

## Preference-based Search and Multi-criteria Optimization

Ulrich Junker

ILOG

1681, route des Dolines

F-06560 Valbonne

ujunker@ilog.fr

### Abstract

Many real-world AI problems (e.g. in configuration) are weakly constrained, thus requiring a mechanism for characterizing and finding the preferred solutions. Preference-based search (PBS) exploits preferences between decisions to focus search to preferred solutions, but does not efficiently treat preferences on defined criteria such as the total price or quality of a configuration. We generalize PBS to compute balanced, extreme, and Pareto-optimal solutions for general CSP's, thus handling preferences on and between multiple criteria. A master-PBS selects criteria based on trade-offs and preferences and passes them as optimization objective to a sub-PBS that performs a constraint-based Branch-and-Bound search. We project the preferences of the selected criterion to the search decisions to provide a search heuristics and to reduce search effort, thus giving the criterion a high impact on the search. The resulting method will particularly be effective for CSP's with large domains that arise if configuration catalogs are large.

**Keywords:** preferences, nonmonotonic reasoning, constraint satisfaction, multi-criteria optimization, search.

### Introduction

In this paper, we consider combinatorial problems that are weakly constrained and that lack a clear global optimization objective. Many real-world AI problems have these characteristics: examples can be found in configuration, design, diagnosis, but also in temporal reasoning and scheduling. An example for configuration is a vacation adviser system that chooses vacation destinations from a potentially very large catalog. Given user requirements (e.g. about desired vacation activities such as wind-surfing, canyoning), compatibility constraints between different destinations, and global 'resource' constraints (e.g. on price) usually still leave a large set of possible solutions. In spite of this, most of the solutions will be discarded as long as more interesting solutions are possible. Preferences on different choices and criteria are an adequate way to characterize the interesting solutions. For example, the user may prefer Hawaii to Florida for doing wind-surfing or prefer cheaper vacations in general.

Different methods for representing and treating preferences have been developed in different disciplines. In AI,

Copyright © 2002, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

preferences are often treated in a qualitative way and specify an order between hypotheses, default rules, or decisions. Examples for this have been elaborated in nonmonotonic reasoning (Brewka 1989) and constraint satisfaction (Junker 2000). Here, preferences can be represented by a predicate or a constraint, which allows complex preference statements (e.g. dynamic preferences, soft preferences, meta-preferences and so on). Furthermore, preferences between search decisions also allow to express search heuristics and to reduce search effort for certain kinds of scheduling problems (Junker 2000).

In our vacation adviser example, the basic decisions consist in choosing one (or several) destinations and we can thus express preferences between individual destinations. However, the user preferences are usually formulated on global criteria such as the total price, quality, and distance which are defined in terms of the prices, qualities, and distances of all the chosen destinations. We thus obtain a multi-criteria optimization problem.

We could try to apply the preference-based search (Junker 2000) by choosing the values of the different criteria before choosing the destinations. However, this method has severe draw-backs:

1. Choosing the value of a defined criterion highly constrains the remaining search problem and usually leads to a thrashing behaviour.
2. The different criteria are minimized in a strict order. We get solutions that are optimal w.r.t. some lexicographic order, but none that represents compromises between the different criteria. E.g., the system may propose a cheap vacation of bad quality and an expensive vacation of good quality, but no compromise between price and quality.

Hence, a naive application of preferences between decisions to multi-criteria optimization problems can lead to thrashing and lacks a balancing mechanism.

Multi-criteria optimization avoids those problem. In Operations Research, a multi-criteria optimization problem is usually mapped to a single or a sequence of single-criterion optimization problems which are solved by traditional methods. Furthermore, there are several notions of optimality such as Pareto-optimality, lexicographic optimality, and lexicographic max-order optimality (Ehrgott 1997). We can thus determine 'extreme solutions' where one criteria is

favoured to another criteria as well as 'balanced solutions' where the different criteria are as close together as possible and that represent compromises. This balancing requires that the different criteria are comparable, which is usually achieved by a standardization method. Surprisingly, the balancing is not achieved by weighted sums of the different criteria, but by a new lexicographic approach (Ehrgott 1997). In order to find a compromise between a good price and a good quality, Ehrgott first minimizes the maximum between (standardized versions of) price and quality, fixes one of the criteria (e.g. the standardized quality) at the resulting minimum, and then minimizes the other criterion (e.g. the price).

In this paper, we will develop a modified version of preference-based search that solves a minimization subproblem for finding the best value of a given criterion instead of trying out the different value assignments. Furthermore, we also show how to compute Pareto-optimal and balanced solutions with new versions of preference-based search.

Multi-criteria optimization as studied in Operations Research also has draw-backs. Qualitative preferences as elaborated in AI can help to address following issues:

1. We would like to state that certain criteria are more important than other criteria without choosing a total ranking of the criteria as required by lexicographic optimality. For example, we would like to state a preference between a small price and a high quality on the one hand and a small distance on the other hand, but we still would like to get a solution where the price is minimized first and a solution where the quality is maximized first.
2. Multi-criteria optimization specifies preferences on defined criteria, but it does not translates them to preferences between search decisions. In general, it is not evident how to automatically derive a search heuristics from the selected optimization objective. Adequate preferences between search decisions provide such a heuristics and also allow to apply preference-based search to reduce the search effort for the subproblem.

