

SMART-SWAPS – DECISION SUPPORT FOR THE PROACT PROCESS WITH THE EVEN SWAPS METHOD

Jyri Mustajoki Raimo P. Hämäläinen



TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKI UNIVERSITY OF TECHNOLOGY
TECHNISCHE UNIVERSITÄT HELSINKI
UNIVERSITE DE TECHNOLOGIE D'HELSINKI

Distribution:

Systems Analysis Laboratory
Helsinki University of Technology
P.O. Box 1100
FIN-02015 HUT, FINLAND
Tel. +358-9-451 3056
Fax. +358-9-451 3096
systems.analysis@hut.fi

This report is available at
www.e-reports.sal.hut.fi/pdf/E20.pdf

Series E - Electronic Reports
www.e-reports.sal.hut.fi

ISBN 951-22-8341-7
ISSN 1456-5218

Title: Smart-Swaps – Decision Support for the PrOACT Process with the Even Swaps Method

Authors: Jyri Mustajoki
Systems Analysis Laboratory
Helsinki University of Technology
P.O. Box 1100, 02015 HUT, FINLAND
jyri.mustajoki@hut.fi
www.sal.hut.fi/Personnel/Homepages/JyriM.html

Raimo P. Hämäläinen
Systems Analysis Laboratory
Helsinki University of Technology
P.O. Box 1100, 02015 HUT, FINLAND
raimo@hut.fi
www.sal.hut.fi/Personnel/Homepages/RaimoP.html

Date: August, 2006

Status: Systems Analysis Laboratory Research Reports E20 August 2006

Abstract: We introduce a decision support tool called Smart-Swaps to support the PrOACT process. This is a relatively new decision analytical approach to support multicriteria decision making. The elicitation of the decision maker's preferences is carried out with the even swaps method which is based on value trade-offs. We demonstrate different ways to support the process and new opportunities provided by the preference programming model on which the software is based. In many practical problems, such decision support can provide substantial help. Often in large problems, it is, in fact, a necessity for carrying out the even swaps process.

Keywords: multicriteria decision analysis, even swaps, preference programming, trade-offs.

1. Introduction

In their book *Smart Choices*, Hammond et al. (1999) describe a structured approach to analyze problems with multiple criteria and alternatives. The core of the approach consists of the PrOACT, which is the acronym for **P**roblem, **O**bjectives, **A**lternatives, **C**onsequences and **T**rade-offs, working phases. These are the essential elements in almost any complex decision making situation. The aim is to provide practical guidance on how to make consistent decisions by taking all the important objective and subjective factors into account. The thinking process is designed for practitioners, and the decision maker (DM) does not need to have a mathematical or decision analytical background to use the approach.

In this approach, preference elicitation is carried out with the even swaps method (Hammond et al. 1998). It is a multicriteria decision analysis (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 DM changes the consequence of an alternative in one attribute, and compensates this change with a preferentially equal change in the consequence of some other attribute. The new virtual alternative with the revised consequences is as preferred as the initial one and thus can be used instead, even though it does not exist for real.

The aim of the even swaps process is to carry out even swaps that either make attributes irrelevant (i.e. all the alternatives have equal consequences on this attribute), or alternatives dominated (i.e. some other alternative is at least as good as this alternative on every attribute). These can be eliminated, and the process continues until only the most preferred alternative remains. Hammond et al. (1998) also introduce the concept of practical dominance. 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. Thus, y can be eliminated in order to reduce the problem in obvious cases without a need to carry out unnecessary even swap tasks.

In this paper, we introduce a new web-based Smart-Swaps software (Hämäläinen et al. 2003). We focus on the opportunities this Java-applet provides to support the PrOACT approach. Especially from the viewpoint of carrying out the even swaps elimination process, such software can be of substantial help. For example, the procedural support of Smart-Swaps for dynamically managing the consequences table can be very useful, as the table changes frequently during the process. The software also implements a new method of Mustajoki and Hämäläinen (2005) to help the DM to identify practical dominances and to find suitable candidates for the next even swaps. This method is based on preference programming (Arbel 1989; Salo and Hämäläinen 1992, 2004) which is a multiattribute value theory approach that models uncertainty in preferences with intervals. Smart-Swaps is available at <http://www.smart-swaps.hut.fi>, and it is a new tool in the family of MCDA software on the Decisionarium site for decision support (Hämäläinen 2000, 2003; <http://www.decisionarium.hut.fi>).

In spite of its simplicity, it seems that the even swaps method has rarely been applied in practice. To our knowledge, the only reported applications in the literature are the one on strategy selection in a rural enterprise (Kajanus et al. 2001) and another on environmental planning (Gregory and Wellman 2001). This lack of use can be partly due to the fact that, so far, there has not been any software to support the approach. We hope that the introduction of the Smart-Swaps software described here will make the approach more popular in real life too.

This paper is organized as follows. Section 2 gives a brief introduction to multicriteria tools. Section 3 discusses the opportunities of computer support in the PrOACT approach and describes the Smart-Swaps software in terms of how these opportunities are adapted in the software. An example of the use of the software is given in Section 4, and Section 5 concludes the paper.

2. Multicriteria decision support systems

Multicriteria decision support systems (MCDSS) or tools are computer-based interactive software designed to support the modelling of multicriteria problems with decision analytical methods. The aim is to provide help for structuring the problem, eliciting preferences and analyzing the results, so that the DM can focus on the core of the problem while the technical issues are taken care of by the computer.

