Available online at www.sciencedirect.com
SCIENCE DIRECT®
European Journal of Operational Research 168 (2006) 584–592
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

Decision Aiding

Nonessential objectives within network approaches for MCDM

Tomas Gal [a], Thomas Hanne [b,*]

[a] Department of Economics, Operations Research, Fern Universität Hagen, In der Krone 17, 58084 Hagen, Germany
[b] Department of Optimization, Fraunhofer Institute for Industrial Mathematics (ITWM), Europaallee 10, 67657 Kaiserslautern, Germany

Received 30 September 2002; accepted 14 April 2004
Available online 14 August 2004

Abstract

In Gal and Hanne [Eur. J. Oper. Res. 119 (1999) 373] the problem of using several methods to solve a multiple criteria decision making (MCDM) problem with linear objective functions *after* dropping nonessential objectives is analyzed. It turned out that the solution does not need be the same when using various methods for solving the system containing the nonessential objectives or not. In this paper we consider the application of network approaches for multicriteria decision making such as neural networks and an approach for combining MCDM methods (called MCDM networks). We discuss questions of comparing the results obtained with several methods as applied to the problem with or without nonessential objectives. Especially, we argue for considering redundancies such as nonessential objectives as a native feature in complex information processing. In contrast to previous results on nonessential objectives, the current paper focuses on discrete MCDM problems which are also denoted as multiple attribute decision making (MADM).

Keywords: Multiple criteria analysis; Multiple attribute decision making; Modeling; Redundancy; Nonessential objectives; Neural networks; MCDM networks

1. Introduction

Although the problem of obtaining well-defined criteria for a multiple criteria decision making (MCDM) problem is well-known (See, e.g., Bouyssou, 1992; Keeney and Raiffa, 1976, pp. 50–53; Keeney, 1992, pp. 82–87, 120; Roy, 1977, Roy and Vincke, 1984; for more general results cf. Gal (1995) and Karwan et al. (1983).), it is often neglected in MCDM theory, methods, and applications (Carlsson and Fullér, 1995; see also Carlsson and Fullér, 1994).

* Corresponding author.
E-mail addresses: tomas.gal@fernuni-hagen.de (T. Gal), hanne@itwm.fhg.de (T. Hanne).

0377-2217/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2004.04.045

One of the approaches dealing with dependent criteria is the concept of nonessential objective functions. The basic ideas of nonessential objective functions and of a methodology to determine them have been elaborated for linear multiobjective optimization problems (Leberling, 1977; Gal and Leberling, 1977; Gal, 1980). Recent results on nonlinear problems are given by Malinowska (2002).

A nonessential objective function could also be called a redundant one from the following point of view: Dropping it, the set of all efficient solutions does not change. In the corresponding literature also the notion of various systems of nonessential objectives and a minimal spanning system of objectives have been defined (Leberling, 1977). Using the algorithm for dropping nonessential objectives published in the above cited literature leads to a saving of computer time for determining the efficient set or a compromise solution because of working with a smaller system of objectives.

In Gal and Hanne (1999) the concept of dropping nonessential objective functions has been reconsidered. The main issue dealt with in this paper is whether, from the point of view of practice or of the decision maker (DM in what follows), such a reduction of the set of objectives is sensible. A central question is whether the DM obtains, when applying an MCDM method, the same solution(s) when analyzing either the original or the reduced problem which are mathematically equivalent in the sense of having the same set of efficient solutions. Assuming constant preferences the multicriteria analysis should lead to the same results for both problems. We analyzed the problem of finding parameters (e.g. weights) for an MCDM problem reduced in the above sense such that the original solution (with the nonreduced set of objectives) is preserved. Empirically, it is quite questionable whether a DM would choose these adapted parameters intuitively. Therefore, from an application oriented point of view, it matters whether nonessential objectives are dropped or not.

In Hanne (2001) concepts of neural networks and other network structures for solving MCDM problems (MCDM networks) have been elaborated. In this paper, we consider the question of dropping nonessential objective functions within the framework of these approaches. The paper focuses on multiobjective problems with a finite number of alternatives. In Section 2, some technical notations are introduced and arguments pro and contra dropping nonessential objectives are considered. In Section 3, the usage of neural networks for MCDM problems is discussed with respect to nonessential objective functions. Similarly, in Section 4, MCDM networks are considered and a simple example problem is discussed. Section 5 finalizes this paper with the conclusions.

