Annals of Operations Research 130, 75–115, 2004

Preference-Based Search and Multi-Criteria Optimization *

ULRICH JUNKER ujunker@ilog.fr
ILOG, 1681, route des Dolines, F-06560 Valbonne, France

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 global criteria such as the total price or quality of a configuration. We generalize PBS to compute balanced, extreme, and Pareto-optimal solutions for general CSPs, thus handling preferences on and between multiple criteria. A master-PBS selects criteria based on trade-offs and preferences and passes them as an 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 heuristic and to reduce search effort, thus giving the criterion a high impact on the search. The resulting method will be particularly effective for CSPs with large domains that arise if configuration catalogues 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 catalogue. 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 have 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, 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; Delgrande and Schaub, 2000)

* Previous versions of this article have been presented at CP-AI-OR'02 and AAAI-02.

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 us 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 of 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 global criterion highly constrains the remaining search problem and usually leads to 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. For example, 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 (MCO) avoids those problems. Operations research provides different methods for solving a multi-criteria optimization problem. For example, the problem can be mapped to a single or to a sequence of single-criterion optimization problems which are then solved by traditional methods. Furthermore, there are several notions of optimality such as Pareto-optimality, lexicographic optimality, and lexicographic max-order optimality. A recent overview of this large research field of multi-criteria optimization can be found in (Ehrgott and Gandibleux, 2000). Based on these methods, we can thus determine ‘extreme solutions’, where some criteria is favoured over other criteria, as well as ‘balanced solutions’, where the different criteria are as close together as possible and which represent compromises. This balancing requires that the different criteria are comparable, which is usually achieved by a standardization method. The balancing is not achieved by weighted sums of the different criteria, but by a new lexicographic approach that has been studied by different authors (cf. Behringer, 1981; Ehrgott, 1997). According to this approach, we have to proceed as follows in order to find a compromise between a good price and a good quality: we first minimize the maximum between (standardized versions of) price and quality, fix one of the criteria (e.g., the standardized quality) at the resulting minimum, and then minimize 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 this new version 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 the 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 would still 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 global criteria, but it does not translate them to preferences between search decisions. In general, it is not evident how to derive a search heuristic automatically from the selected optimization objective. Adequate preferences between search decisions provide such a heuristic and also allow a preference-based search to be applied 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 nonmonotonic reasoning (NMR). There have been two major approaches to treat a strict partial order between default rules. Geffner and Pearl (1992) and Grosof (1991) lift this order to a partial order among solutions and consider the solutions that are the most preferred ones with respect to this order. Brewka (1989) chooses a linearization of the partial order and then compares the solutions lexicographically by using the chosen linearization as base order. Each linearization leads to a single preferred solution. Different preferred solutions can be obtained by choosing different linearizations. In (Junker, 1997), we showed that each preferred solution in the sense of Brewka (B-preferred solution) corresponds to a preferred solution in the sense of Geffner and Grosof (G-preferred solution). In this paper, we adapt these definitions to multi-criteria optimization. Instead of a strict partial order between default rules, we introduce two kinds of preferences:

1. *Preferences on criteria*: for each criterion, we consider a strict partial order between its possible values and we seek a best value w.r.t. this order.

2. *Preferences between criteria*: if a criterion z_1 is more important than z_2, then any value assignment to z_1 is more important than any value assignment to z_2.

If no preferences between criteria are given, the Pareto-optimal solutions correspond to the G-preferred solutions and the lexicographic-optimal solutions correspond to the B-preferred solutions. Preferences between criteria can easily be taken into account by the latter methods. For balanced solutions, we present a variant of Ehrgott's definition

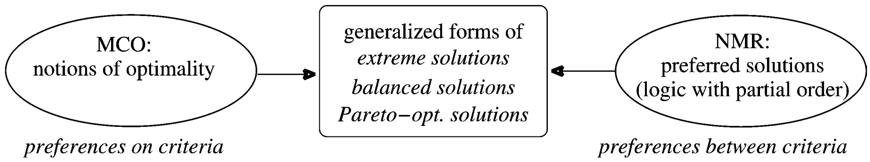

Figure 1. Merging concepts from MCO and NMR.

that additionally respects preferences between criteria and we define preferred solutions in the style of Ehrgott (E-preferred solutions). Thus, we merge concepts from MCO and NMR as illustrated in figure 1. Since the preference-based search method is dedicated to B-preferred solutions, we develop suitable translations of a multi-criteria optimization problem such that the G- and E-preferred solutions of the original problem correspond to the B-preferred solutions of the translations. 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.

The preference-based search method explores the given criteria in different orders that are compatible with the preferences between criteria. When preference-based search selects a criterion, it solves a minimization subproblem to determine the best value for this criterion. Once this value has been found, preference-based search tries out two different possibilities: either it assigns the best value to the selected criterion or it tries to refute this assignment by optimizing other criteria first. Thus, preference-based search sets up a sequence of minimization subproblems with changing objectives. These subproblems can be solved by different methods, for example constraint-based Branch-and-Bound. This method imposes an upper-bound constraint on the objective. Although the upper bound is reduced each time a solution is found, the objective has quite a weak impact on the search space. In particular, the first solution does not depend at all on the objective since no upper bound is given yet.

We can improve the search behaviour by projecting the preferences of the selected criterion to the search decisions. For example, if we want to minimize the price of our trip we will choose cheaper hotels first. We will introduce a general method for preference projection, which we then apply to normal 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. 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. Furthermore, the projected preferences preserve Pareto-optimality and we can reduce search effort by limiting search to the Pareto-optimal solutions that are defined by the projected preferences. This can be achieved by a suitable adaption of preference-based search to the subproblem as indicated in figure 2.

The paper is organized as follows: we first introduce different notions of optimality from multi-criteria optimization (section 1) and then extend them to cover preferences between criteria (section 2). We then show how these preferences can be formulated in a general preference programming framework (section 3). After this, we develop

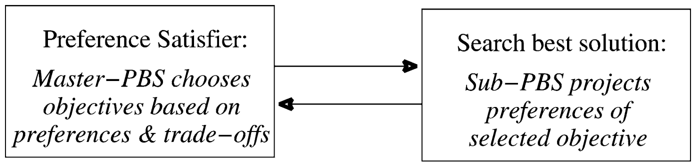

Figure 2. Multiple subsearches driven by Master-PBS.

new versions of preference-based search for computing the different kinds of preferred solutions (section 4). Finally, we introduce preference projection (section 5). The paper supposes some basic background in optimization as well as constraint programming.

1. Preferences on criteria

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

Throughout this paper, we consider combinatorial problems that have the decision variables $\mathcal{X} := (x_1, \ldots, x_m)$, the criteria $\mathcal{Z} := (z_1, \ldots, z_n)$, and the constraints $\mathcal{C}$. We suppose that each decision variable x_i has a fixed domain $D(x_i)$ that is finite and that specifies the possible values for x_i. In our vacation adviser problem, x_i represents the accommodation of the ith vacation day. The constraints in $\mathcal{C}$ have the form $C(x_1, \ldots, x_m)$. 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 stops should be neighbouring cities) 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 the form of a functional constraint $z_i := f_i(x_1, \ldots, x_m)$ and a finite 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 the minimum of the individual qualities and the total distance is the maximum of the individual distances. The prices, qualities, and destinations of the individual accommodations are given by tables such as the catalogue in table 1.

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

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 $<$ for price and distance and $>$ for quality. We write $u \preceq v$ iff $u \prec v$ or $u = v$.

Often, we choose preferences on criteria that satisfy specific properties. Table 2 specifies the properties of strict partial orders, ranked orders, and strict total orders.

Table 1
Catalogue of a fictive hotel chain.

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
...				

Table 2
Properties of strict orders.

Strict partial order: *binary relation* $\prec$ s.t.	
$u \prec v$ implies $u \neq v$	(irreflexivity)
$u \prec v, v \prec w$ implies $u \prec w$	(transitivity)
Ranked order: *strict partial order* $\prec$ s.t.	
$u \prec v,\ v \equiv w$ implies $u \prec w$	
$u \equiv v,\ v \prec w$ implies $u \prec w$	
Strict total order: *strict partial order* $\prec$ s.t.	
$v \prec w$ or $v = w$ or $w \prec v$	

There is a strict hierarchy between these notions: each strict total order is a ranked order and each ranked order is a strict partial order. Ranked orders ensure that incomparable elements can replace each other in comparisons of the form $u \prec v$. Two elements v and w are incomparable, i.e. $v \equiv w$, iff neither $v \prec w$, nor $w \prec v$ is true. Due to this, we can say that a ranked order puts incomparable elements in a layer of same priority. If $\prec$ is a ranked order then there exists a (unique) function $\text{rank}_\prec$ that maps the values to ordinals $1, 2, \ldots, k$ such that the following correspondence holds

$$v \prec w \quad \text{iff} \quad \text{rank}_\prec(v) < \text{rank}_\prec(w) \tag{1}$$

and the largest rank k is as small as possible.

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 optimal in the following sense. If another solution S^* is better than S w.r.t. a criterion z_k then S is better than S^* for some other criterion z_j:

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 $(\mathcal{C}, \mathcal{X})$ such that the following conditions hold:

1. $v_{S^*}(z_k) \prec_{z_k} v_S(z_k)$ for a k, and
2. $v_{S^*}(z_i) \preceq_{z_i} v_S(z_i)$ for all i.

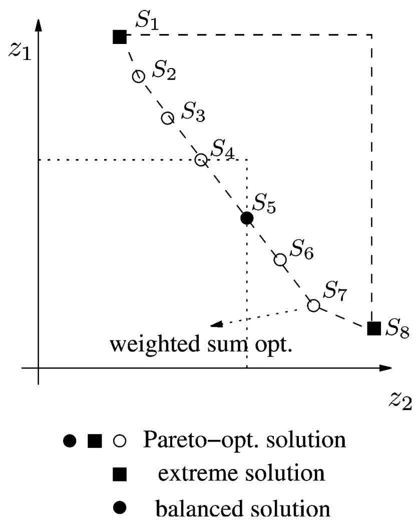

Figure 3. Pareto-optimal solutions.

Pareto-optimal solutions narrow down the solution space since non-Pareto-optimal solutions do not appear to be acceptable. However, their number is usually too large to enumerate them all. Figure 3 shows the Pareto-optimal solutions S_1 to S_8 for the two criteria z_1 and z_2 that need to be minimized. Other solutions are contained in the area that is surrounded by the dashed line.

A lexicographic solution is based on a ranking of the different criteria. We express such a ranking by a permutation π of the positions $1, \ldots, n$. We use π_i for the index of the criterion at the ith position:

$$\pi(\mathcal{Z}) := (z_{\pi_1}, \ldots, z_{\pi_n}). \tag{2}$$

Let $V_S(\pi(\mathcal{Z}))$ be the tuple of values of these criteria:

$$V_S\big(\pi(\mathcal{Z})\big) := \big(v_S(z_{\pi_1}), \ldots, v_S(z_{\pi_n})\big). \tag{3}$$

As an example, consider three criteria z_1, z_2, z_3. We obtain six permutations:

	π_1	π_2	π_3	$\pi(\mathcal{Z})$	$V_S(\pi(\mathcal{Z}))$
1.	1	2	3	z_1, z_2, z_3	$(v_S(z_1), v_S(z_2), v_S(z_3))$
2.	1	3	2	z_1, z_3, z_2	$(v_S(z_1), v_S(z_3), v_S(z_2))$
3.	2	1	3	z_2, z_1, z_3	$(v_S(z_2), v_S(z_1), v_S(z_3))$
4.	2	3	1	z_2, z_3, z_1	$(v_S(z_2), v_S(z_3), v_S(z_1))$
5.	3	1	2	z_3, z_1, z_2	$(v_S(z_3), v_S(z_1), v_S(z_2))$
6.	3	2	1	z_3, z_2, z_1	$(v_S(z_3), v_S(z_2), v_S(z_1))$

Given a permutation π, we compare two solutions S_1 and S_2 by a lexicographic order $\prec_{\text{lex}}$, which uses $\pi(\mathcal{Z})$ as ranking of the criteria. The values at position i are compared w.r.t. the preferences on the criterion $\prec_{z_{\pi_i}}$ at position i. Let $V_{S_1}(\pi(\mathcal{Z})) := (v_1, \ldots, v_n)$ and $V_{S_2}(\pi(\mathcal{Z})) := (w_1, \ldots, w_n)$, We define

$$\begin{aligned}&(v_1, \ldots, v_n) \prec_{\text{lex}}^{\pi} (w_1, \ldots, w_n) \quad \text{iff}\\ &\exists k\colon\ v_k \prec_{z_{\pi_k}} w_k \text{ and } v_i = w_i \text{ for all } i = 1, \ldots, k-1.\end{aligned} \tag{4}$$

Definition 2. Let π be a permutation of $1, \ldots, n$. 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. $V_{S^*}(\pi(\mathcal{Z})) \prec_{\text{lex}}^{\pi} V_S(\pi(\mathcal{Z}))$.

Different rankings lead to different extreme[1] solutions which are all Pareto-optimal. In figure 3, 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 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 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 criteria z_i have a common domain D and that the preference orders $\prec_{z_i}$ of the different criteria are equal to a strict total order $<_D$. This usually requires some scaling or standardization of the different criteria. We also introduce the reverse order $>_D$ which satisfies $z_i >_D z_j$ iff $z_j <_D z_i$. 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 $>_D$. The sorted tuples are then compared by a lexicographic order $\prec_{\text{lex}}$. It is important to note that this sorting can lead to different permutations of the criteria if different solutions are considered. We describe the sorting by a permutation ρ^S that depends on a given solution S and that satisfies two conditions:

1. ρ^S sorts the criteria in a decreasing order: if $v_S(z_{\rho_i^S}) >_D v_S(z_{\rho_j^S})$ then $i < j$.
2. ρ^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 $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, <_D)$ iff there is no other solution S^* of $(\mathcal{C}, \mathcal{X})$ s.t. $V_{S^*}(\rho^{S^*}(\mathcal{Z})) \prec_{\text{lex}} V_S(\rho^S(\mathcal{Z}))$.

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 3, we obtain S_5 as balanced solution. According to Ehrgott, it can be determined as follows: first $\max(z_1, z_2)$ is minimized, i.e. $\max(z_1, z_2)$ is used as objective[2] of the constraint satisfaction problem $(\mathcal{C}, \mathcal{X})$. 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.

2. 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 the form of a strict partial order $\prec_{\mathcal{Z}} \subseteq \mathcal{Z} \times \mathcal{Z}$. These preferences express a notion of relative importance and it is natural to require that this notion is transitive and irreflexive.

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 the following two conditions: 1. If $u \prec_{z_i} v$ then $(z_i = u) \prec (z_i = v)$ 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 (Grosof, 1991; Geffner and Pearl, 1992) follow the first approach, whereas the second approach leads to the B-preferred solutions of (Brewka, 1989; Junker, 1997). We will now adapt the definitions in (Junker, 1997) to the specific preference structure of this paper.

