Pareto based Soft Arc Consistency for Multi-Objective Valued CSPs

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) is to find 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.

Keywords: Multi-Objective Optimization · Multi-Objective Valued Constraint Satisfaction Problems MO-VCSP · Soft local arc consistency · Lower Bound Set · Pareto Dominance.

1 Introduction

Solving a Single-Objective Optimization Problem amounts to determining the best solutions that satisfies 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[11], Scheduling Problems[22], ..., a single objective function may be insufficient. In fact, most of the real-world applications requires 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_1 if s is better than s_1 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 makers 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 [23, 15]. 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 and the valued CSP (VCSP) [25]. The semi-ring framework is slightly more general, but the VCSP framework is simpler, and sufficiently powerful to describe many important classes of problems [30]. 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 artificial intelligence and combinatorial problems that can be expressed as MO-VCSPs. We introduce a generic formalization of multi-objective problems in terms of Valued CSP framework. Several models that uses the soft CSP framework, have been presented in the literature[12, 2, 3, 28]. Our new model of multi-objective valued constraint satisfaction problem is important for two reason; 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 multi-objective VCSP, we characterize the **LowerCostVector** (`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 correspond to the transfered 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 who to formalize a VCSP in the multi-objective case. Section 3 also, presents basic operations over costs and their extension to costs sets. Section 4 we introduce soft local consistencies based on the Pareto principle and the mains differences while considering multiple objective functions. Furthermore in the same section, we describes the multi-objective extension of Soft Arc Consistencies maintaining algorithms (Maintain P-SAC) and show how Maintain P-SAC can be integrated into a BnB search algorithm. Section 5 we give a description of the extension of depth-first branch-and-bound, to solve MO-VCSP problems, that maintain Pareto soft local consistencies during search. Section 6 presents the related work and, at last,we wraps up the paper, presenting our conclusions.

2 Background

2.1 Multi-objective Optimization

Multi-objective optimization problems deal with multiple objectives, which should be simultaneously optimized [10, 13, 4]. Consider a Time-cost trade off problem [21] involves the following two objective functions

- ϕ_c to optimize the cost.
- ϕ_d to optimize the time.

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

The concept of looking for an optimal solution becomes more difficult to define. In this case, 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 multi-objective functions, commonly referred to a multi-objective problem, is to compute the best set of compromise solutions called *Pareto Front*. A multi-objective problem can be defined as a problem which one seeks action that satisfies a constraint set and optimizes a set of objective functions.

A very important task of interest in a MOP is to compute its efficient frontier $\mathcal{E}$.

2.2 Valued constraint satisfaction problem & 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 in 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) [25]. 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, cost lie in a valuation structure (a positive totally-ordered monoid) $\langle \mathtt{E}, \oplus, \preccurlyeq \rangle$ where $\mathtt{E}$ is the set of valuations totally ordered by $\preccurlyeq$ and combined using the aggregation operator $\oplus$ [25]. 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. Formally,

Definition 1 (See [25]). *A* Valued Constraint Satisfaction Problem (VCSP) *is a tuple $\langle X, D, C, S \rangle$ where X is a set of n variables $X = \{1, \dots, n\}$, each variable $x \in X$ has a domain of possible values $D_x \in D$, C is a set of cost functions and $S = \langle \mathtt{E}, \oplus, \preccurlyeq \rangle$ 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 ϕ_σ from the Cartesian product of the domains $D_x (x \in \sigma)$ to $\mathtt{E}$.*

Example 1 (VCSP). Consider the problem depicted in figure 1(a). It has two variables x, y with two values (a, b) in their domains. Unary costs are depicted within small circles. Binary costs are represented by edges connecting the corresponding values. The label of each edges is an integers, represents the corresponding cost. If two values are not connected, the binary cost between them is (0). In this problem the optimal cost (1) and it is attained with the assignment (a, a).