In order to address the first point, we compare the different notions of optimal solutions with the different notions of preferred solutions that have been elaborated in non-monotonic reasoning. If no preferences between criteria are given, the Pareto-optimal solutions correspond to the G-preferred solutions (Grosz 1991; Geffner & Pearl 1992; Junker 1997) and the lexicographic-optimal solutions correspond to the B-preferred solutions (Brewka 1989; Junker 1997). Preferences between criteria can easily be taken into account by the latter methods. For balanced solutions, we present a variant of Ehrgott's definition that respects preferences between criteria as well. The different versions of preference-based search will also respect these additional preferences. We thus obtain a system where the user can express preferences on the criteria and preferences between the criteria and choose between extreme solutions, balanced solutions and Pareto-optimal solutions.

As mentioned above, the new versions of preference-based search solve a minimization subproblem when determining the best value for a selected criterion. We would like to also use preference-based search for solving the sub-

problems. However, the preferences are only expressed on the criteria and not on the search decisions. It therefore is a natural idea to project the preferences on the selected criterion to the search decisions. We will introduce a general method for preference projection, which we then apply to usual objectives such as sum, min, max, and element constraints. It is important to note that these projected preferences will change from one subproblem to the other. The projected preferences will be used to guide the search and to reduce search effort. Depending on the projected preferences, completely different parts of the search space may be explored and, in particular, the first solution depends on the chosen objective. Search effort can be reduced since the projected preferences preserve Pareto-optimality. We therefore adapt the new preference-based search method for Pareto-optimal solutions for solving the subproblems.

The paper is organized as follows: we first introduce different notions of optimality from multi-criteria optimization and then extend them to cover preferences between criteria. After this, we develop new versions of preference-based search for computing the different kinds of preferred solutions. Finally, we introduce preference projection.

## Preferred Solutions

We first introduce different notions of optimality from multi-criteria optimization and then link them to definitions of preferred solutions from nonmonotonic reasoning.

### Preferences on Criteria

Throughout this paper, we consider combinatorial problems that have the decision variables  $\mathcal{X} := (x_1, \dots, x_m)$ , the criteria  $\mathcal{Z} := (z_1, \dots, z_n)$ , and the constraints  $\mathcal{C}$ . Each decision variable  $x_i$  has a domain  $D(x_i)$  from which its values will be chosen. For example,  $x_i$  may represent the vacation destination in the  $i$ -th of  $m = 3$  weeks. The constraints in  $\mathcal{C}$  have the form  $C(x_1, \dots, x_m)$ . Each constraint symbol  $C$  has an associated relation  $R_C$ . In our example, there may be compatibility constraints (e.g., the destinations of two successive vacation destinations must belong to neighbored countries) and requirements (e.g., at least one destination should allow wind-surfing and at least one should allow museum visits). Each criterion  $z_i$  has a definition in form of a functional constraint  $z_i := f_i(x_1, \dots, x_m)$  and a domain  $D(z_i)$ . Examples for criteria are price, quality, and distance (zone). The price is a sum of element constraints:

$$price := \sum_{i=1}^m price(x_i)$$

The total quality is defined as minimum of the individual qualities and the total distance is the maximum of the individual distances. The individual prices, qualities, and destinations are given by tables such as the catalog in figure 1.

A solution  $S$  of  $(\mathcal{C}, \mathcal{X})$  is a set of assignments  $\{x_1 = v_1, \dots, x_m = v_m\}$  of values from  $D(x_i)$  to each  $x_i$  such that all constraints in  $\mathcal{C}$  are satisfied, i.e.  $(v_1, \dots, v_m) \in R_C$  for each constraint  $C(x_1, \dots, x_m) \in \mathcal{C}$ . We write  $v_S(z_i)$  for the value  $f_i(v_1, \dots, v_m)$  of  $z_i$  in the solution  $S$ .

Destination	Price	Quality	Distance	Activities
Athens	60	1	4	museums, wind-surfing
Barcelona	70	2	3	museums, wind-surfing
Florence	80	3	3	museums
London	100	5	2	museums
Munich	90	4	2	museums
Nice	90	4	2	wind-surfing
...				

Figure 1: Catalog of a fictive hotel chain

Furthermore, we introduce preferences between the different values for a criterion  $z_i$  and thus specify a multi-criteria optimization problem. Let  $\prec_{z_i} \subseteq D(z_i) \times D(z_i)$  be a strict partial order for each  $z_i$ . For example, we choose  $\prec$  for price and distance and  $\succ$  for quality. We write  $u \preceq v$  iff  $u \prec v$  or  $u = v$ . Multiple criteria optimization provides different notions of optimality. The most well-known examples are Pareto optimality, lexicographic optimality, and optimality w.r.t. weighted sums.

A Pareto-optimal solution  $S$  is locally optimal. If another solution  $S^*$  is better than  $S$  w.r.t. a criterion  $z_j$  then  $S$  is better than  $S^*$  for some other criterion  $z_k$ :

**Definition 1** A solution  $S$  of  $(\mathcal{C}, \mathcal{X})$  is a Pareto-optimal solution of  $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec_{z_i})$  iff there is no other solution  $S^*$  of  $(\mathcal{C}, \mathcal{X})$  s.t.  $v_{S^*}(z_k) \prec_{z_k} v_S(z_k)$  for a  $k$  and  $v_{S^*}(z_i) \preceq_{z_i} v_S(z_i)$  for all  $i$ .

