DECISION ANALYSIS
Vol. 2, No. 2, June 2005, pp. 110–123
ISSN 1545-8490 | EISSN 1545-8504 | 05 | 0202 | 0110

informs®

DOI 10.1287/deca.1050.0043
© 2005 INFORMS

A Preference Programming Approach to Make the Even Swaps Method Even Easier

Jyri Mustajoki, Raimo P. Hämäläinen
Systems Analysis Laboratory, Helsinki University of Technology, P.O. Box 1100, FIN-02015 HUT, Finland
{jyri.mustajoki@hut.fi, raimo@hut.fi}

This paper introduces an approach to support different phases of the even swaps process by preference programming, which is a framework for modeling incomplete information within multiattribute value theory (MAVT). In the approach, the even swaps process is carried out as usual, but in parallel, the evolution of the preferences of the decision maker is modeled with preference programming. With this model, we can provide information to the even swaps process to help identify practically dominated alternatives, and to find applicable candidate attributes for the next even swap. The approach does not compromise the original idea of an easy-to-use process. Our Smart-Swaps web software is used to demonstrate the method in practice.

Key words: multicriteria decision analysis; even swaps; trade-offs; preference programming; decision support systems
History: Received on November 3, 2004. Accepted by Robert Clemen and Don Kleinmuntz on September 17, 2005, after 2 revisions.

1. Introduction

Multicriteria decision analysis (MCDA) is a structured way to analyze problems with multiple criteria and alternatives. Even swaps (Hammond et al. 1998, 1999) is an MCDA method based on value trade-offs (see e.g., Keeney and Raiffa 1976, Keeney 2002), which are called *even swaps*. In an even swap, the consequence of an alternative in one attribute is changed and this change is compensated with a preferentially equal consequence change in some other attribute. The new alternative with these revised consequences is equally preferred to the initial one, and thus it can be used instead. The aim of the method is to carry out even swaps that either make attributes irrelevant, in the sense that all the alternatives have equal consequences on this attribute, or create dominated alternatives, in the sense that some other alternative is at least as good as this alternative on every attribute. Such attributes and alternatives can be eliminated, and the process continues until one alternative, the most preferred one, remains. One can also eliminate practically dominated alternatives (Hammond et al. 1998) during the process: An alternative **x** practically dominates alternative **y**, if **y** is slightly better than **x** in only one or few attributes, but **x** clearly outranks **y** in several other attributes. The aim of practical dominance is to reduce the problem in obvious cases, and thus eliminate unnecessary even swap tasks.

Interval modeling provides a convenient way to account for preferential uncertainty or imprecision in MCDA. It has been applied in various methods. One of the first ones was Alternative Ranking Interactive Aid based on DomiNance structural information Elicitation (ARIADNE) (Sage and White 1984, White et al. 1984), which allows uncertain judgments through direct constraints on values and weights. Kirkwood and Sarin (1985) developed an algorithm for obtaining a partial rank order of alternatives under partial information about attribute weights, and Kirkwood and Corner (1993) studied the effectiveness of partial information about the weights. Hazen (1986) and Weber (1987) further developed the concepts of dominance and potential optimality for analyzing the results obtained with incomplete information. Preference programming (Arbel 1989; Salo and Hämäläinen 1995, 2004) is a general term for a family of methods using intervals to model uncertainties or incomplete information within MAVT. Preference Assessment by Imprecise Ratio Statements (PAIRS) (Salo and Hämäläinen 1992) is a preference programming method, in which the attributes are compared in pairs, and the alternatives are evaluated, for example, by using value functions.

In this paper, we introduce an approach in which preference programming is applied to support the even swaps process. In our approach, the decision maker (DM) carries out the even swaps process as usual, but in parallel, the evolution of the preferences of the DM is modeled with the PAIRS method. This model is initialized with some weak assumptions about the DM's preferences and updated during the process according to the trade-off information given in even swaps. The additional preference information obtained with this model is used in the even swaps process to produce (1) suggestions for alternatives that could be practically dominated and (2) suitable candidates for attributes on which to carry out the next even swap.

The aim of the proposed approach is to combine the benefits of both the even swaps method and MAVT modeling. The main advantages of the even swaps are that it is simple to use and it does not require making any assumptions about the form of the value function (see e.g., Butler et al. 2001). Thus, the method is also well suited for DMs not having a mathematical background. However, the preference information given in an even swap is only used to change the representation of the alternatives in the comparisons, but not to elicit the DM's preferences over the attributes in general. Consequently, as a result of the process, one only gets the dominance relations between the alternatives. In this respect, MAVT provides a structured process to compose and represent the relative overall scores of the alternatives by means of the weights of the attributes and the ratings of the alternatives with respect to each attribute. A disadvantage of MAVT is that it requires assumptions about the form of a value function. However, in our approach, these assumptions do not restrict the generality of the even swaps process, as the PAIRS model is only operated in the background to produce suggestions to help the process, and the DM eventually decides whether or not to follow these suggestions.

In practice, the implementation of the proposed approach requires computer support to manage and update the PAIRS model during the process. However, the even swaps process benefits from computer support, in general, in managing the consequences table, visualizing the process, keeping the log and tracking the actions taken. In this paper, we will briefly describe the web-based Smart-Swaps software (Hämäläinen et al. 2003), which provides support for all of these tasks. Smart-Swaps is available for free for academic purposes at www.smart-swaps.hut.fi and it is part of the Decisionarium website for decision support (Hämäläinen 2000, 2003).

We expect that the proposed support is most useful in large problems, in which finding practical dominance relations and suitable swaps could otherwise be quite difficult. We also see that the approach should be applied together with practical advice proposed by Hammond et al. (1998, 1999). They suggest, for example, to first make swaps on those attributes that are easy to deal with (such as money), which can usually be easily identified with by the DM. However, to reduce the size of the consequences table, the DM should also make swaps that result in dominated alternatives or irrelevant attributes. The DM can use the approach proposed here to detect these swaps, and then decide whether to make one of the suggested swaps or would she prefer a swap on some easily measurable attribute.

The proposed approach is well suited both for the individual use of DMs and for supported use by decision analysts. In individual use, the DM does not have to study the details of the method, as suggestions can be produced by using the default initialization parameters in the model. Then, the supporting system should, however, provide some help to understand the suggestions. For example, the Smart-Swaps software provides explanations for the DM as to why the given suggestions are worth considering. Yet, an experienced user, such as a decision analyst can customize the approach, as described later.

As far as the authors know, the only reported applications of even swaps in the literature are the ones on strategy selection in a rural enterprise (Kajanus et al. 2001) and on environmental planning (Gregory and Wellman 2001). This indicates lack of use, which can be partly due to the fact that despite the simplicity of the method, there is still a need for computational help in the screening of the consequences, especially in large problems. So far there has not been any software for this before the introduction of the Smart-Swaps software.

This paper is organized as follows. Section 2 presents the framework for the use of the proposed approach and the preference programming model. Section 3 describes how to identify practical dominance relations, and §4 describes how to identify applicable candidates for the next even swap with this model. The Smart-Swaps software is introduced in §5. Section 6 demonstrates the use of the proposed approach with an illustrative example, and §7 compares the approach with MAVT. Section 8 concludes this paper.