2.1. *Generalizing extreme 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. The following definition has been adapted from (Brewka, 1989; Junker, 1997) to our specific preference structure:

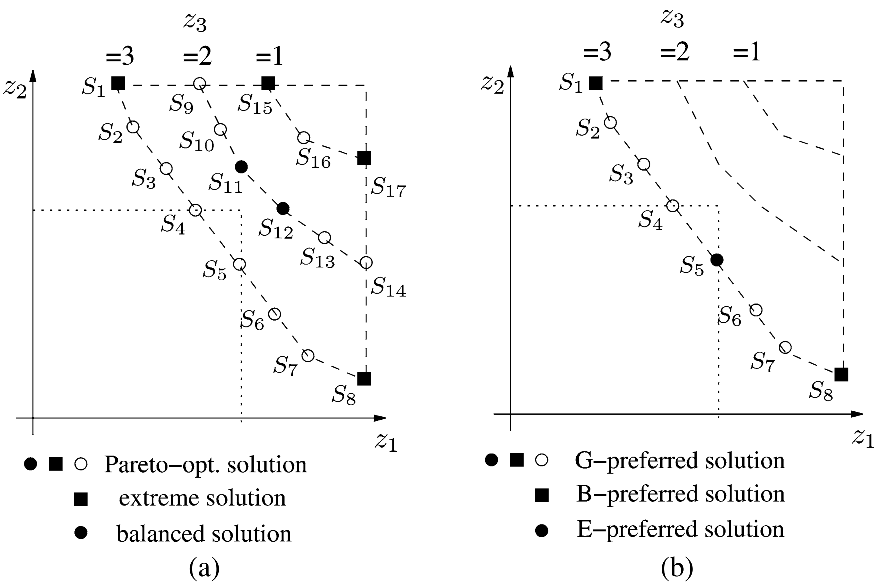

Figure 4. Preferred solutions.

Definition 4. 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 π such that (1) π respects $\prec_{\mathcal{Z}}$ (i.e. $z_{\pi_i} \prec_{\mathcal{Z}} z_{\pi_j}$ implies $i < j$) and (2) there is no other solution S^* of $(\mathcal{C}, \mathcal{X})$ satisfying $V_{S^*}(\pi(\mathcal{Z})) \prec^{\pi}_{\text{lex}} V_S(\pi(\mathcal{Z}))$.

The B-preferred solution for π can be computed by solving a sequence of minimization problems if all $\prec_{z_i}$ are ranked orders. Let $\mathcal{C}_0 := \mathcal{C}$ and

$$\mathcal{C}_i := \mathcal{C}_{i-1} \cup \{z_{\pi_i} = m\},$$

where

$$m = \min_{\prec_{z_{\pi_i}}} \{v_S(z_{\pi_i}) \mid S \text{ is a solution of } (\mathcal{C}_{i-1}, \mathcal{X})\}.$$

Each solution of the resulting set $\mathcal{C}_n$ is a B-preferred solution and each B-preferred solution is a solution of a set $\mathcal{C}_n$ of some permutation π. Figure 4 shows different kinds of preferred solutions for three criteria z_1, z_2, z_3 that are all minimized and that respect the preferences $z_1 \prec_{\mathcal{Z}} z_3$ and $z_2 \prec_{\mathcal{Z}} z_3$. The B-preferred solutions are S_1, S_8 (cf. figure 4(b)). 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, in figure 4(a), S_{15} is an extreme solution, which is obtained if first the distance z_3 is minimized and then the price z_1. However, this ranking of the criteria does not respect the given preferences.

2.2. Generalizing Pareto-optimal solutions

Adapting the G-preferred solutions of (Junker, 1997) to the specific preference structure yields the following definition.

Definition 5. 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 all i with $v_S(z_i) \neq v_{S^*}(z_i)$ satisfy at least one of the following conditions: (1) $v_{S^*}(z_i) \prec_{z_i} v_S(z_i)$ or (2) there exists a j s.t. $z_j \prec_{\mathcal{Z}} z_i$ and $v_{S^*}(z_j) \neq v_S(z_j)$.

A G-preferred solution S is optimal in the following sense. If another solution S^* is better than S w.r.t. a criterion z_i then there exists a more important criterion z_j such that S and S^* differ on z_j. Then either S is better than S^* on z_j or another criterion z_k exists, such that S and S^* differ on z_k. We cannot repeat this argumentation an infinite number of times since $\prec_{\mathcal{Z}}$ does not have infinite descending chains due to the finiteness of $\mathcal{Z}$. Hence, we finally end up with a criterion z_{k^*} that is more important than z_i and for which S is better than S^*. Hence, a criterion of a G-preferred solution can become worse if a more important criterion is improved. In figure 4(b), 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.

Proposition 1. Let $\mathcal{P}$ be $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$. If S is a G-preferred solution of $\mathcal{P}$ then S is a Pareto-optimal solution of $\mathcal{P}$. If S is a Pareto-optimal solution of $\mathcal{P}$ and $\prec_{\mathcal{Z}} = \emptyset$ then S is a G-preferred solution of $\mathcal{P}$.

However, if there are preferences between criteria, certain Pareto-optimal solutions S are not G-preferred. There can be a Pareto-optimal solution S that is better than a G-preferred solution S^* for a criterion z_i, but worse for a more important criterion z_j (i.e. $z_i \prec_{\mathcal{Z}} z_j$). In this case, the G-preferred solution S^* is preferred to S meaning that S is not G-preferred. In figure 4(a), S_9 to S_{17} are Pareto-optimal, but not G-preferred. For example, S_9 is not G-preferred since S_1 and S_9 differ on distance and price. S_1 has a better price than S_9 and thus improves S_9 w.r.t. this criterion. The fact that S_9 has a better distance than S_1 is compensated by the fact that S_1 and S_9 differ on a criterion that is more important than the distance, namely the price.

In general, we may get new G-preferred solutions if we add new constraints to our problem. However, adding upper bounds to best criteria does not add new G-preferred solutions. We say z is a $\prec_{\mathcal{Z}}$-best criterion iff there is no z^* s.t. $z^* \prec_{\mathcal{Z}} z$:

Proposition 2. Let z be a $\prec_{\mathcal{Z}}$-best criterion. S is a G-preferred solution of $(\mathcal{C} \cup \{z \preceq_z u\}, \mathcal{X}, \mathcal{Z}, \prec)$ iff S is a G-preferred solution of $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$ and $v_S(z) \preceq_z u$.

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

Furthermore, we can eliminate a best criterion from a problem by assigning a best value to this criterion:

Proposition 3. Let z be a $\prec_{\mathcal{Z}}$-best criterion and v be a $\prec_z$-best value for z. S is a G-preferred solution of $(\mathcal{C} \cup \{z = v\}, \mathcal{X}, \mathcal{Z}, \prec)$ iff S is a G-preferred solution of $(\mathcal{C}, \mathcal{X}, \mathcal{Z} - \{z\}, \prec)$ and $v_S(z) = v$.

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.

Proposition 4. Let $\mathcal{P}$ be $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$. Each B-preferred solution of $\mathcal{P}$ is also a G-preferred solution of $\mathcal{P}$.

In figure 4(b), S_2 to S_7 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 the following case, where no compromises are possible:

Proposition 5. Let $\mathcal{P}$ be $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$. If $\prec_{\mathcal{Z}}$ is a ranked order and there are no three solutions S_1, S_2, S_3 of $(\mathcal{C}, \mathcal{X})$ such that $v_{s_1}(z) \prec_z v_{s_2}(z)$ and $v_{s_3}(z) \not\preceq_z v_{s_2}(z)$ for a criterion z then each G-preferred solution of $\mathcal{P}$ is also a B-preferred solution of $\mathcal{P}$.

In general, this equivalence does not hold. However, propositions 2 and 5 point out a possibility for mapping G-preferred solutions to B-preferred solutions if the order $\prec_{\mathcal{Z}}$ is ranked. The basic idea is to replace the original criteria by binary criteria that are satisfied if the original criteria are smaller or equal to an upper bound.

Let z be a criterion in $\mathcal{Z}$ and let v be a possible value for z. We introduce a binary criterion $u_{z,v}$ which is equal to 1 if and only if the criterion z has a value smaller or equal to v:

$$u_{z,v} := \begin{cases} 1 & \text{if } z \preceq_z v, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Let $\mathcal{U}$ be the set of these upper-bound criteria. It is important to note that the value of a binary criterion $u_{z,v}$ in a solution S is entirely determined by the value of the criterion z in S. Hence, if we know the values of the criteria in a G-preferred solution, we can determine the values of the binary criteria. Vice versa, if we know the values of the binary criteria, we can determine the values of the original criteria: $v_S(z)$ is equal to the $\prec_z$-smallest value w such that $v_S(u_{z,w})$ is equal to one. We can thus replace the set of original criteria $\mathcal{Z}$ by $\mathcal{U}$ in the translation of the original problem.

We maximize each $u_{z,v}$, which is expressed by the following preferences on the binary values:

$$1 \prec'_{u_{z,v}} 0. \tag{6}$$

The preferences between the criteria $\mathcal{Z}$ are mapped to corresponding preferences between the criteria $\mathcal{U}$. Given two criteria z_i and z_j the following correspondence holds for all upper bounds u_i and u_j:

$$z_i \prec_{\mathcal{Z}} z_j \quad \text{implies} \quad u_{z_i,v_i} \prec'_{\mathcal{U}} u_{z_j,v_j}. \tag{7}$$

If the order $\prec_{\mathcal{Z}}$ is ranked then $\prec'_{\mathcal{U}}$ is ranked as well. We then call the resulting problem $(\mathcal{C}, \mathcal{X}, \mathcal{U}, \prec')$ the *bound-translation* of $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$.

The G-preferred solutions are preserved by this translation.

Proposition 6. Let $\mathcal{P}$ be $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$. S is a G-preferred solution of $\mathcal{P}$ iff S is a G-preferred solution of the bound-translation of $\mathcal{P}$.

The bound-translation matches the conditions of proposition 5 since all criteria in $\mathcal{U}$ are binary and since we suppose that $\prec_{\mathcal{Z}}$ is ranked. Hence, the G-preferred solutions correspond to the B-preferred solutions of the bound-translation:

Theorem 1. Let $\mathcal{P}$ be $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$ and $\prec_{\mathcal{Z}}$ be a ranked order. S is a G-preferred solution of $\mathcal{P}$ iff S is a B-preferred solution of the bound-translation of $\mathcal{P}$.

If the order $\prec_{\mathcal{Z}}$ is not ranked, then the bound-translation is not sufficient to establish a correspondence between B- and G-preferred solution. Future work is needed to address this more general case.

2.3. Generalizing balanced solutions

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 solution 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 mutually disjoint sets $G_1, \ldots, G_k$ 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_is, we can easily define an order $\prec_{\mathcal{Z}}$ between criteria: if $G_1 \prec_G G_2$ and $z_i \in G_1$, $z_j \in G_2$ then $z_i \prec_{\mathcal{Z}} z_j$. Hence, the preferences between criteria are easy to acquire. If we want to balance several criteria we put them into the same group. If there are several groups we will determine a balancing for one group after another. Preferences between groups constrain the possible orderings of the groups. The most natural case is obtained if the groups are totally ordered. Otherwise, multiple orders of the groups can be considered.

We now combine definitions 4 and 3. Again, we assume that the preference orders $\prec_{z_i}$ of the different criteria are equal to a strict total order $<_D$. As in definition 4, we first choose a global permutation π 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 θ^S that depends on a given solution S and that satisfies three conditions:

1. θ^S can only exchange variables that belong to the same balanced group: $G(z_i) = G(z_{\theta_i^S})$.
2. θ^S sorts the criteria of each group in a decreasing order: if $v_S(z_{\theta_i^S}) >_D v_S(z_{\theta_j^S})$ and $G(z_{\theta_i^S}) = G(z_{\theta_j^S})$ then $i < j$.
3. θ^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 π such that (1) π respects $\prec_{\mathcal{Z}}$ (i.e. $z_{\pi_i} \prec_{\mathcal{Z}} z_{\pi_j}$ implies $i < j$) and (2) there is no other solution S^* of $(\mathcal{C}, \mathcal{X})$ s.t. $V_{S^*}(\theta^{S^*}(\pi(\mathcal{Z}))) \prec_{\text{lex}} V_S(\theta^S(\pi(\mathcal{Z})))$.

The concatenation of the two permutations π and θ^{S^*} deserves a short explanation. Firstly, we apply π in order to determine an ordering $z_{\pi_1}, \ldots, z_{\pi_n}$ of the criteria that respects the preferences between criteria. Let us say that this ordering is equal to $p_1, \ldots, p_n$, i.e. let p_i be equal to z_{π_i}. Secondly, we apply θ^{S^*} to $p_1, \ldots, p_n$ resulting in the ordering $p_{\theta_1^{S*}}, \ldots, p_{\theta_n^{S*}}$. It is now easy to see that $V_S(\theta^S(\pi(\mathcal{Z})))$ is the tuple $(v_S(z_{\pi_{\theta_1^S}}), \ldots, v_S(z_{\pi_{\theta_n^S}}))$.

In figure 4(b), S_{11} and S_{12} are balanced solutions w.r.t. the standardized versions of price, quality, and distance. This notion does not take into account that the distance is less important than price and quality ($z_1 \prec_{\mathcal{Z}} z_3$ and $z_2 \prec_{\mathcal{Z}} z_3$). If we determine the E-preferred solutions, we consider two groups. The more important group contains the price and quality, whereas the second group contains the distance. In order to obtain the E-preferred solution S_5, we first compute a balanced solution of group 1 and then minimize the single criterion of group 2. In this case, neither the balanced solutions are E-preferred, nor the E-preferred solutions are balanced. However, if there is only a single group, E-preferred solutions coincide with balanced solutions.

Interestingly, we can map E-preferred solutions to B-preferred solutions if we introduce suitable variables and preferences. We explain the idea for three criteria $\mathcal{Z}_3 := \{z_1, z_2, z_3\}$ and we suppose that the common strict total order $<_D$ is the increasing order on integers, meaning that all three criteria are minimized. The first step consists in minimizing the criterion that has the worst value in a solution. For this purpose, we introduce a new criterion

$$\hat{y}_3 := \max(z_1, z_2, z_3) \tag{8}$$

by using a max-expression.[3] Given the best value v_3 for $\hat{y}_3$, we then know that at least one criterion z_k has the value v_3 in a preferred solution S and that the other criteria in $\mathcal{Z}_2 := \{z_1, z_2, z_3\} - \{z_k\}$ have the same or a better value. We can further compare the remaining solutions by comparing the values of the criteria in $\mathcal{Z}_2$ after setting $\hat{y}_3$ to v_3. We therefore minimize the maximum of the criteria in $\mathcal{Z}_2$. We can do this although we do not know z_k and the elements of $\mathcal{Z}_2$. The trick is to consider all possibilities for $\mathcal{Z}_2$, namely $\{z_1, z_2\}$, $\{z_1, z_3\}$, and $\{z_2, z_3\}$. We determine the maximum of each of these combinations, namely $\max(z_1, z_2)$, $\max(z_1, z_3)$, $\max(z_2, z_3)$, and distinguish two cases:

1. If z_k is in a set $\{z_i, z_j\}$ of two criteria then $\max(z_i, z_j)$ has the value v_3 in a solution.
2. If z_k is not a set $\{z_i, z_j\}$ of two criteria then this set is equal to $\mathcal{Z}_2$ and $\max(z_i, z_j)$ corresponds to our objective, which has a value smaller or equal to v_3.