Pareto-optimal solutions narrow down the solution space since non-Pareto-optimal solution do not appear to be acceptable. However, their number is usually too large in order to enumerate them all. Figure 2 (left) shows the Pareto-optimal solutions  $S_1$  to  $S_8$  for the two criteria  $z_1$  and  $z_2$ .

From now on, we suppose that all the  $\prec_{z_i}$ 's are total orders. This simplifies the presentation of definitions and algorithms. A lexicographic solution requires to choose a ranking of the different criteria. We express it by a permutation of the indices:

**Definition 2** Let  $\pi$  be a permutation of  $1, \dots, n$ . Let  $V_S(\pi(\mathcal{Z})) := (v_S(z_{\pi_1}), \dots, v_S(z_{\pi_n}))$ . A solution  $S$  of  $(\mathcal{C}, \mathcal{X})$  is an extreme solution of  $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec_{z_i})$  iff there is no other solution  $S^*$  of  $(\mathcal{C}, \mathcal{X})$  s.t.  $V_{S^*}(\pi(\mathcal{Z})) \prec_{lex} V_S(\pi(\mathcal{Z}))$ .

Different rankings lead to different extreme<sup>1</sup> solutions which are all Pareto-optimal. In figure 2 (left), we obtain the extreme solutions  $S_1$  where  $z_1$  is preferred to  $z_2$  and  $S_8$  where  $z_2$  is preferred to  $z_1$ . Extreme solutions can be determined by solving a sequence of single-criterion optimization problems starting with the most important criterion.

If we cannot establish a preference order between different criteria then we would like to be able to find compromises between them. Although weighted sums (with equal weights) are often used to achieve those compromises, they do not necessarily produce the most balanced solutions. If

<sup>1</sup>We use the term extreme in the sense that certain criteria have an absolute priority over other criteria.

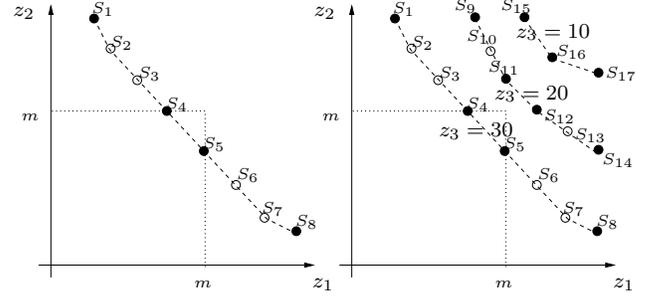


Figure 2: Pareto-optimal solutions for minimization criteria

we choose the same weights for  $z_1$  and  $z_2$ , we obtain  $S_7$  as the optimal solution. Furthermore, if we slightly increase the weight of  $z_1$  the optimal solution jumps from  $S_7$  to  $S_2$ . Hence, weighted sums, despite of their frequent use, do not appear a good method for balancing.

In (Ehrgott 1997), Ehrgott uses lexicographic max-orderings to determine optimal solutions. In this approach, values of different criteria need to be comparable. For this purpose, we assume that the preference orders  $\prec_{z_i}$  of the different criteria are equal to a fixed order  $\prec_D$ . This usually requires some scaling or standardization of the different criteria. We also introduce the reverse order  $\succ_D$  which satisfies  $z_i \succ_D z_j$  iff  $z_j \prec_D z_i$ . When comparing two solutions  $S_1$  and  $S_2$ , the values of the criteria in each solution are first sorted w.r.t. the order  $\succ_D$  before being compared by a lexicographic order. This can lead to different permutations of the criteria for different solutions. We describe the sorting by a permutation  $\rho^S$  that depends on a given solution  $S$  and that satisfies two conditions:

1.  $\rho^S$  sorts the criteria in a decreasing order: if  $v_S(z_i) \succ_D v_S(z_j)$  then  $\rho_i^S < \rho_j^S$ .
2.  $\rho^S$  does not change the order if two criteria have the same value: if  $i < j$  and  $v_S(z_i) = v_S(z_j)$  then  $\rho_i^S < \rho_j^S$ .

**Definition 3** A solution  $S$  of  $(\mathcal{C}, \mathcal{X})$  is a balanced solution of  $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec_D)$  iff there is no other solution  $S^*$  of  $(\mathcal{C}, \mathcal{X})$  s.t.  $V_{S^*}(\rho^{S^*}(\mathcal{Z})) \prec_{lex} V_S(\rho^S(\mathcal{Z}))$ .

Balanced solutions are Pareto-optimal and they are those Pareto-optimal solutions where the different criteria are as close together as possible. In the example of figure 2 (left), we obtain  $S_5$  as balanced solution. According to Ehrgott, it can be determined as follows: first  $\max(z_1, z_2)$  is minimized. If  $m$  is the resulting optimum, the constraint  $\max(z_1, z_2) = m$  is added before  $\min(z_1, z_2)$  is minimized. Balanced solutions can thus be determined by solving a sequence of single-criterion optimization problems.