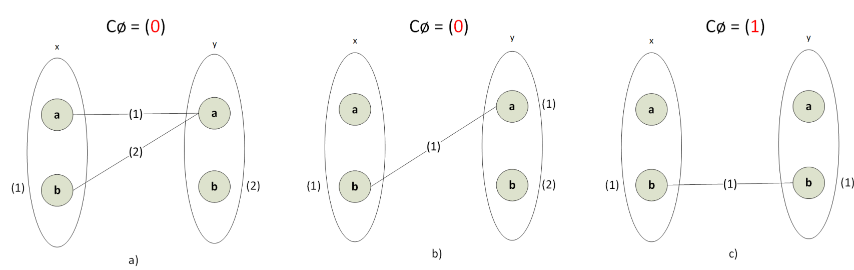

Fig. 1. Three equivalent VCSP instances

EPTs and Soft arc consistency The soft local consistency, we study below, has an important role in the efficient resolution of VCSPs [1, 18, 17, 2]. 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 constraint with $k+1$ variables in its scope [7].

Various consistency notions have been proposed for Valued CSP. Examples include NC* [19], AC* [6, 20], FDAC* [19, 17, 20], EDAC* [14, 20], VAC [5] and OSAC [9]. Existential Directional Arc Consistency *EDAC** is the strongest known polynomial-time achievable form of soft arc consistency. Note that *EDAC** has only been defined in the special case of binary [14] and ternary [20] VCSPs.

Enforcing such local consistencies requires applying equivalence preserving transformations (EPTs) that shift costs between different scopes [6]. EPT is based on three cost transfers operations which are also called SAC operations [14]. *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, *UnaryProject* 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 VCSPs* $\langle X, D, C, S \rangle$*, 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 cost functions* ϕ_σ *such that* $|\sigma| > 1$*, 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, $EDAC$ [16].

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) [13, 27, 26], 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 authorize an important efficiency gains with regard to the usual approach in constraint programming consist on encapsulate 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 [25], for each objective $j = 1, 2, \ldots, k$, we assume that E^j, the set of possible valuations, is a totally ordered set where $\perp^j$ denotes its minimal element and $\top^j$ its maximal element. In addition, we need a monotone and binary operator $\oplus^j$. 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 $\langle E^j, \oplus^j, \preccurlyeq^j \rangle$ such as:*

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

Once the valuation structure 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 $x \in X$.*
- *$\mathcal{S} = (S^1, \ldots, S^k)$ is a set of valuation structure. Each $S^j = (E^j, \oplus^j, \preccurlyeq^j),\ j : 1, \ldots, k$, is a 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 $\Phi_\sigma(t) = (\phi_\sigma^1(t), \ldots, \phi_\sigma^k(t))$.*
  where
  - *t is tuple of values from $\Pi_{x\in\sigma} D_x$,*
  - *k is the number of objectives,*
  - *ϕ_σ^j is the j^{th} objective function.*

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

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

where $t \downarrow \sigma$ denotes the projection of t on variables σ.

For a variable x, we can only assign a value of its domain. The arity of a multi-valued constraint is the size of its scope. The arity of the problem is the maximum arity of its constraints. In this work, we are concerned with binary MO-VCSPs. These are MO-VCSPs whose constraints are exclusively unary and binary.

In this work, we suppose that no two constraints have the same scope. This allows us to identify C with the set of scopes σ of cost functions Φ_σ in the MO-VCSP. We write Φ_x as a shorthand for $\Phi_{\{x\}}$ and Φ_{xy} as a shorthand for $\Phi_{\{x,y\}}$. Without loss of generality, we assume that C contains a unary multi-valued constraint $C_x = \langle x, \Phi_x \rangle$ for every variable $x \in X$ as well as a zero-arity multi-valued constraint $C_\varnothing = \langle \varnothing, \Phi_\varnothing \rangle$.

Purely for notational convenience and readability reasons, in the following and if there is no confusion, we will write Φ_σ, v, $\preccurlyeq$ and $\oplus$ instead of $C_\sigma = \langle \sigma, \Phi_\sigma \rangle$, $\vec{c}$, $\vec{\preccurlyeq}$ and $\vec{\oplus}$, respectively.