2. Even Swaps with Preference Programming

The framework for the use of the proposed approach is illustrated in Figure 1. The idea is that each time the DM makes changes in the consequences table during the even swaps process, this information is updated into the underlying preference programming model. This model is then applied to provide suggestions to the DM about the next even swap and candidates for the alternatives to be eliminated as practically dominated.

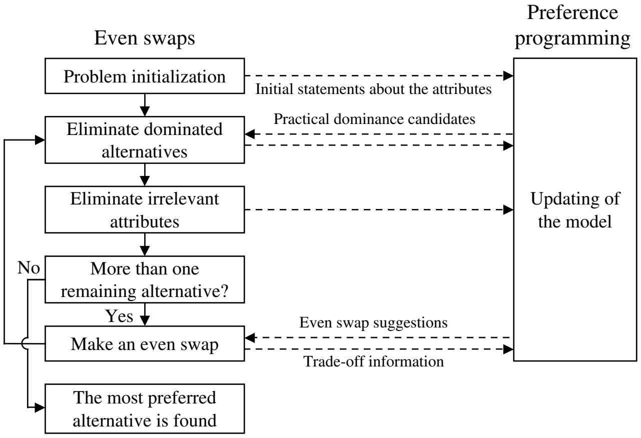
2.1. The Preference Programming Model

The preference programming model builds on two assumptions: (1) the DM's preferences can be modeled with an additive MAVT model and (2) the DM is able to provide some general initial information about her preferences. An additive model can be used if the attributes are mutually preferentially independent (see e.g., Keeney and Raiffa 1976). Then, the overall score of an alternative described by the consequence vector $\mathbf{x} = (x_1, \ldots, x_n)$ is

$$v(\mathbf{x}) = \sum_{i=1}^{n} w_i v_i(x_i), \tag{1}$$

where n is the number of attributes, x_i is the consequence of this alternative with respect to attribute i, $v_i(x_i)$ is its rating on $[0, 1]$ scale, and w_i is the weight of attribute i representing the relative importance of this attribute. The weights are normalized to sum up to 1. Additivity is not assumed in even swaps, but, in general, the assumption of an additive value model is not considered to be very restrictive (see e.g., Keeney and Raiffa 1976, Stewart 1996). The additive model is also the most commonly used form in practice (see e.g., Belton and Stewart 2002).

Figure 1 **Use of Preference Programming to Support the Even Swaps Process**


Even swaps
Preference programming
Problem initialization
Initial statements about the attributes
Eliminate dominated alternatives
Practical dominance candidates
Eliminate irrelevant attributes
Updating of the model
No
More than one remaining alternative?
Yes
Make an even swap
Even swap suggestions
Trade-off information
The most preferred alternative is found

Note. The dotted lines represent the information exchange between these.

According to the basic principles of MAVT, each attribute in the model should have some importance to the DM (see e.g., Keeney 1992). The practical dominance concept also implicitly assumes that even swaps are reasonable in terms of the allowed compensations. Thus, it is justifiable to assume that the ratings and the ratios of the weights are not completely unrestricted. This allows us to initialize the PAIRS model by first assigning general interval bounds for the ratings of the alternatives and for the weight ratios between the attributes so that every rating and weight ratio is assumed to be within these bounds. For details of using PAIRS to model imprecision, see Salo and Hämäläinen (1992), and for details of assigning the bounds in our model, see §§3.1–3.3.

We emphasize that the proposed approach can be used even if the above assumptions are not fulfilled, as the implications of the preference programming model are only treated as suggestions for how to proceed with the even swaps process. The DM is explicitly asked to approve the suggestions, and thus any operations based on too strict assumptions will not be made automatically.

For brevity, we will use the term "alternative $\mathbf{x}$" throughout this paper to denote "the alternative described by the consequence vector $\mathbf{x}$." We shall use the notation $\mathbf{x}$: (i: $x_i \to x_i'$, j: $x_j \to x_j'$) for an even swap, where the way alternative $\mathbf{x}$ is represented is altered by changing attribute i from x_i to x_i', which is compensated for with a change in attribute j from x_j to x_j'.

3. Practical Dominance

In models with incomplete information, the *pairwise dominance* concept is used to analyze relations between the alternatives (see e.g., Hazen 1986, Weber 1987, Salo and Hämäläinen 1992). Alternative **x** dominates alternative **y** in a pairwise sense if the overall score of **x** is at least as high as the overall score of **y** with every feasible combination of the weights and the ratings; that is, if

$$\min_{w\in S}\sum_{i=1}^{n} w_i[\underline{v}_i(x_i)-\bar{v}_i(y_i)]\geq 0, \tag{2}$$

where S is the feasible region of the weights, $\underline{v}_i(x_i)$ and $\bar{v}_i(y_i)$ are the lower and upper bounds for $v_i(x_i)$ and $v_i(y_i)$, respectively, and the inequality is strict at least for some $w=(w_1,\ldots,w_n)\in S$.

When the DM eliminates an alternative using practical dominance during the even swaps process, she implicitly assumes that there are no feasible swaps with which this alternative can be made the most preferred one. With this interpretation, any alternative that is dominated in a pairwise sense in our PAIRS model can also be seen as practically dominated. That is, in the model, there are bounds for the weight ratios and the ratings that define the maximum allowed imprecision in these. These bounds implicitly set bounds for the feasible swaps too. Thus, if an alternative is dominated in a pairwise sense within these bounds, it cannot be made the most preferred one with any feasible swaps, and thus it can be seen as practically dominated.

Assumption 1. *Assume that the DM's preferences are modeled with preference programming as described in §2. Then, an alternative is practically dominated if it is dominated in a pairwise sense.*

3.1. Initial Bounds

The use of pairwise dominance to represent practical dominance requires setting initial general bounds for the weight ratios and for the ratings of the alternatives. We assume (1) a general upper bound for the weight ratios, (2) exponential value function bounds that define the lower and upper bounds for the ratings on each attribute, and (3) general bounds for the rating differences (i.e., for the slope of the feasible value functions). The motivation for using these types of bounds is that weight ratios and exponential value functions are conceptually easy to understand and they are commonly used.

The bounds (1) and (3) can be given directly, but on (2), one gives a general input parameter that defines the curvatures of the exponential functions from which the bounds for the ratings are derived. The same general bounds apply for each weight ratio and rating interval in the model. Thus, the model can be initialized quickly and easily. Naturally, one could assign separate bounds on each attribute and weight ratio, but this would mean a thorough interval-based MAVT assessment of the model.

Setting a general upper bound $r\geq 1$ for the weight ratios leads to a set of constraints

$$\frac{w_i}{w_j}\leq r,\quad \forall\, i,j=1,\ldots,n,\ i\neq j. \tag{3}$$

These constrain the feasible region of the weights S. An example of the feasible region on the weight plane $\sum_{i=1}^{n} w_i=1$ is illustrated in Figure 2.