Recent development of computer technology has provided new opportunities to enhance the use of MCDA (for discussion, see e.g. Shim et al. 2002; Bhargava et al. 2005). For example, with today's multimedia facilities, MCDA methods can be visualized to help preference elicitation and the analysis of the results. Increased computational capabilities have also made it possible to create interactive software with new computationally demanding methods, such as linear programming. In recent years, the proliferation of the World Wide Web has also enabled MCDSSs that are easily available for remote use.

There are various MCDSSs available (see e.g. INFORMS 2000). For recent reviews of these see, for example, Maxwell (2004), Turban et al. (2004) or French and Xu (2004). The software range from customized tools with a single method for some specific application (see e.g. Johnson 2005) to general-purpose systems providing a choice between several different MCDA methods as well as group decision support facilities (see e.g. Mustajoki and Hämäläinen 2000). Some of the systems allow the use of trade-offs as one of the weighting methods, but to our knowledge, the Smart-Swaps software is the first system implementing the even swaps method.

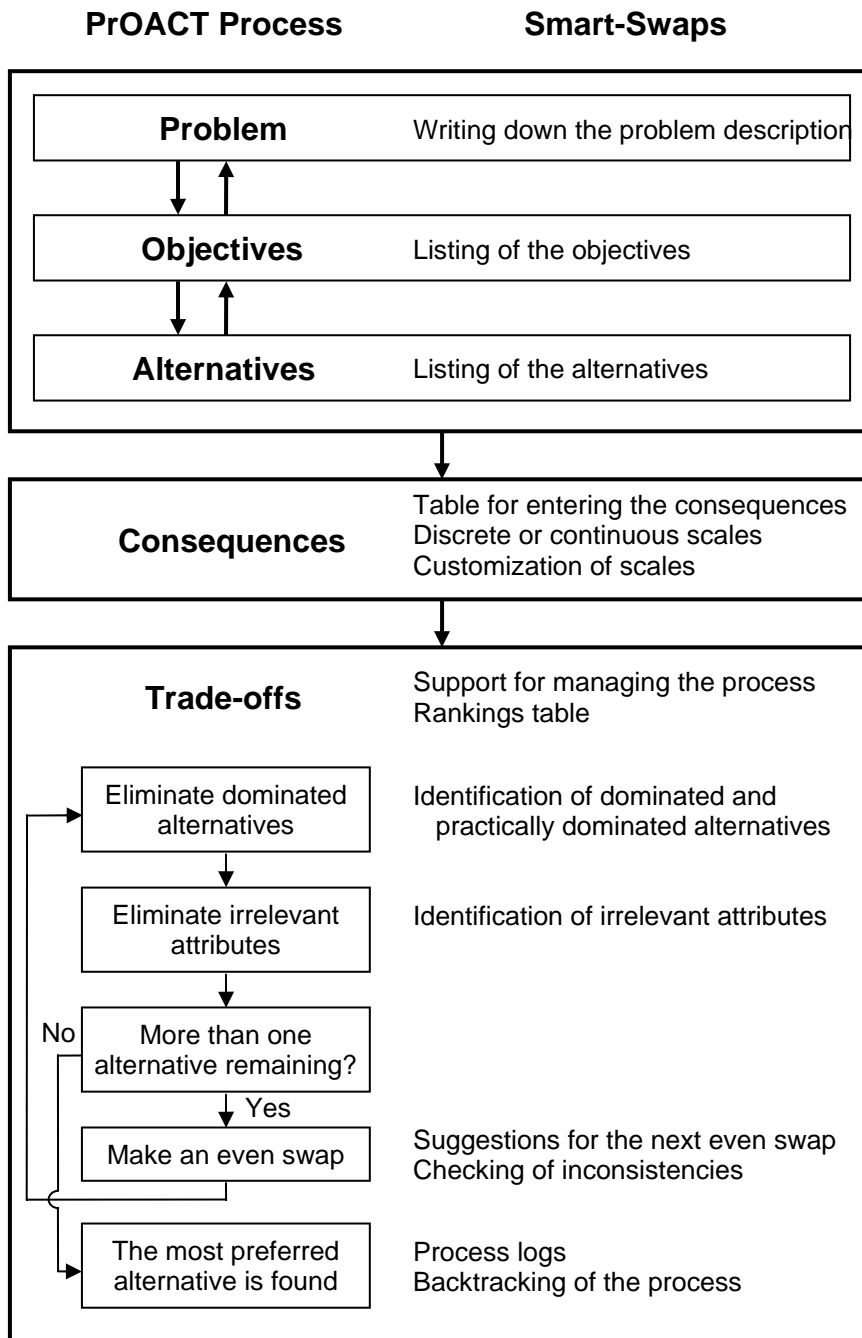
3. The Smart-Swaps decision support tool

3.1. Support for the PrOACT process

Table 1 shows an overview of the support provided by Smart-Swaps in the different phases of the process. By nature, the PrOACT approach is a sequential process based on five clearly distinct phases. To support this sequentiality, the process management in Smart-Swaps is implemented with a tab-panelled user interface which follows the PrOACT working phases (Figure 1). The first three phases

(*Problem, Objectives and Alternatives*) are on the same panel, as active interaction between these is often needed but the *Consequences* and *Trade-offs* phases are on their own panels. The aim is to give the DM a clear indication of the course of the process but, at a same time, allow him/her to easily return to the earlier phases of the process, if necessary.

Table 1. Support provided by the Smart-Swaps software in the different phases of the PrOACT process.