We can thus prove that minimizing

$$\hat{y}_2 := \min\bigl(\max(z_1, z_2), \max(z_1, z_3), \max(z_2, z_3)\bigr) \tag{9}$$

determines the best value v_2 of the second worst criterion in a solution. After assigning v_2 to $\hat{y}_2$, we minimize

$$\hat{y}_1 := \min(z_1, z_2, z_3) \tag{10}$$

in order to find the best value v_1 for the third worst criterion (i.e. the best criterion in this example).

We now discuss the general case. For each group G of cardinality n_G, we use the following min-max-variables $y_{G,n_G}, \ldots, y_{G,1}$. The criterion $y_{G,i}$ is minimized if the best values of all, but i criteria have been found. Since we do not know which of the criteria are remaining we determine the maximum of each subset of size i and take the minimum of these maxima as explained above:

$$y_{G,i} := \min_{<_D}\bigl\{\max_{<_D}(X) \mid X \subseteq G \text{ s.t. } |X| = i\bigr\}, \tag{11}$$

where $\max_{<_D}(X) := \max_{<_D}\{z \mid z \in X\}$. The min-max-variables can directly be expressed in a constraint programming language. Due to the exponential size of the expression, this is only feasible for a small number of criteria. For a large number of criteria, an option for future work is the development of a global constraint. Let $\widehat{\mathcal{Z}} := \{\hat{z}_1, \ldots, \hat{z}_n\}$ be the set of all of these min-max-variables. The $\hat{z}_i$ are arranged in an order that preserves the group of position i: if the criterion z_i belongs to group G then $\hat{z}_i$ also belongs to group G and is equal to $y_{G,j}$ for some j.

We now adapt the preferences $\prec$ to the new criteria and we denote the result by $\hat{\prec}$. The preference order $\hat{\prec}_{y_{G,i}}$ of all criteria $y_{G,i}$ is equal to the strict total order $<_D$. The preferences $\hat{\prec}_{\widehat{\mathcal{Z}}}$ between the criteria $\widehat{\mathcal{Z}}$ satisfy two properties:

1. The following preferences ensure that min-max-variables for larger subsets X are more important:
$$y_{G,i} \hat{\prec}_{\widehat{\mathcal{Z}}}\, y_{G,i-1} \quad \text{for } i = n_G, \ldots, 2. \tag{12}$$
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:
$$y_{G^*,1} \hat{\prec}_{\widehat{\mathcal{Z}}}\, y_{G,n_G}. \tag{13}$$

We call the resulting problem $(\mathcal{C}, \mathcal{X}, \widehat{\mathcal{Z}}, \hat{\prec})$ the *min-max-translation* of $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$.

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

Theorem 2. Let $\mathcal{P}$ be $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$. S is an E-preferred solution of $\mathcal{P}$ iff S is a B-preferred solution of the min-max-translation of $\mathcal{P}$.

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.

3. Preference programming

In the previous section, we defined preferred solutions based on preferences on criteria and between criteria, but we did not discuss how these preferences can be specified. For this purpose, we enhance traditional constraint programming by primitives for stating preferences. We thus obtain a system of preference programming, which is described in (Junker and Mailharro, 2003b) in detail. Preference programming has been implemented in ILOG JCONFIGURATOR 2.0 (ILOG, 2002) and supports different preference-based problem solving tasks such as searching for a solution guided by preferences, finding a preferred explanation, and satisfying user preferences. Furthermore, JCONFIGURATOR combines an expressive constraint language with a description logic for describing the taxonomic and partonomic configuration knowledge (Junker and Mailharro, 2003a).

In this section, we show how preferences on and between criteria can be specified within the preference programming framework. In our approach, preferences constrain the order in which decisions are made. We therefore represent preferences by special kinds of constraints. Preferences between two criteria (e.g., price and distance) can be stated by the following constraint:

```
prefer(price, distance);
```

The strict partial order $\prec_{\mathcal{Z}}$ is then defined as the transitive closure of the set of all tuples (z_i, z_j) for which `prefer`(z_i, z_j) is given. If the transitive closure is not irreflexive then the preference statements are considered inconsistent.

Domain orders $\prec_{z_i}$ can be specified in a compact way. We represent the increasing order on integers by `minFirst` and the decreasing order on integers by `maxFirst`. Ranked orders can be expressed by assigning a priority to each value and strict partial orders can be formulated by `prefer`-statements between values (Junker and Mailharro, 2003b). A domain order is applied to a criterion by a `preferValues`-constraint:

```
preferValues(distance, minFirst());
preferValues(price, minFirst());
preferValues(quality, maxFirst());
```

In certain cases, preferences between criteria can be structured by grouping different criteria together and by stating preferences between these groups. For example, we can introduce a group containing the criteria price and quality and then prefer this group to the criteria distance:

```
group(g1);
contains(g1, price);
contains(g1, quality);
prefer(g1, distance);
```

If a group g_1 is preferred to a group g_2 then all elements of g_1 are preferred to all elements of g_2.

Since the preference programming layer is built on top of a constraint programming system, it is straightforward to achieve balancing of several criteria by the translation of theorem 2. As example, suppose that we want to balance minimization of price and maximization of quality and that this balancing is more important than minimizing the distance. We first introduce a group called `balance` and state that its elements are more important to the distance, which has to be minimized:

```
group(balance);
prefer(balance, distance);
preferValues(distance, minFirst());
```

Next, we describe the contents of the balancing group. First, we need to bring price and quality to the same scale since we cannot balance criteria with different domain orders. For this purpose, we introduce a common scale defined by the ordinals $1, 2, \ldots, d$ and we use the increasing order as preference order on this scale. Let us suppose that the domain of the price is $[0, 300]$ and the domain of the quality is $[1, 6]$. For the sake of simplicity, we provide a linear mapping of the criteria to ordinals.[4] We choose $d = 6$. We divide the price by 60 and add 1, thus obtaining a mapping from the price domain $[0, 300]$ to the ordinals $1, \ldots, 6$. Furthermore, we subtract the quality from $d + 1$ since we need to minimize the standardized criteria:

```
scaledPrice = price / 60 + 1;
scaledQuality = d + 1 - quality;
```

Next we introduce the translated criteria as specified in (11). Since only two criteria are involved the min-max-expressions of (11) can be simplified:

```
criterion1 = max(scaledPrice, scaledQuality);
criterion2 = min(scaledPrice, scaledQuality);
```

Finally, we add the new criteria to the group `balance` and specify their domain orders:

```
contains(balance, criterion1);
contains(balance, criterion2);
preferValues(criterion1, minFirst());
preferValues(criterion2, minFirst());
prefer(criterion1, criterion2);
```

This example shows how balanced solutions can be determined with JCONFIGURATOR 2.0.

4. Preference-based search

We now adapt the preference-based search (PBS) algorithm from (Junker, 2000) to treat preferences on criteria. PBS was designed as a search algorithm that reduces search effort by focusing on preferred choices. If v is a best value for a variable x, PBS either tries the assignment $x = v$ or tries to refute it by making best assignments for other variables. PBS abandons a best choice only if such a refutation succeeds. Otherwise, it fails.

We could apply the original PBS to multi-criteria optimization. In this case, PBS would first choose the values of the criteria before choosing the values of the decision variables. However, the assignments to criteria are often very constraining and we easily get a thrashing behaviour as long as these assignments are not supported by any solution. A better idea is to directly determine the best value of a criterion z by solving a minimization subproblem:

$$\text{minimize}(\mathcal{C}, z, \prec_z) := \min\big\{\text{rank}_{\prec_z}\big(v_S(z)\big) \mid S \text{ is a solution of } (\mathcal{C}, \mathcal{X})\big\}. \tag{14}$$

In order to obtain a traditional minimization problem, we only consider ranked orders $\prec_{z_i}$ throughout this section.

The resulting algorithm is called MCPBS[5] and follows the architecture in figure 5. We explain its basic idea for the example shown in figure 6, 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 minimize$(\mathcal{C}, \textit{price}, <)$ as explained above. In our example, the cheapest solution has a price of 133. We now add the assignment $\textit{price} = 133$ to the initial set $\mathcal{C}$ of constraints. In figure 6, these assignments 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, thus obtaining a first solution.

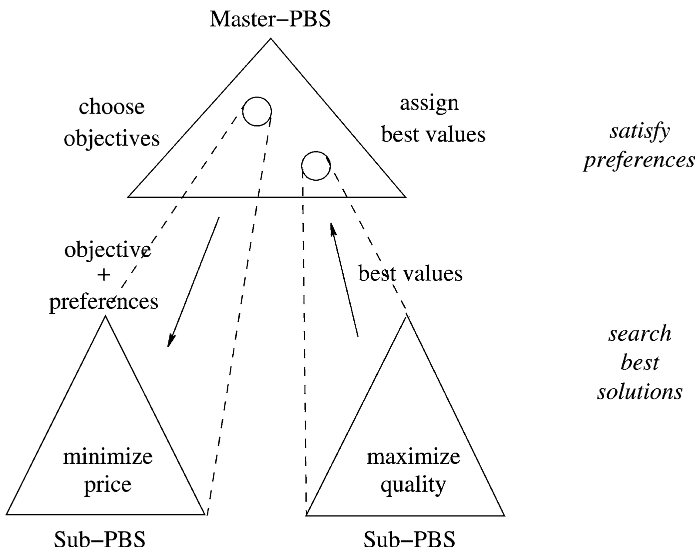

Figure 5. Architecture of PBS-algorithm.

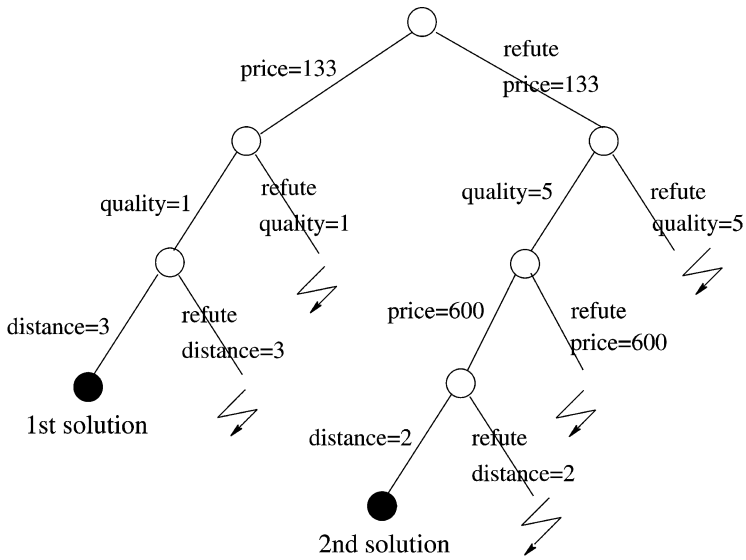

Figure 6. Finding B-preferred solutions.

If a criterion z has several best assignments of same rank r, MCPBS tries them out upon backtracking. If all best assignments have been tried out, MCPBS will search for preferred solutions that assign a worse rank to z. In order to find further preferred solutions, MCPBS introduces a refutation query ϕ of the form $\text{rank}_{\prec_z}(z) = r$. We require that MCPBS determines only those preferred solutions of $\mathcal{C}$ that violate this refutation query ϕ. This is stronger than adding the constraint $\neg\phi$ to $\mathcal{C}$: each preferred solution of

$\mathcal{C}$ that violates ϕ is a preferred solution of $\mathcal{C} \cup \{\neg\phi\}$, but $\mathcal{C} \cup \{\neg\phi\}$ may have preferred solutions, which are not preferred solutions of $\mathcal{C}$ and which must not be determined by McPbs.

We say that a refutation query $\text{rank}_{\prec_z}(z) = r$ is refuted if it becomes inconsistent after assigning values to the 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 after making other assignments.

In our example, 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 133 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 6. This example shows that refutation queries lead to a change of the exploration order of criteria.

The algorithm cannot select a criterion z if it has a refutation query in Q. We denote the set of criteria with refutation queries by

$$Z(Q) := \big\{z \in \mathcal{Z} \mid \exists(\text{rank}_{\prec_z}(z) = q) \in Q\big\}. \tag{15}$$

When adding a new constraint to $\mathcal{C}$, we use a constraint propagation procedure to reduce the possible values for each criterion z. The reduced domain of z is denoted by $\text{dom}_{\mathcal{C}}(z)$. Different constraint propagation algorithms can be used that maintain different degrees of local consistency. A minimal assumption is to maintain node consistency for unary constraints and bound consistency for non-unary constraints. If a criterion has a single possible value in $\text{dom}_{\mathcal{C}}(z)$ then McPbs does not need to try out different assignments for z and can explore it without solving a minimization problem.

The resulting search algorithm is shown in figure 7. For the sake of comprehensibility, we present it as a non-deterministic algorithm that uses try-statements as in OPL (van Hentenryck, Perron, and Puget, 2000) to describe the branching. The statement '**try** A **or** B' performs a simple branching. In the left branch, A is executed, whereas B is executed in the right branch. The statement '**try** A **for each** $v \in V$' creates several branches, namely, one for each value in V.

Furthermore, McPbs uses several subprocedures (see figure 8) that can have different implementations and that may make some global updates of the data structures of McPbs, such as the addition of new constraints. The subprocedure isConsistent checks the consistency of the constraints. The selection of a best criteria is done by the procedure Select. The procedure Minimize determines the best rank for the chosen criterion.