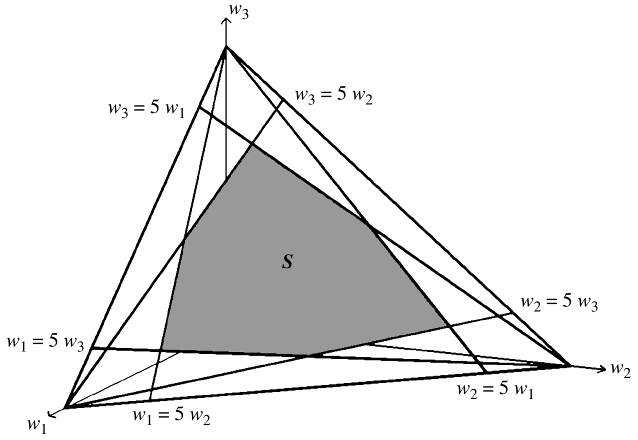
The upper and lower bounds for the ratings $v_i(x_i)$ are derived from the exponential value functions

$$\bar{v}_i(x_i)=\frac{a^{x_i^N}-1}{a-1}\quad\text{and} \tag{4}$$

$$\underline{v}_i(x_i)=\frac{(1/a)^{x_i^N}-1}{(1/a)-1}, \tag{5}$$

respectively, where $x_i^N=(x_i-x_i^0)/(x_i^*-x_i^0)$ is the value of x_i standardized onto range $[0,1]$, and x_i^0 and x_i^*

Figure 2 Feasible Region of the Weights S in the Case of Three Attributes ($r=5$)



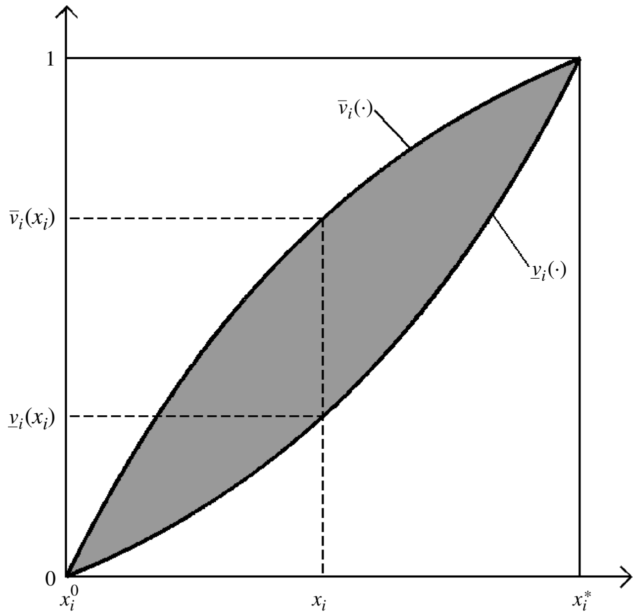


Figure 3 Bounds for the Ratings Derived from Exponential Functions ($a = 0.2$)

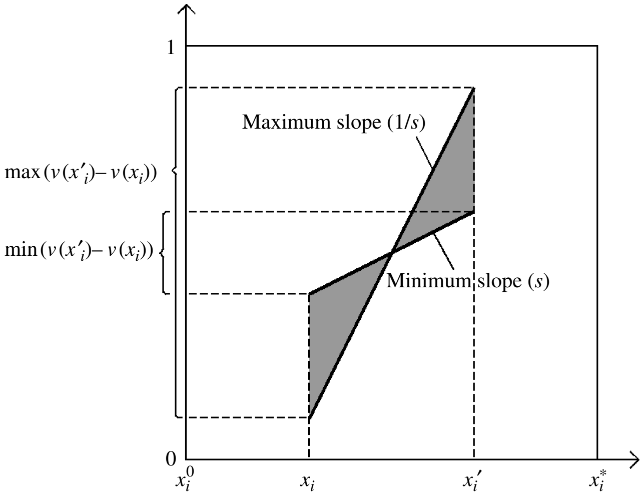


Figure 4 Bounds for the Slope ($s = 0.5$)

stand for the minimum and the maximum values of x_i, respectively, in the initial set of the alternatives. Figure 3 illustrates how to derive the bounds for the rating of consequence x_i from these functions. In a case of a predefined discrete consequence scale, x_i^0 and x_i^* stand for the end points of the scale, and the set of possible consequences is evenly distributed on this scale. Input parameter $a \in (0, 1)$ defines the curvature of the functions, so that the closer it is to 1, the closer the functions are to a linear function. In this paper, we assume that the functions are increasing, but analogous bounds can be derived for decreasing functions. One should also note that these bounds do not explicitly define the shapes of feasible value functions, but any shapes (e.g., linear, exponential, as well as S shaped) within these bounds are allowed.

The general lower bound $s \in [0, 1]$ for the slope of the function leads to constraints

$$\frac{v_i(x'_i) - v_i(x_i)}{\Delta x_i^N} \geq s, \quad \forall\, i = 1, \ldots, n \quad \text{and} \quad \forall\, x'_i, x_i, \qquad (6)$$

where $\Delta x_i^N = x_i'^N - x_i^N = (x'_i - x_i)/(x_i^* - x_i^0)$ is the difference between consequences x'_i and x_i standardized onto range $[0, 1]$ (Figure 4). That is, any feasible component value function has the slope of at least s throughout the function. In practice, it is reasonable to assume that $s \geq 0$ (i.e., the functions are monotonic). Monotonicity is not explicitly assumed in the even swaps method, but in practice the consequences of the attributes are often either maximized or minimized, when the corresponding value functions are monotonic. Yet, in our approach, a nonmonotonic value function can be modeled by using a discrete scale to set a preference order for a set of consequences. An upper bound for the slope can be set similarly as the lower bound, for example, by setting it as $1/s$ ($s \neq 0$) (see Figure 4).

3.2. Updating the Model with the Information Obtained from the Given Even Swaps

Additivity of the proposed model implies that the consequence changes made in an even swap are not assumed to depend on the consequence levels of the other attributes. This allows us to use the trade-off information given in an even swap to represent the DM's general preferences over the attributes. Consequently, we can elicit new bounds for the weight ratios to make the model more precise. That is, by making an even swap **x**: (i: $x_i \to x'_i$, j: $x_j \to x'_j$), the additivity implies that $w_i v_i(x_i) + w_j v_j(x_j) = w_i v_i(x'_i) + w_j v_j(x'_j) \Leftrightarrow w_i(v_i(x'_i) - v_i(x_i)) = w_j(v_j(x_j) - v_j(x'_j))$. As the ratings, and consequently the rating differences between the attributes, are assumed to be within some bounds, we can obtain a new additional upper bound for the weight ratio w_i/w_j as the maximum feasible ratio of these differences:

$$\frac{w_i}{w_j} \leq \max\left(\frac{v_j(x_j) - v_j(x'_j)}{v_i(x'_i) - v_i(x_i)}\right). \tag{7}$$

A corresponding upper bound can be obtained for the weight ratio w_j/w_i.