2. Dropping nonessential objectives

2.1. Some notations

In general, an MCDM problem $P=(X,f)$ with q objective functions can be formulated as

$$\max \quad f(x) \tag{2.1}$$

$$\text{s.t.} \quad x \in X \tag{2.2}$$

with $X \subset R^n$ being the set of feasible alternatives and f: $R^n \to R^q$ representing q objective functions to be maximized. The concept of nonessential objective functions has been introduced for MCDM problems with q linear objectives, i.e. f defined by

$$f(x) = Cx \tag{2.3}$$

with $C \in R^{q \times n}$. Special assumptions about X are not necessary although the corresponding methods have been developed for the case of X being a convex polyhedron given as

$$X = \{x \in R^n : Ax \leqq b, \ x \geqq 0\} \tag{2.4}$$

with $A \in R^{m \times n}$, $b \in R^m$ (multiple objective linear programming, MOLP).

Another case is when X is discrete. Should the set of alternatives be a finite set, $X=\{x_1,\ldots,x_p\}$, an MCDM problem can be defined in a more simple way. The objective or criteria values for each feasible alternative can simply be represented by a decision matrix

$$Z = (z_{ij}), \quad i \in \{1,\ldots,p\}, \quad j \in \{1,\ldots,q\}, \tag{2.5}$$

where z_{ij} is the value of objective j for alternative x_i, $z_{ij}=f_j(x_i)$. Such a problem is usually called a multiple attribute decision making (MADM) problem.

In a mathematical sense, usually the set of efficient alternatives is considered as the solution of an MCDM problem. This set is defined as

$$\mathrm{Eff}(X,f) := \{x \in X : \nexists y \in X \text{ such that } f(y) \geqslant f(x)\}, \tag{2.6}$$

where $f(y) \geqslant f(x)$ means that $f_j(y) \geqslant f_j(x)$ for all $j \in \{1,\ldots,q\}$ and $f_j(y) > f_j(x)$ for at least one $j \in \{1,\ldots,q\}$. For $x, y \in X$ with $f(y) \geqslant f(x)$, we also write $y \succ_f x$ or, if clear, simply $y \succ x$.

An objective function f_j, $j \in \{1,\ldots,q\}$, j fixed, is then called *nonessential* if the set of efficient solutions does not change when f_j is removed from the problem. The Gal and Leberling method for finding nonessential functions in an MOLP (Gal and Leberling (1977), Gal (1980)) leads to a dropping of (some) nonessential objectives (if there are any) such that P defined by (2.1)–(2.4) can be replaced by P' defined by

$$\begin{aligned} \max \quad & C'x \\ \text{s.t.} \quad & x \in X \end{aligned} \tag{2.7}$$

with $C' \in R^{q' \times n}$, $q' \leqslant q$.

C' can be expressed as $C' = TC$ where $T \in R^{q'} \times q$ is a transformation matrix. The rows in T which correspond to nonessential, dropped objectives are 0 vectors while the other rows are unit vectors.

Let E be the index set of nondropped objectives (essential objectives and possibly some nonessential objectives which are not dropped) and N the index set of dropped objectives ($E \cup N = \{1,\ldots,q\}$, $E \cap N = \emptyset$). With respect to the Gal and Leberling theory, nonessential objectives can be expressed as nonnegative linear combinations of other objective functions (as follows also from the cone dominance theory by Yu (1974)): For all $k \in N$: $\exists \lambda_{jk} \geqslant 0$, $j \in E$, such that

$$\sum_{j \in E} \lambda_{jk} c_j = c_k, \tag{2.8}$$

where c_j and c_k are the jth and kth row of C respectively.

In the case of an MADM problem we can similarly apply the idea of dropping nonessential objectives in the sense of deleting the corresponding columns from the decision matrix Z.