### Preferences between Criteria

If many criteria are given it is natural to specify preferences between different criteria as well. For example, we would like to specify that a (small) price is more important than a (short) distance without specifying anything about the quality. We therefore introduce preferences between criteria in form of a strict partial order  $\prec_{\mathcal{Z}} \subseteq \mathcal{Z} \times \mathcal{Z}$ .

Preferences on criteria and between criteria can be aggregated to preferences between assignments of the form  $z_i = v$ . Let  $\prec$  be the smallest relation satisfying following two conditions: 1. If  $u \prec_{z_i} v$  then  $(z_i = u) \prec (z_i = v)$  for all  $u, v$  and 2. If  $z_i \prec_{\mathcal{Z}} z_j$  then  $(z_i = u) \prec (z_j = v)$  for all  $u, v$ . Hence, if a criteria  $z_i$  is more important than  $z_j$  then any assignment to  $z_i$  is more important than any assignment to  $z_j$ . In general, we could also have preferences between individual value assignments of different criteria. In this paper, we simplified the structure of the preferences in order to keep the presentation simple.

In nonmonotonic reasoning, those preferences  $\prec$  between assignments can be used in two different ways:

1. as specification of a preference order between solutions.
2. as (incomplete) specification of a total order (or ranking) between all assignments, which is in turn used to define a lexicographic order between solutions.

The ceteris-paribus preferences (Boutilier *et al.* 1997) and the G-preferred solutions of (Grosz 1991; Geffner & Pearl 1992) follow the first approach, whereas the second approach leads to the B-preferred solutions of (Brewka 1989; Junker 1997). We adapt the definitions in (Junker 1997) to the specific preference structure of this paper:

**Definition 4** A solution  $S$  of  $(\mathcal{C}, \mathcal{X})$  is a G-preferred solution of  $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$  if there is no other solution  $S^*$  of  $(\mathcal{C}, \mathcal{X})$  such that  $v_S(z_k) \neq v_{S^*}(z_k)$  for some  $k$  and for all  $i$  with  $v_S(z_i) \prec_{z_i} v_{S^*}(z_i)$  there exists a  $j$  s.t.  $z_j \prec_{\mathcal{Z}} z_i$  and  $v_{S^*}(z_j) \prec_{z_j} v_S(z_j)$ .

Hence, a criterion can become worse if a more important criterion is improved. In figure 2 (right),  $S_1$  to  $S_8$  are G-preferred if  $z_1 \prec_{\mathcal{Z}} z_3$  and  $z_2 \prec_{\mathcal{Z}} z_3$  are given. Each G-preferred solution corresponds to a Pareto-optimal solution. If there are no preferences between criteria, each Pareto-optimal solution corresponds to some G-preferred solution. However, if there are preferences between criteria, certain Pareto-optimal solutions  $S$  are not G-preferred. There can be a G-preferred solution  $S^*$  that is better than  $S$  for a criterion  $z_i$ , but worse for a less important criterion  $z_j$  (i.e.  $z_i \prec_{\mathcal{Z}} z_j$ ).

In general, we may get new G-preferred solutions if we add new constraints to our problem. However, adding upper bounds on criteria does not add new G-preferred solutions:

**Proposition 1**  $S$  is a G-preferred solution of  $(\mathcal{C} \cup \{z_i \prec_{z_i} u\}, \mathcal{X}, \mathcal{Z}, \prec)$  iff  $S$  is a G-preferred solution of  $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$  and  $v_S(z_i) \prec_{z_i} u$ .

Although this property appears to be trivial it is not satisfied for the B-preferred solutions which will be introduced next. It will be essential for computing G-preferred solutions.

In the definition of lexicographic optimal solutions, a single ranking of the given criteria is considered. In the definition of B-preferred solutions, we consider all rankings that respect the given preferences between the criteria. Following definition has been adapted from (Brewka 1989; Junker 1997) to our specific preference structure:

**Definition 5** A solution  $S$  of  $(\mathcal{C}, \mathcal{X})$  is a B-preferred solution of  $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$  if there exists a permutation  $\pi$  such

that 1.  $\pi$  respects  $\prec_{\mathcal{Z}}$  (i.e.  $z_i \prec_{\mathcal{Z}} z_j$  implies  $\pi_i < \pi_j$ ) and 2. there is no other solution  $S^*$  of  $(\mathcal{C}, \mathcal{X})$  satisfying  $V_{S^*}(\pi(\mathcal{Z})) \prec_{lex} V_S(\pi(\mathcal{Z}))$ .

The B-preferred solution for  $\pi$  can be computed by solving a sequence of minimization problems: Let  $A_0 := \emptyset$  and

$$A_i := A_{i-1} \cup \{z_{\pi_i} = m\}$$

where

$$m = \min_{\prec_{z_{\pi_i}}} \{v \mid \mathcal{C} \cup A_{i-1} \cup \{z_{\pi_i} = v\} \text{ has a solution}\}$$

In figure 2 (right),  $S_1$  and  $S_8$  are B-preferred (for  $z_1 \prec_{\mathcal{Z}} z_3$  and  $z_2 \prec_{\mathcal{Z}} z_3$ ). Each B-preferred solution corresponds to an extreme solution. If there are no preferences between criteria, each extreme solution corresponds to some B-preferred solution. If there are preferences between criteria certain extreme solutions may not be B-preferred. For example,  $S_{15}$  is an extreme solution, which is obtained if first the distance is minimized and then the price. However, this ranking of criteria does not respect the given preferences.