3.3. Remarks on the Bounds

The bounds for the weight ratios in (3) and (7) define the maximum relative differences between the importances of the attributes, which should always be considered with respect to the ranges used. In our approach, the end points of the ranges in (4)–(6) are elicited as the minimum and the maximum of the attributes' consequences in the initial set of the alternatives. Thus, the model assumes that the variations in the attributes' values over their ranges impact on the preference for the alternatives. One should also note that the current minimum and maximum consequences of the attributes may change during the even swaps process because of changes of attribute values in even swaps, or because of the elimination of alternatives. Thus, to keep the interpretation of the weights the same, the initially defined $[x_i^0, x_i^*]$ ranges have to be used throughout the modeling process.

The consequences of the alternatives may also exceed the initially defined $[x_i^0, x_i^*]$ ranges when making an even swap. The proposed bounds can also be used outside this range. However, then the exponential lower and upper bounds for the ratings have to swap places with each other (i.e., (4) has to be set as the lower bound and (5) as the upper bound), as they cross with each other at the points x_i^0 and x_i^*.

3.4. Computational Considerations

Computationally, one has to go through $m \times (m-1)/2$ alternative pairs (in which m is the number of the alternatives) to check all the possible practical dominance relations between the alternatives. The bounds for the rating differences can be obtained directly from the value function constraints (4)–(6), but solving the optimization problems (2) requires linear programming (LP) methods.

In large problems, standard LP methods may appear to be too slow to solve the optimization problems. However, the characteristics of the model make it possible to speed up the calculation process with certain techniques. For example, if alternative **x** dominates alternative **y**, it is not necessary to check whether **y** dominates **x**. Also, when using iterative techniques (e.g., the simplex method), the calculation process can be interrupted once a negative value for (2) is found, as this implies that the minimum is also negative. The dominance relations can also be checked in some specific order, for example, in the order based on the value difference between the alternatives at some specific point of the feasible region. Then, the dominated alternatives are more likely to be found in earlier phases of the checking process, when one does not have to check the other possible dominance relations on these alternatives.

Especially, if the additional constraints (7) derived from the given even swaps are not used, one can substantially speed up calculations by using Theorem 1 to check the possible pairwise dominance relations.

Theorem 1. *If the feasible region of the weights is bounded by weight ratio constraints $w_i/w_j \leq r$ that have the same upper bound $r \geq 1$ for $\forall\, i, j = 1, \ldots, n,\ i \neq j$, then alternative* **x** *dominates alternative* **y** *in a pairwise sense if*

$$\sum_{i=1}^{n} c_i[\underline{v}_i(x_i) - \bar{v}_i(y_i)] > 0,$$

$$\textit{where} \quad \begin{cases} c_i = 1 & \textit{if } [\underline{v}_i(x_i) - \bar{v}_i(y_i)] \geq 0, \\ c_i = r & \textit{if } [\underline{v}_i(x_i) - \bar{v}_i(y_i)] < 0. \end{cases} \tag{8}$$

Proof. See the appendix.

Theorem 1 is based on the symmetry of the feasible region, as the constraints (3) are all of the same form. It does not require solving optimization problems. Even if the additional constraints (7) are used, Theorem 1 can be used as a preliminary check for pairwise dominance. That is, if pairwise dominance was reached without using the constraints (7), it would also be reached with these, as then the feasible region is the subset of the initial one.

4. Suggestions for the Next Even Swap

In a typical even swaps process, the DM has to make several swaps until the most preferred alternative is found. Usually, there are several possibilities for an even swap to be carried out next, and different sequences of swaps can substantially differ from each other with respect to how quickly the most preferred alternative is found. In this section, we discuss how to help the DM select applicable swaps to make the process efficient in the sense of making alternatives dominated and attributes irrelevant with as few swaps as possible. However, as mentioned earlier, there are also other ways to measure the quality of the swaps, for example, with respect to their easiness. We emphasize that the best use of the proposed aids is obtained by using them with common sense by also addressing the other considerations when selecting the next swap.

In our approach, the primary candidates for the next swap are those swaps with which some alternative could become dominated or some attribute irrelevant. If there are no such swaps, we next consider the swaps with which some alternative (or attribute) could be made to be one swap apart from being dominated (or irrelevant), and so on. However, usually there are several swaps that are equal with respect to how few swaps apart from obtaining dominance relations or irrelevant attributes we can reach with this swap. On the swaps leading to irrelevant attributes, we can next study with which of these swaps we could also make alternatives dominated. However, on the swaps that may lead to dominance relations, we cannot be sure that an alternative will actually become dominated with a particular swap, as we do not know the precise preferences of the DM, and consequently, the amount of the consequence change in the compensating attribute. For example, if alternative **y** outranks alternative **x** only in one attribute i, we can try to make **y** dominated with a swap $\mathbf{x}$: $(i\colon x_i \to y_i,\ j\colon x_j \to x'_j)$. However, the attribute-wise ranking of the alternatives in attribute j may change because of this swap, in which case, alternative **y** will not become dominated.

4.1. Applicability of an Even Swap to Make an Alternative Dominated

Preference programming provides a way to evaluate the applicability of the swaps in terms of how close to making an alternative dominated one would get with each swap. We propose an *applicability index* that indicates how large a consequence change the DM is allowed to make in the compensating attribute to reach dominance, in proportion to the estimated consequence change that would be made in this attribute.

Definition 1. Assume that the DM's preferences are modeled with preference programming as described in §§2 and 3, and alternative **y** outranks alternative **x** only on attribute i. Then, the applicability index of an even swap $\mathbf{x}$: $(i\colon x_i \to y_i,\ j\colon x_j \to x'_j)$ to make alternative **x** dominate alternative **y** is

$$d(\mathbf{x}\to\mathbf{y}, i, j) = \min\left(\frac{v_j(x_j) - v_j(y_j)}{v_j(x_j) - v_j(x'_j)}\right)$$

$$= \min\left(\frac{v_j(x_j) - v_j(y_j)}{(w_i/w_j)[v_i(y_i) - v_i(x_i)]}\right), \quad (9)$$

where the compensation x'_j in attribute j is unknown, and the extremes for the weight ratio w_i/w_j are based on (3) and (7) and the extremes for the rating differences on (4)–(6).

Thus, $d(\mathbf{x}\to\mathbf{y}, i, j)$ (or d hereafter) is the ratio between the minimum feasible rating difference $(v_j(x_j) - v_j(y_j))$ in attribute j with which the dominance would still be reached, and the maximum feasible rating change $(v_j(x_j) - v_j(x'_j))$ that could be made in this attribute. We do not know the compensation x'_j in attribute j, but the difference $v_j(x_j) - v_j(x'_j)$ is estimated from the additivity assumption $w_j(v_j(x_j) - v_j(x'_j)) = w_i(v_i(y_i) - v_i(x_i))$.

The bigger the applicability index d is, the more likely the consequence x'_j stays above y_j (i.e., within the area in which **x** dominates **y**). By default, Definition 1 gives a worst-case value for d, as it is obtained as the minimum of the feasible ratios. Thus, $d > 1$ indicates that any feasible compensation in attribute j makes alternative **x** dominate **y**. However, we can also calculate an average-case value for d by deriving the rating differences in (9) from linear value functions and by estimating the weight ratio w_i/w_j as a geometric average of its lower and upper bounds. This value may often be more useful to the DM and then $d = 1$ indicates that it is estimated to be equally likely to reach or not to reach dominance with this swap. Yet, one should note that even if we apply linear value functions in (9) to model the average of all the possible value functions, the preference programming model is still needed to update the new preference information obtained from the given swaps into the weight ratios.