Help system available at each phase

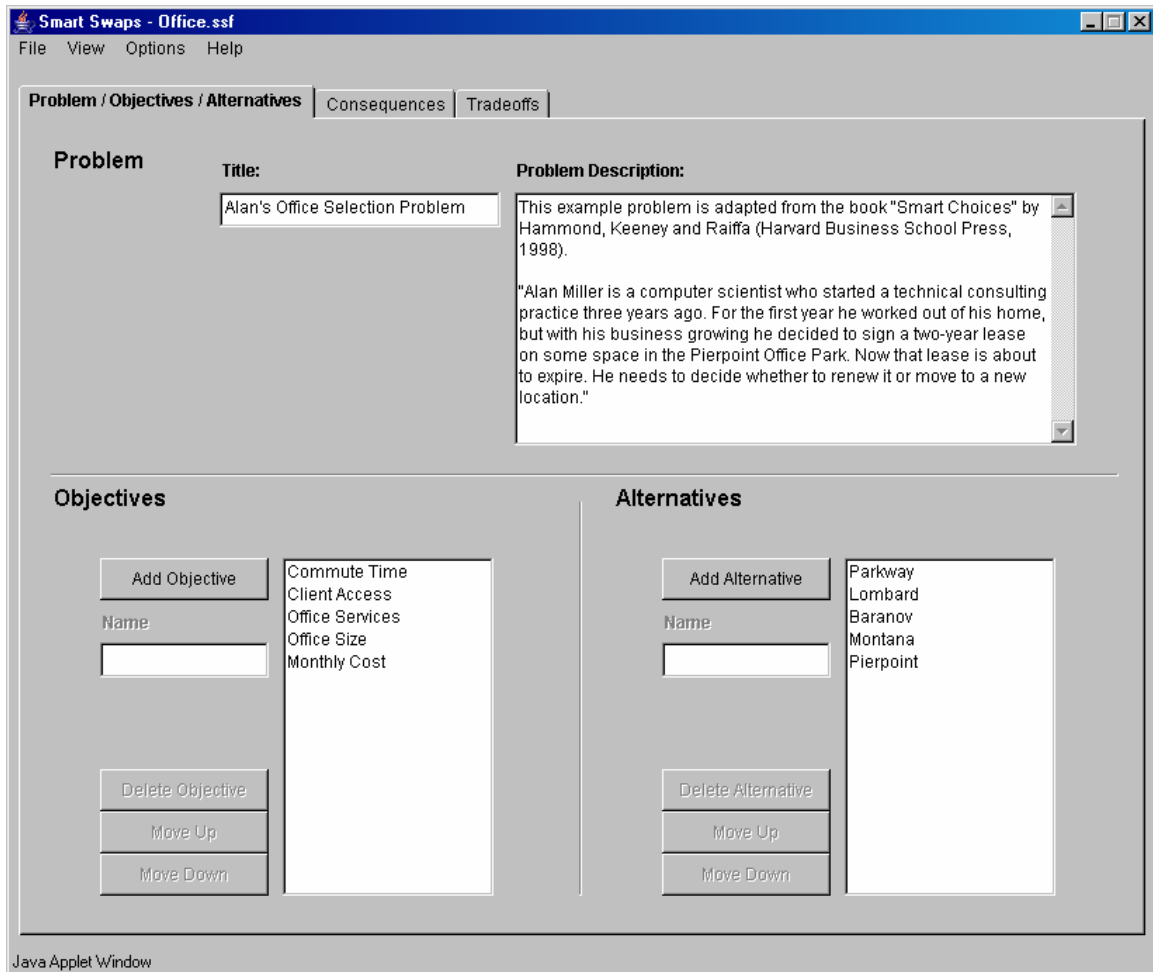


Figure 1. *The user interface for the identification of the Problem, Objectives and Alternatives.*

3.2. Problem, objectives and alternatives

The first phase of the process is to set a problem framework on which the decision will be made. Consideration of the problem from different perspectives is essentially important to get a view of what the real problem is. The next phases are the identification of objectives and alternatives. Objectives define what the DM wants to achieve, and alternatives describe the possible actions that can be taken. There are different procedures for identifying these, and for a related discussion, see Hammond et al. (1998) or Keeney (1992).

Our software provides a section for each of these phases in which the DM can write down the problem description and list and order the objectives or alternatives (Figure 1). However, the greatest challenge in these phases is to help the DM to take all the different viewpoints into account and still construct a practically useful and concise model. For example, providing information about how to avoid possible biases in the modelling is one way of guiding the DM. In Smart-Swaps, the methodological guidance is implemented with help facilities providing detailed information for all the phases of the PROACT process. For details, see Section 3.5.

One should note that the order of these first three phases is not fixed and the most creative ideas are usually found by considering these phases interchangeably (Hammond et al. 1999). For example, a thorough analysis of the alternatives can bring in new views that put the problem into a completely new perspective. In such cases, the DM should go back and reframe the problem according to this new information.

3.3. Initialization of the consequences table

In the *Consequences* phase, the DM creates the consequences table in which the performance of each alternative is measured with respect to each attribute. Sometimes the attributes for measuring the achievements of the objectives are natural ones (e.g. on minimizing the costs) but on other cases they have to be constructed (see e.g. Keeney and Gregory 2005).

Smart-Swaps supports both discrete and continuous scales for the attributes. For continuous scales, the DM can use any decimal number to represent the performance levels of the attributes associated with each alternative. For discrete scales, the DM can use either one of the predefined scales (e.g. "Very Good – Good – Intermediate – Bad – Very Bad"), or create a scale of his/her own. This opportunity for customizing the scale allows the DM to construct scales indicating the preference order for any set of consequences. The DM can, for example, create a preference order for a set of numerical consequences, in which the preference over the consequences does not initially increase in a numerical order.