Finding an assignment that optimize all objective simultaneously is ideal. However, in general such an ideal assignment does not exist or cannot be reached, since trade-off exist among objectives. Thus, the (Optimal) solution of MO-VCSP is characterized by using the concept of *Pareto Optimality.*

Definition 5 (Dominance). *For a MO-VCSP and two vector $\mathcal{V}(t)$ and $\mathcal{V}(t')$ obtained by assignment t and t', we say that $\mathcal{V}(t)$ dominates $\mathcal{V}(t')$, denoted by $\mathcal{V}(t) \prec \mathcal{V}(t')$, iff $\mathcal{V}(t)$ is partially less than $\mathcal{V}(t')$, i.e., (i) $\phi^j(t) \preccurlyeq^j \phi^j(t')$ holds true for all objective $j \in 1..k$. And (ii) there exist at least one objective j such that $\phi^j(t) \prec^j \phi^j(t')$.*

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

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.*

Example 2 (MO-VCSP). Consider the problem depicted in figure 2(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 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 with the assignment (a, a). And the cost of solution 2 is the pair $(2, 200)$ and it is attained with the assignment (a, b).

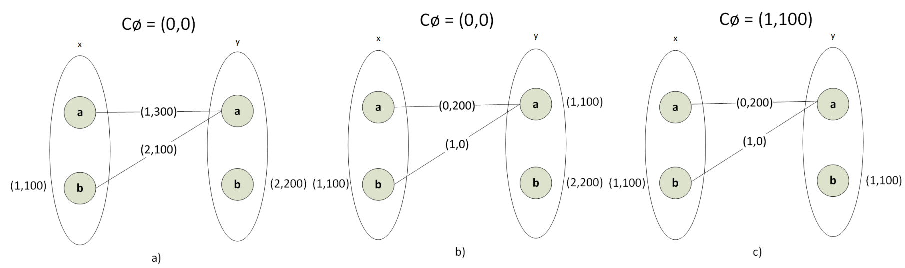

Fig. 2. Three equivalent MO-WCSP instances (T = (4,400))

3.2 Operations over costs and their extensions

The following definitions require that valuations structures considered for the MO-VCSP to be *fair*[6]. 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, that considered valuations structures must be a lattice which mean that any pairs of valuation (costs) must have a lower bound, denoted `LC`. 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 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. $\perp = (\perp_1, \ldots, \perp_k)$ is a k-vector, where each $\perp_j \in E_j$ 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 $\vec{u}$ and $\vec{v}$ be two distinct k-vectors.

- The aggregation of two costs value for an objective j is defined as :
$$u^j \oplus^j v^j \stackrel{\mathsf{def}}{=} \begin{cases} \top^j, & \text{if } u^j \oplus^j v^j = \top^j. \\ (u^j \oplus^j v^j), & \text{otherwise.} \end{cases}$$
- The aggregation of two k-ary cost vectors is defined as :
$$u \oplus v \stackrel{\mathsf{def}}{=} \begin{cases} \top, & \text{if } \exists j, u^j \oplus^j v^j = \top^j. \\ (u^1 \oplus^1 v^1, \ldots, u^k \oplus^k v^k), & \text{otherwise.} \end{cases}$$
- The subtraction of a cost value for an objective j, is defined as :
For two cost value $u^j, v^j \in E^j$, *such that* $u^j \succeq^j v^j$, the subtraction of v^j from u^j we have :
$$u^j \ominus^j v^j \stackrel{\mathsf{def}}{=} \begin{cases} u^j \ominus^j v^j, & \text{if } u^j \prec^j \top^j. \\ \perp^j, & \text{otherwise.} \end{cases}$$

– The subtraction of a vector of cost 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 :

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

We say that u *dominates* v (noted $u \prec_D v$) if $\forall j, u^j \preccurlyeq^j v^j$. Let S be a set of k-ary cost vectors. We define its non-domination closure as

$$\langle S \rangle = \{u \in S | \forall v \in S, v \not\prec_D u\}.$$

Let S_1 and S_2 be two sets closed under non-domination. We say that S_1 *dominate* S_2 (noted $S_1 \prec_D S_2$) if $\forall v \in S_2, \exists u \in S_1\ s.t\ u \prec_D v$.

Definition 8. *Lower Cost* (LC) *Let* s *be a set of cost, an element* $c \in \mathtt{s}$ *is the Lower Cost of* s *iff:* $c \preceq x, \forall x \in \mathtt{s}$.