In Definition 1, we evaluate the applicability index of an even swap $\mathbf{x}$: $(i\colon x_i \to y_i,\ j\colon x_j \to x_j')$. However, dominance can also be reached by making the consequences the same in attribute j and compensating this change in attribute i (i.e., with a swap $\mathbf{x}$: $(j\colon x_j \to y_j,\ i\colon x_i \to x_i')$). Then, the attributewise ranking of the alternatives in attribute i must change to reach dominance. However, in terms of evaluating the applicability of an even swap to reach dominance, it is irrelevant in which way the even swap is made because in an additive model the extents of the rating changes do not depend on the levels of the other attributes. Thus, for the swap $\mathbf{x}$: $(j\colon x_j \to y_j,\ i\colon x_i \to x_i')$, we can use the same value of the applicability index as for the swap $\mathbf{x}$: $(i\colon x_i \to y_i,\ j\colon x_j \to x_j')$. In both cases, an even swap can also be carried out so that the change is made in alternative $\mathbf{y}$, but we can use the same index value also in those cases.

In Definition 1, we assume that alternative $\mathbf{y}$ may be made dominated with one swap, but similar indices can be calculated for swaps with which alternative $\mathbf{y}$ may be made to be one (or more) swaps apart from being dominated in a case where $\mathbf{y}$ outranks $\mathbf{x}$ in two (or more) attributes. However, one should note that then the strategy of selecting the swap with the highest applicability index might not be the best choice if the possible subsequent swaps are also considered. The DM may, for example, prefer a compensation that comes as close as possible to the threshold value y_j—but does not exceed this—so that dominance can be reached with the following swap.

The DM may also make some other alternative than $\mathbf{y}$ (say, alternative $\mathbf{z}$) dominated with a swap $\mathbf{x}$: $(i\colon x_i \to y_i,\ j\colon x_j \to x_j')$. In this case, we can evaluate the applicability index of this swap to also make $\mathbf{z}$ dominated, by replacing the threshold value y_j in (9) by the corresponding consequence z_j.

5. Smart-Swaps Software

The Smart-Swaps (Hämäläinen et al. 2003) software follows the original *Problem, Objectives, Alternatives, Consequences, Trade-offs* (PrOACT) working phases (Hammond et al. 1999). The aim of the software is that the DM can carry out the even swaps process independently with the help provided by the software.

The *Trade-offs* phase is supported by the preference programming approach described in this paper. In practice, before starting to make an even swap, the DM can let the software suggest candidates either for a swap that could lead to irrelevant attributes or for a swap that could lead to dominated alternatives. The suggested swaps are always the first ones in the shortest paths of swaps needed to make an alternative dominated or an attribute irrelevant, and they are suggested in the order based on their applicability index to reach dominance. To keep the task of the DM as simple as possible, the software gives the suggestions one at a time. With each suggestion, the DM is informed of what can be achieved with this swap; that is, which alternatives can possibly become dominated and which attributes can become irrelevant with this swap. This will help the DM understand the process and the logic behind the suggestions. To keep the number of suggestions reasonable, only one swap for a given pair of attributes and alternatives is presented, and there is a limit of six suggestions.

The initial input parameters for setting the bounds in (3)–(6) default to $r = 5$, $a = 0.2$, and $s = 0.5$, and these can be adjusted by the DM. This has an effect on the number of candidates for practical dominance relations shown. For advanced users, the software also provides several other options to customize how the swaps are suggested. One can, for example, define whether to use the average or worst-case values for the applicability indices. One can also choose to show the applicability index, as this can be useful information, especially for a decision analyst.

After each swap, the software automatically identifies the dominated and practically dominated alternatives. The computation of practical dominance relations is carried out with the proposed preference programming approach and the simplex method, and it is speeded up with the techniques described in §3.4. After each swap, the software also checks whether there are any inconsistencies in the weight ratios obtained from the given swaps, as in practice, the bounds derived from the given even swaps may become conflicting. If this is the case, the software informs the DM about the attributes between which there are inconsistencies and suggests checking the consistency of the given swaps on these attributes by backtracking the process. If the DM does not redefine the given swaps, the assumptions of the model are likely to be too strong and the software does not apply the additional constraints in the future suggestions.

6. Example

We illustrate the use of the proposed approach with the Smart-Swaps software by an office selection problem adapted from Punkka and Salo (2004). There are 12 possible locations for the office of a medium-sized company and the decision is based on eight attributes: (1) *size* of the office, (2) *rental costs*, (3) *renovation need*, (4) *car park* opportunities, (5) availability of *public transport*, (6) *distance* to city center, (7) *other facilities* (e.g., restaurants, etc.), and (8) *habitability* of the office and the neighborhood. We have made some minor modifications to the original example, for instance, the attribute values that were originally given as optional (e.g., *renovation need* of Alternative 7 was *small* or *considerable*) have been changed into single values. Table 1 shows the consequences as they are in our example.

We assume that before making a swap, the DM looks into the suggestions provided by the software for a swap that could make some alternatives dominated. We also assume that the DM always makes one of the suggested swaps, or a corresponding swap in the other direction. Naturally, in practice, she can decide to make other swaps as well. The suggestions are derived with the default initialization parameters of the PAIRS model and the average-case values are calculated for the applicability indices.

At the start of the process, the software scans through the consequences table and finds out that Alternative 12 might be practically dominated by Alternative 4. By comparing these alternatives explicitly, the DM can see that Alternative 12 is equal to or worse than Alternative 4 in all the other attributes except in *distance*, in which it is only a little more preferable. Thus the DM eliminates Alternative 12 based on practical dominance.

After eliminating Alternative 12, there are no more dominated alternatives and the process continues by making an even swap. When asked for suggestions for the next swap, the software suggests making a swap between *car park* and *renovation need* of Alternative 8. The software also informs the DM that with this swap, Alternative 8 may become dominated by Alternative 9, and that this suggestion is the first of six suggestions. This suggestions seems to be a quite reasonable one (the applicability index $d = 3.0$), and thus the DM makes a swap Alternative 8: (*car park: intermediate → great, renovation need: intermediate → considerable*). As a result, Alternative 8 becomes dominated by Alternative 9 and is thus eliminated.