2.2. Pros and cons

One of the most obvious ambitions in dropping nonessential objective functions is to present a more 'consistent' MCDM problem to the DM (see, e.g., Decision Analysis Society, 2002). Even in general guidelines for decision making (see, e.g., Johnson et al., 2002) a multiobjective decision problem affected by nonessential or redundant criteria is assumed to lead to some misleading 'double counting'. On the other hand, it is clear that human information processing and biological information processing in general (e.g., Zurada et al., 1997) is based on a huge amount of redundant information. Biological nerve nets, for instance, have evolved as a powerful tool for coping with redundant information. Often, redundant information, in nature as well as in technical applications, is used for getting more security about the results of information processing. Some important examples from different areas are the utilization of genetic code, image processing, the processing of natural language, or the prediction of bankruptcies and other economic issues. For instance, a DM might want to consider highly correlated objectives such as ratio numbers in balance-sheet analysis with similar meanings for gaining more safety in rating a company (Hanne, 1995). Thus, it is not clear whether a DM analyzing a problem with similar and/or redundant criteria actually does some double counting or whether the criteria are purposefully chosen. It is therefore not clear whether a DM actually could provide more reliable data (e.g. weights) after dropping nonessential objectives if he/she still thinks in categories of the unreduced problem, i.e. including nonessential objective functions (see Gal and Hanne, 1999).

As a more technical reason for dropping nonessential objective functions one might consider aspects of computer efficiency. It should be possible to solve a 'smaller' MCDM problem with less computational requirements, esp. less CPU time. Also, a DM could benefit from determining a fewer number of parameters when applying an MCDM method (see Gal and Hanne, 1999).

Especially in the context of interactive MCDM methods, a user should welcome to work with a smaller number of objective functions or criteria.

In the following, we will defer these general questions concerning the usefulness of dropping nonessential criteria. Instead, we discuss problems concerning the equivalence of results when analyzing problems with or without dropping nonessential objectives in the context of using complex methods such as neural networks or MCDM networks.

3. Neural networks and nonessential objectives

Neural networks are a promising tool capable of machine learning which can be applied to multiobjective decision problems (Malakooti and Zhou, 1994). Such networks, especially those of the feedforward type, are used usually for calculating a scalarizing function for evaluating alternatives characterized by several objectives (see, e.g., Wang and Malakooti, 1992). In this case, it is assumed that the neural network gets q inputs, hence the objective values $f_1(x),\ldots,f_q(x)$ of an alternative x to be aggregated to a result $F(f(x)) \in R$. The neural network function F: $R^q \to R$, in general, depends on weights and other parameters of the neural network. As is usual in the literature[1] concerning MCDM applications, the neural network is assumed to be constructed, e.g., by adapting the weights, such that it represents the DM's preferences. This is best accomplished if $F(f(x)) > F(f(y))$ is equivalent to xPy for all $x, y \in X$ where xPy means that alternative x is preferred to alternative y.

Basically, a feedforward neural network can be considered as a finite directed graph of nodes (corresponding to neurons) which are organized in layers. Edges (and thus information transfer) which are marked by weights are only allowed from one layer to the subsequent one. Usually, it is assumed that neighboring layers are 'fully connected' as shown in Fig. 1. The output S_j of a neuron j depends on its input values. The input values of the first layer neurons are the input values of the neural network. Those of the other neurons are the output values of the neurons of the preceding layer.

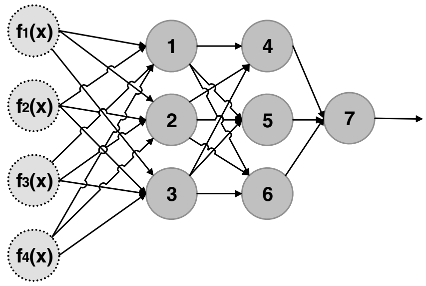

Fig. 1. A feedforward neural network for evaluating alternatives characterized by four criteria.

Usually, S_j is calculated by a function

$$S_j = \psi\left(\sum_{i \in V(j)} w_{ij} \cdot S_i - \theta_j\right), \tag{3.1}$$

where $V(j)$ is the set of nodes or network inputs preceding j and S_i are the states of the preceding neurons or the input values for first layer neurons. The w_{ij} are weights and θ_j is a threshold value. The weights $w_{ij} \in [-1, 1]$[2] indicate in which proportion the output of the ith neuron (or input) is considered as exciting ($w_{ij} > 0$) or inhibitive ($w_{ij} < 0$) for calculating a weighted sum as the output value of the jth neuron. The output function ψ is an increasing function with $\lim_{x\to\infty}\psi(x) = 1$ and, depending on the network model, $\lim_{x\to-\infty}\psi(x) = 0$ or -1. For instance, the following function with a parameter $\beta > 0$ is a valid model for ψ:

$$\psi(x) = (1 + \mathrm{e}^{-2\beta x})^{-1}. \tag{3.2}$$

Further details on neural networks can, for instance, be found in Hecht-Nielsen (1989). As assumed above, the neural network is used for

