Pareto-based Soft Arc Consistency for Multi-objective Valued CSPs

Limeme Ben Ali[1], Maher Helaoui[2] and Wady Naanaa[3]
[1]*Faculty of Economics and Management of Sfax, University of Sfax, Tunisia*
[2]*Higher Institute of Business Administration, University of Gafsa, Tunisia*
[3]*National Engineering School of Tunis, University Tunis El Manar, Tunisia*

Keywords: Multi-objective Optimization, Multi-objective Valued Constraint Satisfaction Problems MO-VCSP, Soft Local Arc Consistency, Lower Bound Set, Pareto Dominance.

Abstract: A valued constraint satisfaction problem (VCSP) is a soft constraint framework that can formalize a wide range of applications related to Combinatorial Optimization and Artificial Intelligence. Most researchers have focused on the development of algorithms for solving mono-objective problems. However, many real-world satisfaction/optimization problems involve multiple objectives that should be considered separately and satisfied/optimized simultaneously. Solving a Multi-Objective Optimization Problem (MOP) consists of finding the set of all non-dominated solutions, known as the Pareto Front. In this paper, we introduce multi-objective valued constraint satisfaction problem (MO-VCSP), that is a VCSP involving multiple objectives, and we extend soft local arc consistency methods, which are widely used in solving Mono-Objective VCSP, in order to deal with the multi-objective case. Also, we present multi-objective enforcing algorithms of such soft local arc consistencies taking into account the Pareto principle. The new Pareto-based soft arc consistency (P-SAC) algorithms compute a Lower Bound Set of the efficient frontier. As a consequence, P-SAC can be integrated into a Multi-Objective Branch and Bound (MO-BnB) algorithm in order to ensure its pruning efficiency.

1 INTRODUCTION

Solving a Single-Objective Optimization Problem amounts to determining the best solutions that satisfy a set of constraints and optimize an objective function defined by the user. The best solution, also known as the optimal solution, is the solution with the highest assessment against the defined objective. Such a problem can be formulated in terms of a Valued Constraint Satisfaction Problem (VCSP). However, when dealing with real-world problems such as Supply Chain Problem, Production Management Problems, Communication Problems, Time-cost trade-off problem (Afruzi et al., 2013), Scheduling Problems (Hazır et al., 2010) and many others, a single objective function may be insufficient. In fact, most of the real-world applications require the integration of multiple simultaneous objective functions, often conflicting. When considering multiple objectives functions, the notion of optimal solution from single-objective optimization does not apply anymore, and instead one must rely on the notion of *Pareto Dominance*. A solution s is better, in the Pareto sense, than another solution s' if s is better than s' for at least one objective and not worse for any of the remaining ones. If none of the two solutions is better than the other, they represent two different trade-offs of the objectives function that, without knowledge of the decision maker's preferences, are considered to be equally valuable. A very important task of interest in a multi-objective optimization problem (MOP) is to compute its efficient frontier $\mathcal{E}$ (and, possibly, one or all efficient solutions for each of its elements). In order to present a more powerful modeling to these real problems, we propose a generalization of VCSPs to Multi-Objective Valued Constraint Satisfaction Problems (MOVCSPs).

The classical Constraint Satisfaction Problem (CSP) model considers only the feasibility of satisfying a collection of simultaneous requirements (van Beek and Manchak, 1996; Jeavons and Cooper, 1995). Various extensions have been proposed to this model, to allow it to deal with different kinds of optimization criteria, or preferences between different feasible solutions. Two very general extended frameworks that have been proposed are the semi-ring CSP (Bistarelli et al., 1999) and the valued CSP (VCSP) (Schiex et al., 1995). The semi-ring framework is slightly more general, but the VCSP framework is

Ben Ali, L., Helaoui, M. and Naanaa, W.
Pareto-based Soft Arc Consistency for Multi-objective Valued CSPs.
DOI: 10.5220/0007401802940305
In *Proceedings of the 11th International Conference on Agents and Artificial Intelligence (ICAART 2019)*, pages 294-305
ISBN: 978-989-758-350-6

simpler, and sufficiently powerful to describe many important classes of problems (Cohen et al., 2008). In this framework every constraint has an associated cost function which assigns a cost to every tuple of values for the variables in the scope of the constraint. In the literature, many local consistency algorithms have been proposed for soft CSPs. Soft Consistency algorithms work by making explicit the inconsistency level originally implicit in the problem. The general idea is to *safely move* costs (i.e., without changing the level of consistency of the solution) from high arity constraint to smaller arity ones.

In this paper we address combinatorial problems that can be expressed as MOVCSPs. We introduce a generic formalization of multi-objective problems in terms of Valued CSP framework. Several models using the soft CSP framework have been presented in the literature (Emma and Javier, 2006; Bistarelli et al., 2008; Bistarelli et al., 2012; Wilson et al., 2015). Our new MOVCSP model is important for two reasons; first we pick up an understanding of the nature of multi-objective optimization problems, and we accede to some theoretical results from the Valued CSP. Given a MO-VCSP, we define the *Lower Cost Vector* (LCV) operator on cost. Furthermore, we present its generalization to be applicable over sets of k-ary cost functions. Also, we show how to use the LCV value (*i.e.*, the value returned by LCV) inside the soft arc consistency techniques in order to deal with the multi-objective case. Thereafter, we introduce new definitions for the support notion based on LCV. As consequence, the LCV value corresponds to the transferred cost vector in the new Pareto-based soft arc consistency operations commonly known as Equivalence Preserving Transformation (EPT).

The rest of the paper is organized as follows. Section 2 summarizes the background notions about valued constraint satisfaction problems and multi-objective optimization. Section 3 shows how to extend the VCSP formalism to model multi-objective optimization problems. Section 3 also, presents basic operations over costs and their extension to costs sets. In Section 4, we introduce soft local consistencies based on the Pareto principle and the main differences while considering multiple objective functions. Furthermore in the same section, we describe the multi-objective extension of Soft Arc Consistencies maintaining algorithms (Maintain P-SAC). In Section 5, we give a description of the extension of depth-first branch-and-bound, to solve MOVCSP problems, that maintain Pareto soft local consistencies during search. Section 5 also presents the related work and a discussion of potential extensions. At last, Section 6 wraps up the paper, presenting our conclusions.

2 BACKGROUND

In this section, we point out the specific features of the multi-objective problems and recall the main definitions of the VCSP framework.

2.1 Multi-objective Optimization

Multi-objective Optimization Problems deal with multiple objectives, which should be simultaneously optimized (Deb and Kalyanmoy, 2001; Le Thi et al., 2008; Chiandussi et al., 2012).

Example 1. Π PROJECT MODELED AS A BI-OBJECTIVES DISCRETE TIME COST TRADE-OFF PROBLEM (VANHOUCKE AND DEBELS, 2007). *Let* Π *be a project defined as follows:*

Π *is a project comprised of* 6 *tasks: A, B, C, D, E and F. The predecessors of each task are defined in column* 2 *(Preds) of Table 1. The various options of the executions times and the relative costs of each tasks (option* 1*, option* 2*, option* 3*) are defined in columns* 3*,* 4 *and* 5*.*

Solving the problem is equivalent to finding one option for each task such that:

1. *The precedence constraints are satisfied.*
2. *Both global costs and global makespan are optimized.*

Table 1: Π project.

Tasks	Preds	option 1	option 2	option 3
A	–	(5,100)	(3,250)	(1,500)
B	A	(5,100)	(4,300)	(2,900)
C	A	(5,100)	(3,350)	(2,600)
D	A	(10,200)	(8,500)	(7,800)
E	B,C	(5,100)	(3,300)	(1,600)
F	D,E	(5,100)	(4,580)	(2,2500)

The project network $G = (V, A)$ *of the project* Π *is depicted in Fig.1 where* $V = \{A, B, C, D, E, F\}$ *is the set of Tasks and A is the set of arcs representing the precedence constraints. Each task can be executed in three options (modes). Each node* V_i *of G is labeled by a set of pairs (time,cost) representing time, cost values of each option of the task* V_i*.*

Optimizing simultaneously two functions can be contradictory, since reducing the cost, ϕ_c, often increases the project execution period, ϕ_t, and conversely.