In practice, the DM directly enters the consequences into the consequences table (Figure 2). One should, however, note that the attributes in the table represent the objectives defined during the *Objectives* phase. Thus, in the *Objectives* phase, the DM should make sure that for each objective there is a corresponding measurable attribute.

	Commute Time	Client Access	Office Services	Office Size	Monthly Cost
Scale	Continuous	Continuous	A-C	Continuous	Continuous
Min/Max	Minimize	Maximize		Maximize	Minimize
Unit	min	%		sq. ft.	\$
Parkway	45	50	A	800	1850
Lombard	25	80	B	700	1700
Baranov	20	70	C	500	1500
Montana	25	85	A	950	1900
Pierpoint	30	75	C	700	1750

Figure 2. The user interface for the Consequences phase.

3.4. Support for the even swaps process

The final phase of the ProACT process is to elicit the preferences of the DM. Both in the Smart Choices book and in the Smart-Swaps software, the even swaps method is used for this task but in principle any other MCDA method can be used as well. The even swaps process is conceptually simple, as the DM does not have to explicitly define the preferences over the attributes in general. It does not require making any restricting mathematical assumptions either, for example, about the form of a value function (Butler et al. 2001).

In Smart-Swaps, the interface for supporting the even swaps elimination process includes a window showing the current consequences table and buttons for the possible actions. In addition, there is an information window showing any additional information that may be needed to give to the DM during the process (Figure 3).

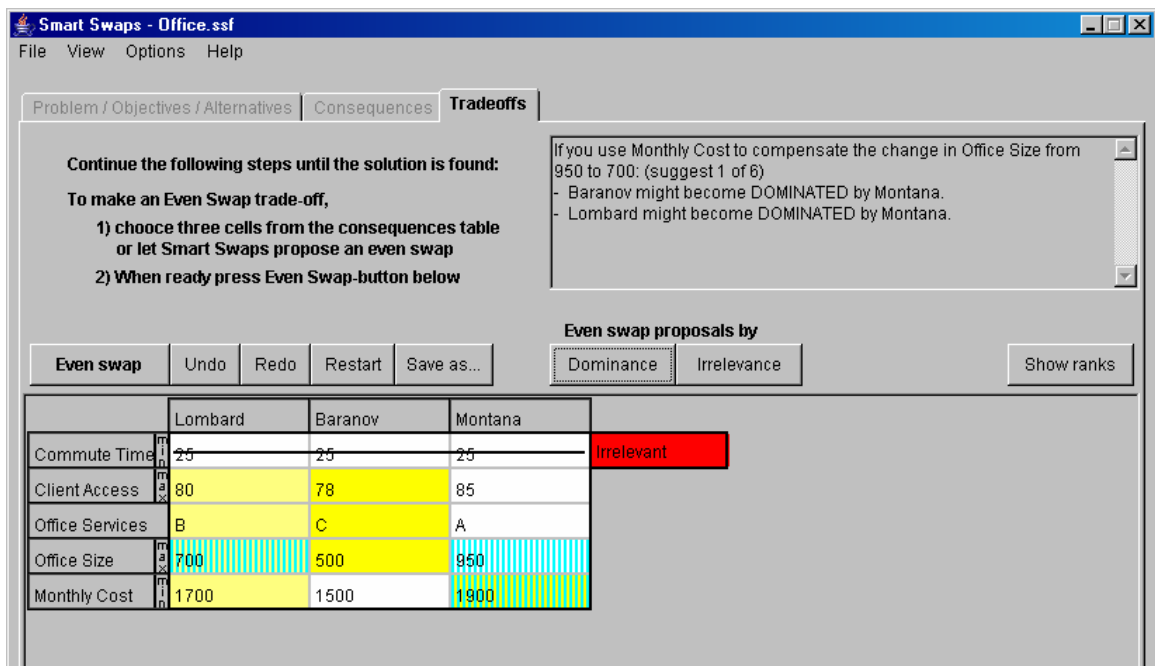


Figure 3. The user interface for the Even Swaps process.

As a result of the process, the DM ends up with the most preferred alternative. However, with respect to transparency and justifiability of the result, it is important to know, how this conclusion has been reached. Smart-Swaps documents the whole process by keeping log of the actions made by the DM during the process. This history of log saves all the information about the swaps made by the DM and about the eliminated attributes and alternatives, as well as the state of the consequences table after each swap (see Figure 4).

The DM can undo and redo the actions that have been made. The DM can, for example, return to the beginning of the process to check whether the new virtual alternative with the revised attribute values indeed corresponds to the initial alternative. The DM can also carry out sensitivity analyses to see whether different sequences of swaps produce different final results. That is, the DM can backtrack to

some intermediate point of the process and restart from this point by making some other sequence of swaps to see whether s/he ends up to the same solution.