Definition 9. *Lower Cost Vector* (LCV) *Let* $\mathcal{S} = S^1, \ldots, S^m$ *be a set of* m *cost vectors. The Lower Cost Vector of* $\mathcal{S}$ *denoted by* (LCV)$(\mathcal{S})$ *is defined as follow:*

$$\mathtt{LCV}(\mathcal{S}) = (\mathtt{LC}(S^1), \ldots, \mathtt{LC}(S^m))$$

We extend our previous definitions to deal with the multi-objective case.

Definition 10. *Lower Cost* ($\mathtt{LC^j}$) *Let* $\mathtt{s}^j$ *be a subset of* E^j*, an element* $c \in \mathtt{s}^j$ *is the Lower Cost for the objective* j *of* $\mathtt{s}^j$ *iff:* $c \preceq^j x, \forall x \in \mathtt{s}^j$.

j denote the considered objective.

Definition 11. *Lower Cost Vector* (LCV) *Let* $\mathcal{S} = S_1, \ldots, S_m$ *be a set of* m *k-cost vector, where each* $S_i = (S_i^1, \ldots, S_i^k)$*. The Lower Cost Vector of* $\mathcal{S}$ *denoted by* $\mathtt{LCV}(\mathcal{S})$ *is defined as follow:*

$$\mathtt{LCV}(\mathcal{S}) = \Big(\mathtt{LCV}(S_1), \ldots, \mathtt{LCV}(S_m)\Big)$$

i.e.,

$$\mathtt{LCV}(\mathcal{S}) = \Big(\big(\mathtt{LC}^1(S_1^1), \ldots, \mathtt{LC}^1(S_m^1)\big), \ldots, \big(\mathtt{LC}^k(S_1^k), \ldots, \mathtt{LC}^k(S_m^k)\big)\Big)$$

k denote the number of objectives

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, we study 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 compute 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 such 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 muti-objective case is based on three basic operations (project, extend and unary project). The main difference is that the transfered data between constraints are now k-ary cost vector.
The main Pareto based EPT (P-EPT) is defined bellow and described as Algorithm1. This is a an extension of the standard version defined in [7] to the multi-objective case.

Definition 12. *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 13. *The sub-problem of a MO-VCSP $(X, D, C, \mathcal{S})$ induced by $\mathcal{F} \subseteq C$ is the problem MO-VCSP($\mathcal{F}$) $= (X_{\mathcal{F}}, D_{\mathcal{F}}, C, \mathcal{S})$, where $X_{\mathcal{F}} = \bigcup_{\Phi_\sigma \in F} \sigma$ and $D_{\mathcal{F}} = \{D_i | i \in X_F\}$.*

Definition 14. *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($\mathcal{F}$) into an equivalent MO-VCSP.*

When F contains at most one k-ary cost functions Φ_σ such that $|\sigma| > 1$, such an P-EPT is called a Pareto Soft Arc Consistency (P-SAC) *operation.*

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

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

Algorithm1 gives three basic P-EPT which are also P-SAC operations [19]. *P-Project* projects costs vector from a set of costs functions (on two or more variables) to a set of unary costs functions. *P-Extend* performs the inverse operation, sending costs vectors from a set of unary costs functions to a set of higher-order costs functions. Each cost vector contains k-ary cost function for each objective $j \in 1 \dots k$. Finally *P-UnaryProject* projects costs vector from a set of unary costs functions to the nullary costs functions set $\Phi_\varnothing$ which is a set of k lower bound on the value of any solution. For each of the P-SAC operations given in Algorithm 1, a precondition is given which guarantees that costs value, for each objective, remain non-negative after the Pareto-EPT has been applied.

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 techniques of P-SAC, we need to introduce some new concepts related to the support notion. The main idea in our extension is take advantage of the `LCV` operator in the definition of *pareto support*.

In the multi objective case, we define the support of a variable x as being the set of values from the domain of x whose unary costs k-vector are included in a set of k-ary costs vectors S, maximum in terms of cardinality such that: $\mathtt{LCV}(S) = LCV_{u \in S}(\{\Phi_x(u)\}) = \perp$. we denote this set $\mathcal{PS}$ a shorthand for *Pareto Support Set*.