The concept of looking for an optimal solution becomes more difficult to define. In this case, and in accordance with the Pareto optimal, the desired optimal solution is no longer a single point, but a set of non-dominated solutions. Otherwise, solving a problem with several objective functions, commonly referred

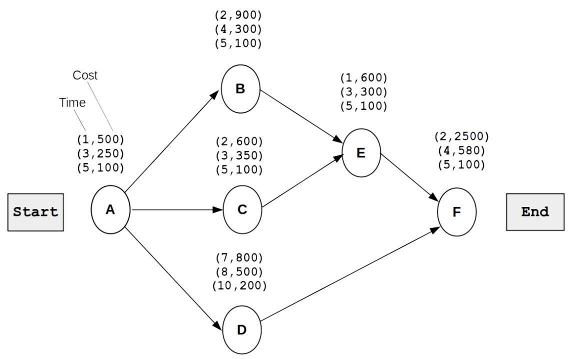

Figure 1: Project network of Π.

to a multi-objective problem, is to compute the best set of trade-off solutions called the *Pareto Front*.

The Pareto front $\mathcal{E}_\Pi$ of the Project Π is shown in Fig.2. The set of non-dominated solutions after removing redundant solutions is composed of 19 solutions marked with red points in Fig.2.

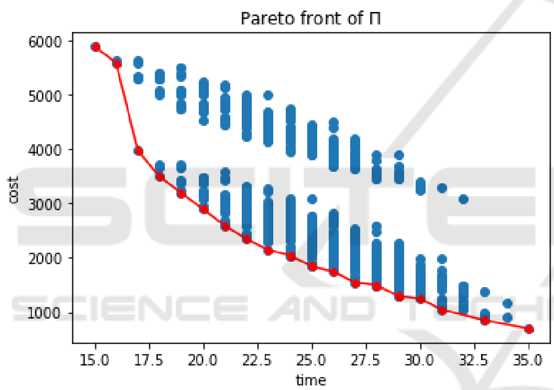

Figure 2: The Pareto front of Π.

A multi-objective problem can be defined as a problem where one seeks action that satisfies a constraint set and optimizes a set of objective functions. A very important task of interest in a multi-objective problem is to compute its efficient frontier $\mathcal{E}$.

2.2 Valued Constraint Satisfaction Problem and Soft Local Consistency

The Valued CSP (VCSP) framework is a generic optimization framework with a wide range of applications. Soft arc consistency operations transform a VCSP into an equivalent problem by shifting weights between cost functions. The principal aim is to produce a good lower bound on the cost of solutions, an essential ingredient of a branch-and-bound search.

Valued Constraint Satisfaction Problem. The Constraint Satisfaction Problem (CSP) consists of finding an assignment to n finite-domain variables such that a set of constraints are satisfied. Crisp constraints in the CSP are replaced by cost functions in the Valued Constraint Satisfaction Problem (VCSP) (Schiex et al., 1995). A cost function returns a valuation (a cost, a weight or penalty) for each combination of values for the variables in the scope of the function. Crisp constraints can still be expressed by, for example, assigning an infinite cost to inconsistent tuples. In the most general definition of a VCSP, costs lie in a valuation structure (a positive totally-ordered monoid) $(E, \oplus, \preccurlyeq)$, where E is the set of valuations which are totally ordered by $\preccurlyeq$ and combined using the aggregation operator $\oplus$ (Schiex et al., 1995) and $\top$ and $\perp$ denotes maximum and minimum elements of E given by $\preccurlyeq$. In this paper we only consider integer or rational costs.

A Valued Constraint Satisfaction Problem can be seen as a set of valued constraints, which are simply cost functions placed on particular variables. One can propose this definition.

Definition 1 ((Schiex et al., 1995)). *A Valued Constraint Satisfaction Problem (VCSP) is a tuple (X, D, C, S), where X is a set of n variables, each variable $x \in X$ has a domain of possible values $D_x \in D$, C is a set of cost functions and $S = (E, \oplus, \preccurlyeq)$ is a valuation structure. Each cost function $\langle \sigma, \phi_\sigma \rangle \in C$ is defined over a tuple of variables $\sigma \subseteq X$ (its scope) as a function ϕ_σ from the Cartesian product of the domains $\Pi_{x \in \sigma} D_x$ to E.*

As we search for assignments with minimal valuation computed by combining violated constraints by $\oplus$, one may see that the element $\top$ corresponds to unacceptable violation and is used to express hard constraints while $\perp$ element corresponds to complete satisfaction.

EPTs and Soft Arc Consistency. The soft local consistency, we study below, has an important role in the efficient resolution of VCSPs (Allouche et al., 2016; Larrosa and Schiex, 2004; Bistarelli et al., 2008). By definition, *local consistency* is a family of increasingly harder properties about a Soft Constraint Satisfaction Problem. The control parameter is the size of the sub-network (*i.e.*, the number of variable tuples) involved. The larger the tuples, the harder the property is. The simplest form of local consistency is node consistency, which only takes into account unary constraint. The next one is arc consistency, which takes into account binary constraint. In general, k-consistency takes into account constraints with k variables in their scope (Cooper, 2005).

Various consistency notions have been proposed for Valued CSP. Examples include NC* (Larrosa and

Schiex, 2004), AC* (Lee and Leung, 2012), FDAC* (Larrosa and Schiex, 2004; Lee and Leung, 2012), EDAC* (de Givry et al., 2005), VAC (Cooper et al., 2008) and OSAC (Cooper et al., 2010). Existential Directional Arc Consistency *EDAC** is the strongest known polynomial-time achievable form of soft arc consistency. Note, however, that *EDAC** has only been defined in the special case of binary (de Givry et al., 2005) and ternary (Lee and Leung, 2012) VCSPs.

Enforcing such local consistencies requires applying equivalence preserving transformations (EPTs) that shift costs between different scopes (Cooper and Schiex, 2004). EPT is based on three cost transfers operations which are also called *SAC* operations (de Givry et al., 2005). *Project* operation which projects costs from a cost function (on two or more variables) to a unary cost function. *Extend* performs the inverse operation, sending costs from a unary cost function to a higher-order cost function. Finally, *ProjectUnary* projects costs from a unary cost function to the nullary cost function $\phi_\varnothing$ which is a lower bound on the value of any solution.

Definition 2. *For a VCSP* (X,D,C,S)*, an* equivalence preserving transformation *(EPT) on* $F \subseteq C$ *is an operation which transforms the sub-problem* $VCSP(F)$ *into an equivalent* $VCSP$*. When* F *contains, at most, one non unary constraint, such an* equivalence-preserving transformation *is called a* Soft Arc Consistency (SAC) *operation.*

For simplicity, we restrict ourselves to binary VCSP. A binary VCSP is *AC** , *DAC**, $FDAC^*$, *EAC**, *EDAC** if it is *NC* and respectively *AC*, *DAC*, $FDAC$, *EAC*, $EDAC$ (Larrosa, 2002).

Local consistency properties are used to transform problems into equivalent simpler ones. From a practical point of view, the effect of applying local consistencies at each node of the search tree of a branch and bound algorithm is to prune values and to compute good lower bounds.

3 MULTI-OBJECTIVE VALUED CONSTRAINT SATISFACTION PROBLEM

Compared to Integer Linear Programming (ILP) (Le Thi et al., 2008; Teghem, 2009), the VCSP approach is an interesting alternative way to treat complex (multi-objective) optimization problems. VCSPs are a pragmatic extension of the CSP dedicated to optimization which authorizes an important efficiency gains with regard to the usual approach in constraint programming. The latter approach consists in encapsulating the objective function into a variable. In this section, we formalize a Multi-Objective Valued Constraint Satisfaction Problem (MO-VCSP). Furthermore, we introduce operations over costs and their extension to deal with MO-VCSP.