In (Junker 1997), it has been shown that each B-preferred solution is a G-preferred one, but that the converse is not true in general. In figure 2 (right),  $S_2$  to  $S_6$  are G-preferred, but not B-preferred. These solutions assign a worse value to  $z_1$  than the B-preferred solution  $S_1$ , but a better value than  $S_8$ . Similarly, they assign a better value to  $z_2$  than  $S_8$ , but a worse value than  $S_1$ . It is evident that such a case cannot arise if each criteria has only two possible values. Hence, we get an equivalence in following case, where no compromises are possible:

**Proposition 2** If there is no  $z_i$  such that  $v_1 \prec_{z_i} v_2 \prec_{z_i} v_3$  and  $\mathcal{C} \cup \{z = v_i\}$  has a solution for  $i = 1, 2, 3$  then each G-preferred solution of  $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$  is also a B-preferred solution of  $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$ .

So far, we simply adapted existing notions of preferred solutions to our preference structure and related them to well-known notions of optimality. We now introduce a new kind of preferred solutions that generalizes the balanced solutions. We want to be able to balance certain criteria, e.g. the price and the quality, but prefer these two criteria to other criteria such as the distance. Hence, we limit the balancing to certain groups of criteria instead of finding a compromise between all criteria. For this purpose, we partition  $\mathcal{Z}$  into disjoint sets  $G_1, \dots, G_k$  of criteria. Given a criterion  $z$ , we also denote its group by  $G(z)$ . The criteria in a single group  $G_i$  will be balanced. The groups themselves are handled by using a lexicographic approach. Thus, we can treat preferences between different groups, but not between different criteria of a single group. Given a strict partial order  $\prec_G$  between the  $G_i$ 's, we can easily define an order  $\prec_{\mathcal{Z}}$  between criteria: if  $G_1 \prec G_2$  and  $z_i \in G_1, z_j \in G_2$  then  $z_i \prec_{\mathcal{Z}} z_j$ .

We now combine definitions 5 and 3. As in definition 5, we first choose a global permutation  $\pi$  that respects the preferences between groups. We then locally sort the values of each balancing group in a decreasing order. We describe this local sorting by a permutation  $\theta^S$  that depends on a given solution  $S$  and that satisfies three conditions:

1.  $\theta^S$  can only exchange variables that belong to the same balanced group:  $G(z_i) = G(z_{\theta^S_i})$ .

**Algorithm Extreme-PBS1**( $\mathcal{C}, \mathcal{Z}, \prec$ )

1.  $A := \emptyset; U := \mathcal{Z}; Q := \emptyset;$
2. **while**  $Q \cup U \neq \emptyset$  **do**
3.   **for all**  $(z = q) \in Q$  **do**
4.     let  $m$  be *minimize*( $\mathcal{C} \cup A, z, \prec_z$ )
5.     **if**  $q \prec_z m$  **then**  $Q := Q - \{z = q\};$
6.     **if**  $U = \emptyset$  and  $Q \neq \emptyset$  **then fail;**
7.      $B := \{y \in U \mid \nexists y^* \in U : y^* \prec_z y \text{ and}$
8.          $\nexists (y = q) \in Q\};$
9.     **if**  $B = \emptyset$  **then fail else**
10.     select  $z \in B;$
11.     let  $m$  be *minimize*( $\mathcal{C} \cup A, z, \prec_z$ );
12.     **choose**  $A := A \cup \{z = m\}; U := U - \{z\}$
13.     **or**  $Q := Q \cup \{z = m\};$
14. **return**  $A;$

Figure 3: Algorithm Extreme-PBS1

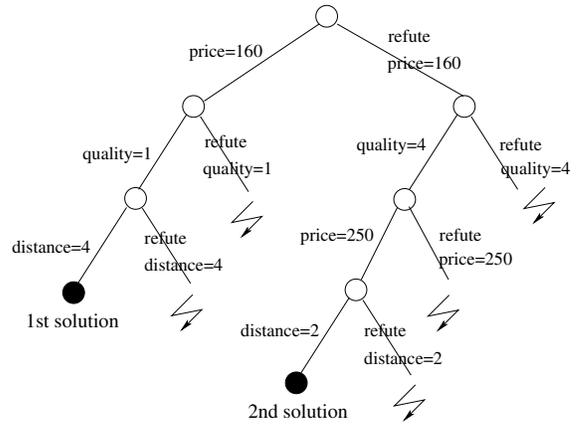


Figure 4: Finding extreme solutions

2.  $\theta^S$  sorts the criteria of each group in a decreasing order: if  $v_S(z_i) \succ_D v_S(z_j)$  and  $G(z_i) = G(z_j)$  then  $\theta_i^S < \theta_j^S$ .
3.  $\theta^S$  does not change the order if two criteria of the same group have the same value: if  $i < j$ ,  $v_S(z_i) = v_S(z_j)$ , and  $G(z_i) = G(z_j)$  then  $\theta_i^S < \theta_j^S$ .