The algorithm maintains a state $\sigma := (\mathcal{C}, Q, U)$ consisting of the set $\mathcal{C}$ of constraints, the set Q of refutation queries, and the set U of unexplored criteria. We say that S is a *B-preferred solution of the state* σ iff S is a B-preferred solution of the problem $\mathcal{P}_\sigma := (\mathcal{C}, \mathcal{X}, U, \prec)$ and S violates each $(\text{rank}_{\prec_z}(z) = q)$ in Q. Given a state σ, we consider the best unexplored criteria

$$B_\sigma := \big\{z \in U \mid \nexists z^* \in U\colon\, z^* \prec_{\mathcal{Z}} z\big\} \tag{16}$$

procedure MCPBS (**in** $\mathcal{C}$: constraints; $\mathcal{X}$: variables; $\mathcal{Z}$: criteria; $\prec$: preferences)
$U := \mathcal{Z}; Q := \emptyset;$
while $U \neq \emptyset$ **do**
  if *not* ISCONSISTENT *($\mathcal{C}$, $\mathcal{X}$)* **then** fail **else**
    $B := \{z \in U \mid \nexists z^* \in U : z^* \prec_{\mathcal{Z}} z\};$
    $B' := B;$
    while $B' \neq \emptyset$ *and* $B \subseteq U$ **do**
      $z :=$ SELECT(B', Q); $B' := B' - \{z\};$
      $m :=$ MINIMIZE$(\mathcal{C}, \mathcal{X}, z, \prec_z);$
      for each $(\mathit{rank}_{\prec_z}(z) = q) \in Q$ **do**
        if $q < m$ **then** $Q := Q - \{\mathit{rank}_{\prec_z}(z) = q\};$
        else if q *is the only rank in* $\mathit{dom}_{\mathcal{C}}(z)$ **then** fail;
      if $(\mathit{rank}_{\prec_z}(z) = m) \notin Q$ **then**
        if $\mathit{dom}_{\mathcal{C}}(z)$ *has a single value* **then** $U := U - \{z\}$ **else**
          $V := \{v \in \mathit{dom}_{\mathcal{C}}(z) \mid \mathit{rank}_{\prec_z}(z) = m\};$
          try
            try $\mathcal{C} := \mathcal{C} \cup \{z = v\}; U := U - \{z\};$
            for each $v \in V$
          or $Q := Q \cup \{\mathit{rank}_{\prec_z}(z) = m\}$
    if $B \subseteq U$ **then** fail;
return $\mathcal{C}$;

Figure 7. Algorithm for computing B-preferred solutions.

procedure ISCONSISTENT (**in** $\mathcal{C}$: constraints; $\mathcal{X}$: variables)
if *there is a support* S^* *and* S^* *is a solution of* $(\mathcal{C}, \mathcal{X})$ **then**
  return true
search a solution S of $(\mathcal{C},\mathcal{X})$;
if *search failed* **then return** false;
set S^* to S;
return true;

procedure SELECT (**in** B: best criteria; Q: refutation queries)
if $B - Z(Q) \neq \emptyset$ **then** select $z \in B - Z(Q)$; return z;
else select $z \in B$; return z;

procedure MINIMIZE (**in** $\mathcal{C}$: constraints; $\mathcal{X}$: variables; z: criterion; $\prec_z$: preferences)
if *there is no support* S_z *for* z *or* S_z *is not a solution of* $(\mathcal{C}, \mathcal{X})$ **then**
  determine a solution S of $(\mathcal{C}, \mathcal{X})$ with smallest $\mathit{rank}_{\prec_z}(z)$;
  let $\mathcal{C} := \mathcal{C} \cup \{\mathit{rank}_{\prec_z}(z) \geq m\};$
  set S_z to S and set S^* to S;
return $v_{S_z}(z)$;

Figure 8. Subprocedures for extreme solutions.

and their best ranks:

$$\text{rank}_\sigma(z) := \min\bigl\{\text{rank}_{\prec_z}\bigl(v_S(z)\bigr) \mid S \text{ is a solution of } \mathcal{C}\bigr\}. \tag{17}$$

If the rank $\text{rank}_{\prec_z}(v)$ of a value v is equal to $\text{rank}_\sigma(z)$ then there is no solution S that assigns a better value to z. This means $v_S(z) \prec_z v$ is not possible in this case.

MCPBS is based on the following properties of a state σ:

- *Success*: If $U = \emptyset$ and $Q = \emptyset$ then each solution of $(\mathcal{C}, \mathcal{X})$ is a B-preferred solution of σ.
- *Refutation query*: Let $z \in U$ and q be any value. S is a B-preferred solution of $(\mathcal{C}, Q \cup \{\text{rank}_{\prec_z}(z) = q\}, U)$ iff S is a B-preferred solution of $(\mathcal{C}, Q, U)$ and S violates $\text{rank}_{\prec_z}(z) = q$.
- *Choice*: Let $z \in B_\sigma$ and v be a value having the rank $\text{rank}_\sigma(z)$. S is a B-preferred solution of $(\mathcal{C} \cup \{z = v\}, Q, U - \{z\})$ iff S is a B-preferred solution of $(\mathcal{C}, Q, U)$ and S satisfies $z = v$.
- *Refutation*: Suppose $(\text{rank}_{\prec_z}(z) = q) \in Q$ and $q < \text{rank}_\sigma(z)$. Then S is a B-preferred solution of $(\mathcal{C}, Q, U)$ iff S is a B-preferred solution of $(\mathcal{C}, Q - \{\text{rank}_{\prec_z}(z) = q\}, U)$.
- *Global fail*: If $U \neq \emptyset$ and for all $z \in B_\sigma$ we have $(\text{rank}_{\prec_z}(z) = \text{rank}_\sigma(z)) \in Q$ then $(\mathcal{C}, Q, U)$ has no B-preferred solution.

MCPBS has an inner loop and an outer loop. In each iteration of the outer loop, it first determines the set B of best criteria. In the inner loop, it either assigns a value to a single criterion z of B or it ensures that each criterion in B has a refutation query for its best rank. In the first case, MCPBS removes z from U and immediately leaves the inner loop. B is updated in the next iteration of the outer loop. In the second case, the conditions of a global fail are satisfied and MCPBS fails (cf. line 18 in figure 7: please note that $B \subseteq U$ holds in the second case, but not in the first case since z from B is no longer in U). We are now able to show that MCPBS is correct and complete:

Theorem 3. Suppose that all $\prec_{z_i}$ are ranked. Algorithm MCPBS$(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$ always terminates. Each successful run returns a set X of constraints, the solutions of which are B-preferred solution of $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$ and each such B-preferred solution is the solution of the result X of exactly one successful run.

MCPBS can solve several minimization subproblems for the same criterion z if it has a refutation query for z in Q. In the worst case, such a minimization problem is solved each time an assignment is added to $\mathcal{C}$. The number of subproblems can be reduced by keeping previous solutions as supports. Suppose that rank q for criterion z needs to be refuted. MCPBS determined a solution S_z with $\text{rank}_{\prec_z}(v_S(z)) = q$ before it added the refutation query. We can keep this solution as support for the best rank q of z. If the refutation query is checked again after adding the constraint Δ to $\mathcal{C}$, we first test whether S_z is a solution of $\mathcal{C} \cup \Delta$. If yes, then the best rank for z in $\mathcal{C} \cup \Delta$ is q and the refutation query has not been refuted by the additional constraints. As long as the supporting solution S_z of the refutation query satisfies all constraints, we do not need to solve a minimization problem. The subprocedures in figure 8 incorporate supports. We also introduce a support S^* for the consistency check of $\mathcal{C}$. Future work will be devoted to improve the pruning of the search tree by exploiting further properties of refutation queries and solution supports. For example, we can post a constraint that avoids that S_z is computed twice and that prunes solutions dominated by S_z as in (Gavanelli, 2002).

We analyze the complexity for the case where each criterion z has d possible values and a strict total order $\prec_z$. In each successful iteration of the inner-loop, the algorithm McPbs either assigns a value to a criterion or adds a refutation query requiring that this assignment will be violated. Hence, each non-deterministic run needs at least $O(n)$ iterations (where n is the number of criteria) and at most $O(n \cdot d)$ iterations. Since each iteration corresponds to a branching in a search tree, we obtain a search tree with a branching factor of 2 and a maximal depth of $O(n \cdot d)$. Hence, $O(2^{n \cdot d})$ is an upper bound for the number of search nodes. Furthermore, the number of search states is also bounded by the number of permutations of criteria, namely, $O(n!)$.

Given the bound-translation of a problem, we can use McPbs to determine all Pareto-optimal solutions. We consider a very simple example involving two criteria a, b that have both the domain $[1, d]$ and that are both minimized. Suppose that there are three Pareto-optimal solutions. The first one is $v_{S_1}(a) = 1$, $v_{S_1}(b) = 5$, the second one is $v_{S_2}(a) = 2$, $v_{S_2}(b) = 2$, and the third one is $v_{S_3}(a) = 5$, $v_{S_3}(b) = 1$. In spite of the simplicity of the example, we have $2d$ binary criteria, namely $u_{a,1}, \ldots, u_{a,d}$ and $u_{b,1}, \ldots, u_{b,d}$. There are no preferences between these binary criteria, but there are logical dependencies that can be exploited when running McPbs on the bound translation and that have a high impact on the form of the search tree. The search tree in figure 9 is obtained by the following observations:

1. If we add the constraint $u_{z,v} = 1$ to $\mathcal{C}$ then constraint propagation reduces the domains of the criteria $u_{z,w}$ with $v \prec_z w$ to 1 meaning that these criteria are skipped by McPbs in line 13. For example, when adding $u_{a,1} = 1$ in the left branch, constraint propagation deduces that the criteria $u_{a,i+1}, \ldots, u_{a,d}$ are all equal to 1 and McPbs does not do any branching for them.

2. Let m be the best rank for criterion z. If we add the constraint $\mathrm{rank}_{\prec_z}(z) \geqslant m$ to $\mathcal{C}$, then constraint propagation reduces the domains of the criteria $u_{z,w}$ with $\mathrm{rank}_{\prec_z}(w) < m$ to 0. For example, $u_{b,1}, \ldots, u_{b,4}$ are all 0 after adding $u_{b,5} = 1$ and need not be considered.

3. If there is a refutation query $u_{z,v} = 1$ in Q and $w \prec_z v$, then violating $u_{z,v} = 1$ will violate $u_{z,w} = 1$ as well. Hence, $u_{z,w} = 1$ needs to be refuted as well in this case. We therefore will not select $u_{b,1}, \ldots, u_{b,4}$ when seeking a refutation of $u_{b,5} = 1$.

We thus select a binary criterion $u_{z,v}$ with a best bound v for z in order to reconstitute the search behaviour of McPbs on the original problem when running it on the bound-translation. Furthermore, we can improve the refutation behaviour by exploiting the supporting solutions of refutation queries:

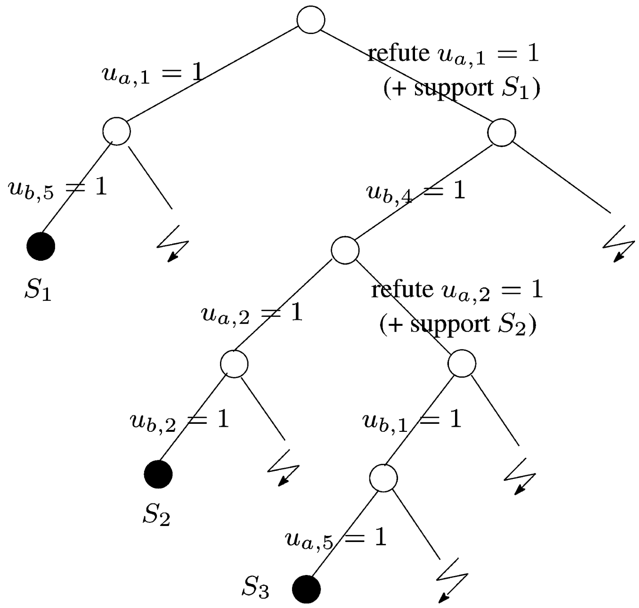

Figure 9. Determining Pareto-optimal solutions.

1. If there is a refutation query $u_{z,v} = 1$ in Q and $u_{z,v} = 1$ is not refuted by $\mathcal{C}$, then there exists a solution S of $\mathcal{C}$ that satisfies $u_{z,v} = 1$. It also satisfies $u_{z,w} = 1$ for $w \succ_z v$. Hence, adding $u_{z,w} = 1$ to $\mathcal{C}$ will not produce a state where $u_{z,v} = 1$ is refuted. We therefore will not select $u_{b,6}, \ldots, u_{b,d}$ when seeking a refutation of $u_{b,5} = 1$.

2. As a consequence, a refutation query $u_{z,v} = 1$ in Q cannot be refuted if all binary criteria in B have the form $u_{z,v'}$, i.e. concern the original criterion z. Therefore, $u_{b,5} = 1$ cannot be refuted.

3. If a refutation query $u_{z,v} = 1$ in Q is satisfied by the solution S_z and S_z is a solution of $\mathcal{C}$, then we select a criterion u_{z^*,v^*} such that adding $u_{z^*,v^*} = 1$ will invalidate S_z The constraint $z^* \preceq_z v^*$ is satisfied by S_z if and only if $v_{S_z}(z^*) \preceq_z v^*$. Hence, $u_{z^*,v^*} = 1$ invalidates S_z if $v_{S_z}(z^*) \not\preceq_z v^*$. We choose a $\preceq_z$-largest value v^* satisfying this condition. For example, we choose $u_{b,4} = 1$ to invalidate the support S_1 for $u_{a,1} = 1$ and $u_{b,1} = 1$ to invalidate the support S_2 for $u_{a,2} = 1$.
   However, if no such criterion z^* and value v^* exist then $u_{z,v} = 1$ cannot be refuted, meaning that the current state has no G-preferred solution. In this case, a failure occurs.

We have thus seen how solution supports for refutation queries can guide the selection of binary criteria and ensure that refutations can be found. Moreover, solution supports for refutation queries could be used to do additional propagations and deductions: if there is only one possibility to invalidate a support for a refutation query, then this possibility must be true. A detailed elaboration of those rules is a subject of future work.

procedure SELECT (**in** B: best criteria; Q: refutation queries)
select a $u_{z,v}$ in B and access the original criterion z and its order $\prec_z$;
if *there exists q s.t.* $(u_{z,q} = 1) \in Q$ **then**
    consider $(u_{z,q} = 1) \in Q$ with a $\prec_z$-smallest q;
    let S_z be the support for $u_{z,q} = 1$ (i.e. $v_{S_z}(z) \preceq_z q$);
    if S_z *is a solution of* $(\mathcal{C}, \mathcal{X})$ **then**
        if *there is a* $u_{z^*,v^*} \in B$ *with* $z \neq z^*$ *and* $v^* \prec_{z^*} v_{S_z}(v^*)$ **then**
            select a $u_{z^*,v^*} \in B$ with $\prec_{z^*}$-greatest value v^* s.t. $z \neq z^*$ and $v^* \prec_{z^*} v_{S_z}(v^*)$;
            return u_{z^*,v^*};
        else fail
$m := \text{MINIMIZEBOUND}(\mathcal{C}, \mathcal{X}, z, \prec_z)$;
return $u_{z,m}$;

procedure MINIMIZEBOUND (**in** $\mathcal{C}$: constraints; $\mathcal{X}$: variables; z: criterion; $\prec_z$: preferences)
if *there is no support* S_z *for* z *or* S_z *is not a solution of* $(\mathcal{C}, \mathcal{X})$ **then**
    determine a solution S of $(\mathcal{C}, \mathcal{X})$ with smallest $\mathit{rank}_{\prec_z}(z)$;
    let $\mathcal{C} := \mathcal{C} \cup \{\mathit{rank}_{\prec_z}(z) \geq m\}$;
    set S_z to S and set S^* to S;
return $v_{S_z}(z)$;

procedure MINIMIZE (**in** $\mathcal{C}$: constraints; $\mathcal{X}$: variables; z: criterion; $\prec_z$: preferences)
return 1;

Figure 10. Subprocedures for Pareto-optimal solutions.