3.1 Model

In a *Multi-Objectives Valued Constraint Satisfaction Problem (MO-VCSP)*, as for a VCSP (Schiex et al., 1995), for each objective $j = 1,2,\ldots,k$, we assume that E^j, the set of possible valuations for objective j, is a totally ordered set with $\perp^j$ as minimal element and $\top^j$ as maximal element. We also need a monotone and binary operator $\oplus^j$ to aggregate valuations for objective j. These components can be gathered in k valuation structures each one can be specified as follows:

Definition 3 (Valuation Structure)**.** *Each valuation structure* S^j *of a MO-VCSP is the triple* $(E^j,\oplus^j,\preccurlyeq^j)$ *such as:*

- E^j *is a set of valuations for objective function* j*;*
- $\preccurlyeq^j$ *is a total order on* E^j*;*
- $\oplus^j$ *is commutative, associative and monotone.*

Proposition 1. *Let* $\{S^j = (E^j,\preccurlyeq^j,\oplus^j,\perp^j,\top^j)\}_{j=1}^k$ *be a family of valuations structures. Then, the structure* $\mathcal{S} = (E,\oplus,\preccurlyeq,\perp,\top)$*, where* $E = E^1 \times \ldots \times E^k, \oplus = \langle\oplus^1,\ldots,\oplus^k\rangle, \preccurlyeq = \langle\preccurlyeq^1,\ldots,\preccurlyeq^k\rangle, \perp = (\perp^1,\ldots,\perp^k)$ *and* $\top = (\top^1,\ldots,\top^k)$*, is a valuation structure. The relation between valuations vectors* $\preccurlyeq$ *is based on* Pareto dominance relation (Teghem, 2009)*. For distinction,* $\mathcal{S}$ *will be called a* multi-valuation structure.

Once the valuation structure $\mathcal{S}$ is specified, the multi-objective valued constraint satisfaction problem (MO-VCSP) can be defined as follows:

Definition 4 (MO-VCSP)**.** *A multi-objective valued constraint satisfaction problem (MO-VCSP) is defined by the tuple* $(X,D,C,\mathcal{S})$ *such as:*

- X *is a finite set of variables;*
- D *is a finite set of domains, such that* $D_x \in D$ *denotes the domain of* $x \in X$*.*
- $\mathcal{S}=(E^1 \times \ldots \times E^k,(\oplus^1,\ldots,\oplus^k),(\preccurlyeq^1,\ldots,\preccurlyeq^k))$ *is a multi-valuation structure.*
- C *is a set of multi-valued constraints. Each constraint is an ordered pair* (σ,Φ_σ) *where* $\sigma \subseteq X$ *is the scope of the constraint and* Φ_σ *is a function from* $\Pi_{x\in\sigma}D_x$ *to* $\Pi_{j=1}^k E^j$*, such that* $\Phi_\sigma(t) = (\phi_\sigma^1(t),\ldots,\phi_\sigma^k(t))$*.*

For a variable x, we can only assign a value of its domain. The valuation $\mathcal{V}$ of an assignment t to a subset of variables $V \subseteq X$ is obtained by

$$\mathcal{V}(t) \quad = \quad \bigoplus_{(\sigma,\Phi)\in C, \sigma\subseteq V} \Phi(t \downarrow \sigma)$$

which can be written as

$$\mathcal{V}(t) \;=\; \left(\bigoplus_{(\sigma,\Phi)\in C, \sigma\subseteq V} \phi^1(t \downarrow \sigma), \ldots, \bigoplus_{(\sigma,\Phi)\in C, \sigma\subseteq V} \phi^k(t \downarrow \sigma) \right)$$

where $\phi^j = \Phi[j]$ and $t \downarrow \sigma$ denotes the projection of t on variables σ.

The arity of a multi-valued constraint is the size of its scope. The arity of the problem is the maximum arity over all its constraints. In this work, we are concerned with binary MO-VCSPs. These are MO-VCSPs whose constraints are exclusively unary or binary. Moreover, we suppose that all constraints have distinct scopes. This allows us to identify every constraint (σ, Φ) of C with its scope-indexed vector Φ_σ. We write Φ_x as a shorthand for $\Phi_{\langle x \rangle}$ and Φ_{xy} as a shorthand for $\Phi_{\langle x,y \rangle}$. Without loss of generality, we assume that C contains a unary multi-valued constraint Φ_x for every variable $x \in X$ as well as a zero-arity multi-valued constraint $\Phi_\varnothing$.

Finding an assignment that optimizes all objectives simultaneously is not always possible. Indeed, in general, such an ideal assignment does not exist or cannot be reached, because of the trade-off between the objectives. Thus, the (optimal) solution of MO-VCSP can be characterized by using the concept of *Pareto Optimality*.

Example 2. *We return again to the DTCT project Π presented in Example 1. This project Π can be modeled as a bi-objectives VCSP P_1 defined as follows:*

1. *X_i is a finite set of variables such that $i = \{Tasks\}$;*
2. *$\mathcal{D}$ is a set of finite domains, where $D_i \in \mathcal{D}$ denotes the set of options that can be taken by task i; we therefore have $D_i = \{v_1, v_2, v_3\}$.*
3. *$S = (E^t \times E^c, \langle \oplus^t, \oplus^c \rangle, \langle \preceq^t, \preceq^c \rangle)$ is a multi-valuation structure, where $\oplus^t$ and $\oplus^c$ is the sum operator over time and cost values respectively.*
4. *C is a set of valued constraints. Each valued constraint C_i is an ordered pair (σ, Φ_σ) where $\sigma \subseteq X_i$ is the scope of C_i and $\Phi_\sigma(v) = (\phi_t(v), \phi_c(v))$.*

The predecessors of each task are defined in column 2 (Preds) of Table 2 which can be expressed by crisp constraint C_p taking bi-value $(\bot_t, \bot_c)$ if precedence is satisfied and $(\top_t, \top_c)$ otherwise.

We get

$$P_1 = (X, D, S, C \cup C_p)$$

Table 2: Bi-objectives VCSP $P_1 = \Pi$.

X_i	Preds	$\Phi_i(v_1)$	$\Phi_i(v_2)$	$\Phi_i(v_3)$
X_A	–	(5,100)	(3,250)	(1,500)
X_B	X_A	(5,100)	(4,300)	(2,900)
X_C	X_A	(5,100)	(3,350)	(2,600)
X_D	X_A	(10,200)	(8,500)	(7,800)
X_E	X_B,X_C	(5,100)	(3,300)	(1,600)
X_F	X_D,X_E	(5,100)	(4,580)	(2,2500)

The Dominance relation among valuation vectors is defined as follows:

Definition 5 (Dominance)**.** *Let $\mathcal{V}$ and $\mathcal{V}'$ be two k-sized valuation vectors, and let $\mathcal{V}[j]$ ($\mathcal{V}'[j]$) be the j^{th} component of $\mathcal{V}$ (resp. $\mathcal{V}'$). We say that $\mathcal{V}$ dominates $\mathcal{V}'$, denoted by $\mathcal{V} \prec_D \mathcal{V}'$, iff (i) $\mathcal{V}[j] \preccurlyeq^j \mathcal{V}'[j]$ holds true for all objective $j \in 1..k$. And (ii) there exist at least one objective $j \in 1..k$ such that $\mathcal{V}[j] \prec^j \mathcal{V}'[j]$.*

A solution t is a complete assignment. It is said to be *efficient* or *Pareto optimal* if it respects the definition 6 below.

Definition 6 (Pareto Optimal Solution)**.** *For a MO-VCSP and a complete assignment t, we say t is a* Pareto optimal solution *(resp. a* non-dominated solution*) iff there does not exist another assignment t', such that $\mathcal{V}(t') \prec_D \mathcal{V}(t)$.*

Solving MO-VCSP is to find all Pareto Optimal solutions representing the *Pareto front* corresponding to the set off all non-dominated solutions called *NDS*.

