a$ $\iff a \preceq_{A_j} b$ or $b \preceq_{A_j} a$.

Given an orientation $A_j$ for each objective $j$ of multi objective inconsistency graph, the relation $\preceq_{A_j}$ defines a total order on the domain of each variable.

**Lemma 4.** $\preceq_{A_j}$ is a total order on $D_x$.

**Proof Lemma:** $\preceq_{A_j}$ is trivially reflexive. We prove that $\preceq_{A_j}$ is also transitive. To this end, assume the opposite and proceed to obtain a contradiction. Let $u, v, w \in D_x$ such that for each objective $j u \preceq_{A_j} v \wedge v \preceq_{A_j} w$ This means that $u \preceq_{A_j} v \wedge v \preceq_{A_j} w$
but $\exists j$ such that $u \not\preceq_{A_j} w$. This means that, for all $u' \in D_x$, we have for each objective $j$

$$\Phi_j(u) \succeq_j \Phi_j(v) \land \left( \Phi_j(u), \Phi_j(u') \right) \preceq_{A_j} \Phi_j(v, u') \land \Phi_j(v) \succeq_j \Phi_j(u, u') \land \Phi_j(u) \preceq_j \Phi_j(v, u')$$

as $\exists j$ such that $u \preceq_{A_j} w$ and $\preceq$ is a total order, it must exists $u' \in D_x$ such that it must exists $j$ where:

$$\Phi_j(u) \succ_j \Phi_j(v) \lor \left( \Phi_j(u), \Phi_j(u') \right) \preceq_{A_j} \Phi_j(v, u') \lor \Phi_j(u) \preceq_j \Phi_j(v, u') \land \Phi_j(u) \preceq_j \Phi_j(v, u')$$

From (3) and (4) we get for each objective $j$

$$\Phi_j(u) \preceq_j \Phi_j(w) \land \left( \Phi_j(u), \Phi_j(u') \right) \not\in A_j \lor \left( \Phi_j(w), \Phi_j(u') \right) \not\in A_j \lor \Phi_j(u, u') \succ_j \Phi_j(u, u')$$

which contradicts (5).

Thus, $\preceq_{A_j}$ is transitive. \qed

The binary relation $\sim_{A_j}$ is defined on $D_x$ as follows: $u \sim_{A_j} v$ if and only if $u \preceq_{A_j} v$ and $v \preceq_{A_j} u$.

**Lemma 5.** $\sim_{A_j}$ is an equivalence relation on $D_x$.

Thus, each domain $D_x$ can be divided into subsets $D_{x,k} = D_x \cup \ldots \cup D_{x,s}$ such that the elements of each $D_{x,k}$, $k = 1, \ldots, s$ are all comparable, that is, to say, that for any $a, b \in D_{x,k}$, we have $a \preceq_{A_j} b$ or $b \preceq_{A_j} a$. Each $D_{x,k}$ is a chain of value totally ordered by $\preceq_{A_j}$.

In each chain $D_{x,k}$, we can distinguish the subset of directional dominant elements denoted by $D_{x,k}^+$:

$$D_{x,k}^+ = \{ a \in D_{x,k} \lor b \in D_{x,k}, a \preceq_{A_j} b \}$$

### 3.3 Tractability of the Directional Substitutable MOCSOP Class

In what follows, we study the tractability of binary MOCSOP.

The following approach identifies a tractable class of binary MOCSOP which is based on the Directional Substitutable functions.

We denote that if $\mathcal{P}$ is in $\mathcal{L}(\text{MODS})$ then it is with Directional Substitutable Valuation Functions only.

**Theorem 1.** The class of binary Multi-Objective Directional Substitutable Valued Constraint Satisfaction Problem with only crisp binary constraint ($\mathcal{L}(\text{MODS})$) is tractable.

**Proof Theorem 1.** First we will make $\mathcal{P}$ arc-consistent. Then, we show that by selecting a dominant element (see (6)) of each domain value, an optimal solution is obtained. This means that any $n$-tuple $t \in \Pi_{x \in X} D_{x,k}^+$ is an optimal solution. Referring to the Definition 8 and the fact that the function $\Phi$ is computable in polynomial time, this selection can be done in polynomial time.

Suppose that $t \in \Pi_{x \in X} D_{x,k}^+$ is not an optimal solution. This means that $t$ is inconsistent or that $\Phi(t)$ is not dominate.

Suppose that $t$ is inconsistent. Therefore $t$ must include, at least, a pair of incompatible values. Let $a \in D_{x,k}^+$ and $b \in D_{x,k}^+$ such a pair. Since $a$ and $b$ are inconsistent, for each objective $j$ such that $(\Phi_j(a), \Phi_j(b)) \in A_j$ or $(\Phi_j(b), \Phi_j(a)) \in A_j$.

Assume without loss of generality that $(\Phi_j(a), \Phi_j(b)) \in A_j$. (otherwise we can reason on $\Phi_j(b)$ rather than $\Phi_j(a)$ and obtain the same result).