We now adapt the subprocedures to the bound-translation and incorporate the selection rules for the bound translation. The resulting subprocedures are shown in figure 10. Please note that minimization subproblems and solution supports S_z refer to the original criteria z and not to the binary criteria $u_{z,v}$. These changes ensure that MCPBS provides an adequate search behaviour for enumerating Pareto-optimal solutions. For two criteria, we get a behaviour that is similar[6] to the algorithm of Van Wassenhove and Gelders (1980), which determines all Pareto-optimal solutions for bicriteria-problems. On the one hand, we thus derive a well-known algorithm from more basic principles and we can understand it as a method that deduces bound-constraints from solution supports of refutation queries. On the other hand, we can understand MCPBS (on the bound-translation) as a method that generalizes the algorithm of Van Wassenhove and Gelders to more than two criteria.

According to theorem 2, we can also use MCPBS to compute balanced solutions if we apply it to the min-max-translation. Table 3 shows the values of the three criteria price, quality, and distance for the two B-preferred solutions and the unique E-preferred solution, which is determined w.r.t. the two groups $G_1 := \{\mathit{scaledPrice}, \mathit{scaledQuality}\}$ and $G_2 := \{\mathit{scaledDistance}\}$, where G_1 is more important than G_2. We see that the

Table 3
Comparing B-preferred and E-preferred solutions.

Solution	Price	Quality	Distance	#Subsearches	Total effort
1st B-preferred sol.	133	1	3	3	2872 choices
2nd B-preferred sol.	600	5	2	3	61 choices
E-preferred sol.	254	4	2	3	4872 choices

E-preferred solution provides a good compromise between the B-preferred solution. Typically, a user might ask for such a E-preferred solution whenever the gap between the different B-preferred solutions is too high. Table 3 also shows the number of subsearches needed for each solution, which is 3 in all cases. We use constraint-based Branch-and-Bound to solve the subproblems. The total effort, i.e. the number of the choices of all subsearches, highly depends on the first criterion that is chosen. It is interesting to note that the effort for E-preferred solutions is not too high compared to the effort for the first B-preferred solution. This demonstrates that constraint-based Branch-and-Bound can well minimize min-max-variables as long as two criteria need to be balanced. Future investigations are needed to measure the behaviour for more criteria.

Other CSP-based approaches to multi-criteria optimization do a single Branch-and-Bound search for all criteria, which requires maintaining 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, MCPBS does not need dominance checking, but uses refutation queries to avoid non-preferred solutions. Furthermore, MCPBS does not preform a single search with multiple criteria, but multiple searches with a single criterion, and thus follows well-established methods for lexicographical optimization that are broadly used in Operations Research. This paper provides the theoretical background for MCPBS. Therefore, a detailed empirical evaluation of MCPBS and its comparison with Branch-and-Bound search for multiple criteria is beyond its scope and a subject of future work.

5. 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 heuristic should be adapted as well. Thus, the explored search tree of each minimization subproblem will strongly depend on the selected objective as figure 11 illustrates.

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.

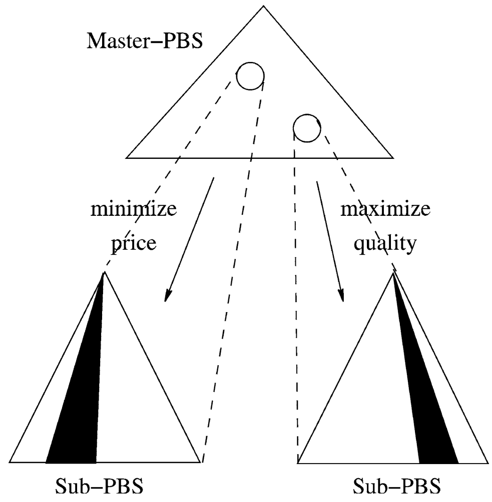

Figure 11. Explored subtree depends on projected preferences.

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

$$\text{if } u_k \prec_{x_k} v_k \quad \text{then } f_j(u_1, \ldots, u_m) \preceq_{z_j} f_j(v_1, \ldots, v_m). \tag{18}$$

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

The projected preferences preserve Pareto-optimality:

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

We give some examples for preference projections satisfying the conditions of definition 7:

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.

Table 4
Reduction of search effort by preference projection.

Objective	Projection	Distance to opt. (1st. sol.)	Effort (best sol.)	Effort (opt. proof)
minimize	no	248%	462 choices	1203 choices
price	yes	24%	251 choices	675 choices
maximize	no	20%	27 choices	27 choices
quality	yes	0%	16 choices	16 choices

3. Given an element constraint (i.e. an arbitrary functional 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 $f(x)$:

$$u \prec_x v \quad \text{iff} \quad f(u) < f(v). \tag{19}$$

Hence, projecting the increasing or decreasing order on integers via an element constraint yields a ranked order.

Table 4 shows the impact of preference projection on our vacation adviser problem. We consider the subsearches for B-preferred solutions that are performed for the first selected criterion, namely the quality or the price. These criteria are defined by different element constraints and the projected preferences completely differ depending on the selected objective. If we want to minimize the price, the projected preferences ensure that cheaper hotels are selected first for each vacation stop. If we want to maximize the quality, the projected preferences will favour hotels of better quality in each stop.

In this example, preference projection helps to reduce the number of choices around 45%. More importantly, it improves the degree of optimality of the first solution. If no preference projection is used the first solution has a distance of 248% to the best price and a distance of 20% to the best quality. If preference projection is used, the first solution depends on the chosen objective. If price is minimized first the distance to the best price reduces from 248% to 24%. If quality is maximized first the distance to the best quality reduces from 20% to 0%. Hence, these two solutions are completely different. Since the problem is weakly constrained, the first solutions are found rapidly, meaning that different trade-offs can indeed be produced in a small time frame if preference projection is used.

This shows that preference projection is of high importance for interactive configuration problems where time is limited. If only a single solution can be determined by each subsearch, standard constraint-based Branch-and-Bound will always return the same solution independent of the selected objective. In this case, preference projection ensures that the selected objective is taken into account and that different solutions are determined. Again, this is important for interactive configuration, where we want to determine several solutions of different characteristics in a short time frame.

Since extreme and balanced solutions are Pareto-optimal, we can additionally use the projected preferences to reduce search effort when solving a subproblem. For this purpose, we can apply the algorithm MCPBS to the decision variables $x_1, \ldots, x_m$ and the bound-translation. An investigation of the possible gains of this method is a subject of future work.

6. 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 the 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 and Thomason, 1999; Bacchus and Grove, 1995; Boutilier et al., 1997; Domshlak, Brafman, and Shimony, 2001). We further generalized PBS, such that not only extreme solutions are computed, but also balanced and Pareto-optimal solutions. 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). Furthermore, we project the preferences on the selected criterion to preferences between the search decisions, which provides an adapted search heuristic for the optimization objective and which allows the search effort to be reduced further. Hence, different regions of the search space will be explored depending on the selected objective. The master-PBS has been implemented in ILOG JCONFIGURATOR V2.1 and adds multi-criteria optimization functionalities to this constraint-based configuration tool.

Future work will be devoted to improving the pruning behaviour of the new PBS procedures w.r.t. the master problem as well as the subproblems. We will also examine whether PBS can be used to determine preferred solutions as defined by soft constraints (Bistarelli et al., 1999; Khatib et al., 2001).

Acknowledgments

I would like to thank my colleagues Xavier Ceugniet, Olivier Lhomme, Daniel Mailharro, Jean-François Puget, Philippe Refalo, and Jean-Charles Régin for numerous discussions and comments. Furthermore, I am also grateful to Mark Wallace who challenged the lexicographical approach and thus stimulated the responses given in this paper. Finally, I would like to thank the anonymous reviewers of this article for their excellent comments and Jane Morcombe for polishing my English.

Appendix A. Proofs

This appendix contains detailed proofs for the propositions of the article. For the sake of readability, we are using the following short-hand for formulae:

$$\begin{aligned}
S_1 \text{ is better than } S_2 \text{ on criterion } z &\quad\Leftrightarrow\quad v_{S_1}(z) \prec_z v_{S_2}(z),\\
S_1 \text{ and } S_2 \text{ agree on criterion } z &\quad\Leftrightarrow\quad v_{S_1}(z) = v_{S_2}(z),\\
z_i \text{ is a } \prec_{\mathcal{Z}}\text{-best criterion} &\quad\Leftrightarrow\quad \text{there is no } z_j \text{ s.t. } z_j \prec_{\mathcal{Z}} z_i,\\
v \text{ is a } \prec_z\text{-best value} &\quad\Leftrightarrow\quad \text{there is no } w \text{ s.t. } v \prec_z w.
\end{aligned}$$

Proof of proposition 1. Let $\mathcal{P}$ be $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$. We show that each G-preferred solution S of $\mathcal{P}$ is Pareto-optimal by a contradiction proof. Assume that S is not a Pareto-optimal solution. According to definition 1, there exists a solution S^* of $\mathcal{P}$ such that $v_{S^*}(z_i) \preceq_{z_i} v_S(z_i)$ for all i and $v_{S^*}(z_i) \prec_{z_i} v_S(z_i)$ for at least one i. Consider a $\prec_{\mathcal{Z}}$-best criterion z_k such that S^* is different from S for z_k. Since S is G-preferred, one of the following conditions holds: (1) $v_S(z_k) \prec_{z_k} v_{S^*}(z_k)$ or (2) there exists a j with $z_j \prec_{\mathcal{Z}} z_k$ such that S and S^* differ on z_j. The properties of S^* imply that $v_{S^*}(z_k) \preceq_{z_k} v_S(z_k)$. Since S^* and S are different for z_k, we obtain $v_{S^*}(z_k) \prec_{z_k} v_S(z_k)$. Hence, the first condition does not hold since $\prec_{z_k}$ is irreflexive. The second condition implies that S^* and S differ for a criterion that is better than z_k. However, z_k is a $\prec_{\mathcal{Z}}$-best criterion satisfying this condition. Hence, we get a contradiction in both cases, meaning that S is Pareto-optimal.

We now show that each Pareto-optimal solution is G-preferred if there are no preferences between criteria, i.e. $\prec_{\mathcal{Z}} = \emptyset$. Consider a solution S of $\mathcal{P}$ that is not G-preferred. Hence, there exists a solution S^* of $\mathcal{P}$ such that $v_S(z_k) \neq v_{S^*}(z_k)$ for some k and for all i with $v_S(z_i) \neq v_{S^*}(z_i)$ one of the following conditions holds: (1) $v_{S^*}(z_i) \prec_{z_i} v_S(z_i)$ or (2) there exists a j with $z_j \prec_{\mathcal{Z}} z_i$ such that S and S^* differ on z_j. Since there are no preferences between criteria, no such j can exist and the second condition does not hold. As a consequence, we obtain $v_{S^*}(z_i) \prec_{z_i} v_S(z_i)$ for all z_i where S^* and S differ. As a consequence, $v_{S^*}(z_i) \preceq_{z_i} v_S(z_i)$ for all z_i. Furthermore, we know that there is at least one z_k such that S and S^* differ on z_k. Hence, we obtain $v_{S^*}(z_k) \prec_{z_k} v_S(z_k)$. This means that S is G-preferred if it is Pareto-optimal. $\square$

Proof of proposition 2. Given the problem $\mathcal{P} := (\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$, a $\prec_{\mathcal{Z}}$-best criterion z_i, and a value u, we introduce the problem $\mathcal{P}^* := (\mathcal{C} \cup \{z_i \preceq_{z_i} u\}, \mathcal{X}, \mathcal{Z}, \prec)$.

Consider a solution S of $\mathcal{P}$ such that $v_S(z_i) \preceq_{z_i} u$. We will show that S is not a G-preferred solution of $\mathcal{P}$ if it is not a G-preferred solution of $\mathcal{P}^*$. Since S satisfies the constraint $z_i \preceq_{z_i} u$ in addition to $\mathcal{C}$, it is a solution of $\mathcal{P}^*$. Assume that S is not a G-preferred solution of $\mathcal{P}^*$. Hence, there is another solution S^* of $\mathcal{P}^*$ such that $v_S(z_k) \neq v_{S^*}(z_k)$ for some k and for all l with $v_S(z_l) \neq v_{S^*}(z_l)$ one of the following conditions holds: (1) $v_{S^*}(z_l) \prec_{z_l} v_S(z_l)$ or (2) there exists a j s.t. $z_j \prec_{\mathcal{Z}} z_l$ and $v_{S^*}(z_j) \neq v_S(z_j)$. Since the constraints of $\mathcal{P}$ are a subset of the constraints of $\mathcal{P}^*$, S^* is also a solution of $\mathcal{P}$ and we obtain that S is not a G-preferred solution according to definition 5.

Now consider a solution S of $\mathcal{P}^*$. We will show that S is not a G-preferred of $\mathcal{P}^*$ if it is not a G-preferred solution of $\mathcal{P}$. Since S satisfies the constraint $z_i \preceq_{z_i} u$, we obtain $v_S(z_i) \preceq_{z_i} u$. Furthermore, S is a solution of $\mathcal{P}$. Suppose that S is not a G-preferred solution of $\mathcal{P}$. Hence, there is another solution S^* of $\mathcal{P}$ such that $v_S(z_k) \neq v_{S^*}(z_k)$ for some k and for all l with $v_S(z_l) \neq v_{S^*}(z_l)$ one of the following conditions holds: (1) $v_{S^*}(z_l) \prec_{z_l} v_S(z_l)$ or (2) there exists a j s.t. $z_j \prec_{\mathcal{Z}} z_l$ and $v_{S^*}(z_j) \neq v_S(z_j)$.

Assume that $v_{S^*}(z_i) \preceq_{z_i} u$ is not satisfied. Since $v_S(z_i) \preceq_{z_i} u$, this means that S^* and S cannot be equal for z_i. Since z_i is a $\prec_{\mathcal{Z}}$-best criterion, there does not exist a j s.t. $z_j \prec_{\mathcal{Z}} z_i$ and $v_{S^*}(z_j) \neq v_S(z_j)$. Since S^* and S differ for z_i and the condition 2 does not hold for z_i, condition 1 must be true: $v_{S^*}(z_i) \prec_{z_i} v_S(z_i)$. Since $v_S(z_i) \preceq_{z_i} u$, we also get $v_{S^*}(z_i) \preceq_{z_i} u$, which contradicts the assumption. Therefore, the assumption was wrong and we deduce $v_{S^*}(z_i) \preceq_{z_i} u$. Hence, S^* is a solution of $\mathcal{P}^*$. As a consequence, S is not a G-preferred solution of $\mathcal{P}^*$ since S^* satisfies the conditions stated in definition 5. $\square$

Proof of proposition 3. Given the problem $\mathcal{P} := (\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$, a $\prec_{\mathcal{Z}}$-best criterion, and a $\prec_{z_i}$-best value v for z_i, we define $\mathcal{P}^* := (\mathcal{C} \cup \{z_i = v\}, \mathcal{X}, \mathcal{Z} - \{z_i\}, \prec)$.