Definition 7 (Pareto Front)**.** *For a MO-VCSP, a set of cost vector obtained by* Pareto optimal solution *is called* Pareto Front*. Solving a MO-VCSP is to find the Pareto front.*

3.2 Operations over Costs and their Extensions

The following definitions require that assumed valuation structures for the MO-VCSP are *fair* (Cooper and Schiex, 2004). A valuation structure S^j is *fair* if for any valuation pair $\alpha, \beta \in E^j$, if $\alpha \preceq^j \beta$, there is a maximum difference between β and α. The only maximum difference between β and α is noted by $\beta \ominus^j \alpha$. Another requirement for the purpose of this paper, is that the valuation structures must be a lattice which mean that any pairs of valuation (costs) must have a lower bound, denoted LC. In order to generalize, we will stretch this notion of LC on costs sets.

In the multi-objective case, operations overs costs will be extended to costs vectors. Let us consider problems with k objectives. The only difference is that cost are now k-vectors and cost functions are now k-functions. $\top = (\top^1, \ldots, \top^k)$ is a k-vector, where each $\top^j \in E^j$ is the maximum acceptable cost for the objective j. $\bot = (\bot^1, \ldots, \bot^k)$ is a k-vector, where each $\bot^j \in E^j$ is the lowest acceptable cost for the objective j.

A k-vector $u = (u^1, \ldots, u^k)$ is a vector of k-components, where each $u^j \in E^j$ and $u^j \preccurlyeq^j \top^j$. Let u and v be two distinct k-vectors.

- The aggregation of two costs values for an objective j is defined as:

$$u^j \oplus^j v^j \stackrel{\text{def}}{=} \begin{cases} u^j \oplus^j v^j, & \text{if } u^j \oplus^j v^j \prec \top^j. \\ \top^j, & \text{otherwise.} \end{cases}$$

- The aggregation of two k-ary cost vectors is defined as:

$$u \oplus v \stackrel{\text{def}}{=} \begin{cases} \top, & \text{if } \exists j, u^j \oplus^j v^j = \top^j. \\ (u^1 \oplus^1 v^1, \ldots, u^k \oplus^k v^k), & \text{otherwise.} \end{cases}$$

- For two cost values $u^j, v^j \in E^j$, *such that* $u^j \succeq^j v^j$, the subtraction of v^j from u^j for an objective j is given by:

$$u^j \ominus^j v^j \stackrel{\text{def}}{=} \begin{cases} u^j \ominus^j v^j, & \text{if } u^j \prec^j \top^j. \\ \top^j, & \text{otherwise.} \end{cases}$$

- The subtraction of a cost vector v from a cost vector u, such that, $\forall j \in \{1, \ldots, k\}, u^j \in u$, *and* $v^j \in v$ we have $u^j \succeq^j v^j$ is defined as:

$$u \ominus v \stackrel{\text{def}}{=} \begin{cases} \top, & \text{if } \exists j, u^j \ominus^j v^j = \top^j. \\ (u^1 \ominus^1 v^1, \ldots, u^k \ominus^k v^k), & \text{otherwise.} \end{cases}$$

Let V be a set of k-ary cost vectors. We define its non-domination closure as

$$V^* = \{u \in V \mid \forall v \in V, v \not\prec_D u\}.$$

Let V_1 and V_2 be two sets closed under non-domination. We say that V_1 *dominate* V_2 (noted $V_1 \prec_D V_2$) if $\forall v \in V_2, \exists u \in V_1$ *s.t* $u \prec_D v$.

Theorem 1. *Let $\mathcal{S}$ a multi-valuation structure. If each S^j is a fair valuation structure, then so is $\mathcal{S}$.*

The result above ensures that the equivalence preserving transformation applied during soft arc consistency operations for mono-objective VCSP can still be applied for multi-objective VCSP.

Definition 8. *Lower Cost* ($\mathtt{LC^j}$) *Let F^j be a subset of E^j, an element $c \in F^j$ is minimal in F^j iff: $c \preceq^j x, \forall x \in F^j$. The set of all minimal elements of F^j will be denoted by $\mathtt{LC}^j(F^j)$.*

The ($\mathtt{LC}$) operator can be applied over cost vectors as follows:

Definition 9. *Lower Cost Vector* ($\mathtt{LCV}$) *Let $\mathcal{L} = \{L_1, \ldots, L_m\}$ be a set of k-cost vectors, where $L_i = (L_i^1, \ldots, L_i^k)$. The Lower Cost Vector of $\mathcal{L}$ denoted by $\mathtt{LCV}(\mathcal{L})$ is defined as follow:*

$$\mathtt{LCV}(\mathcal{L}) = \Big(\mathtt{LC}^1(L_1^1, \ldots, L_m^1), \ldots, \mathtt{LC}^k(L_1^k, \ldots, L_m^k)\Big)$$

Similarly, we define the Upper Cost Vector ($\mathtt{UCV}$) of set of m k-cost vector $\mathcal{U} = \{U_1, \ldots, U_m\}$ as being the k-vector $\mathtt{UCV}(\mathcal{U})$ corresponding to the upper value, for each objective $j \in 1..k$, of the j^{th} components of $U_1, \ldots, U_m$.

Note that, if $k = 1$, all previous definitions reduces to the classical ones.

4 PARETO-BASED SOFT LOCAL ARC CONSISTENCY (P-SAC)

The Pareto-based soft local arc consistency, presented below, has an important role in the efficient resolution of MO-VCSPs. We propose to extend and adapt soft arc consistency techniques for MO-VCSPs. P-SAC, computing a lower bound set of the cost of the Pareto optimal solutions set, avoids unnecessary explored branches and accelerates the convergence to the Pareto front (see Definition 7).

4.1 Pareto-based Equivalence Preserving Transformation (P-EPT)

Enforcing local arc consistencies requires applying equivalence preserving transformations (EPTs) that shift costs between different scopes. As for mono-objective case, *equivalence preserving transformation* in the multi-objective case is based on three basic operations (project, extend and project-unary). The main difference is that the transferred data between constraints is now a k-ary cost vector.

The main Pareto-based EPT (P-EPT) is defined below and described in Algorithm 1. This is an extension of the standard EPT version defined in (Cooper, 2005) for the multi-objective case.

Definition 10. *Two MO-VCSP $P = (X, D, C, \mathcal{S})$, $P' = (X', D', C', \mathcal{S}')$ are equivalent if for all complete assignment t, we have: $\mathcal{V}_P(t) = \mathcal{V}_{P'}(t)$.*

Definition 11. *The sub-problem of a MO-VCSP $(X, D, C, \mathcal{S})$ induced by $F \subseteq C$ is the MO-VCSP(F) = $(X_F, D_F, F, \mathcal{S})$, where $X_F = \bigcup_{\Phi_\sigma \in F} \sigma$ and $D_F = \{D_i \mid i \in X_F\}$.*

For a MO-VCSPs $(X, D, C, \mathcal{S})$, a Pareto-based equivalence preserving transformation (P-EPT) on

$F \subseteq C$ is an operation which transforms the multi-objective sub-problem MO-VCSP(F) into an equivalent MO-VCSP. When F contains at most one k-ary cost functions Φ_σ such that $|\sigma| \geq 2$, such a P-EPT is called a *Pareto Soft Arc Consistency (P-SAC)* operation.