Definition 15. *Let* $(\langle x \rangle, \Phi_x)$ *a unary multi-objective constraint. A set of values* $\mathcal{S} \subseteq D_x$ *is a* Pareto Support *for* x *if* $\mathcal{S} \subseteq D_x$ *is maximal subset*[1] *of* D_x *and* $\mathtt{LCV}(\mathcal{S}) = \mathtt{LCV}_{u \in \mathcal{S}}(\Phi_x(u)) = \perp$. *we will denote the* Pareto Support Set *by* $\mathcal{PS}$. *Otherwise,*

$$\mathcal{PS}(x) \stackrel{def}{=} \begin{cases} a \in \mathcal{PS}(x), & si\ \exists j, \phi_{j,x}(a) = \perp_j. \\ a \notin \mathcal{PS}(x), & otherwise\ (\forall j, \phi^j_x(a) \succ \perp^j). \end{cases}$$

Definition 16. *Let* $(\langle x, y \rangle, \Phi_{xy})$ *a binary multi-objective constraint. A set of values* $\mathcal{S} \subseteq D_y$ *is a* Pareto Simple Support *for a value* $u\ inD_x$ *if* $\mathcal{S}$ *is maximal subset* [2] *and* $\mathtt{LCV}(\mathcal{S}) = \mathtt{LCV}_{v \in \mathcal{S}}(\Phi_{xy}(u, v)) = \perp$. *we will denote the* Pareto Simple Support Set *by* $\mathcal{PSS}$.

Definition 17. *Let* $(\langle x, y \rangle, \Phi_{xy})$ *a binary multi-objective constraint. A set of values* $\mathcal{S} \subseteq D_y$ *is a* Pareto Full Support *for a value* $u\ inD_x$ *if if* $\mathcal{S}$ *is maximal subset* [3] *and* $\mathtt{LCV}(\mathcal{S}) = \mathtt{LCV}_{v \in \mathcal{S}}(\Phi_{xy}(u, v) \bigoplus \Phi_y(v)) = \perp$. *we will denote the* Pareto Full Support Set *by* $\mathcal{PFS}$

[1] We say that $\mathcal{S}$ is a maximal subset of D_x if $\mathcal{S} \subseteq D_x$ and there doesn't exist $c' \in \mathcal{S}$ such that $\nexists j : 1..k,\ \phi^j_x(c') = \perp_j$

[2] We say that $\mathcal{S}$ is a maximal subset of D_y if $\mathcal{S} \subseteq D_y$ and there doesn't exist $c' \in \mathcal{S}$ such that $\nexists j : 1..k, \phi^j_{xy}(u, c') = \perp_j$

[3] We say that $\mathcal{S}$ is a maximal subset of D_y if $\mathcal{S} \subseteq D_y$ and there doesn't exist $c' \in \mathcal{S}$ such that $\nexists j : 1..k, \phi^j_{xy}(u, c') \oplus \phi_y(c') = \perp_j$

Definition 18 (Pareto Soft Node Consistency.). *A variable x is pareto soft node consistent (P-NC*) if each value $u \in D_x$ satisfies $\Phi_\varnothing \oplus \Phi_x(u) \prec \vec{\top}$ and $\Phi_\varnothing \oplus LCV_{u \in D_x}(\{\Phi_x(u)\}) = \Phi_\varnothing$.*

A MO-VCSP is P-NC iff all variables are P-NC*.*

Example 3. The variable x presented in the Figure 3 **(a)** is not Pareto soft node consistent. $LCV(\{\Phi_x(a)|a \in D_x\}) = (1,2) \neq \vec{\bot} = (0,0)$. While after applying $ProjectUnary$ operation, the variable x in Figure 3 **(b)** become Pareto soft node consistent. $LCV(\{\Phi_x(a)|a \in D_x\}) = (0,0) = \vec{\bot}$ and $\mathcal{PS}(x) = \{u, v\}$