It follows that $(\forall j) \Phi_j(a, b) = T_j$, and since $a \in D_{x,k}^+$ then for all $a' \in D_x$, we must have $(\forall j) \Phi_j(a', b) = T_j$. This means that $(\forall j) b$ has no support in $D_x$ and so that $\mathcal{P}$ is not arc-consistent, hence a contradiction.

**3.4 Usefulness of Directional Substitutable MOVCSP Class**

Given a MOVCSP $\mathcal{P}$ not in a Directional Substitutable MOVCSP class ($\mathcal{L}(\text{MODS})$), is it possible to use $\mathcal{L}(\text{MODS})$ class to solve $\mathcal{P}$? A problem decom-

**Function** ORDER$^+(\Phi, D_{x,k}^+, v, A_j) : \bar{v}$

```
D_{x,k}^+ \leftarrow D_{x,k}^+ \setminus v
\bar{v} \leftarrow \emptyset
while D_{x,k}^+ do
  u \leftarrow \text{MINCOST}(\Phi, D_{x,k}^+)
  D_{x,k}^+ \leftarrow D_{x,k}^+ \setminus u
  Order \leftarrow true
  for u' \in D do
    for v \in \bar{v} do
      for j \in k do
        if \Phi_j(v, u') \preceq_{A_j} \Phi_j(u, u') \land \Phi_j(v) \succ_{A_j} \Phi_j(u)
          Order \leftarrow false
          break
    if Order then \bar{v} \leftarrow \bar{v} \cup \{u\}
```

223
position scheme for MOVCSPs that takes advantage of Directional Substitutable Valuation Functions even when the studied problem is not limited to these Functions can be solved within a backtrack-based search.

The Algorithm DS–MOEDAC

1. computes $P'$ the dual of $P$ (line 1)
2. computes the subset of directional dominant elements denoted by $D'$ (line 2)
3. in order to update the Pareto set of non dominated solution $s^*$, according to Definition 3 and Lemma 1, if a solution $s$ dominates one solution $s_i$ from $s^*$, deletes $s_i$ from $s^*$ and adds $s$ to $s^*$ (lines 3, 4).

4. calls the Function ORDER$^+$ to identify in $O(ked^3)$ (where $k$ the number of objectives and $e$ is the number of constraints) a tractable sub-problem $P''$ of $P'$ such that $P''$ is in $\mathcal{L}(\text{MODS})$, (line 5) For a value $v$ there may be more than one $\bar{v}$ partitions. As a result, we can use any partition strategy. For example, the partition that promotes values which minimize cost.


This decomposition scheme can be distinguished by the possibility of instantiating variables by assigning to each one of them a subset of values in $\mathcal{L}(\text{MODS})$ instead of single values for the $P'$.

Example 4. Let $P_3$ a bi-Objectives binary VCSV composed of three variables $x_1$, $x_2$ and $x_3$. (see Figure 3). The domain $D_i$ of each variable $x_i$ is formed of two values $D_i = \{a,b\}$. This bi-Objectives VCSV has three constraints $c_1 = ((x_1,x_2), \Phi), c_2 = ((x_2,x_3), \Phi)$ and $c_3 = ((x_1,x_3), \Phi)$ whose valuations functions $\Phi$, are defined in Table 5. We suppose that for the both objectives $j = 1,2 \ \alpha_j \preceq \beta_j$ Referring to the Definition 9 $P_3$ is in a bi-Objectives $\mathcal{L}(\text{MODS})$, Referring to the orientation $A$ given in Figure 3 and referring to the proof of Theorem 1, if we affect

![Figure 3: $P_3$ in a bi-objectives $\mathcal{L}(\text{MODS})$ class.](image)

of an value $a$ to each variable $x_i$ we get the optimal solution $P_3$ since for each variable $x_i$: $a \preceq \alpha_i \preceq \beta_j \preceq b$.

Let $P$ a MOVCSLP. We deduce the tractability of MOVCSLPs through their duals.

**Corollary 1.** The MOVCSLP $P$ is tractable if

\[ \text{Dual}(\text{MOVCSLP}) = P' \in \mathcal{L} (\text{MODS}) \]

**Proof** Corollary 1. From Lemma 3 the Dual( MOVCSLP ) is MOCSLP. By Theorem 1 we have that $\mathcal{L}(\text{MODS})$ is a tractable class of MOCSLP. This means that the MOVCSLP $P$ such that Dual( $P$ ) = $P'$ $\in \mathcal{L} (\text{MODS})$ is tractable.

**4 CONCLUSION**

In this paper we have proposed a Soft Directional Substitutable based Decompositions for MOVCSLP. Despite the NP-hardness of MOVCSLP we have presented a tractable classes by forcing the allowable val-

---

0x0
ulation functions to have specific mathematical properties. This is the case of MOV CSP classes MOV CSPs with Directional Substitutable Valuation Functions only \((\text{MODS})\). As usefulness of \((\text{MODS})\) MOV CSP class even when the studied problem is not limited to these functions, we have proposed a Directional Substitutable decomposition algorithm. As a natural extension of this work, in order to validate the practical use of \((\text{MODS})\), we will compare a problem decomposition scheme for MOV CSPs that uses Pareto-based Soft Arc Consistency only with a decomposition scheme which, in addition, takes advantage of \((\text{MODS})\).

REFERENCES