The Pareto projection operation is defined as follows:

Definition 12. *Let* α *be the* Lower Cost Vector *of* $u \in D_x$ *with respect to* Φ_{xy}.

$$\alpha = \texttt{LCV}\{\Phi_{xy}(u,v)\}_{v \in D_y}$$

The Pareto Projection (P-Project) consists of adding α *to* $\Phi_x(u)$ *as follows:*

$$\Phi_x(u) \oplus \alpha, \quad \forall u \in D_x.$$

and subtracting α *from* $\Phi_{xy}(u,v)$ *as follows:*

$$\Phi_{xy}(u,v) \ominus \alpha, \quad \forall v \in D_y, \quad \forall u \in D_x.$$

the inverse operation is Pareto extend operation defined as follows:

Definition 13. *Let* β *be a cost k-vector such that* β *is the* Lower Cost Vector *of* $u \in D_x$ *with respect to* Φ_x.

$$\beta = \texttt{LCV}\{\Phi_x(u)\}_{u \in D_x}$$

The Pareto Extension (P-Extend) consists of adding β *to* $\Phi_{xy}(u,v)$*, as follows:*

$$\Phi_{xy}(u,v) \oplus \beta, \quad \forall v \in D_y.$$

and subtracting β *from* $\Phi_x(u)$*,*

$$\Phi_x(u) \ominus \beta, \quad \forall u \in D_x.$$

Theorem 2. *Given any fair binary MO-VCSP* $P = (X,D,C,S)$*, for any* $\Phi_\sigma \in C, x \in \sigma$ *and* $u \in D_x$*, the application of P-Project or P-Extend on P yields an equivalent MO-VCSP.*

Algorithm 1 gives three basic P-EPT which are also P-SAC operations (Cooper and Schiex, 2004). *P-Project* projects cost vectors from a set of cost functions (on two or more variables) on a set of unary cost functions. *P-Extend* performs the inverse operation, sending cost vectors from a set of unary costs functions to a set of higher arity cost functions. Each cost vector contains k-ary cost function for each objective $j \in \{1..k\}$. Finally *P-ProjectUnary* projects cost vectors from a set of unary cost functions to the nullary k-ary cost function $\Phi_\varnothing$. Observe that $\Phi_\varnothing$ is a k lower bound vector on the value of any solution. For each of the P-SAC operations given in Algorithm 1, a precondition is given which guarantees that cost values, for each objective, remain non-negative after the Pareto-EPT has been applied.

Algorithm 1: The basic equivalence-preserving transformations required to establish different forms of soft arc consistency.

Precondition: $(\alpha \prec \texttt{LCV}\{\Phi_{xy}(u,v)\}_{v \in D_y})$
procedure P-PROJECT(x,u,y,α)
  $\Phi_x(u) \leftarrow \Phi_x(u) \oplus \alpha$
  for each $v \in D_y$ **do**
    $\Phi_{xy}(u,v) \leftarrow \Phi_{xy}(u,v) \ominus \alpha$

Precondition: $(\alpha \prec \texttt{LCV}\{\Phi_x(u)\}_{u \in D_x})$
procedure P-EXTEND(x,u,y,α)
  for each $v \in D_y$ **do**
    $\Phi_{xy}(u,v) \leftarrow \Phi_{xy}(u,v) \oplus \alpha$
  $\Phi_x(u) \leftarrow \Phi_x(u) \ominus \alpha$

Precondition: $(\alpha \prec \texttt{LCV}\{\Phi_x(u)\}_{u \in D_x})$
procedure P-PROJECTUNARY(x,α)
  for each $(u \in D_x)$ **do**
    $\Phi_x(u) \leftarrow \Phi_x(u) \ominus \alpha$
  $\Phi_\varnothing \leftarrow \Phi_\varnothing \oplus \alpha$

Example 3 (MO-VCSP)**.** *Consider the problem depicted in Figure 3(a). It has two variables x, y with two values* a,b *in their domains. Unary multi-objective costs are depicted within small circles. Binary multi-objective costs are represented by edges connecting the corresponding values. The label of each edges, which is a pair of integers, represents the corresponding cost for each objective. If two values are not connected, the binary pair of cost between them is* $(0,0)$*. In this problem, there are two Pareto optimal solutions. The cost of solution* 1 *is the pair* $(1,300)$ *and it is attained by the assignment* (a,a)*. The cost of solution* 2 *is the pair* $(2,200)$ *and it is attained by the assignment* (a,b)*.*

4.2 P-SAC Techniques

In this section we extend previously-defined notions of soft arc consistency to deal with the multi-objective case. To describe our P-SAC, we need to introduce some new concepts related to the support notion. The main idea in our extension is to take advantage of the LCV operator in the definition of *Pareto support*.

Definition 14. *(Maximal Subset) Let* $(\langle x \rangle, \Phi_x)$ *be a unary multi-objective constraint and* $(\langle x,y \rangle, \Phi_{xy})$ *a binary multi-objective constraint. We define a maximal subset as follows:*

- $D_x^{'}$ *is a maximal subset of* D_x *if* $D_x^{'} \subseteq D_x$ *and* $\forall c' \in D_x^{'}$*,* $\exists j \in 1..k$*;* $\phi_x^j(c') = \perp^j$*.*
- $D_y^{'}$ *is a simple maximal subset of* D_y*, for a value* $u \in D_x$*, if* $D_y^{'} \subseteq D_y$ *and* $\forall\ c' \in D_y^{'}$*,* $\exists j \in 1..k$*;*

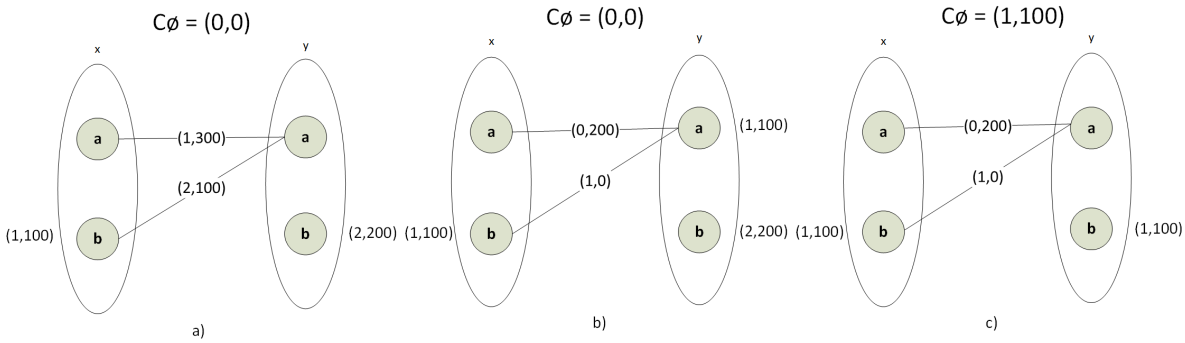

Figure 3: Three equivalent MO-VCSP instances ($\top = (4,400)$).

$\phi^j_{xy}(u,c') = \perp^j$.

- D'_y *is a full maximal subset of* D_y*, for a value* $u \in D_x$*, if* $D'_y \subseteq D_y$ *and* $\forall\ c' \in D'_y$*,* $\exists j \in 1..k$*;* $\phi^j_{xy}(u,c') \oplus \phi^j_y(c') = \perp^j$.

Definition 15. *Let* $(\langle x\rangle, \Phi_x)$ *be a unary multi-objective constraint. A subset of values* $D'_x \subseteq D_x$ *is a* Pareto Support *for* x *if* $D'_x \subseteq D_x$ *is maximal subset of* D_x *and* $\mathtt{LCV}(D'_x) = \mathtt{LCV}\{\Phi_x(u)\}_{u\in D'_x} = \perp$*. Equivalently,*

$$\mathcal{PS}(x) \overset{\text{def}}{=} \begin{cases} a \in \mathcal{PS}(x), & \text{if } \exists j, \phi^j_x(a) = \perp^j. \\ a \notin \mathcal{PS}(x), & \forall j, \phi^j_x(a) \succ \perp^j. \end{cases}$$

We will denote the Pareto Support Set *by* $\mathcal{PS}$.