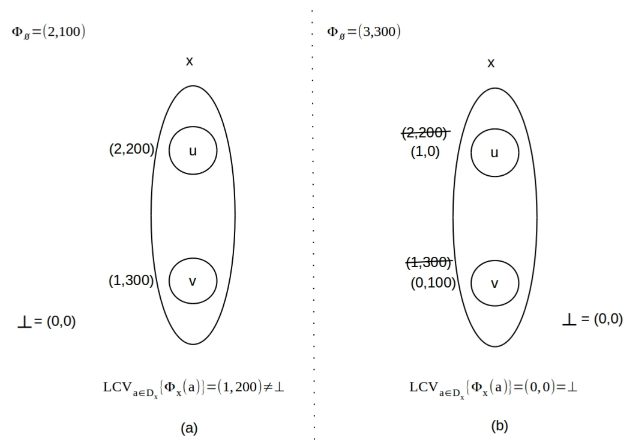

Fig. 3. Example of enforcing Pareto Soft NC* property (T = (4,400))

Definition 19 (Pareto Soft Arc Consistency.). *A variable x is Pareto soft arc consistent if for all value $u \in D_x$, $\Phi_x(u) \oplus LCV_{v \in D_y}(\{\Phi_{xy}(u,v)\}) = \Phi_x(u)$.*

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

Definition 20 (Pareto directional arc consistency.). *A variable x is* Pareto directional arc consistent (P-DAC*) *if $\forall u \in D_x, \forall y$ such that $y > x$:*

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

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

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 their establishment produces 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 [**?**]).

Definition 22 (Pareto Soft Existential Arc-consistency.). *A variable x is Pareto soft existential arc-consistency (P-EAC*) if $\exists$ a set $\mathcal{PS}(x) \subseteq D_x$ such that,*

$$\Phi_\varnothing \oplus LCV_{u\in\mathcal{PS}(x)}\{\Phi_x(u) \bigoplus_{\Phi_{xy}\in C \ s.t. \ y<x} LCV_{v\in D_y}\{\Phi_{xy}(u,v)\oplus\Phi_y(v)\}\} = \Phi_\varnothing$$

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 require applying P-EPT. Any Multi-objective Valued CSP can be transformed into an equivalent instance having P-NC* property by projecting any unary multi-valued constraint towards the zero-arity constraint $Phi_\varnothing$ and subsequently pruning every unfeasible value. 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. Algorithm2 introduces enforcing algorithms of various Pareto based soft local arc consistencies.

5 Discussion & Future works

Algorithm that compute lower bound such as mini-bucket elimination MBE [12, 18] or Existential Directional Arc consistency EDAC* [14] 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[17, 14].

The elementary operations made during applying P-SAC on MO-VCSPs are deleting values the projection and extension cost vector. All these operations can not add to the problem of binary constraints on which the filtering technique P-SAC* is applied. So the problem remains binary. 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-VCSPs, PSAC$*$ 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. About application domains of MO-VCSP, we believe that *Sensor Networks* would be a promising area. This problem can be viewed as a *Ressource Allocation Problem* which can

Algorithm 2 Enforcement algorithms of P-NC*, P-AC*, P-DAC*, P-FDAC*, P-EAC*

procedure ENFORCE P-$NC^*(x)$
  $PruneVar(x)$
  if $\Phi_\varnothing \oplus LCV_{u \in D_x}\{\Phi_x(u)\} \neq \Phi_\varnothing$ **then**
    $ProjectUnary(x)$

function PRUNEVAR(x)
  $flag \leftarrow False$
  for each $u \in D_x$ **do**
    if $\exists j,\ s.t.,\ \phi_x^j(u) = \top_j$ **then**
      $D_x \leftarrow D_x \setminus \{u\}$
      $flag \leftarrow True$
  return flag

procedure ENFORCE P-$AC^*(x, y)$
  for each $u \in D_x$ **do**
    $\alpha \leftarrow LCV_{v \in D_y}\{\Phi_{xy}(u, v)\}$
    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**
    $P[u] \leftarrow LCV_{v \in D_y}\{\Phi_{xy}(u, v) \oplus \Phi_y(v)\}$
    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-$EAC^*(x)$
  $\alpha \leftarrow LCV_{u \in \mathcal{PS}(x)}\{\Phi_x(u) \bigoplus_{\Phi_{xy} \in C\ s.t.\ y < x} LCV_{v \in D_y}\{\Phi_{xy}(u, v) \oplus \Phi_y(v)\}\}$
  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-$FDAC(x, y)$

be formalized as a COP and VCSP. For example, consider a sensor network in a territory, where each sensor can sense a certain area in this territory. When we consider this problem with multiple objective, e.g., electrical consumption, data management and quantity/quality of observation data, it can be formalized as a MO-VCSP.