From now on, the process continues in the same way; that is, the DM repeats making even swaps and eliminating dominated alternatives and irrelevant attributes until the best alternative is found. Table 2 shows one possible realization of the process, which results in Alternative 9 being the most preferred alternative. In this example, the DM makes, with certain exceptions, the first swaps suggested by the software. However, in some situations, it could be useful to also check the other suggestions. For example, on swap Number 3, the applicability index for the first suggested swap is only 0.93, and thus it may not be very likely to reach dominance with this swap. The applicability indices for the next two suggestions are

Table 1 **The Office Selection Problem Adapted from Punkka and Salo (2004)**

Alternatives	Size (m^2)	Rent ($)	Renovation need	Car park	Public transport	Distance to center (km)	Other facilities	Habitability
1	180	2,000	Considerable	Good	Quite bad	12	Intermediate	Great
2	240	3,000	No	Good	Good	15	Good	Bad
3	210	2,800	Intermediate	Bad	Great	0	Great	Good
4	214	2,000	Very small	Great	Bad	25	Intermediate	Good
5	300	3,200	Considerable	Great	Good	4	Great	Very good
6	170	1,800	Considerable	Quite bad	Good	0	Great	Good
7	250	2,600	Considerable	Great	Intermediate	7	Good	Intermediate
8	260	2,650	Intermediate	Intermediate	Good	10	Intermediate	Intermediate
9	262	2,400	Big	Great	Good	10	Intermediate	Very good
10	241	2,500	Small	Very good	Intermediate	7	Good	Good
11	198	2,200	Considerable	Good	Bad	17	Good	Good
12	201	2,000	Intermediate	Quite bad	Bad	22	Quite bad	Intermediate

Table 2 **Even Swaps Made During the Process**

		Even swap			
Number		Change	Compensation	Suggestion	Resulting dominance relations
Initial scanning through the consequences table					Alternative 12 practically dominated by Alternative 4
1	Alternative 8	Car park: intermediate → great	Renovation need: intermediate → considerable	1 of 6	Alternative 8 dominated by Alternative 9
2	Alternative 11	Renovation need: considerable → small	Rent: $2,200 → $2,800	1 of 6	Alternative 11 dominated by Alternative 10
3	Alternative 7	Rent: $2,600 → $3,200	Size: 250 → 290	3 of 5	Alternative 7 dominated by Alternative 5
4	Alternative 6	Distance: 0 → 4	Size: 170 → 175	1 of 6	
5	Alternative 6	Rent: $1,800 → $3,200	Size: 175 → 260	1 of 5	Alternative 6 dominated by Alternative 5
6	Alternative 2	Habitability: bad → very good	Rent: $3,000 → $3,300	1 of 6	Alternative 1 practically dominated by Alternative 5
7	Alternative 2	Renovation need: no → considerable	Size: 240 → 305	1 of 5	Alternative 2 practically dominated by Alternative 5
8	Alternative 5	Renovation need: considerable → big	Facilities: great → intermediate	2 of 6	
9	Alternative 5	Size: 300 → 262	Rent: $3,200 → $2,600	1 of 2	
10	Alternative 5	Distance: 4 → 10	Rent: $2,600 → $2,500	1 of 1	Alternative 5 dominated by Alternative 9
11	Alternative 4	Rent: $2,000 → $2,400	Size: 214 → 240	1 of 6	
12	Alternative 4	Public transport: bad → good	Renovation need: very small → intermediate	1 of 4	
13	Alternative 4	Renovation need: intermediate → big	Distance: 25 → 15	1 of 3	Alternative 4 dominated by Alternative 9
14	Alternative 3	Car park: bad → very good	Distance: 0 → 30	2 of 6	
15	Alternative 3	Facilities: great → good	Distance: 30 → 23	1 of 6	
16	Alternative 3	Public transport: great → intermediate	Distance: 23 → 5	1 of 4	Alternative 3 practically dominated by Alternative 10
17	Alternative 10	Car park: very good → great	Distance: 7 → 15	2 of 6	
18	Alternative 10	Facilities: good → intermediate	Distance: 15 → 12	2 of 6	
19	Alternative 10	Renovation need: small → big	Size: 241 → 265	1 of 5	Alternative 10 practically dominated by Alternative 9
Alternative 9 is the only remaining alternative → It is the most preferred one.					

almost the same (0.93 and 0.90, respectively), and thus it would be useful to compare all these suggestions to find out the most suitable swap. Based on this comparison, the DM decides to make the third suggested swap between *rent* and *size* of Alternative 7, as then Alternative 7 becomes dominated.

This example demonstrates how the preference information given in the even swaps can give the DM new information about the relations of the alternatives. For example, as a result of the swap Number 6, Alternative 1 becomes practically dominated by Alternative 5, although no changes have been made in either of these alternatives during the process. However, the swaps Numbers 5 and 6 result in new preference information about the attributes, of which we elicit new constraints $1.21 \leq w_{size}/w_{rent} \leq 2.29$ and $1.60 \leq w_{rent}/w_{habitability} \leq 8.45$ to the PAIRS model. Based on this updated preference information, the advantage of Alternative 1 in *rent* and *habitability* is not enough to compensate for the advantage of Alternative 5 in the other attributes, and thus Alternative 1 becomes practically dominated.

One can follow different strategies in defining the initial bounds of the PAIRS model. The basic idea is that the bounds reflect all the possible imprecision in the model. In practical decision support, the bounds can, however, be used to customize the process. Tighter bounds will produce more candidates for practical dominance relations. For example, with the parameter values $r = 3$, $a = 0.4$, and $s = 0.75$, Alternatives 8 and 11 also become identified as practically dominated in the initial scanning of the consequences table. However, because of these tighter assumptions of the model, the DM has to carefully consider whether to eliminate these alternatives using practical dominance or not. If she decides to do this, she escapes making the first two swaps in Table 2.

This example shows that the even swaps process can be easily carried out in relatively large problems, assuming that some kind of support, such as the approach presented in this paper, is available. However, we think that without any support, it would be difficult to apply the even swaps method to problems of this size, and in this respect, the proposed approach provides a useful way to support the even swaps method. Yet, it is an empirical question as to how much easier the use of the even swaps method can become with the proposed approach in real situations with real DMs. These tests are beyond the scope of this paper, and are a topic of further research.

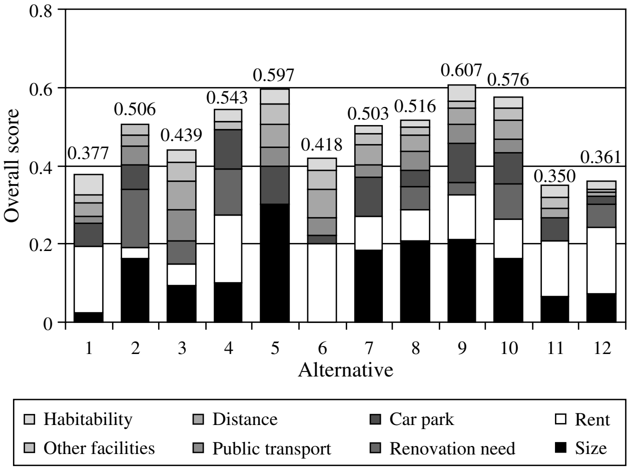
7. Comparison with MAVT

Next, we analyze the same example with a traditional MAVT approach and compare the pros and cons between even swaps and MAVT. We focus on four characteristics of the approaches: (1) assumptions needed, (2) elicitation burden, (3) transparency of the process and outcome, and (4) computational issues.