Let S be a G-preferred solution of $\mathcal{P}$ and suppose $v_S(z_i) = v$. Then S is a solution of $\mathcal{P}^*$. Now suppose that S is not a G-preferred solution of $\mathcal{P}^*$. Hence, there is another solution S^* of $\mathcal{P}^*$ such that $v_S(z_k) \neq v_{S^*}(z_k)$ for some $z_k \in \mathcal{Z} - \{z_i\}$ and for all $z_l \in \mathcal{Z} - \{z_i\}$ with $v_S(z_l) \neq v_{S^*}(z_l)$ one of the following conditions holds: (1) $v_{S^*}(z_l) \prec_{z_l} v_S(z_l)$ or (2) there exists a $z_j \in \mathcal{Z} - \{z_i\}$ s.t. $z_j \prec_{\mathcal{Z}} z_l$ and $v_{S^*}(z_j) \neq v_S(z_j)$. Since S^* is a solution of $\mathcal{P}^*$, it also satisfies $v_{S^*}(z_i) = v$. Hence, S and S^* agree on z_i. Hence, we can state that $v_S(z_k) \neq v_{S^*}(z_k)$ for some $z_k \in \mathcal{Z}$ and for all $z_l \in \mathcal{Z}$ with $v_S(z_l) \neq v_{S^*}(z_l)$ one of the following conditions holds: (1) $v_{S^*}(z_l) \prec_{z_l} v_S(z_l)$ or (2) there exists a $z_j \in \mathcal{Z}$ s.t. $z_j \prec_{\mathcal{Z}} z_l$ and $v_{S^*}(z_j) \neq v_S(z_j)$. According to this, S is not a G-preferred solution of $\mathcal{P}$, which is a contradiction.

Let S be a G-preferred solution of $\mathcal{P}^*$. Then $v_S(z_i) = v$. Now suppose that S is not a G-preferred solution of $\mathcal{P}$. Hence, there is another solution S^* of $\mathcal{P}$ such that $v_S(z_k) \neq v_{S^*}(z_k)$ for some $z_k \in \mathcal{Z}$ and for all $z_l \in \mathcal{Z}$ with $v_S(z_l) \neq v_{S^*}(z_l)$ one of the following conditions holds: (1) $v_{S^*}(z_l) \prec_{z_l} v_S(z_l)$ or (2) there exists a $z_j \in \mathcal{Z}$ s.t. $z_j \prec_{\mathcal{Z}} z_l$ and $v_{S^*}(z_j) \neq v_S(z_j)$.

Assume $v_{S^*}(z_i) \neq v$. Hence, S^* and S differ on z_i. Hence S^* is better than S on z_i or there exists a $z_j \in \mathcal{Z}$ s.t. $z_j \prec_{\mathcal{Z}} z_i$ and $v_{S^*}(z_j) \neq v_S(z_j)$. The first case is not satisfied since $v_S(z_i)$ is equal to the $\prec_{z_i}$-best value v for z_i. The second case is not satisfied since z_i is a $\prec_{\mathcal{Z}}$-best criterion. We obtain a contradiction in both cases and conclude that $v_{S^*}(z_i) = v$. Hence, we can state that $v_S(z_k) \neq v_{S^*}(z_k)$ for some $z_k \in \mathcal{Z} - \{z_i\}$ and for all $z_l \in \mathcal{Z} - \{z_i\}$ with $v_S(z_l) \neq v_{S^*}(z_l)$ one of the following conditions hold: (1) $v_{S^*}(z_l) \prec_{z_l} v_S(z_l)$ or (2) there exists a $z_j \in \mathcal{Z} - \{z_i\}$ s.t. $z_j \prec_{\mathcal{Z}} z_l$ and $v_{S^*}(z_j) \neq v_S(z_j)$. Hence, S is not a G-preferred solution of $\mathcal{P}^*$, which is a contradiction. $\square$

Proof of proposition 4. Consider the problem $\mathcal{P} := (\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$. We show that a B-preferred solution S of $\mathcal{P}$ is G-preferred by a contradiction proof. Since S is a B-preferred solution then there exists a permutation π such that (1) π respects $\prec_{\mathcal{Z}}$ and (2) there is no other solution S' of $(\mathcal{C}, \mathcal{X})$ such that $V_{S'}(\pi(Z)) \prec^{\pi}_{\text{lex}} V_S(\pi(Z))$. Now, assume that S is not G-preferred. Hence, there exists a solution S^* such that $v_S(z_k) \neq v_{S^*}(z_k)$ for some k and for all i with $v_S(z_i) \neq v_{S^*}(z_i)$ one of the following conditions holds: (1) $v_{S^*}(z_i) \prec_{z_i} v_S(z_i)$ or (2) there exists a j s.t. $z_j \prec_{\mathcal{Z}} z_i$ and $v_{S^*}(z_j) \neq v_S(z_j)$. Consider the smallest k such that $v_S(z_{\pi_k}) \neq v_{S^*}(z_{\pi_k})$. We consider two cases, both leading to a contradiction:

1. Suppose S^* is better than S on z_{π_k}. Since both solutions agree on $z_{\pi_1}, \ldots, z_{\pi_{k-1}}$ we thus obtain that $V_{S^*}(\pi(\mathcal{Z})) \prec^{\pi}_{\text{lex}} V_S(\pi(\mathcal{Z}))$. Hence, S is not B-preferred, which is a contradiction.
2. Suppose S^* is not better than S on z_{π_k}. Hence, condition 1 is false for z_{π_k} and there exists a π_j with $z_{\pi_j} \prec_{\mathcal{Z}} z_{\pi_k}$ and $v_{S^*}(z_{\pi_j}) \neq v_S(z_{\pi_j})$. Since k is the smallest index such that S and S^* differ on z_{π_k}, we obtain $j \geqslant k$. Since π respects the preferences, $z_{\pi_j} \prec_{\mathcal{Z}} z_{\pi_k}$ implies $j < k$, which is a contradiction. □

Proof of proposition 5. We need the following lemma to prove the proposition.

Lemma 1. Suppose $\prec_{\mathcal{Z}}$ is a ranked order. If S_1 is a G-preferred solution for $\mathcal{P} := (\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$ and S_2 is a solution of $\mathcal{P}$ that is better than S_1 for a $\prec_{\mathcal{Z}}$-best criterion z_k then there exists another $\prec_{\mathcal{Z}}$-best criterion z_j such that S_1 and S_2 differ on z_j and S_2 is not better than S_1 on z_j.

Proof. Let S_1 and S_2 as supposed in the lemma. Assume that $v_{S_2}(z) \preceq_z v_{S_1}(z)$ for all $\prec_{\mathcal{Z}}$-best criteria. Let z_i be an arbitrary criterion such that S_1 and S_2 are different on z_i and suppose that S_2 is not better than S_1 on z_i. According to the assumption, z_i cannot be a $\prec_{\mathcal{Z}}$-best criterion. Hence, there exists a $\prec_{\mathcal{Z}}$-best criterion z_j such that $z_j \prec_{\mathcal{Z}} z_i$. Since z_j and z_k are both $\prec_{\mathcal{Z}}$-best elements, they are incomparable w.r.t. $\prec_{\mathcal{Z}}$. According to the properties of a ranked order, we get $z_k \prec_{\mathcal{Z}} z_i$ in this case and we know that S_1 and S_2 do not agree on z_k. Hence, we can state that for all criteria z_i with $v_{S_1}(z_i) \neq v_{S_2}(z_i)$ one of the following conditions holds: (1) $v_{S_2}(z_i) \preceq_{z_i} v_{S_1}(z_i)$ or (2) $z_k \prec_{\mathcal{Z}} z_i$ $v_{S_2}(z_k) \neq v_{S_1}(z_k)$. As a consequence, S_1 is not G-preferred, which is a contradiction. Hence, we cannot have $v_{S_2}(z) \preceq_z v_{S_1}(z)$ for all $\prec_{\mathcal{Z}}$-best criteria and there exists a $\prec_{\mathcal{Z}}$-best criterion z_j with $v_{S_2}(z_j) \not\preceq_{z_j} v_{S_1}(z_j)$. Since $v_{S_2}(z_k) \prec_{z_k} v_{S_1}(z_k)$, the criterion z_j is different from z_k. □

Let $\mathcal{P}$ be a problem $(\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$ such that there do not exist three solutions S_1, S_2, S_3 for $\mathcal{P}$ and a criterion z_i with $v_{s_1}(z_i) \prec_{z_i} v_{s_2}(z_i)$ and $v_{s_3}(z_i) \not\preceq_{z_i} v_{s_2}(z_i)$. Furthermore let $\prec_{\mathcal{Z}}$ be a ranked order.

We show that a G-preferred solution S of $\mathcal{P}$ is a B-preferred solution by an induction on the number n of criteria.

If $n = 0$ then $\mathcal{Z}$ is empty and all solutions are incomparable w.r.t. to any lexicographical order. Hence, each solution of $\mathcal{P}$ is B-preferred since there is no better solution w.r.t. a lexicographical order.

If $n \geqslant 1$ suppose that the proposition is true for all problems with $n-1$ or less criteria. We first show that there exists a $\prec_{\mathcal{Z}}$-best criterion z_k such that there is no solution that is better than S on z_k. For this purpose, consider any $\prec_{\mathcal{Z}}$-best criteria z_i. We distinguish two cases:

1. If there does not exist a solution S^* that is better than S on z_i we choose $k := i$.

2. Otherwise, there exists a solution S^* that is better than S on z_i. According to lemma 1, there exists a $\prec_{\mathcal{Z}}$-best criteria z_j such that S is different from S^* on z_j and S^* is not better than S on z_j. According to the prerequisites of the proposition, there cannot be a third solution that is better than S on z_j. Hence, we choose $k = j$.

Thus, we know that there is a k such that no solution is better than S on z_k.

Since z_k is a $\prec_{\mathcal{Z}}$-best criterion there exists a permutation π that respects the preferences between criteria and that chooses z_k as first criterion, i.e. $\pi_1 = k$. Assume that S is not a B-preferred solution of $\mathcal{P}$. Hence, there exists a solution S^* such that

$$V_{S^*}\big(\pi(\mathcal{Z})\big) \prec_{\text{lex}}^{\pi} V_S\big(\pi(\mathcal{Z})\big). \tag{A.1}$$

From (A.1), we obtain $v_{S^*}(z_{\pi_1}) \preceq_{z_{\pi_1}} v_S(z_{\pi_1})$. Since $\pi_1 = k$ and no solution is better than S on z_k, we conclude that

$$v_{S^*}(z_{\pi_1}) = v_S(z_{\pi_1}). \tag{A.2}$$

From the properties (A.1) and (A.2), we obtain

$$\big(v_{S^*}(z_{\pi_2}), \ldots, v_{S^*}(z_{\pi_n})\big) \prec_{\text{lex}}^{\pi} \big(v_S(z_{\pi_2}), \ldots, v_S(z_{\pi_n})\big). \tag{A.3}$$

Now consider the problem $\mathcal{P}'$ that remains after assigning the value $v_S(z_{\pi_1})$ to the criterion z_{π_1}:

$$\mathcal{P}' := \big(\mathcal{C} \cup \big\{z_k = v_S(z_{\pi_1})\big\}, \mathcal{X}, \mathcal{Z} - \{z_k\}, \prec\big).$$

Since S and S^* satisfy $z_k = v_S(z_{\pi_1})$, they are both solutions of $\mathcal{P}'$. According to proposition 3, S is also a G-preferred solution of $\mathcal{P}'$. Property (A.3) implies that S is not a B-preferred solution of $\mathcal{P}'$, which contradicts the induction hypothesis. Thus, we have shown that S is a B-preferred solution of $\mathcal{P}$. $\square$

Proof of proposition 6. Let $\mathcal{P} := (\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$ and $\mathcal{P}^* := (\mathcal{C}, \mathcal{X}, \mathcal{U}, \prec')$. We observe the following properties:

1. Since $\mathcal{P}$ and $\mathcal{P}^*$ have exactly the same constraints, S is a solution of $\mathcal{P}$ if and only if S is a solution of $\mathcal{P}^*$.

2. $z^* \prec_{\mathcal{Z}} z$ iff there exists values v, v^* s.t. $u_{z^*,v^*} \prec'_{\mathcal{U}} u_{z,v}$.

3. Consider two solution S_1 and S_2 of $\mathcal{P}^*$ and a binary criterion $u_{z,v} \in \mathcal{U}$. Suppose S_1 and S_2 assign the same value v' to z. If $v' \preceq_z v$, then $v_{S_1}(u_{z,v})$ and $v_{S_2}(u_{z,v})$ are both equal to 1. In the other case, $v_{S_1}(u_{z,v})$ and $v_{S_2}(u_{z,v})$ are both equal to 0. Hence, S_1 and S_2 assign the same value to $u_{z,v}$.
As a consequence, S_1 and S_2 do not agree on z if they do not agree on $u_{z,v}$.

4. Consider two solutions S_1 and S_2 of $\mathcal{P}^*$ and let $v_1 := v_{S_1}(z)$ and $v_2 := v_{S_2}(z)$. We suppose that S_1 and S_2 agree on u_{z,v_1} and on u_{z,v_2}. The first supposition implies $v_2 = v_{S_2}(z) \preceq_z v_1$ and the second supposition implies $v_1 = v_{S_1}(z) \preceq_z v_2$. Due to the irreflexivity of $\preceq_z$, we obtain v_1 and v_2 are equal, meaning that S_1 and S_2 agree on z.
As a consequence, S_1 and S_2 differ on u_{z,v_1} or on u_{z,v_2} if S_1 and S_2 differ on z.

Let S be a G-preferred solution of $\mathcal{P}$. We show that S is a G-preferred solution of $\mathcal{P}^*$ by a contradiction proof. Suppose S is not a G-preferred solution of $\mathcal{P}^*$. Hence, there is another solution S^* of $\mathcal{P}^*$ such that $v_S(u_{z,v}) \neq v_{S^*}(u_{z,v})$ for some $u_{z,v} \in \mathcal{U}$ and for all $u_{z,v} \in \mathcal{U}$ with $v_S(u_{z,v}) \neq v_{S^*}(u_{z,v})$ one of the following conditions holds: (1) $v_{S^*}(u_{z,v}) \prec'_{u_{z,v}} v_S(u_{z,v})$ or (2) there exists a $u_{z^*,v^*} \in \mathcal{U}$ s.t. $u_{z^*,v^*} \prec'_{\mathcal{U}} u_{z,v}$ and $v_{S^*}(u_{z^*,v^*}) \neq v_S(u_{z^*,v^*})$. If S and S^* do not agree on $u_{z,v}$ they do not agree on z as shown above. Since there exists such a $u_{z,v}$, we can state that S and S^* do not agree on some z.