**Definition 6** A solution  $S$  of  $(\mathcal{C}, \mathcal{X})$  is an E-preferred solution of  $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$  if there exists a permutation  $\pi$  such that 1.  $\pi$  respects  $\prec_{\mathcal{Z}}$  (i.e.  $z_i \prec_{\mathcal{Z}} z_j$  implies  $\pi_i < \pi_j$ ) and 2. there is no other solution  $S^*$  of  $(\mathcal{C}, \mathcal{X})$  s.t.  $V_{S^*}(\theta^{S^*}(\pi(\mathcal{Z}))) \prec_{lex} V_S(\theta^S(\pi(\mathcal{Z})))$ .

In figure 2 (right),  $S_5$  is E-preferred ( $z_1 \prec_{\mathcal{Z}} z_3$  and  $z_2 \prec_{\mathcal{Z}} z_3$ ). Each E-preferred solution corresponds to a balanced solution. If there are no preferences between criteria, each balanced solution corresponds to an E-preferred solution.

Interestingly, we can map E-preferred solutions to B-preferred solutions if we introduce suitable variables and preferences. For each group  $G$  of cardinality  $n_G$ , we use following min-max-variables  $y_{G,n_G}, \dots, y_{G,1}$ :

$$y_{G,i} := \min\{\max(X) \mid X \subseteq G \text{ s.t. } |X| = i\} \quad (1)$$

where  $\max(X) := \max\{z \mid z \in X\}$ . Let  $\hat{\mathcal{Z}}$  be the set of all of these min-max-variables. Following preferences ensure that min-max-variables for larger subsets  $X$  are more important:

$$y_{G,i} \hat{\succ} y_{G,i-1} \text{ for } i = n_G, \dots, 2 \quad (2)$$

A preference between a group  $G^*$  and a group  $G$  can be translated into a preference between the last min-max-variable of  $G^*$  and the first one of  $G$ :

$$y_{G^*,1} \hat{\succ} y_{G,n_G} \quad (3)$$

The E-preferred solutions then correspond to the B-preferred solutions of the translated criteria and preferences:

**Theorem 1**  $S$  is an E-preferred solution of  $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$  iff  $S$  is a B-preferred solution of  $(\mathcal{C}, \mathcal{X}, \hat{\mathcal{Z}}, \hat{\prec})$ .

We have thus established variants of Pareto-optimal, extreme, and balanced solutions that take into account preferences between criteria. On the one hand, we gain a better understanding of the existing preferred solutions by this comparison with notions from multi-criteria optimization. On

the other hand, we obtain a balancing mechanism that fits well into the qualitative preference framework.

### Preference-based Search

We now adapt the preference-based search algorithm from (Junker 2000) to treat preferences on criteria and to compute Pareto-optimal and balanced solutions as well.

### Extreme Solutions

The algorithm in (Junker 2000) can easily be adapted to preferences on criteria. The resulting algorithm is shown in figure 3. We explain its basic idea for the small example shown in figure 4, where price and quality are preferred to distance. The algorithm maintains a set  $U$  of unexplored criteria, which is initialized with the set of all criteria (i.e. price, quality, and distance). In each step, the algorithm selects a best criterion  $z$  of  $U$  (e.g. the price). Instead of trying to assign different values to the total price, we determine the cheapest price by solving a minimization subproblem:

$$\min_{\prec_z} \{v \mid A \cup \{z = v\} \text{ has a solution}\} := \text{minimize}(A, z, \prec_z) \quad (4)$$

In order to obtain a unique result, we assume that the orders  $\prec_{z_i}$  are strict total orders throughout the entire section. In our example, the cheapest solution has a price of 160. We now add the assignment  $price = 160$  to an initially empty set  $A$  of assignments. In figure 3, the elements of  $A$  occur as labels of the left branches. We then determine the best quality under this assignment. Once the price and quality have been determined we can determine a distance as well.

In order to find further solutions, *Extreme-PBS1* does not add the negation of assignments, but introduces a refutation query for each assignment  $z = v$ . We say that  $z = v$  is refuted if it becomes inconsistent after assigning values to unexplored criteria that may precede  $z$ . The refutation queries are added to a set  $Q$ . We can remove an element from  $Q$  if it has been refuted by further assignments. The assignment to the distance cannot be refuted since there are no further unexplored criteria. The quality of 1 cannot be refuted since the single non-explored criterion distance cannot precede the

quality. However, we can refute the price of 160 by first maximizing the quality. After this, we can again minimize the price and the distance, which leads to a new solution as shown in figure 3.

**Theorem 2** *Algorithm Extreme-PBS1( $\mathcal{C}, \mathcal{Z}, \prec$ ) always terminates. Each successful run returns a B-preferred solution of  $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$  and each such B-preferred solution is returned by exactly one successful run.*

According to theorem 1, we can use this algorithm to also compute balanced solutions supposed we provide it with the adequately translated criteria and preferences.

### Pareto-optimal Solutions

The algorithm for B-preferred solutions is thus relatively simple. Computing G-preferred solutions turns out to be more subtle. Interestingly, most operations of algorithm *Extreme-PBS1* are also valid for G-preferred solutions except for the rules in lines 6 and 8, where the algorithm backtracks since no B-preferred solution exists that is compatible with the given assignments and refutation queries. An algorithm for G-preferred solutions cannot backtrack in this case since there may be G-preferred solutions, which are not B-preferred. In order to obtain an algorithm for computing G-preferred solutions, we have to avoid such a situation. The basic idea is to add additional constraints that produce the equivalence between G-preferred and B-preferred solutions that is stated in proposition 2. We need to reduce the domain of each criterion  $z$  such that all values are either a best or a worst element of the domain. If there are intermediate values  $u$  between a best value  $q$  and a worst value  $v$  then we consider two possibilities: either we impose  $u$  as upper bound on  $z$  by adding a constraint  $z \preceq_z u$  or we require that  $z = u$  is refuted. In general, adding constraints can introduce new G-preferred solutions. Proposition 1 states that this is not the case if upper bounds are added.