In MAVT, the ratings for the alternatives in (1) are typically derived from attributewise value functions and the weights for the attributes elicited with some specific weighting method such as Simple Multiattribute Rating Technique (SMART) (Edwards 1977, von Winterfeldt and Edwards 1986) or Swing (von Winterfeldt and Edwards 1986). The weight elicitation typically requires at least $n-1$ preference statements, where n is the number of attributes. For a comparison of different weighting methods, see, for example, Borcherding et al. (1991) or Pöyhönen and Hämäläinen (2001). Here, we do not consider differences in the weighting methods, but assume the weights to be $w_{size} = 0.3$, $w_{rent} = 0.2$, $w_{renovation\ need} = 0.15$, $w_{car\ park} = 0.10$, $w_{public\ transport} = 0.08$, $w_{distance} = 0.07$, $w_{facilities} = 0.05$, and $w_{habitability} = 0.05$, which are consistent with the even swaps made in Table 2. The ratings are obtained by using linear value functions on the attribute values over their ranges. As a result, we get the overall scores for the alternatives. These can be decomposed into components describing the contribution of different attributes (Figure 5). The best alternatives are Alternative 9 (overall score 0.607), Alternative 5 (0.597), and Alternative 10 (0.576). This result is consistent with the result of the even swaps process, in which Alternative 9 is the most preferred alternative, and Alternatives 5 and 10 are both eliminated during the process because of the dominance by Alternative 9.

As mentioned before, one of the main advantages of the even swaps method over MAVT is that it does not require making any assumptions about the form of the value function. Additive value functions are typically used in MAVT, but if the mutual preferential independence between the attributes cannot

Figure 5 **Overall Scores of the Alternatives and the Contribution of Different Attributes to the Overall Scores**



be assumed, some other form, for example, a multiplicative value function should be used. This makes, however, the elicitation of the MAVT model more complex and less attractive.

In the even swaps process, one does not know in advance how many swaps are needed to find out the most preferred alternative. Depending on the characteristics of the problem, the number of the swaps needed can become high. In our example, we need 19 swaps until the best alternative is found, whereas in MAVT, the weight elicitation, for example, by SMART requires seven preference statements between the attributes. In practice, making trade-offs in the weighting of the attributes is a comparable task as making even swaps. In MAVT, one additionally needs an attributewise value function for each attribute. In many applications, these are assumed to be linear for simplicity, but, in general, the elicitation of value functions also requires that the DM makes value comparisons between consequence changes on the attributes. Typically, this would mean at least one elicitation question for each attribute. One should note that in MAVT, the elicitation burden does not increase with the number of alternatives. In even swaps, this is not the case, but more steps are usually needed when there are more alternatives.

In MAVT, the value comparisons are made over the whole range of all the attributes. One might say that this is more difficult than the even swaps process, where the comparisons are only made on those attribute values that are involved in the even swaps. However, in even swaps, the DM may have to make several comparisons on the same pair of attributes during the process. Also, the DM has to decide which swaps to make, whereas in MAVT, all the steps needed are known in advance. In this respect, MAVT can be said to be clearly more structured. To summarize, the elicitation burden, when counting the number of statements, with an additive MAVT model is likely to be lower than in the even swaps process. One should, however, note that the questions to be answered in MAVT are entirely different.

The results of MAVT analyses are clear to interpret, and from the overall scores, one can easily see the origins of the differences in the preferences between the alternatives. In even swaps, the process reveals the dominance relations between the most preferred alternative and the other alternatives, but it does not produce the relative scores of the alternatives. This can be a problem, especially if there are alternatives close to each other, all of which could be considered as the most preferred one. We feel that the most problematic feature in even swaps is that during the process, the alternatives' outcomes are replaced by artificial outcomes, which are changed with each trade-off. This is a characteristic of the process only, and it can be difficult to understand that when the best alternative is found, it will indeed be the original one with the original outcomes.

Computationally, the MAVT and even swaps processes can both be carried out easily, for example, on a spreadsheet program. The use of the proposed preference programming approach requires solving linear programs, but with the computational capacity of current PCs, this is not a critical issue. For example, the initial calculation of practical dominances in our example takes 12 seconds with the Smart-Swaps software on a 2.4 GHz computer.

Overall, both methods have their pros and cons. We think that a traditional MAVT is always required when you need to have a transparent process and explain and justify the choice. This is important in group and public policy decisions. The even swaps method suits personal decision making, where you are only looking for the best alternative for yourself. It has appealing characteristics such as freedom from assumptions, but the use of it requires understanding the nature of trade-offs. In addition, the elicitation burden can become high in large problems, but in this respect, the approach proposed here should make the use of the even swaps method easier. Yet, the question about the relative usefulness of the methods in different kinds of situations is an open issue. The effectiveness and efficiency of the methods need to be evaluated in experiments, preferably with real DMs, and this remains a topic of further research.

8. Conclusions

We have introduced an approach to help the DM carry out the even swaps process in practice. The original process of swapping remains the same, but the proposed preference programming approach makes the process easier by giving helpful suggestions during the process. The greatest benefits of using the approach are obtained in large problems, in which a thorough screening of the problem is practically impossible without computational help.

Even swaps is a relatively new method, and thus its practical applicability remains to be demonstrated. The availability of the Smart-Swaps software is likely to increase the number of real-life applications. The basic philosophy of the even swaps process is to keep the elicitation burden of the DM as low as possible. Our approach contributes to this by making use of all the possible partial preference information obtained during the process.

Acknowledgments

The authors acknowledge Pauli Alanaatu, Ville Karttunen, and Arttu Arstila for their contributions in programming the Smart-Swaps software and the anonymous referees for their constructive comments. Jyri Mustajoki acknowledges the financial support from the Academy of Finland (project 32641), and the Jenny and Antti Wihuri Foundation.

Appendix. Proof of Theorem 1

The minimum of the function is found in an extreme point of the feasible region of the weights S, which is now constrained by only one type of constraints $w_i/w_j \le r$ for $\forall\, i, j = 1, \ldots, n,\ i \ne j$. Thus, in each extreme point, at least one weight ratio is on its bound $w_i/w_j = r$, as otherwise, this point would not be an extreme point. Correspondingly, each other weight w_k for $\forall\, k = 1, \ldots, n,\ k \ne i,\ k \ne j$ on this point is on bound $w_i/w_k = r$ or $w_k/w_j = r$, and thus the weights of the attributes are w_j, $w_i = r \times w_j$, and either $w_k = w_i/r = w_j$ or $w_k = r \times w_j$ for $\forall\, k = 1, \ldots, n,\ k \ne i,\ k \ne j$. The set of all the extreme points can be elicited by taking all the points $c = (c_1, \ldots, c_n)$, where $c_i = 1$ or $c_i = r$ for $\forall\, i = 1, \ldots, n$, and normalizing the sum to $\sum_{i=1}^n c_i = 1$. (In fact, the point where $c_1 = c_2 = \cdots = c_n$ should be excluded from this set, but in the computation of the minimum, this point can be included in the set of extreme points as it is clearly an interior point.) As a result, we get the set of non-normalized extreme points $E^C = \{c = (c_1, \ldots, c_n) \mid c_i = 1 \text{ or } c_i = r,\ \forall\, i = 1, \ldots, n\}$, and by normalizing the sum of the weights to one, we get the set of extreme points $E^S = \{w = (c_1/\sum c_i, \ldots, c_n/\sum c_i) \mid c \in E^C\}$.