The first level of Pareto local consistency is Pareto node consistency. It is defined as follows:

Definition 16 (Pareto Soft Node Consistency)**.** *A variable* x *is Pareto soft node consistent (P-NC*) if each value* $u \in D_x$ *satisfies* $\Phi_\varnothing \oplus \Phi_x(u) \prec \top$ *and* $\mathcal{PS}(x) \neq \varnothing$*. (i.e,.* $\Phi_\varnothing \oplus \mathtt{LCV}\{\Phi_x(u)\}_{u\in D_x} = \Phi_\varnothing$*). A MO-VCSP is P-NC* iff all variables are P-NC*.*

Pareto soft node consistency can be established by repeated calls to P-ProjectUnary until convergence.

Example 4. *The variable* x *presented in the Figure* ***4-a*** *is not Pareto soft node consistent.* $\mathtt{LCV}\{\Phi_x(a)\}_{a\in D_x} = (1,2) \neq \perp = (0,0)$*. After applying* $ProjectUnary$ *operation, variable* x *(see Figure* ***4-b****) become Pareto soft node consistent.* $\mathtt{LCV}\{\Phi_x(a)\}_{a\in D_x} = (0,0) = \perp$ *and* $\mathcal{PS}(x) = \{u,v\}$.

Following (Cooper and Schiex, 2004), Pareto soft arc consistency is based on the notion of Pareto Simple Support.

Definition 17. *Let* $(\langle x,y\rangle, \Phi_{xy})$ *a binary multi-objective constraint. A subset of values* $D'_y \subseteq D_y$ *is a* Pareto Simple Support *for a value* $u \in D_x$*, if* D'_y *is a simple maximal subset of* D_y *and* $\mathtt{LCV}(D'_y) = \mathtt{LCV}\{\Phi_{xy}(u,v)\}_{v\in D'_y} = \perp$*. We will denote the* Pareto Simple Support Set *by* $\mathcal{PSS}$.

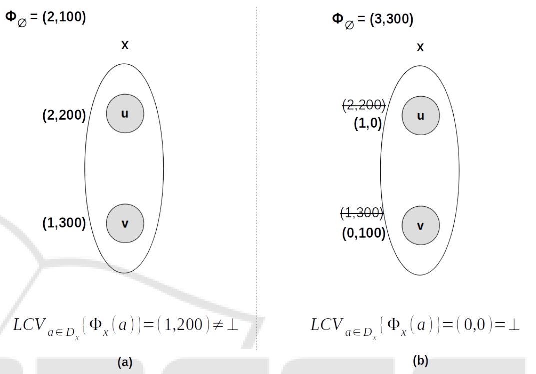

Figure 4: Example of enforcing Pareto Soft NC* property ($\perp = (0,0)$, $\top = (4,400)$).

A Pareto soft arc consistent problem is defined as follows:

Definition 18 (Pareto Soft Arc Consistency)**.** *A variable* x *is Pareto soft arc consistent if for every* $u \in D_x$ *has a non empty Pareto Simple Support set (*$\mathcal{PSS}(x(u)) \neq \emptyset$*), equivalently,* $\Phi_x(u) \oplus \mathtt{LCV}\{\Phi_{xy}(u,v)\}_{v\in D_y} = \Phi_x(u)$.

A MO-VCSP is Pareto Soft arc-consistency (P-AC) if all variables are Pareto soft node consistent and Pareto soft arc-consistent.*

Stronger Pareto local arc consistency levels rely on the notion of Pareto Full Support.

Definition 19. *Let* $(\langle x,y\rangle, \Phi_{xy})$ *be a binary multi-objective constraint. A set of values* $D'_y \subseteq D_y$ *is a* Pareto Full Support *for a value* $u \in D_x$ *if* D'_y *is full maximal subset of* D_y *and* $\mathtt{LCV}(D'_y) = \mathtt{LCV}\{\Phi_{xy}(u,v) \bigoplus \Phi_y(v)\}_{v\in D'_y} = \perp$*. We will denote the* Pareto Full Support set *by* $\mathcal{PFS}$

Pareto Directional Arc Consistency consists in combining the binary costs and unary cost in the calculation of the minimum valuation to be projected.

This consistency level requires a total order on the variables.

Definition 20 (Pareto Directional Arc Consistency)**.** *A variable x is* Pareto directional arc consistent (P-DAC*) *if* $\forall u \in D_x, \quad \forall y,\ y > x,\ \mathcal{PFS}(x(u)) \neq \emptyset$. *Equivalently,*

$$\Phi_x(u) \oplus \mathtt{LCV}\{\Phi_{xy}(u,v) \oplus \Phi_y(b)\}_{v \in D_y} = \Phi_x(u)$$

A MO-VCSP is Pareto Soft directional arc-consistency (P-DAC*) if all variables are P-DAC* and P-NC*.

Inspired by the work of (de Givry et al., 2005), P-FDAC* is an improvement of P-AC* and P-DAC*.

Definition 21 (Pareto Full Directional Arc-consistency)**.** *A MO-VCSP is Pareto FDAC (P-FDAC*) with respect to an order $<$ on the variables if it is P-AC$*$ and P-DAC$*$ with respect to $<$.*

Pareto full supports can be established in two directions if this can produce an increase in the lower bound set. This is a local natural consistency property, called Pareto soft existential arc-consistency (P-EAC* inspired by the work of (de Givry et al., 2005)). Pareto Existential arc consistency (P-EAC) is independent of a variable order. For each variable x in turn, P-EAC shifts costs to Φ_x if this can lead to an immediate increase in $\Phi_\varnothing$ via P-ProjectUnary.

Definition 22 (Pareto Soft Existential Arc-consistency)**.** *A variable x is Pareto soft existential arc-consistent (P-EAC*) if $\mathcal{PS}(x) \neq \varnothing$, and $\forall \Phi_{xy} \in C, \exists a \in D_x, \mathcal{PFS}((x,a)) \neq \varnothing$.*

A MO-VCSP is Pareto soft existential arc-consistency (P-EAC*) if all variables are Pareto soft node consistent and Pareto soft existential arc-consistent. A MO-VCSP is P-EDAC* if it is P-FDAC* and P-EAC*.

4.3 Enforcing Pareto Soft Arc Consistencies

Enforcement of such a Pareto local consistency property previously defined requires applying P-EPT. Any Multi-objective Valued CSP can be transformed into an equivalent instance having the P-NC* property by projecting any unary multi-valued constraint towards the zero-arity multi-objective constraint $\Phi_\varnothing$ and subsequently pruning every unfeasible value.

Enforcing P-NC* is described in Algorithm 2.

Procedure *Enforce P-NC** (see Algorithm 2) enforce Pareto NC*, where *ProjectUnary()* applies EPTs that move unary costs towards $\Phi_\varnothing$ while keeping the solution unchanged, and *PruneVar()* remove unfeasible values.

Algorithm 2: Enforce P-NC*.

procedure ENFORCE P-*NC**($\mathcal{X}$)
  for each $x \in \mathcal{X}$ **do**
    $ProjectUnary(x)$
  for each $x \in \mathcal{X}$ **do**
    $PruneVar(x)$
procedure PROJECTUNARY((x))
  $\alpha \leftarrow \mathtt{LCV}_{u \in D_x}(\Phi_x(u))$
  for each $u \in D_x$ **do**
    $\Phi_x(u) \leftarrow \Phi_x(u) \ominus \alpha$
  $\Phi_\varnothing \leftarrow \Phi_\varnothing \oplus \alpha$
procedure PRUNEVAR(x)
  for each $u \in D_x,\ s.t., \Phi_x(u) \oplus \Phi_\varnothing \prec_D \{s \in NDS\}$ **do**
    $D_x \leftarrow D_x \setminus \{u\}$

Theorem 3. *Given a set of non-dominated solutions NDS found during the exploration of the search space: The value $a \in D_x$ deleted by the function* PRUNEVAR *may only participate in solutions dominated by some solutions in NDS.*

Likely, P-AC* can be enforced by projecting binary multi-valued constraints towards unary multi-valued constraints and thereafter enforcing P-NC*. Since enforcing P-NC* may prune some domain values, some variables may have become Pareto soft arc inconsistent. Therefore, the entire process is repeated until no changes are performed. Algorithm 3 allows enforcing various levels of previously defined Pareto-based soft local arc consistencies.