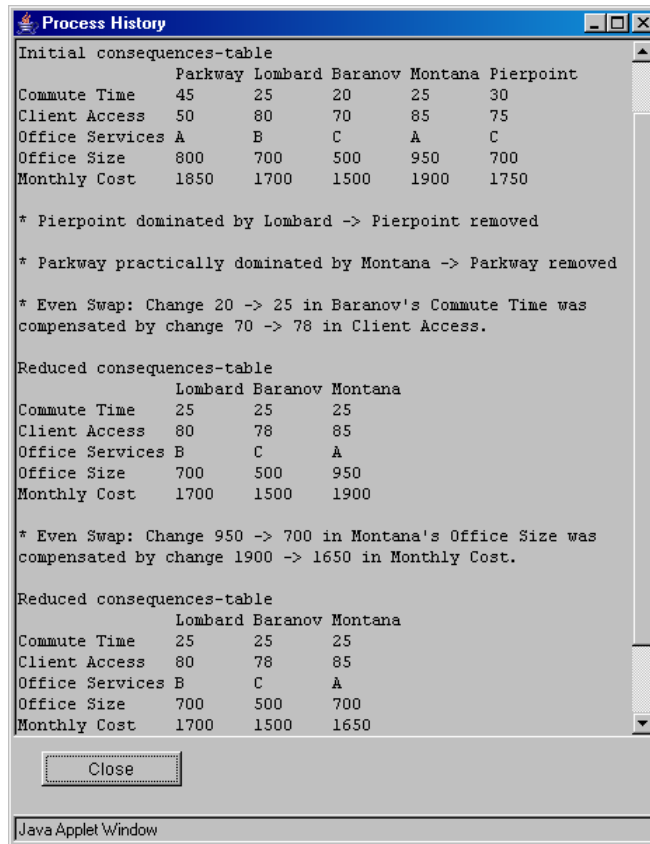


Figure 4. An example of the process log.

3.4.1. Making an even swap

To make an even swap, the DM first selects the three cells in the consequences table between which s/he wants to carry out an even swap. These include (i) a cell in which the consequence change is made, (ii) a reference cell indicating against which value this consequence is traded, and (iii) a cell in which the change is compensated (striped cells in Figure 3). Once these have been selected, the information window shows what can be achieved with this swap, that is, which alternatives can possibly become dominated and which attributes irrelevant. This information can be very useful, as it might otherwise be difficult to see especially which alternatives may become dominated. By changing the selected cells, the DM can easily study the implications of different possible swaps before confirming the final swap to be made.

The actual even swap is defined in a separate dialog window. The DM is asked to define which consequence change in attribute j of alternative x would compensate the given consequence change in attribute i (Figure 5). The phrasing in the dialog window helps the DM understand whether s/he should decrease or increase the current consequence. However, if the DM accidentally makes a swap into the wrong direction, the software informs him/her about this and requests to redefine the swap.

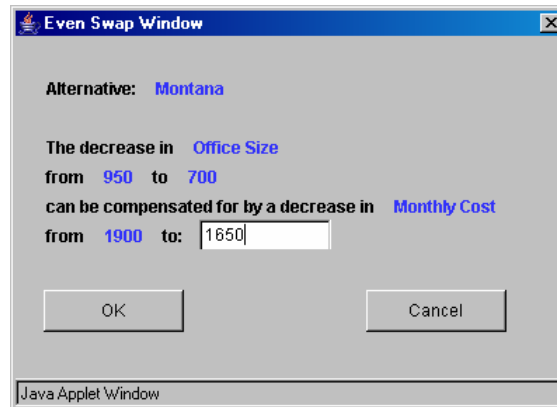


Figure 5. A dialog for making an even swap.

3.4.2. Derivation of the suggestions for the next swap

In a typical case, there are numerous possibilities for the next swap, and it can be very difficult to decide which one of these should be carried out. Hammond et al. (1998) provide practical advice for selecting the next swap, for example, they suggest making the easiest swaps first (e.g. on attribute money). On the other hand, the idea of the even swaps process is to carry out those swaps that make attributes irrelevant or alternatives dominated. In practice, the identification of such swaps can be excessively difficult, especially in problems with several attributes and alternatives.

In our software, the DM has an option to let the software suggest suitable candidates for the following swap. The software scans through the problem and calculates the efficiency of each possible swap by estimating the minimum number of swaps needed after this swap to make an attribute irrelevant or an alternative dominated. The software creates and continuously updates lists of suitable candidate swaps both for making attribute(s) irrelevant and alternative(s) dominated. The DM can ask the software to suggest the next swap from either of these lists.

The software creates the list of candidate swaps for making attribute(s) irrelevant in the following way. The list of candidates for making alternative(s) dominated is created analogously.

- Step 1. The software identifies all the swaps that would make any attribute irrelevant. If there are no such swaps, the software lists those swaps that could make some attribute to be one swap, or two swaps, etc., apart from being irrelevant.
- Step 2. The obtained list is ordered according to the applicability index of each swap to reach dominance (see Mustajoki and Hämäläinen, 2005; Salo and Hämäläinen, 1992, 2004).
- Step 3. Of the swaps involving the same two attributes and two alternatives, the software excludes all but the first one from the list. The reason is that once one of these has been identified, the others can be easily seen.
- Step 4. The software excludes all but the first six swaps from the list, as giving too many options may complicate the DM's task rather than make it easier.

In Step 2, the ordering is needed as there can be many sequences of swaps that reach dominance or irrelevance with a given number of swaps. The higher applicability index value a swap has, the more likely it is that dominance would be reached with this swap. The applied preference programming model is initialized with some weak assumptions about the DM's preferences, and updated during the process with the trade-off information obtained from the given even swaps. For details, see Mustajoki and Hämäläinen (2005).

The software only provides one suggestion at a time by highlighting those cells of the consequences table that are involved in the swap. It also informs the DM what can be achieved with this swap to help the DM see the logic behind the suggestion. The DM needs to choose whether to make this swap or ask the software to give the next suggestion on the list. Naturally, the DM can also select the swap by him/herself, if s/he prefers making a swap, for example, on some easily measurable attribute.