[1] See Malakooti and Zhou (1994), Malakooti and Raman (2000) and Sun et al. (1996) for further results on using neural networks for MCDM. Hanne (2001) provides additional references on the usage of neural networks for MCDM. Feedforward networks processing complete representations of discrete MCDM problems (instead of those processing alternatives separately) have been considered there as well.

[2] These weights which may be negative should not be confused with weights in the context of traditional scalarization approaches in MCDM which are required to be positive in order to guarantee the calculation of an efficient solution.

calculating a scalarizing function. Thus, there is only one neuron in the final layer and its output is interpreted as the result of the neural network.

If nonessential objectives of the MCDM problem are dropped then a neural network can be adapted (see below) such that it calculates a result in accordance with the result in case of not dropping nonessential objectives. First of all, it is assumed that the adapted neural network has just $q' < q$ input values, each of them corresponding to one of the q' objective values of the MCDM problem with dropped nonessential objectives. Secondly, the neurons of the first layer of the neural network have to be adapted as stated in the following lemma.

Lemma. *Assume that an MCDM problem with nonessential objectives $k \in N$ is processed by a feedforward network as specified above ((2.8) and (3.1)). Then for the reduced problem the same results are obtained if each first layer neuron j processes q' input values applying modified weights w'_{ij} defined as follows*:

$$w'_{ij} = w_{ij} + \sum_{k \in N} w_{kj} \lambda_{ik}. \tag{3.3}$$

Proof. Consider one of the input neurons of the modified neural network for an MCDM problem with dropped nonessential objectives. Such a neuron j calculates an output value $S_j = \psi(\sum_{i \in V(j)} w'_{ij} S_i - \theta_j)$. Let the weights w'_{ij} be defined as $w'_{ij} = w_{ij} + \sum_{k \in N} w_{kj} \lambda_{ik}$. Assuming input values $z_i = f_i(x)$, $i \in \{1, \ldots, q\}$, for some $x \in X$, the aggregated input of a first layer neuron j then is:

$$\begin{aligned}
\sum_{i \in E} w'_{ij} z_i &= \sum_{i \in E} \left(w_{ij} + \sum_{k \in N} w_{kj} \lambda_{ik} \right) z_i \\
&= \sum_{i \in E} \left(w_{ij} z_i + \sum_{k \in N} w_{kj} \lambda_{ik} z_i \right) \\
&= \sum_{i \in E} w_{ij} z_i + \sum_{i \in E} \sum_{k \in N} w_{kj} \lambda_{ik} z_i \\
&= \sum_{i \in E} w_{ij} z_i + \sum_{k \in N} w_{kj} \sum_{i \in E} \lambda_{ik} z_i \\
&= \sum_{i \in E} w_{ij} z_i + \sum_{k \in N} w_{kj} z_k = \sum_{i=1}^{q} w_{ij} z_i.
\end{aligned}$$

Note that (2.8) has been used in the last but one step of the transformation. Since the input nodes calculate the same results as those of the neural network for the original MCDM problem, the output of the modified neural network is also the same. □

Note that the weights calculated according to (3.3) do not necessarily fulfill the required condition $w'_{ij} \in [-1, 1]$. This cannot be changed in an easy way because a rescaling of weights would lead to a different output of the neurons according to (3.1).

For neural networks, it is often emphasized (see, e.g., Torgo, 1993) that they have specific advantages for processing information affected by redundancies. Such redundancies can, for instance, be found in sensory information. Neural networks process such information efficiently and can eliminate redundancies. Therefore, it is questionable whether pre-processing inputs for eliminating redundant information is useful (e.g. considering running time aspects of computer software). For purposes of MCDM, this means that the application of a method for dropping nonessential objectives is possibly not necessary.