Property 1. *The complexity of P-EDAC* = $|NDS| * O(ed^2 max\{nd, max\{|E_j|\}_{j=1}^k\})$, where n, e, k, E_j and d are the number of variables, the number of constraints, the number of objectives, the set of possible valuations for the objective j and larger domain size (de Givry et al., 2005).*

Property 2. *On a problem with a single objective function (i.e., k=1), the enforcement algorithms of P-AC*, P-DAC*, P-FDAC* and P-EAC*, are equivalent to classical soft arc consistency algorithms AC*, DAC*, FDAC* and EAC*.*

5 DISCUSSION AND FUTURE WORKS

Multi-Objective Branch-and-Bound (MO-BB) is a general search scheme for multi-objective constraint optimization problems. The search space is represented as a tree. The algorithm searches in a depth-first manner the tree defined by the problem. Its outputs are the set of Non Dominated Solutions (NDS).

Algorithm 3: Enforcement algorithms of P-AC*, P-DAC*, P-FDAC*, P-EAC*.

procedure ENFORCE P-$AC^*(x,y)$
  for each $u \in D_x$ **do**
    $\alpha \leftarrow \texttt{LCV}\{\Phi_{xy}(u,v)\}_{v\in D_y}$
    if $\Phi_x(u) \oplus \alpha \neq \Phi_x(u)$ **then**
      $Project(x,u,y,\alpha)$
  $ProjectUnary(x)$
  $PruneVar(x)$
procedure ENFORCE P-$DAC^*(x,y)$
  for each $u \in D_x$ **do**
    $P[u] \leftarrow \texttt{LCV}\{\Phi_{xy}(u,v) \oplus \Phi_y(v)\}_{v\in D_y}$
    for each $v \in D_y$ **do**
      $Extend(y,u,x,P[u] \vec{\ominus} \Phi_{xy}(u,v))$
    $Project(x,u,y,P[u])$
  $ProjectUnary(x)$
  $PruneVar(x)$
procedure ENFORCE P-$FDAC^*(x,y)$
  if $x < y$ **then**
    Enforce P-$DAC(x,y)$
    Enforce P-$AC(y,x)$
procedure ENFORCE P-$EAC^*(x)$

$$\alpha \leftarrow \texttt{LCV}\{\Phi_x(u) \bigoplus_{\Phi_{xy}\in C \ s.t. \ y<x} \texttt{LCV}\{\Phi_{xy}(u,v) \oplus \Phi_y(v)\}_{v\in D_y}\}_{u\in D_x}$$

  if $\Phi_\varnothing \oplus \alpha \neq \Phi_\varnothing$ **then**
    Enforce P-$FDAC(x,y)$
procedure ENFORCE P-$EDAC^*(x,y)$
  if $x < y$ **then**
    Enforce P-$EAC(x,y)$
    Enforce P-$FDAC(x,y)$

The efficiency of the algorithm greatly depends on its pruning ability which, in turns, depends on the computation of a good lower bound set at each visited node.

Algorithms that compute lower bound such as mini-bucket elimination MBE (Emma and Javier, 2006; Larrosa and Schiex, 2004) or Existential Directional Arc consistency EDAC* (de Givry et al., 2005) are a fundamental component of mono-objective Branch and Bound because they can be executed at every search node in order to detect infeasible nodes (Larrosa and Schiex, 2003; de Givry et al., 2005). The elementary operations made during applying P-SAC on MO-VCSPs are deleting values, the projection and extension of costs vector. All these operations cannot add to the problem of binary constraints on which the filtering technique P-SAC* is applied. So the problem remains binary. In addition, generally the filtering algorithms are incremental. This means, if a consistency is established in the search tree node then determining the local consistency in a son node can be done by considering only the changes between it and the parent node. This property is very useful since the filtering is performed at each node of the search tree. Another key property is that the filtering technique P-SAC* computes a lower bound set for the cost of the optimal solution. For the resolution of MO-VCSP, Pareto SAC* can be used to obtain good quality lower bounds set (LB) or it can be integrated into multi-objective branch and bound in order to increase its pruning efficiency to generate the set of all non-dominated solutions.

In practice, this set of solutions can be important (risk of memory explosion) and it is going to slow down Pareto-NC*. In order to reduce this set we can calculate a lower bound and/or an upper bound for every objective on the set of non-dominated solutions then, we compare to an under/over-approximation of several solutions at once.

Alternatively, we can proceed for a decomposition scheme of an initial problem in order to solve small instances of MO-VCSP. As a first step, we want to identify some new tractable classes of MO-VCSP. Where, we can solve instances of MO-VCSP in polynomial time with MO-BnB+P-SAC by restrictions of objectives functions to be in a specific class $\mathcal{C}$ (such as; modular objective functions (Helaoui and Naanaa, 2013), sub-modular objective functions (Helaoui and Naanaa, 2012) or even Directional Substitutable Valuation Functions (Naanaa, 2008)). As a natural extension of this work, we will propose a problem decomposition scheme for MO-VCSPs that takes advantage of restricted objective Functions even when the studied problem is not limited to these Functions. This decomposition scheme can work within a backtrack-based search and consists in decomposing the original problem into a set of $\bigcup_{i=1}^{|NDS|} \mathcal{P}_i \in \mathcal{C}$, and then tractable sub-problems. This decomposition scheme can be distinguished by the possibility of instantiating variables by assigning to them subsets of values instead of single values for the $\bigcup_{i=1}^{|NDS|} \mathcal{P}_i$, where each one is in $\mathcal{C}$. On a more practical side, we plan to implement our algorithms and to integrate them into an existing VCSP solver.

6 CONCLUSION

The Valued Constraint Satisfaction Problem (VCSP) is a generic optimization problem consisting in a network of local cost functions defined over discrete variables. It has applications in Artificial Intelligence, Operations Research, Bio-informatics and has been used to tackle optimization problems in other graphical models (including discrete Markov Random Fields and Bayesian Networks).

In this paper, we introduce a Multi-Objective VCSP (MO-VCSP), that is a VCSP involving multiple objectives. We propose a new extension of local arc consistency to the MO-VCSP. The incremental lower bounds set produced by Pareto-based soft local arc consistency can be used for pruning inside Branch and Bound search. The latter algorithm enables the calculation of the set of all Pareto Optimal (PO) solutions, an algorithm that enforces a Pareto soft local arc consistency property takes into account the Pareto principle by updating the set of Non-Dominated Solutions during a Branch and Bound search.

REFERENCES

Afruzi, E. N., Roghanian, E., Najafi, A., and Mazinani, M. (2013). A multi-mode resource-constrained discrete time–cost tradeoff problem solving using an adjusted fuzzy dominance genetic algorithm. *Scientia Iranica*, 20(3):931 – 944.

Allouche, D., Bessiere, C., Boizumault, P., de Givry, S., Gutierrez, P., Lee, J. H., Leung, K. L., Loudni, S., Métivier, J.-P., Schiex, T., and Wu, Y. (2016). Tractability-preserving transformations of global cost functions. *Artif. Intell.*, 238(C):166–189.

Bistarelli, S., Gadducci, F., Larrosa, J., and Rollon, E. (2008). A soft approach to multi-objective optimization. In *Proceedings of the 24th International Conference on Logic Programming*, ICLP '08, pages 764–768, Berlin, Heidelberg. Springer-Verlag.

Bistarelli, S., Gadducci, F., Larrosa, J., Rollon, E., and Santini, F. (2012). Local arc consistency for non-invertible semirings, with an application to multi-objective optimization. *Expert Syst. Appl.*, 39(2):1708–1717.

Bistarelli, S., Montanari, U., Rossi, F., Schiex, T., Verfaillie, G., and Fargier, H. (1999). Semiring-based csps and valued csps: Frameworks, properties, and comparison. *Constraints*, 4(3):199–240.