Assume $\min \sum_{i=1}^n c_i[\underline{v}_i(x_i) - \bar{v}_i(y_i)] > 0$, $c \in E^C$, i.e.,

$$\sum_{i=1}^n c_i[\underline{v}_i(x_i) - \bar{v}_i(y_i)] > 0, \quad \forall\, c \in E^C.$$

As

$$\sum c_i > 0 \;\Rightarrow\; \sum_{i=1}^n \frac{c_i}{\sum c_i}[\underline{v}_i(x_i) - \bar{v}_i(y_i)] > 0, \quad \forall\, c \in E^C.$$

As

$$\begin{aligned} E^S &= \left\{ w = \left(\frac{c_1}{\sum c_i}, \ldots, \frac{c_n}{\sum c_i} \right) \,\middle|\, c \in E^C \right\} \\ &\Rightarrow \sum_{i=1}^n w_i[\underline{v}_i(x_i) - \bar{v}_i(y_i)] > 0, \quad \forall\, w \in E^S. \end{aligned} \tag{10}$$

Thus, $\mathbf{x}$ dominates $\mathbf{y}$.

Values c_i in $c = (c_1, \ldots, c_n) \in E^C$ do not depend on each other $\Rightarrow \min \sum_{i=1}^n c_i[\underline{v}_i(x_i) - \bar{v}_i(y_i)] = \sum_{i=1}^n \min(c_i[\underline{v}_i(x_i) - \bar{v}_i(y_i)])$. On the other hand, $\min(c_i[\underline{v}_i(x_i) - \bar{v}_i(y_i)])$ for any i is directly obtained at the point $c_i = 1$ if $(\underline{v}_i(x_i) - \bar{v}_i(y_i)) \ge 0$, and at the point $c_i = r$ if $(\underline{v}_i(x_i) - \bar{v}_i(y_i)) < 0$. □

References

Arbel, A. 1989. Approximate articulation of preference and priority derivation. *Eur. J. Oper. Res.* **43**(3) 317–326.

Belton, V., T. J. Stewart. 2002. *Multiple Criteria Decision Analysis. An Integrated Approach*. Kluwer Academic Publishers, Norwell, MA.

Borcherding, K., T. Eppel, D. von Winterfeldt. 1991. Comparison of weighting judgments in multiattribute utility measurement. *Management Sci.* **37**(12) 1603–1619.

Butler, J., D. J. Morrice, P. W. Mullarkey. 2001. A multiple attribute utility theory approach to ranking and selection. *Management Sci.* **47**(6) 800–816.

Edwards, W. 1977. How to use multiattribute utility measurement for social decisionmaking. *IEEE Trans. Systems, Man, Cybernetics* **7**(5) 326–340.

Gregory, R., K. Wellman. 2001. Bringing stakeholder values into environmental policy choices: A community-based estuary case study. *Ecological Econom.* **39**(1) 37–52.

Hämäläinen, R. P. 2000. *Decisionarium—Global Space for Decision Support*. Systems Analysis Laboratory, Helsinki University of Technology, http://www.decisionarium.hut.fi.

Hämäläinen, R. P. 2003. Decisionarium—Aiding decisions, negotiating and collecting opinions on the web. *J. Multi-Criteria Decision Making* **12**(2–3) 101–110.

Hämäläinen, R. P., J. Mustajoki, P. Alanaatu, V. Karttunen, A. Arstila. 2003. *Smart-Swaps—Smart Choices with Even Swaps*. Computer software, Systems Analysis Laboratory, Helsinki University of Technology, http://www.smart-swaps.hut.fi.

Hammond, J. S., R. L. Keeney, H. Raiffa. 1998. Even swaps: A rational method for making trade-offs. *Harvard Bus. Rev.* **76**(2) 137–149.

Hammond, J. S., R. L. Keeney, H. Raiffa. 1999. *Smart Choices. A Practical Guide to Making Better Decisions*. Harvard Business School Press, Boston, MA.

Hazen, G. B. 1986. Partial information, dominance, and potential optimality in multiattribute utility theory. *Oper. Res.* **34**(2) 296–310.

Kajanus, M., J. Ahola, M. Kurttila, M. Pesonen. 2001. Application of even swaps for strategy selection in a rural enterprise. *Management Decision* **39**(5) 394–402.

Keeney, R. L. 1992. *Value-Focused Thinking. A Path to Creative Decisionmaking*. Harvard University Press, Cambridge, MA.

Keeney, R. L. 2002. Common mistakes in making value trade-offs. *Oper. Res.* **50**(6) 935–945.

Keeney, R. L., H. Raiffa. 1976. *Decisions with Multiple Objectives. Preferences and Value Tradeoffs*. John Wiley and Sons, Inc., New York.

Kirkwood, C. W., J. L. Corner. 1993. The effectiveness of partial information about attribute weights for ranking alternatives in multiattribute decision making. *Organ. Behavior Human Decision Processes* **54**(3) 456–476.

Kirkwood, C. W., R. K. Sarin. 1985. Ranking with partial information: A method and an application. *Oper. Res.* **33**(1) 38–48.

Pöyhönen, M., R. P. Hämäläinen. 2001. On the convergence of multiattribute weighting methods. *Eur. J. Oper. Res.* **129**(3) 569–585.

Punkka, A., A. Salo. 2004. Preference programming with incomplete ordinal information. Working paper, Systems Analysis Laboratory, Helsinki University of Technology, http://www.sal.hut.fi/Publications/pdf-files/mpun04.pdf.

Sage, A. P., C. C. White. 1984. ARIADNE: A knowledge-based interactive system for planning and decision support. *IEEE Trans. Systems, Man, Cybernetics* **14**(1) 35–49.

Salo, A., R. P. Hämäläinen. 1992. Preference assessment by imprecise ratio statements. *Oper. Res.* **40**(6) 1053–1061.

Salo, A., R. P. Hämäläinen. 1995. Preference programming through approximate ratio comparisons. *Eur. J. Oper. Res.* **82**(3) 458–475.

Salo, A., R. P. Hämäläinen. 2004. Preference programming. Working paper, Systems Analysis Laboratory, Helsinki University of Technology, http://www.sal.hut.fi/Publications/pdf-files/msal03b.pdf.

Stewart, T. 1996. Robustness of additive value function methods in MCDM. *J. Multi-Criteria Decision Anal.* **5**(4) 301–309.

von Winterfeldt, D., W. Edwards. 1986. *Decision Analysis and Behavioral Research*. Cambridge University Press, Cambridge, UK.

Weber, M. 1987. Decision making with incomplete information. *Eur. J. Oper. Res.* **28**(1) 44–57.

White, C. C., A. P. Sage, S. Dozono. 1984. A model of multiattribute decisionmaking and trade-off weight determination under uncertainty. *IEEE Trans. Systems, Man, Cybernetics* **14**(2) 223–229.