Now consider a $z \in \mathcal{Z}$ such that $v_S(z) \neq v_{S^*}(z)$. Let $v := v_S(z)$ and $v^* := v_{S^*}(z)$. By definition, $u_{z,v}$ is equal to 1 in S and that u_{z,v^*} is equal to 1 in S^*. We consider two cases:

1. Suppose $u_{z,v}$ is equal to 1 in S^*. This means that $v_{S^*}(z)$ is smaller or equal to $v = v_S(z)$. Since S and S^* do not agree on z we obtain $v_{S^*}(z) \prec_z v_S(z)$ in this case.

2. Suppose $u_{z,v}$ is equal to 0 in S^*. This means that S and S^* differ on $u_{z,v}$. Furthermore, S assigns the preferred value 1 of $u_{z,v}$, whereas S^* assigns the non-preferred value 0. Hence, we have $v_S(u_{z,v}) \prec'_{u_{z,v}} v_{S^*}(u_{z,v})$. This means that $v_{S^*}(u_{z,v}) \prec'_{u_{z,v}} v_S(u_{z,v})$ is false. Hence, there exists a $u_{z^*,v^*} \in \mathcal{U}$ s.t. $u_{z^*,v^*} \prec'_{\mathcal{U}} u_{z,v}$ and $v_{S^*}(u_{z^*,v^*}) \neq v_S(u_{z^*,v^*})$. There are only preferences between binary criteria if there is a preference between the original criteria to which they refer. Hence, $z^* \prec_{\mathcal{Z}} z$. Since S and S^* do not agree on u_{z^*,v^*}, they do not agree on z^*.

In both cases, we have shown that S is not a G-preferred solution of $\mathcal{P}$, which is a contradiction.

Let S be a G-preferred solution of $\mathcal{P}^*$. We show that S is a G-preferred solution of $\mathcal{P}$ by a contradiction proof. Suppose S is not a G-preferred solution of $\mathcal{P}$. Hence, there is another solution S^* of $\mathcal{P}$ such that $v_S(z) \neq v_{S^*}(z)$ for some $z \in \mathcal{Z}$ and for all $z_i \in \mathcal{Z}$ with $v_S(z_i) \neq v_{S^*}(z_i)$ one of the following conditions holds: (1) $v_{S^*}(z_i) \prec_{z_i} v_S(z_i)$ or (2) there exists a $z_j \in \mathcal{Z}$ s.t. $z_j \prec_{\mathcal{Z}} z_i$ and $v_{S^*}(z_j) \neq v_S(z_j)$.

Since there is a $z \in \mathcal{Z}$ with $v_S(z) \neq v_{S^*}(z)$, we conclude that S and S^* differ on $u_{z,v_S(z)}$ or on $u_{z,v_{S^*}(z)}$. Hence, we know that S and S^* differ on some binary criterion.

Consider a $u_{z,v}$ such that S and S^* differ on $u_{z,v}$. Suppose that S^* is not better than S on $u_{z,v}$. This means that S assigns the better value 1 to $u_{z,v}$ and S^* the worse value 0. As a consequence, $v_S(z) \preceq_z v$ and $v_{S^*}(z) \not\preceq_z v$. In this case, we cannot have $v_{S^*}(z) \preceq_z v_S(z)$. According to the properties of S^*, there exists a $z^* \in \mathcal{Z}$ s.t. $z^* \prec_{\mathcal{Z}} z$ and $v_{S^*}(z^*) \neq v_S(z)$. We then know that S and S^* differ on $u_{z^*,v_S(z^*)}$ or on $u_{z^*,v_{S^*}(z^*)}$. Furthermore, $u_{z^*,v_S(z^*)} \prec'_{\mathcal{U}} u_{z,v}$ and $u_{z^*,v_{S^*}(z^*)} \prec'_{\mathcal{U}} u_{z,v}$ Hence, S is not a G-preferred solution of $\mathcal{P}^*$, which is a contradiction. □

Proof of theorem 1. We first show that $\prec'_{\mathcal{U}}$ is a ranked order. Suppose $u_{z_1,v_1} \prec'_{\mathcal{U}} u_{z_2,v_2}$. Then $z_1 \prec_{\mathcal{Z}} z_2$. Furthermore, suppose neither $u_{z_2,v_2} \prec'_{\mathcal{U}} u_{z_3,v_3}$ nor $u_{z_3,v_3} \prec'_{\mathcal{U}} u_{z_2,v_2}$ is the case. Hence, neither $z_2 \prec_{\mathcal{Z}} z_3$ nor $z_3 \prec_{\mathcal{Z}} z_2$ is the case. Since $\prec_{\mathcal{Z}}$ is ranked we obtain $z_1 \prec_{\mathcal{Z}} z_3$. As a consequence, $u_{z_1,v_3} \prec'_{\mathcal{U}} u_{z_1,v_3}$. Similarly, we can show that $z_1 \prec_{\mathcal{Z}} z_3$ is obtained if $z_2 \prec_{\mathcal{Z}} z_3$ and neither $z_1 \prec_{\mathcal{Z}} z_2$ nor $z_2 \prec_{\mathcal{Z}} z_1$ is the case.

Since the $u_{z,v}$'s are all binary it is easy to see that they match the conditions of theorem 5. We can thus prove the theorem by combining propositions 6 and 5. □

Proof of theorem 2. Let $\mathcal{P} := (\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$ and $\widehat{\mathcal{P}} := (\mathcal{C}, \mathcal{X}, \widehat{\mathcal{Z}}, \hat{\prec})$. We observe that $\mathcal{Z}$ and $\widehat{\mathcal{Z}}$ have the same cardinalities. Since $\mathcal{P}$ and $\widehat{\mathcal{P}}$ have the same constraints they have the same set of solutions.

Lemma 2. Let S be a solution of $(\mathcal{C}, \mathcal{X})$ and let π, ρ be two permutations. If $G(z_{\pi_i}) = G(z_{\rho_i})$ for all i then $V_S(\theta^S(\pi(\mathcal{Z}))) = V_S(\theta^S(\rho(\mathcal{Z})))$.

Proof is straightforward from the definition of θ^S.

Lemma 3. If $\hat{\pi}$ is a permutation respecting $\hat{\prec}_{\hat{\mathcal{Z}}}$ then $\hat{\pi}$ respects $\prec_{\mathcal{Z}}$.

Proof. Assume that $z_{\hat{\pi}_i} \prec_{\mathcal{Z}} z_{\hat{\pi}_j}$ and $j \geqslant i$ for some i, j. We then know that $G(z_{\hat{\pi}_i}) \neq G(z_{\hat{\pi}_j})$ and that $G(z_{\hat{\pi}_i}) = G(\hat{z}_{\hat{\pi}_i})$ is preferred to $G(z_{\hat{\pi}_j}) = G(\hat{z}_{\hat{\pi}_j})$. Hence, there is a preference $\hat{z}_{\hat{\pi}_i} \prec_{\hat{\mathcal{Z}}} \hat{z}_{\hat{\pi}_j}$. Since $\hat{\pi}$ satisfies these preferences, we get $i < j$, which is a contradiction. □

Lemma 4. If π is a permutation respecting $\prec_{\mathcal{Z}}$ then there exists a permutation $\hat{\pi}$ respecting $\hat{\prec}_{\hat{\mathcal{Z}}}$ such that $G(z_{\pi_i}) = G(z_{\hat{\pi}_i})$ for all i.

Proof. We define $\hat{\pi}$ such that

1. π and $\hat{\pi}$ agree on the groups: $G(z_{\pi_i}) = G(z_{\hat{\pi}_i})$,
2. if $G(z_{\hat{\pi}_i}) \neq G(z_{\hat{\pi}_j})$ and $\hat{z}_{\hat{\pi}_i} \hat{\prec}_{\hat{\mathcal{Z}}} \hat{z}_{\hat{\pi}_j}$ then $i < j$.

We now show that $\hat{\pi}$ respects $\hat{\prec}_{\hat{\mathcal{Z}}}$. Assume that $\hat{z}_{\hat{\pi}_i} \hat{\prec}_{\hat{\mathcal{Z}}} \hat{z}_{\hat{\pi}_j}$, but $j \geqslant i$. We then know that $G(z_{\hat{\pi}_i}) \neq G(z_{\hat{\pi}_j})$ and that $G(z_{\hat{\pi}_i}) = G(z_{\pi_i})$ is preferred to $G(z_{\hat{\pi}_j}) = G(z_{\pi_j})$. Hence, there is a preference $z_{\pi_i} \hat{\prec}_{\mathcal{Z}} z_{\pi_j}$. Since π satisfies these preferences, we get $i < j$, which is a contradiction. □

Lemma 5. Let S be a solution of $(\mathcal{C}, \mathcal{X})$. If π respects the preferences $\hat{\precsim}$ then $V_S(\pi(\widehat{\mathcal{Z}})) = V_S(\theta^S(\pi(\mathcal{Z})))$.

Proof. Consider the tuples $(v_S(\hat{z}_{\pi_1}), \ldots, v_S(\hat{z}_{\pi_n}))$ and $(v_S(z_{\pi_{\theta_1^S}}), \ldots, v_S(z_{\pi_{\theta_n^S}}))$ and assume that the lemma is wrong. Let k be the largest index such that $v_S(\hat{z}_{\pi_k}) \neq v_S(z_{\pi_{\theta_k^S}})$. According to property (1) of θ^S, we know that $z_{\pi_{\theta_k^S}}$ and z_{π_k} belong to the same group G. Hence, $\hat{z}_{\pi_k}$ also belongs to G as stated in the definition of $\hat{z}$. The criterion $\hat{z}_{\pi_k}$ is equal to $y_{G,j}$ for some j.

Since π respects the preferences $\hat{\precsim}$, we know that j members of G belong to the tuple $(v_S(\hat{z}_{\pi_k}), \ldots, v_S(\hat{z}_{\pi_n}))$. Since $v_S(\hat{z}_{\pi_i})$ and $v_S(z_{\pi_{\theta_i^S}})$ belong to the same groups, we also know that j members of G belong to the tuple $(v_S(z_{\pi_{\theta_k^S}}), \ldots, v_S(z_{\pi_{\theta_n^S}}))$. Let X be the set of these latter members. From property (2) of θ^S we know that $v_S(z_{\pi_{\theta_k^S}}) \geqslant_D v_S(z)$ for all $z \in X$. Since the cardinality of X is j, we get:

$$\begin{aligned} v_S(y_{G,j}) &= \min\bigl\{v_S\bigl(\max\bigl(X'\bigr)\bigr) \mid X' \subseteq G \text{ s.t. } |X| = j\bigr\} \\ &\leqslant_D v_S\bigl(\max(X)\bigr) \leqslant_D v_S(z_{\pi_{\theta_k^S}}). \end{aligned} \tag{A.4}$$

Since $v_S(y_{G,j}) = v_S(\hat{z}_{\pi_k})$ is different from $v_S(z_{\pi_{\theta_k^S}})$, we deduce that $v_S(y_{G,j}) <_D v_S(z_{\pi_{\theta_k^S}})$. According to the definition of $y_{G,j}$, there must exist a subset Y of G with cardinality j such that

$$v_S(y_{G,j}) = v_S\bigl(\max(Y)\bigr). \tag{A.5}$$

Obviously, Y cannot be equal to X. Since they have the same cardinality, there exists an element $z_{\pi_{\theta_t^S}}$ that is in Y, but not in X. Since $z_{\pi_{\theta_t^S}}$ belongs to the group G and it is not in X, it must belong to the tuple $(v_S(z_{\pi_{\theta_1^S}}), \ldots, v_S(z_{\pi_{\theta_{k-1}^S}}))$, meaning that $t < k$. Since $v_S(z_{\pi_{\theta_t^S}}) \leqslant_D v_S(\max(Y)) = v_S(y_{G,j})$ we get

$$v_S(z_{\pi_{\theta_t^S}}) <_D v_S(z_{\pi_{\theta_k^S}}). \tag{A.6}$$

Property (2) of θ^S implies $t > k$ in this case, which is a contradiction. □

These lemmas then allow us to transform the conditions on π and S^* as stated in the definition of B-preferred solutions into the analogue conditions of the definition of E-preferred solutions.

Proof of theorem 3. We first proof the properties on which MCPBS is based:

Lemma 6. If $U = \emptyset$ and $Q = \emptyset$ then each solution of $(\mathcal{C}, \mathcal{X})$ is a B-preferred solution of σ.

Proof. Let S be a solution of $(\mathcal{C}, \mathcal{X})$. Since the problem $\mathcal{P}_\sigma$ has no criteria in U, each of its solutions, including S, is B-preferred. Since Q is empty, S violates all elements of Q. Hence, S is a B-preferred solution of $\sigma := (\mathcal{C}, Q, U)$. □

Lemma 7. Let $z \in U$ and q be any value. S is a B-preferred solution of $(\mathcal{C}, Q \cup \{\text{rank}_{\prec_z}(z) = q\}, U)$ iff S is a B-preferred solution of $(\mathcal{C}, Q, U)$ and S violates $\text{rank}_{\prec_z}(z) = q$.

Proof. This proof is obvious. We simply reformulate the condition that S violates each $\text{rank}_{\prec'_z}(z') = q'$ in $Q \cup \{\text{rank}_{\prec_z}(z) = q\}$ by two conditions, namely that S violates each $\text{rank}_{\prec'_z}(z') = q'$ in Q and that S violates $\text{rank}_{\prec_z}(z) = q$. □

Lemma 8. Let $z \in B_\sigma$ and v be a value such that $\text{rank}_{\prec_z}(v) = \text{rank}_\sigma(z)$. S is a B-preferred solution of $(\mathcal{C} \cup \{z = v\}, Q, U - \{z\})$ iff S is a B-preferred solution of $(\mathcal{C}, Q, U)$ and S satisfies $z = v$.

Proof. We consider a permutation π s.t. $z_{\pi_1} = z$. Such a permutation exists since z is in B_σ meaning that it is a $\prec_{\mathcal{Z}}$-best element of U.