6 Conclusion & Future works

The Valued Constraint Satisfaction Problem (VCSP) is a generic optimization problem defined by a network of local cost functions defined over discrete variables. It has applications in Artificial Intelligence, Operations Research, Bioinformatics 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 objective. We propose a new extensions of local arc consistency to the MO-VCSP. The incremental lower bounds set produced by pareto based local consistency can be used for pruning inside Branch and Bound search.

References

1. Allouche, D., Bessiere, C., Boizumault, P., de Givry, S., Gutierrez, P., Lee, J.H., Leung, K.L., Loudni, S., Métivier, J.P., Schiex, T., Wu, Y.: Tractability-preserving transformations of global cost functions. Artif. Intell. **238**(C), 166–189 (Sep 2016). https://doi.org/10.1016/j.artint.2016.06.005, `https://doi.org/10.1016/j.artint.2016.06.005`
2. Bistarelli, S., Gadducci, F., Larrosa, J., Rollon, E.: A soft approach to multi-objective optimization. In: Garcia de la Banda, M., Pontelli, E. (eds.) Logic Programming. pp. 764–768. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
3. Bistarelli, S., Gadducci, F., Larrosa, J., Rollon, E., Santini, F.: Local arc consistency for non-invertible semirings, with an application to multi-objective optimization. Expert Syst. Appl. **39**(2), 1708–1717 (Feb 2012). https://doi.org/10.1016/j.eswa.2011.06.062, `http://dx.doi.org/10.1016/j.eswa.2011.06.062`
4. Chiandussi, G., Codegone, M., Ferrero, S., Varesio, F.: Comparison of multi-objective optimization methodologies for engineering applications. Computers and Mathematics with Applications **63**(5), 912 – 942 (2012). https://doi.org/https://doi.org/10.1016/j.camwa.2011.11.057, `http://www.sciencedirect.com/science/article/pii/S0898122111010406`
5. Cooper, M., De Givry, S., Sanchez, M., Schiex, T., Zytnicki, M.: Virtual arc consistency for weighted csp. In: Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 1. pp. 253–258. AAAI'08, AAAI Press (2008), `http://dl.acm.org/citation.cfm?id=1619995.1620037`
6. Cooper, M., Schiex, T.: Arc consistency for soft constraints. Artif. Intell. **154**(1-2), 199–227 (Apr 2004). https://doi.org/10.1016/j.artint.2003.09.002, `http://dx.doi.org/10.1016/j.artint.2003.09.002`
7. Cooper, M.C.: High-order consistency in valued constraint satisfaction. Constraints **10**(3), 283–305 (Jul 2005). https://doi.org/10.1007/s10601-005-2240-3, `https://doi.org/10.1007/s10601-005-2240-3`