The resulting algorithm is given in figure 5. It needs the set of values that have not yet been eliminated by some upper bound:

$$Pos(A, z) := \{v \mid \exists(z \preceq u) \in A : u \prec_z v\} \quad (5)$$

The algorithm 5 determines all G-preferred solutions:

**Theorem 3** *Algorithm Pareto-PBS1( $\mathcal{C}, \mathcal{Z}, \prec$ ) always terminates. Each successful run returns a G-preferred solution of  $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$  and each G-preferred solution is returned by exactly one successful run.*

Adding upper bounds thus helps to control the behaviour of PBS and to avoid that it jumps from one extreme solution to the other. In figure 4, minimizing the quality refutes the cheapest price of 160, as it was required. However, we now obtain a very high price of 250. In order to obtain compromises between price and quality, we constrain the price to be strictly smaller than 250 before minimizing the quality. This discussion indicates that we need not consider all upper bounds. We first determine an extreme solution that leads to a high price and we use this value to impose an upper bound on the price. Hence, we can extract the values of the upper bounds from the previous solutions found so far.

### Algorithm Pareto-PBS1( $\mathcal{C}, \mathcal{Z}, \prec$ )

```

1.  A := ∅; U := Z; Q := ∅;
2.  while Q ∪ U ≠ ∅ do
3.    for all (z = q) ∈ Q do
4.      let m be minimize(C ∪ A, z, <_z)
5.      if q <_z m then Q := Q - {z = q};
6.    if U = ∅ and Q ≠ ∅ then fail;
7.    B := {y ∈ U | ∃y* ∈ U : y* <_z y and
8.           ∃(y = q) ∈ Q};
9.    if there is (z = q) ∈ Q and v, w ∈ Pos(A, z)
10.   s.t. q <_z v <_z w then
11.     select a <_z-minimal element u
12.     s.t. u ∈ Pos(A, z) and q <_z u <_z w;
13.     choose A := A ∪ {z ≼_z u};
14.     or Q := Q ∪ {z = u};
15.   else if B = ∅ then fail else
16.     select z ∈ B;
17.     let m be minimize(C ∪ A, z, <_z);
18.     A := A ∪ {z ≼_z m};
19.     choose A := A ∪ {z = m}; U := U - {z}
20.     or Q := Q ∪ {z = m};
21. return A;
```

Figure 5: Algorithm Pareto-PBS1

### Preference Projection

A multi-criteria optimization problem is often solved by a sequence of single-criterion optimization problems having different objectives. We can, for example, solve each of these subproblems by a constraint-based Branch-and-Bound which maintains the best objective value found so far. Now, when changing the objective, the search heuristics should be adapted as well. It is a natural idea to project the preference order of the objective to the decision variables that appear in its definition. We define preference projection as follows:

**Definition 7**  $\prec_{x_k}$  is a projection of  $\prec_{z_j}$  via  $f_j(x_1, \dots, x_m)$  to  $x_k$  if and only if following condition holds for all  $u_1, \dots, u_m$  and  $v_1, \dots, v_m$  with  $u_i = v_i$  for  $i = 1, \dots, k - 1, k + 1, \dots, m$ :

$$\text{if } u_k \prec_{x_k} v_k \text{ then } f_j(u_1, \dots, u_m) \preceq_y f_j(v_1, \dots, v_m) \quad (6)$$

**Definition 8**  $\prec_{x_1}, \dots, \prec_{x_m}$  is a projection of  $\prec_{z_1}, \dots, \prec_{z_n}$  via  $f_1, \dots, f_n$  to  $x_1, \dots, x_m$  if  $\prec_{x_i}$  is a projection of  $\prec_{z_j}$  via  $f_j(x_1, \dots, x_m)$  to  $x_i$  for all  $i, j$ .

The projected preferences preserve Pareto-optimality:

**Theorem 4** Let  $\prec_{x_1}, \dots, \prec_{x_m}$  be a projection of  $\prec_{z_1}, \dots, \prec_{z_n}$  via  $f_1, \dots, f_n$  to  $x_1, \dots, x_m$ . If  $S$  is a Pareto-optimal solution w.r.t. the criteria  $z_1, \dots, z_n$  and the preferences  $\prec_{z_1}, \dots, \prec_{z_n}$  then there exists a solution  $S^*$  that 1. is a Pareto-optimal solution w.r.t. the criteria  $x_1, \dots, x_m$  and the preferences  $\prec_{x_1}, \dots, \prec_{x_m}$  and 2.  $v_{S^*}(z_i) = v_S(z_i)$  for all criteria  $z_i$ .

Since extreme and balanced solutions are Pareto-optimal we can additionally use the projected preferences to reduce search effort when solving a sub-problem. For this purpose, we adapt the algorithm Pareto-PBS1 to the decision

variables  $x_1, \dots, x_m$  and do consistency checks instead of minimize-calls.