There are several options for suggesting the swaps. The DM can, for example, adjust the parameters of the preference programming model that define the initial assumptions about his/her preferences, and choose whether or not to show the applicability index value for each swap. The default settings for all the options are such that an inexperienced DM should be able to proceed without resetting them. For example, by default the value of the applicability index is not shown, as it may confuse an inexperienced DM rather than help him/her. However, an advanced DM or a decision analyst can choose to show it to get more detailed information about the efficiency of the suggested swaps.

3.4.3. Identification of irrelevant attributes and dominated alternatives

After each swap, the Smart-Swaps software automatically identifies irrelevant attributes and dominated alternatives. Especially in detecting the dominance relations, the DM can benefit from this support, as this task requires comparing all the possible pairs of the alternatives. According to the basic idea of the method, irrelevant attributes and dominated alternatives should naturally be eliminated. Smart-Swaps asks the DM to confirm these eliminations, as this helps the DM to fully understand the process and the reasons for elimination. The software also provides an option to retain the irrelevant attributes in the consequences table but marked as eliminated. This may help to keep the big picture in mind and contribute to the overall understanding of the process.

Candidates for practically dominated alternatives are also identified and labelled during the process. The details of the related preference programming technique are presented in Mustajoki and Hämäläinen (2005). By clicking on the labels, a comparison dialog will open, in which the possible practically dominated alternative is presented side-by-side with an alternative that dominates it (Figure 6). The reasoning for the practical dominance is explained, and based on this information the DM is asked to confirm whether the alternative should be eliminated.

After each swap, the software checks whether the preference information given in this swap is conflicting with any preference information given earlier under the assumptions of the preference programming model. The software informs the DM of possible inconsistent statements and suggests checking the consistency of the given

swaps by backtracking the process. If the DM does not find any inconsistencies, the assumptions of the model are likely to be too strong. Then, the conflicting preference information obtained from the previous swaps can not naturally be used in the model and, consequently, any subsequent suggestions can not be assumed to be very accurate.

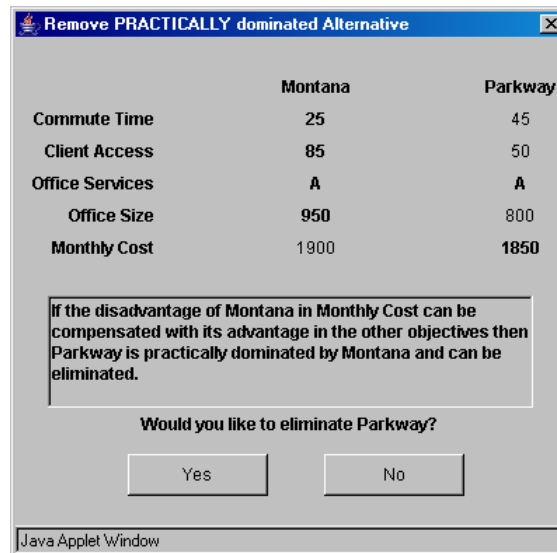


Figure 6. A comparison whether to remove a practically dominated alternative.

3.4.4. Rankings table

Hammond et al. (1999) suggest using a rankings table in which the attribute-wise rankings of the alternatives are shown to get an overview of the overall performances alternatives. In Smart-Swaps, the DM can switch between the two views showing the consequences table and the rankings table. In addition, the software provides an option to visually indicate the rankings of the alternatives by a colour coding on the cells of the consequences table. The best alternative with respect to each attribute is shown on white and the worst alternative on yellow. The colours of the other alternatives follow a linear white–yellow colour scale so that the lower ranking of the alternative has, the darker shade of yellow there is. In this way, the DM can easily see the ranking of the alternatives from the consequences table.

3.4.5. Computational issues

The preference programming model requires using linear programming to solve the related optimization problems. Since there are constraints only on the pairwise ratios of the attribute weights, the graph based algorithm of Salo (1990) can be used to quickly solve these problems. This algorithm is also implemented in the Smart-Swaps software. For example, in a case adapted from Punkka and Salo (2004) with eight attributes and twelve alternatives, the identification of dominances with Smart-Swaps takes less than two seconds with a 2.4GHz computer. Yet, in larger problems the graph algorithm is not very efficient. For this reason, the software uses a basic simplex algorithm in problems with ten or more attributes.

3.5. The help facilities

The ProACT process is intended to be available for non-experts too, which means that we need good help facilities. In Smart-Swaps, the help pages can be configured to follow the DM during the process so that the help screen is automatically updated to show the relevant information. This feature can be very valuable for an inexperienced DM carrying out the process for the first time. The first three phases are on the same panel window, and the help system automatically refers to the section of the panel which is being studied.

There is help separately for theory and for practice (Figure 7). The theory section explains the theoretical background of each task and gives guidance on what should be taken into account in the task. The practice section provides detailed information on how to carry out the current task.