8. Cooper, M.C., Schiex, T.: Arc consistency for soft constraints. Artif. Intell. **154**(1-2), 199–227 (2004). https://doi.org/10.1016/j.artint.2003.09.002, http://dx.doi.org/10.1016/j.artint.2003.09.002
9. Cooper, M., de Givry, S., Sanchez, M., Schiex, T., Zytnicki, M., Werner, T.: Soft arc consistency revisited. Artificial Intelligence **174**(7), 449 – 478 (2010). https://doi.org/https://doi.org/10.1016/j.artint.2010.02.001, http://www.sciencedirect.com/science/article/pii/S0004370210000147
10. Deb, K., Kalyanmoy, D.: Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, Inc., New York, NY, USA (2001)
11. E, N.A., E, R., AA, N., M, M.: A multi-mode resource-constrained discrete time-cost tradeoff problem solving using an adjusted fuzzy dominance genetic algorithm. Scientia Iranica E (2013)
12. E, R., J, L.: Bucket elimination for multiobjective optimization problems. J. Heuristics (2006)
13. El-Amine Chergui, M., Moulaï, M., Zohra Ouaïl, F.: Solving the multiple objective integer linear programming problem. In: Le Thi, H.A., Bouvry, P., Pham Dinh, T. (eds.) Modelling, Computation and Optimization in Information Systems and Management Sciences. pp. 69–76. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
14. de Givry, S., Zytnicki, M., Heras, F., Larrosa, J.: Existential arc consistency: Getting closer to full arc consistency in weighted csps. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence. pp. 84–89. IJCAI'05, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2005), http://dl.acm.org/citation.cfm?id=1642293.1642307
15. Jeavons, P.G., Cooper, M.C.: Tractable constraints on ordered domains. Artificial Intelligence **79**(2), 327 – 339 (1995). https://doi.org/http://dx.doi.org/10.1016/0004-3702(95)00107-7, http://www.sciencedirect.com/science/article/pii/0004370295001077
16. Larrosa, J.: Node and arc consistency in weighted csp. In: Eighteenth National Conference on Artificial Intelligence. pp. 48–53. American Association for Artificial Intelligence, Menlo Park, CA, USA (2002), http://dl.acm.org/citation.cfm?id=777092.777103
17. Larrosa, J., Schiex, T.: 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. pp. 239–244 (2003), http://ijcai.org/Proceedings/03/Papers/035.pdf
18. Larrosa, J., Schiex, T.: Solving weighted CSP by maintaining arc consistency. Artif. Intell. **159**(1-2), 1–26 (2004). https://doi.org/10.1016/j.artint.2004.05.004, http://dx.doi.org/10.1016/j.artint.2004.05.004
19. Larrosa, J., Schiex, T.: Solving weighted csp by maintaining arc consistency. Artif. Intell. **159**(1-2), 1–26 (Nov 2004). https://doi.org/10.1016/j.artint.2004.05.004, http://dx.doi.org/10.1016/j.artint.2004.05.004
20. Lee, J., Leung, K.: Consistency techniques for flow-based projection-safe global cost functions in weighted constraint satisfaction. J. Artif. Intell. Res. (JAIR) **43**, 257–292 (2012). https://doi.org/10.1613/jair.3476
21. M, V.: New computational results for the discrete time/cost trade-off problem with time-switch constraints. European Journal of Operational Research. (2005)
22. O, H., M, H., E, E.: Robust scheduling and robustness measures for the discrete time/cost trade-off problem. European Journal of Operational Research. (2010)
23. P, V.B., DW, M.: The design and experimental analysis of algorithms for temporal reasoning. J. Artif. Intell. Res. (1996)

24. Schiex, T.: Arc consistency for soft constraints. In: Proceedings of the 6th International Conference on Principles and Practice of Constraint Programming. pp. 411–424. CP '02, Springer-Verlag, London, UK, UK (2000), http://dl.acm.org/citation.cfm?id=647487.726635
25. Schiex, T., Fargier, H., Verfaillie, G.: Valued constraint satisfaction problems: Hard and easy problems. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1. pp. 631–637. IJCAI'95, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1995), http://dl.acm.org/citation.cfm?id=1625855.1625938
26. Teghem, J.: Multi-objective integer linear programming. Springer US, Boston, MA (2009). https://doi.org/10.1007/978-0-387-74759-0420, https://doi.org/10.1007/978-0-387-74759-0420
27. Villarreal, B., Karwan, M.H.: Multicriteria integer programming: A (hybrid) dynamic programming recursive approach. Mathematical Programming **21**(1), 204–223 (Dec 1981). https://doi.org/10.1007/BF01584241, https://doi.org/10.1007/BF01584241
28. Wilson, N., Razak, A., Marinescu, R.: Computing possibly optimal solutions for multi-objective constraint optimisation with tradeoffs. In: Proceedings of the 24th International Conference on Artificial Intelligence. pp. 815–821. IJCAI'15, AAAI Press (2015), http://dl.acm.org/citation.cfm?id=2832249.2832362