Chiandussi, G., Codegone, M., Ferrero, S., and Varesio, F. (2012). Comparison of multi-objective optimization methodologies for engineering applications. *Computers and Mathematics with Applications*, 63(5):912 – 942.

Cohen, D., Jeavons, P., and Zivny, S. (2008). The expressive power of valued constraints: Hierarchies and collapses. *Theoretical Computer Science TCS*, 409:137–153.

Cooper, M., De Givry, S., Sanchez, M., Schiex, T., and Zytnicki, M. (2008). Virtual arc consistency for weighted csp. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 1*, AAAI'08, pages 253–258. AAAI Press.

Cooper, M., de Givry, S., Sanchez, M., Schiex, T., Zytnicki, M., and Werner, T. (2010). Soft arc consistency revisited. *Artificial Intelligence*, 174(7):449 – 478.

Cooper, M. and Schiex, T. (2004). Arc consistency for soft constraints. *Artif. Intell.*, 154(1-2):199–227.

Cooper, M. C. (2005). High-order consistency in valued constraint satisfaction. *Constraints*, 10(3):283–305.

de Givry, S., Zytnicki, M., Heras, F., and Larrosa, J. (2005). Existential arc consistency: Getting closer to full arc consistency in weighted csps. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, pages 84–89, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Deb, K. and Kalyanmoy, D. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York, NY, USA.

Emma, R. and Javier, L. (2006). Bucket elimination for multiobjective optimization problems. *J. Heuristics*.

Hazır, O., Haouari, M., and Erel, E. (2010). Robust scheduling and robustness measures for the discrete time/cost trade-off problem. *European Journal of Operational Research*, 207(2):633 – 643.

Helaoui, M. and Naanaa, W. (2012). A submodular-based decomposition strategy for valued csps. *The Sixth "Starting Artificial Intelligence Research" Symposium. Montpellier, France*.

Helaoui, M. and Naanaa, W. (2013). Modularity-based decompositions for valued csp. *Annals of Mathematics and Artificial Intelligence*.

Jeavons, P. G. and Cooper, M. C. (1995). Tractable constraints on ordered domains. *Artificial Intelligence*, 79(2):327 – 339.

Larrosa, J. (2002). Node and arc consistency in weighted csp. In *Eighteenth National Conference on Artificial Intelligence*, pages 48–53, Menlo Park, CA, USA. American Association for Artificial Intelligence.

Larrosa, J. and Schiex, T. (2003). In the quest of the best form of local consistency for weighted CSP. In *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*, pages 239–244.

Larrosa, J. and Schiex, T. (2004). Solving weighted CSP by maintaining arc consistency. *Artif. Intell.*, 159(1-2):1–26.

Le Thi, H. A., Bouvry, P., and Pham Dinh, T., editors (2008). *Solving the Multiple Objective Integer Linear Programming Problem*, Berlin, Heidelberg. Springer Berlin Heidelberg.

Lee, J. and Leung, K. (2012). Consistency techniques for flow-based projection-safe global cost functions in weighted constraint satisfaction. *J. Artif. Intell. Res. (JAIR)*, 43:257–292.

Naanaa, W. (2008). Substitutability based domain decomposition for constraint satisfaction. *CP*.

Schiex, T., Fargier, H., and Verfaillie, G. (1995). Valued constraint satisfaction problems: Hard and easy problems. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 631–637, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Teghem, J. (2009). *Multi-objective integer linear programming*. Springer US, Boston, MA.

van Beek, P. and Manchak, D. W. (1996). The design and experimental analysis of algorithms for temporal reasoning. *J. Artif. Intell. Res.*, 4:1–18.

Vanhoucke, M. and Debels, D. (2007). The discrete time/cost trade-off problem: extensions and heuristic procedures. *Journal of Scheduling*, 10(4):311–326.

Wilson, N., Razak, A., and Marinescu, R. (2015). Computing possibly optimal solutions for multi-objective constraint optimisation with tradeoffs. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 815–821. AAAI Press.

APPENDIX

Proof of Theorem 1

Proof. We have for all $j \in 1..k$, S^j is fair then for any valuation pair $\alpha, \beta \in E^j$, if $\alpha \preceq^j \beta$, there is a maximum difference between β and α, denoted $\beta \ominus^j \alpha$. For each pair of multi-valuation $u, v \in \Pi_{j=1}^{k} E^j$, if $\forall j,\ u^j \preceq^j v^j$ which implies by definition that $u \prec_D v$ (u *dominates* v), there is a maximum difference between v and u which is equal to $(u^1 \ominus^1 v^1, \ldots, u^k \ominus^k v^k)$. Where k denotes the number of objectives. □

Proof of Theorem 2

Proof. Following (Cooper and Schiex, 2004), and under assumption that every valuation structure S^j is *fair*, to demonstrate equivalence, it is sufficient to prove that the cost vector $\Phi_{xy}(u,v) \oplus \Phi_x(u)$ is an invariant of P-Project(x,u,y,α) and P-Extend(y,u,x,α). For any $v \in D_y$, let γ be the initial k-cost value of $\Phi_{xy}(u,v)$ and δ the initial k-cost value of $\Phi_x(u)$. After the execution of P-Project , we have $\Phi_{xy}(a,b) \oplus \Phi_x(a) = (\gamma \ominus \alpha) \oplus (\delta \oplus \alpha) = \gamma \oplus \delta$. After the execution of P-Extend, we have $\Phi_{xy}(a,b) \oplus \Phi_x(a) = (\gamma \oplus \alpha) \oplus (\delta \ominus \alpha) = \gamma \oplus \delta$. This proves the invariance. □

Proof of Theorem 3

Proof. Given a set of non-dominated solutions NDS found during the exploration of the search space:

Denote by $SP_{u \in D_x}$ the set of solutions of a MO-VCSP to which a value $u \in D_x$ may participate.

The function PRUNEVAR* deletes a value $u \in D_x$ if and only if for each objective i it $\exists$ a non dominated solution $s \in NDS$ such that $\Phi_x(u) \oplus \Phi_\varnothing = \top^s$

Since $s \in NDS$ then $\top^s$ is an element of the solution $s \in NDS$, so if $\Phi_x(u) \oplus \Phi_\varnothing = \top^s$ (element of the solution $s \in NDS$) then unary cost of the value $u \in D_x$ denoted by $\Phi_x(u)$ combined with the total of the minimum unary costs of other variables denoted by $\Phi_\varnothing$ is dominated by at least one solution $s \in NDS$:
for each objective $j\ \exists\ s \in NDS$ such that

$$\bigoplus_{v \in (y \in s)} \phi_y^j(v) \preceq \phi_x^j(u) \oplus_i \phi_\varnothing^j$$

$$\Rightarrow \exists\, s \in NDS \prec_D \Phi_x(u) \oplus \Phi_\varnothing$$

Secondly, any solution where $u \in D_x$ denoted by $s_u \in SP_{u \in D_x}$ is dominated by unary cost of value $u \in D_x$: $\Phi_x(u)$ combined with the total of the minimum unary costs of other variables denoted by $\Phi_\varnothing$.

$$\Phi_x(u) \oplus \Phi_\varnothing \prec_D \forall\, s_u \in SP_{u \in D_x}$$

And since the unary cost combined with the unary total minimum cost of the other variables is dominated by a solution $s \in NDS$ we have:

$$\exists\, s \in NDS \prec_D \Phi_x(u) \oplus \Phi_\varnothing \wedge$$
$$\Phi_x(u) \oplus \Phi_\varnothing \prec_D \forall\, s_u \in SP_{u \in D_x}$$

We can therefore conclude, as there is a partial order between dominated solution.

$$\exists\, s \in NDS \prec_D \forall\, s_u \in SP_{u \in D_x}$$

And thereafter the value $u \in D_x$ may only participate in solutions dominated by at least one of the solutions $s \in NDS$.

□