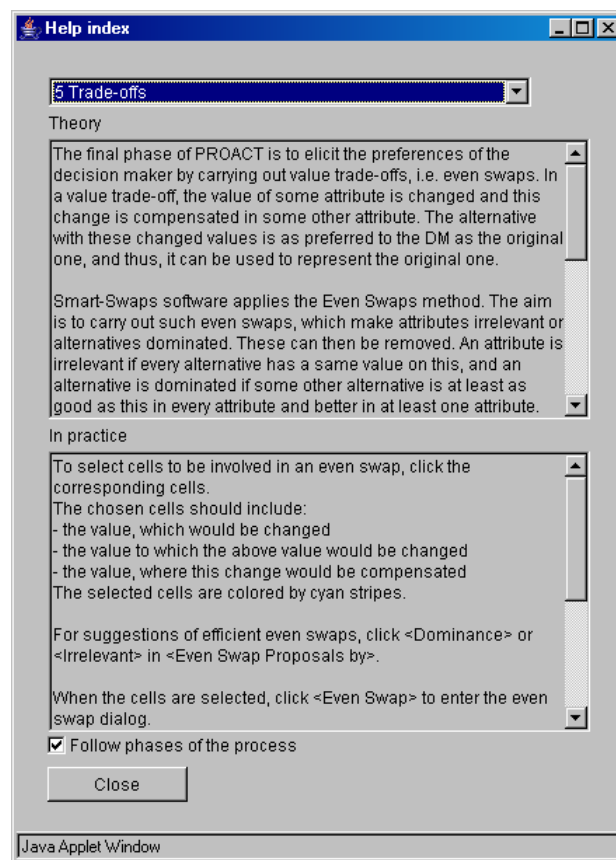


Figure 7. The help information window.

3.6. Model management

The software allows saving the models on the server computer. Thus, one can open models from different Internet locations without a need to transfer the models to remote computers. For each model, the process history will also be saved. This allows the DM to later analyze the process and study path dependence by comparing the results of the processes with some other sequences of even swaps.

The server provides a public folder or the DM can create a private password-protected working folder for him/her. For beginners, there are also sample models available including those described in the Smart Choices book.

4. Example

We illustrate the opportunities to support the even swaps process with the Alan Miller's office selection problem considered in Hammond et al. (1998, 1999) (Figure 8). The first phases of the ProACT process are quite straightforward and, thus, we only demonstrate the support provided for the even swaps elimination process. In the example, the DM wants to minimize the consequence levels of *Commute time* and *Monthly cost* and maximize the level of the other attributes.

	Parkway	Lombard	Baranov	Montana	Pierpoint
Commute Time	45	25	20	25	30
Client Access	50	80	70	85	75
Office Services	A	B	C	A	C
Office Size	800	700	500	950	700
Monthly Cost	1850	1700	1500	1900	1750
	Prac. Dom.				Dominated

Figure 8. Alan Miller's office selection problem (Hammond et al., 1998, 1999).

At the start of the even swaps process, the software scans through the consequences table and suggests eliminating *Pierpoint*, as it is dominated by *Lombard* (Figure 8). Thus, the DM eliminates it. The software also identifies *Parkway* as practically dominated by *Montana*. The DM carefully compares these alternatives side-by-side (Figure 6) and notices that *Montana* outranks *Parkway* in each attribute except in *Monthly cost*, in which it is only 50 dollars weaker. Based on this comparison, the DM decides to eliminate *Parkway*.

Since there are no more dominated alternatives, the DM continues the process by making an even swap. However, s/he first asks the software to give suggestions for a swap that would make attributes irrelevant. Smart-Swaps provides three suggestions to make *Commute time* irrelevant by changing *Commute time* of *Baranov* from 20 to 25 and by compensating this change in *Client access*, *Office services* or *Office size*. Each of these swaps seems to be quite reasonable and the DM makes a swap in which the change in the *Commute time* of *Baranov* from 20 to 25 is compensated with a change in *Client access* from 70 to 78. Consequently, *Commute time* becomes irrelevant, and is eliminated (Figure 9).

The consequences table changes with this swap and the DM asks for new suggestions for the next swap. Now, no attribute can be made irrelevant with a single swap but any one of the attributes can be made irrelevant with two swaps. Thus, no suggestions for making an attribute irrelevant are given. However, when asking for a suggestion to make an alternative dominated, the software suggests six possible swaps. One should note that there are in fact 36 such swaps but only the ones with

the highest applicability index values are given. The first suggestion is a swap in which *Monthly cost* is used to compensate a change in *Office size* of *Montana* from 950 to 700 (Figure 3). With this swap, both *Lombard* and *Baranov* can become dominated. However, by evaluating this swap the DM sees that it is not easy to make both alternatives dominated, as this would require that the consequence of *Montana* in *Monthly cost* is chosen to be below 1500. Nevertheless, the DM makes this swap so that the suggested change in *Office size* of *Montana* is compensated with a change in *Monthly cost* from 1900 to 1650. This results in an elimination of *Lombard* as a dominated alternative but *Baranov* still remains non-dominated.

	Lombard	Baranov	Montana	
Commute Time	25	25	25	Irrelevant
Client Access	80	78	85	
Office Services	B	C	A	
Office Size	700	500	950	
Monthly Cost	1700	1500	1900	

Figure 9. The consequences table after making *Commute Time* irrelevant.

When asking for a new even swap suggestion for making an alternative dominated, the software suggests a change in *Office services* of *Montana* from A to C and a compensation in *Monthly cost*. The DM makes this swap with a compensation in the *Monthly cost* of *Montana* from 1650 to 1350. As a result, *Baranov* is eliminated, and *Montana* is the most preferred alternative.

To see whether the result is sensitive to changes in the sequence of the swaps, the DM restarts from the situation in which *Pierpoint* and *Parkway* were eliminated (Figure 8 without *Pierpoint* and *Parkway*). Initially, the DM carried out a swap that made *Commute time* irrelevant. Now s/he asks the software to suggest a swap that could make some alternatives dominated. There are three suggestions with which *Lombard* could become dominated by *Montana* by making an even swap between *Monthly cost* and either *Office size*, *Office services* or *Client access*. Of these, the DM selects a swap in which s/he changes *Monthly cost* of *Lombard* from 1700 to 1900 and compensates this with a change in *Office size* from 700 to 900, when *Lombard* becomes dominated by *Montana*. After this s/he makes two more swaps suggested by the software. These are the change in *Commute time* of *Baranov* from 20 to 25 and the compensation in *Office size* from 500 to 550, and the change in *Monthly cost* of *Montana* from 1900 to 1500 and the compensation in *Office size* from 550 to 950. As a result, *Baranov* is eliminated and *Montana* is found to be the most preferred alternative also through this other sequence of swaps. This convinces the DM of the result.