Let S be a B-preferred solution of $(\mathcal{C} \cup \{z = v\}, Q, U - \{z\})$. Then $v_S(z) = v$ and the rank of $v_S(z)$ is equal to $\text{rank}_\sigma(z)$. Now assume that S is not a B-preferred solution of σ. Hence, there exists a solution S^* of $(\mathcal{C}, \mathcal{X})$ such that

$$\big(v_{S^*}(z_{\pi_1}), \ldots, v_{S^*}(z_{\pi_n})\big) \prec^{\pi}_{\text{lex}} \big(v_S(z_{\pi_1}), \ldots, v_S(z_{\pi_n})\big). \tag{A.7}$$

Hence, $v_{S^*}(z_{\pi_1}) \preceq_z v_S(z_{\pi_1})$. If S^* were strictly better than S on z_{π_1} then the rank of $v_{S^*}(z)$ would be strictly smaller than the rank of $v_S(z)$. Since $v_S(z)$ has the best rank of all solutions of $(\mathcal{C}, \mathcal{X})$, this cannot be the case. Hence, we conclude that S and S^* agree on z_{π_1}. As a consequence, S^* satisfies $z = v$ and is a solution of $(\mathcal{C} \cup \{z = v\}, \mathcal{X})$. Since $v_{S^*}(z) = v_S(z)$, property (A.7) simplifies to

$$\big(v_{S^*}(z_{\pi_2}), \ldots, v_{S^*}(z_{\pi_n})\big) \prec^{\pi}_{\text{lex}} \big(v_S(z_{\pi_2}), \ldots, v_S(z_{\pi_n})\big). \tag{A.8}$$

Hence, S is not a B-preferred solution of $(\mathcal{C} \cup \{z = v\}, \mathcal{X}, U - \{z\}, \prec)$, which is a contradiction. Therefore, S is a B-preferred solution of σ.

Now let S be a B-preferred solution of $(\mathcal{C}, Q, U)$ that satisfies $z = v$. Hence, $v_S(z) = v$ and S is a solution of $(\mathcal{C} \cup \{z = v\}, \mathcal{X})$. Assume that S is not a B-preferred solution of $(\mathcal{C} \cup \{z = v\}, \mathcal{X}, U - \{z\}, \prec)$. Hence, there exists a solution S^* of this problem such that

$$\big(v_{S^*}(z_{\pi_2}), \ldots, v_{S^*}(z_{\pi_n})\big) \prec^{\pi}_{\text{lex}} \big(v_S(z_{\pi_2}), \ldots, v_S(z_{\pi_n})\big). \tag{A.9}$$

Since S^* satisfies $z = v$, we obtain $v_{S^*}(z) = v = v_S(z)$. Hence

$$\big(v_{S^*}(z_{\pi_1}), \ldots, v_{S^*}(z_{\pi_n})\big) \prec^{\pi}_{\text{lex}} \big(v_S(z_{\pi_1}), \ldots, v_S(z_{\pi_n})\big) \tag{A.10}$$

which means that S is not a B-preferred solution of σ, which contradicts our supposition. □

Lemma 9. Suppose $(\text{rank}_{\prec_z}(z) = q) \in Q$ and $q < \text{rank}_\sigma(z)$. Then S is a B-preferred solution of $(\mathcal{C}, Q, U)$ iff S is a B-preferred solution of $(\mathcal{C}, Q - \{\text{rank}_{\prec_z}(z) = q\}, U)$.

Proof. Let S be a B-preferred solution of $(\mathcal{C}, Q, U)$. Since S violates each element of Q it is also violates each element of $Q - \{\text{rank}_{\prec_z}(z) = q\}$. Hence, we immediately see that S is a B-preferred solution of $(\mathcal{C}, Q - \{\text{rank}_{\prec_z}(z) = q\}, U)$.

Now let S be a B-preferred solution of $(\mathcal{C}, Q - \{\text{rank}_{\prec_z}(z) = q\}, U)$. Assume that S satisfies $\text{rank}_{\prec_z}(z) = q$. This means that the rank of $v_S(z)$ is equal to q and thus strictly better than $\text{rank}_\sigma(z)$. Since S is a solution of $(\mathcal{C}, \mathcal{X})$ this contradicts the fact that $\text{rank}_\sigma(z)$ is the best rank for z in all solutions of $(\mathcal{C}, \mathcal{X})$. Hence, S violates $\text{rank}_{\prec_z}(z) = q$ and is a B-preferred solution of σ. $\square$

Lemma 10. If $U \neq \emptyset$ and for all $z \in B_\sigma$ we have $(\text{rank}_{\prec_z}(z) = \text{rank}_\sigma(z)) \in Q$ then $(\mathcal{C}, Q, U)$ has no B-preferred solution.

Proof. Suppose that $(\mathcal{C}, Q, U)$ has a B-preferred solution S. Hence, there exists a permutation π such that 1. π respects $\prec_{\mathcal{Z}}$ and 2. there is no other solution S^* of $\mathcal{P}_\sigma$ such that $V_{S^*}(\pi(\mathcal{Z})) \prec_{\text{lex}}^{\pi} V_S(\pi(\mathcal{Z}))$.

Now consider z_{π_1}. Since $\text{rank}_\sigma(z_{\pi_1})$ is the best rank for z_{π_1}, we have $\text{rank}_\sigma(z_{\pi_1}) \leqslant \text{rank}_{\prec_z}(v_S(z_{\pi_1}))$. Consider a solution S^* with $\text{rank}_{\prec_z}(v_{S^*}(z_{\pi_1})) = \text{rank}_\sigma(z_{\pi_1})$. Suppose the rank of $v_{S^*}(z_{\pi_1})$ is strictly smaller than the rank of $v_S(z_{\pi_1})$. Then S^* is better than S on z_{π_1} and we get $V_{S^*}(\pi(\mathcal{Z})) \prec_{\text{lex}}^{\pi} V_S(\pi(\mathcal{Z}))$, which contradicts the fact that S is B-preferred. Furthermore, the rank of $v_{S^*}(z_{\pi_1})$ cannot be strictly greater than the rank of $v_S(z_{\pi_1})$ since it is equal to the best rank $\text{rank}_\sigma(z_{\pi_1})$ of a solution. Therefore, the rank of $v_S(z_{\pi_1})$ is equal to this best rank as well:

$$\text{rank}_{\prec_z}\big(v_{S^*}(z_{\pi_1})\big) = \text{rank}_\sigma(z_{\pi_1}). \tag{A.11}$$

Since π respects the preferences, there is no $z_{\pi_j} \in U$ such that $z_{\pi_j} \prec z_{\pi_1}$. Hence, z_{π_1} is an element of B_σ. Due to the prerequisites of the lemma, $(\text{rank}_{\prec_z}(z) = \text{rank}_\sigma(z))$ is a refutation query in Q. Hence, S violates this query, meaning that the rank of $v_S(z_{\pi_1})$ cannot be equal to $\text{rank}_\sigma(z)$), which contradicts (A.11). Hence, S is not a B-preferred solution of σ. $\square$

These properties permit us to show that algorithm 7 is correct. In each outer loop iteration, the algorithm applies one of these rules and maps the initial state σ to a reduced state σ'. If the condition of line 2 is not satisfied then lemma 6 applies. The set B computed in line 4 is equal to B_σ at the beginning of each inner loop iteration (i.e. line 7): if an element of B is removed from U by an inner loop iteration then the inner loop is stopped and B is updated. In all other cases, the inner loop iteration either fails or modifies Q, which does not require an update of B. If the condition of line 10 is true then lemma 9 is valid. Line 13 checks whether z has a single possible value. In this case, each B-preferred solution of σ will assign this value to z and we can apply lemma 8. A preferred solution of σ either satisfies the constraint $\text{rank}_{\prec_z}(z) = m$ or violates it:

1. In the first case, each B-preferred solution of σ assigns a value v of rank m to z since m is the best rank and can thus reduce the problem according to lemma 8. Since MCPBS does not know which of these best values is equal to the value of z it tries them all in different branches.

2. In the second case, lemma 7 is applied to reduce the problem as described in line 17.

If the inner loop has been terminated without a change of U then all criteria in B have a refutation query $\text{rank}_{\prec_z}(z) = m$ in Q where m is their best rank. Hence, the conditions of lemma 10 are met in line 18 and the algorithm fails.

In each successful iteration of the inner-loop, the algorithm MCPBS either assigns a value to a criterion or puts an assignment $z = m$ to the refutation query meaning that m is no longer possible for z. Since each criterion only has a finite number of values, each nondeterministic run terminates after a polynomial number of loop-iterations.

Proof of theorem 4. Let $\mathcal{P} := (\mathcal{C}, \mathcal{X}, \mathcal{Z}, \prec)$. Let $\prec'_{x_1}, \ldots, \prec'_{x_m}$ be a *projection* of $\prec_{z_1}, \ldots, \prec_{z_n}$ via $f_1, \ldots, f_n$ to $x_1, \ldots, x_m$. Let S be a Pareto-optimal solution of $\mathcal{P}$.

Now consider $\mathcal{P}' := (\mathcal{C}, \mathcal{X}, \mathcal{X}, \prec')$. If S is a Pareto-optimal solution of $\mathcal{P}'$, then the theorem is trivially satisfied.

Otherwise, there exists a solution S^* of $\mathcal{P}'$ such that $v_{S^*}(x_k) \prec'_{x_k} v_S(x_k)$ for a k and $v_{S^*}(x_i) \preceq'_{x_i} v_S(x_i)$ for all i. At least one of these solutions S^* is Pareto-optimal. Otherwise, we would be able construct an infinite descending chain $\ldots, S_t, S_{t-1}, \ldots, S_1$, where $S_1 := S$ and where each S_j is better than S_{j-1} for some variable and at least as good as S_{j-1} for all variables. However, this would imply that the domain of at least one variable is infinite, which has been excluded. Hence, we can suppose that S^* is Pareto-optimal without any loss of generality.

We consider each criterion z_i, which is defined by $f_i(x_1, \ldots, x_m)$. Since $\prec'_{x_1}, \ldots, \prec'_{x_m}$ are *projection* of $\prec_{z_i}$ via f_i and $v_{S^*}(x_j) \preceq'_{x_j} v_S(x_j)$, we obtain

$$\begin{aligned} &f_i\big(v_{S^*}(x_1), \ldots, v_{S^*}(x_{j-1}), v_{S^*}(x_j), v_S(x_{j+1}), \ldots, v_S(x_j)\big) \\ &\quad \preceq_{z_i} f_i\big(v_{S^*}(x_1), \ldots, v_{S^*}(x_{j-1}), v_S(x_j), v_S(x_{j+1}), \ldots, v_S(x_j)\big). \end{aligned} \tag{A.12}$$

Hence, we obtain

$$f_i\big(v_{S^*}(x_1), \ldots, v_{S^*}(x_m)\big) \preceq_{z_i} f_i\big(v_S(x_1), \ldots, v_S(x_m)\big) \tag{A.13}$$

which results in $v_{S^*}(z_i) \preceq_{z_i} v_S(z_i)$ for all criteria z_i. If S^* were better than S on a criterion z_i, then S would not be a Pareto-optimal solution of $\mathcal{P}$. Hence, we obtain $v_{S^*}(z_i) = v_S(z_i)$ for all criteria. □

Notes

1. We use the term extreme in the sense that certain criteria have an absolute priority over other criteria.
2. In constraint programming, any numerical expression that can be formulated in the constraint language can be used as objective.

3. Expressions such as $\min(e_1, e_2)$ and $\max(e_1, e_2)$ on two integer expressions e_1, e_2 are standard in constraint programming tools, which usually maintain bound consistency for these numerical expressions.
4. More complex standardization methods can be formulated by using other arithmetic expressions of the constraint library. We can also represent arbitrary mappings by using the so-called element constraints.
5. For multi-criteria preference-based search.
6. Except for the right branches that fail immediately and that could be avoided by some additional propagation rules.

References

Bacchus, F. and A. Grove. (1995). "Graphical Models for Preference and Utility." In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, Montreal, pp. 3–10.

Behringer, F. (1981). "A Simplex Based Algorithm for the Lexicographically Extended Linear Maxmin Problem." *European Journal of Operational Research* 7, 274–283.

Bistarelli, S., U. Montanari, F. Rossi, T. Schiex, G. Verfaillie, and H. Fargier. (1999). "Semiring-Based CSPs and Valued CSPs: Frameworks, Properties, and Comparison." *Constraints* 4(3), 199–240.

Boutilier, C., R. Brafman, C. Geib, and D. Poole. (1997). "A Constraint-Based Approach to Preference Elicitation and Decision Making." In J. Doyle and R.H. Thomason (eds.), *Working Papers of the AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning*, Menlo Park, CA, pp. 19–28.

Brewka, G. (1989). "Preferred Subtheories: An Extended Logical Framework for Default Reasoning." In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Detroit, MI, pp. 1043–1048. Morgan Kaufmann.

Delgrande, J.P. and T. Schaub. (2000). "Expressing Preferences in Default Logic." *Artificial Intelligence* 123(12), 41–87.

Domshlak, C., R. Brafman, and S.E. Shimony. (2001). "Preference-Based Configuration of Web-Page Content." In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, Seattle, pp. 1451–1456. Morgan Kaufmann.

Doyle, J. and R.H. Thomason. (1999). "Background to Qualitative Decision Theory." *Artificial Intelligence 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*, Egelsbach, pp. 193–202. Häsel-Hohenhausen.

Ehrgott, M. and X. Gandibleux. (2000). "A Survey and Annoted Bibliography of Multiobjective Combinatorial Optimization." *Operations Research Spektrum* 22, 425–460.

Gavanelli, M. (2002). "An Implementation of Pareto Optimality in CLP(FD)." In *CP-AI-OR'02*, Le Croisic, France, pp. 49–63.

Geffner, H. and J. Pearl. (1992). "Conditional Entailment: Bridging Two Approaches to Default Reasoning." *Artificial Intelligence* 53(2–3), 209–244.

Grosof, B.N. (1991). "Generalizing Prioritization." In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, Cambridge, MA, pp. 289–300. Morgan Kaufmann.

ILOG. (2002). "ILOG JConfigurator V2.0." `www.ilog.com/products/jconfigurator`.

Junker, U. (1997). "A Cumulative-Model Semantics for Dynamic Preferences on Assumptions." In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, Nagoya, pp. 162–167. Morgan Kaufmann.

Junker, U. (2000). "Preference-Based Search for Scheduling." In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI)*, Austin, pp. 904–909. AAAI Press.

Junker, U. and D. Mailharro. (2003a). "The logic of ILOG (J)Configurator: Combining Constraint Programming with a Description Logic." In *IJCAI-03 Workshop on Configuration*, Acapulco, pp. 13–20.

Junker, U. and D. Mailharro. (2003b). "Preference Programming: Advanced Problem Solving for Configuration." *AIEDAM* 17(1), 13–29.

Khatib, L., P. Morris, R.A. Morris, and F. Rossi. (2001). "Temporal Constraint Reasoning with Preferences." In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, Seattle, pp. 322–327. Morgan Kaufmann.

van Hentenryck, P., L. Perron, and J.-F. Puget. (2000). "Search and Strategies in OPL." *ACM Transactions on Computational Logic* 1(2), 1–36.

Van Wassenhove, L.N. and L.F. Gelders. (1980). "Solving a Bicriterion Scheduling Problem." *European Journal of Operational Research* 4(1), 42–48.