We give some examples for preference projections:

1. The increasing order  $<$  is a projection of  $<$  via sum, min, max, and multiplication with a positive coefficient.
2. The decreasing order  $>$  is a projection of  $<$  via a multiplication with a negative coefficient.
3. Given an element constraint of the form  $y = f(x)$  that maps each possible value  $i$  of  $x$  to a value  $f(i)$ , the following order  $\prec_x$  is a projection of  $<$  to  $x$  via  $f(x)$ :

$$u \prec_x v \quad \text{iff} \quad f(u) < f(v) \quad (7)$$

In our vacation example, the price, quality, and distance of a destination are all defined by element constraints. If we change the objective, we project preferences over a different element constraint. Since the projected preferences depend on the values  $f$  (e.g. price, quality), changing the objective will completely change the order of the value assignments  $x_i = u$ . Thus, the objective will have a strong impact on the search heuristics. On the one hand, the first solution detected is influenced by the projected preferences and thus may change when the objective is changed. On the other hand, search effort is reduced depending on the objective.

## Conclusion

Although Preference-based Search (Junker 2000) provided an interesting technique for reducing search effort based on preferences, it could only take into account preferences between search decisions, was limited to combinatorial problems of a special structure, and did not provide any method for finding compromises in absence of preferences. In this paper, we have lifted PBS from preferences on decisions to preferences on criteria as they are common in qualitative decision theory (Doyle & Thomason 1999; Bacchus & Grove 1995; Boutilier *et al.* 1997; Domshlak, Brafman, & Shimony 2001). We further generalized PBS such that not only extreme solutions are computed, but also balanced and Pareto-optimal solutions. Surprisingly, balanced solutions can be computed by a modified lexicographic approach (Ehrgott 1997) which fits well into a qualitative preference framework as studied in nonmonotonic reasoning and qualitative decision theory.

Our search procedure consists of two modules. A master-PBS explores the criteria in different orders and assigns optimal values to them. The optimal value of a selected criterion is determined by a sub-PBS, which performs a constraint-based Branch-and-Bound search through the original problem space (i.e. the different value assignments to decision variables). We furthermore project the preferences on the selected criterion to preferences between the search decisions, which provides an adapted search heuristics for the optimization objective and which allows to reduce search effort further. Hence, different regions of the search space will be explored depending on the selected objective. Our approach has been implemented in ILOG JCONFIGURATOR V2.0 and adds multi-criteria optimization functionalities to this constraint-based configuration tool.

Other CSP-based approaches to multi-criteria optimization are doing a single Branch-and-Bound search for all criteria, which requires to maintain a set of non-dominated solutions (cf. (Boutilier *et al.* 1997), (Gavanelli 2002)) instead of a single bound. Dominance checking ensures that non-preferred solutions are pruned. Interestingly, the Master-PBS does not need dominance checking, but uses refutation queries to avoid non-preferred solutions.

Future work will be devoted to improve the pruning behaviour of the new PBS procedures by incorporating the conflict checking methods of (Junker 2000). We will also examine whether PBS can be used to determine preferred solutions as defined by soft constraints (Khatib *et al.* 2001).

## Acknowledgements

For helpful comments and discussions, I would like to thank Olivier Lhomme, Xavier Ceugniet, Daniel Mailharro, Mark Wallace, as well as the anonymous reviewers.

## References

- Bacchus, F., and Grove, A. 1995. Graphical models for preference and utility. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, 3–10.
- Boutilier, C.; Brafman, R.; Geib, C.; and Poole, D. 1997. A constraint-based approach to preference elicitation and decision making. In Doyle, J., and Thomason, R. H., eds., *Working Papers of the AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning*.
- Brewka, G. 1989. Preferred subtheories: An extended logical framework for default reasoning. In *IJCAI-89*, 1043–1048.
- Domshlak, C.; Brafman, R.; and Shimony, E. 2001. Preference-based configuration of web-page content. In *IJCAI-2001*.
- Doyle, J., and Thomason, R. H. 1999. Background to qualitative decision theory. *AI Magazine* 20(2):55–68.
- Ehrgott, M. 1997. A characterization of lexicographic max-ordering solutions. In *Methods of Multicriteria Decision Theory: Proceedings of the 6th Workshop of the DGOR Working-Group Multicriteria Optimization and Decision Theory*, 193–202. Egelsbach: Häsel-Hohenhausen.
- Gavanelli, M. 2002. An implementation of Pareto optimality in CLP(FD). In *CP-AI-OR'02*.
- Geffner, H., and Pearl, J. 1992. Conditional entailment: Bridging two approaches to default reasoning. *Artificial Intelligence* 53:209–244.
- Grosz, B. 1991. Generalizing prioritization. In *KR'91*, 289–300. Cambridge, MA: Morgan Kaufmann.
- Junker, U. 1997. A cumulative-model semantics for dynamic preferences on assumptions. In *IJCAI-97*, 162–167.
- Junker, U. 2000. Preference-based search for scheduling. In *AAAI-2000*, 904–909.
- Khatib, L.; Morris, P.; Morris, R. A.; and Rossi, F. 2001. Temporal constraint reasoning with preferences. In *IJCAI*, 322–327.