The example shows how Smart-Swaps can provide convenient help for the even swaps process. Besides procedural support in making the even swaps, the software helps the DM by providing information about the efficiency of the available swaps and about the attributes and alternatives that could be eliminated. Even in a relatively small problem there are numerous possibilities to carry out the even swaps process

but, the larger the problem is, the more useful the software will be in the screening of all the different possibilities.

5. Conclusions

Despite the superficial simplicity of the even swaps method, the procedure benefits from computer support both in the PrOACT process and in making trade-offs and managing the consequences table during the even swaps process. We demonstrate how the Smart-Swaps software provides this support and show the effectiveness of computer support with the new preference programming approach.

Even swaps is a relatively new method and its applicability in practice remains to be demonstrated. The Smart-Swaps software is designed so that a novice should be able to use the method without expert help. This should increase the number of real life applications. It also brings new attractive features to the method itself by allowing rapid backtracking as well as documentation of the steps in the process. These can be important features in practice to increase the credibility and transparency of the method. These features also make interesting future research possible, for example, on the behavioural aspects including the role of different starting strategies and the possible path dependence of the results.

Acknowledgements

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

References

- Arbel, A. (1989). Approximate Articulation of Preference and Priority Derivation, *European Journal of Operational Research*, 43, 317-326.
- Bhargava, H.K., Power, D.J. and Sun, D. (2005). Progress in Web-based Decision Support Technologies, *Decision Support Systems*. (to appear)
- Butler, J., Morrice, D.J. and Mullarkey, P.W. (2001). A Multiple Attribute Utility Theory Approach to Ranking and Selection, *Management Science*, 47(6), 800-816.
- French, S. and Xu, D.-L. (2004). Comparison Study of Multi-Attribute Decision Analytic Software, *MCDM 2004 Conference*, Whistler, B.C., Canada, August 6-11.
- Gregory, R. and Wellman, K. (2001). Bringing Stakeholder Values into Environmental Policy Choices: A Community-based Estuary Case Study, *Ecological Economics*, 39, 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, *Journal of Multi-Criteria Decision Making*, 12(2-3), 101-110.

- Hämäläinen, R.P., Mustajoki, J., Alanaatu, P., Karttunen, V. Arstila, A. and Nissinen, J. (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., Keeney, R.L. and Raiffa, H. (1998). Even Swaps: A Rational Method for Making Trade-offs, *Harvard Business Review*, 76(2), 137-149.
- Hammond, J.S., Keeney, R.L. and Raiffa, H. (1999). *Smart Choices, A Practical Guide to Making Better Decisions*, Harvard Business School Press, Boston, MA.
- INFORMS (2000). OR/MS Resource Collection: Computer Programs. Institute for Operations Research and the Management Sciences. http://www.informs.org/Resources/Computer_Programs/
- Johnson, M.P. (2005). Spatial Decision Support for Assisted Housing Mobility Counseling, *Decision Support Systems*, 41(1), 296-312.
- Kajanus, M., Ahola, J., Kurttila, M. and Pesonen, M. (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, *Operations Research*, 50(6), 935-945.
- Keeney, R.L. and Gregory, R.S. (2005). Selecting Attributes to Measure the Achievement of Objectives, *Operations Research*, 53(1), 1-11.
- Keeney, R.L. and Raiffa, H. (1976). *Decisions with Multiple Objectives. Preferences and Value Tradeoffs*, John Wiley & Sons, Inc., New York.
- Maxwell, D.T. (2004). Decision Analysis: Aiding Insight VII, *OR/MS Today*, 31(5), 44-55.
- Mustajoki, J. and Hämäläinen, R.P. (2000). Web-HIPRE: Global Decision Support by Value Tree and AHP analysis, *INFOR*, 38(3), 208-220.
- Mustajoki, J. and Hämäläinen, R.P. (2005). A Preference Programming Approach to Make the Even Swaps Method Even Easier, *Decision Analysis*, 2(2), 110-123. (Downloadable at <http://www.sal.hut.fi/Publications/pdf-files/pm05b.pdf>)
- Punkka, A. and Salo, A. (2004). Preference Programming with Incomplete Ordinal Information. Manuscript. (Downloadable at <http://www.sal.hut.fi/Publications/pdf-files/mpun04.pdf>)
- Salo, A. (1990). *Approximate Preferences in Hierarchical Decision Models*. Licentiate Thesis. Systems Analysis Laboratory, Helsinki University of Technology.
- Salo, A. and Hämäläinen, R.P. (1992). Preference Assessment by Imprecise Ratio Statements, *Operations Research*, 40(6), 1053-1061.
- Salo, A. and Hämäläinen, R.P. (2004). Preference Programming, Manuscript. (Downloadable at <http://www.sal.hut.fi/Publications/pdf-files/msal03b.pdf>)

Shim, J.P., Warkentin, M., Courtney, J.F., Power, D.J., Sharda, R. and Carlsson, C. (2002). Past, Present, and Future of Decision Support Technology, *Decision Support Systems*, 33, 111-126.

Turban, E., Aronson, J.E. and Liang, T.-P. (2004). *Decision Support Systems and Intelligent Systems*, 7th Edition, Prentice Hall Inc., New Jersey.