4. MCDM networks and nonessential objectives

4.1. MCDM Networks

MCDM networks have been introduced as a novel approach for a combined application of several MCDM methods to a given MCDM problem (Hanne, 2001). In this approach, MCDM methods are associated to nodes organized by a network structure or a directed graph. Just like in neural networks these nodes may process information in a parallel and/or sequential way. However, each node does not process a vector of scalar inputs (like a neuron) but a complete MCDM problem assuming a finite set of alternatives (i.e. an MADM problem) represented by appropriate complex data structures. For instance, a sequential composition of the methods 'eliminate inefficient alternatives' and 'apply utility function' corresponds to the idea of first eliminating infeasible alternatives and then applying a utility function to the remaining alternatives (see below for a more complex example).

The basic idea of MCDM networks is that methods corresponding to nodes of the first layer process the input MCDM problem of the network. Those method nodes from subsequent layers process the aggregated results of the preceding method nodes. Such a process is interpreted as a *meta decision problem* (see Hanne, 2001) in the sense that several methods (instead of just one) may be applied to a given MCDM problem. This should reduce the DM's incertitude concerning the selection of an MCDM method from the multitude of available ones which, usually, would lead to different solutions. A more formal treatment of MCDM networks is given in Hanne (2001). Below, we discuss an example for illustrating this approach.

Besides traditional MCDM methods, this concept may also include neural networks which may correspond to *one* node of the graph. This concept was implemented in a decision support system called LOOPS (Learning Object-Oriented Problem Solver) which includes, besides other features, a mechanism for machine learning based on evolutionary algorithms.

When considering necessary modifications for dealing with MCDM problems reduced by dropping nonessential criteria, the following result is rather obvious: Each of the methods corresponding to the first layer nodes (respectively its parameters) has to be adapted, if this is possible for that method, such that it calculates the same result for the modified MCDM problem as for the original one in order to make the MCDM network calculate the same solution for the modified problem.

This proceeding, however, does not guarantee that within MCDM networks intermediate results including nonessential objectives are calculated. It is also possible that the result of the MCDM network (i.e. the solution of the considered MCDM problem described by the results of the method nodes) has nonessential objectives although the input problem is free of them, e.g. by dropping nonessential objectives. Let us illustrate these findings related to nonessential objective functions by an example of MCDM networks.

4.2. An example

The simple MADM problem shown in Table 1 concerns the choice of a notebook. As criteria we consider various attributes expressing the power of the notebook and its price. The price is to be minimized while all other criteria are to be maximized.

Note that a minimization criterion, i.e. the price, can easily be converted into a maximization criterion multiplying it by −1.

The efficient set of the problem consists of n1, n2, n3, and n5. n4 is inefficient because it is dominated by n3. Criterion 3, the display size, is nonessential because dropping it the efficient set does not change. Therefore, we consider the dropping of this criterion.

Note that, in general, it is not possible to adapt the considered methods such that the same results are obtained as without dropping a nonessential objective. For instance, an aspiration level (see below) defined for a nonessential objective which can be expressed as a positive linear combination of at least two other objective functions cannot simply be replaced by changing the aspiration levels on the other objective functions (see Gal and Hanne, 1999). Therefore, we do not explicitly consider the problem of parameter adaptation for the dropping of criterion 3. Instead, parameters are directly determined for the reduced MADM problem as shown below.

For the application of an MCDM method we apply a standardization of each criterion to the interval [0, 1]. This is done by the transformation

$$f'_k(x) = \frac{f_k(x) - f_k^{\min}}{f_k^{\max} - f_k^{\min}}$$

Table 1
Example problem on notebook selection

Notebook	Speed (MHz)	RAM (Mbytes)	Display (inches)	Price (Euro)
n1	1200	128	14	1119
n2	1200	256	14	1399
n3	1300	256	15	1449
n4	1200	256	12	1449
n5	1600	256	15	1949

Table 2
Standardized notebook selection problem

Criterion	Speed (MHz) c1	RAM (Mbytes) c2	Price (Euro) c4
n1	0	0	1
n2	0	1	0.66
n3	0.25	1	0.6
n4	0	1	0.6
n5	1	1	0

with $f_k^{\min} = \min_{x \in X} f_k(x)$ and $f_k^{\max} = \max_{x \in X} f_k(x)$ for $k \in \{1, \ldots, q\}$. The resulting MADM problem is shown in Table 2.

Consider the following three MCDM methods which can be used for solving the MADM problem:

1. Simple Additive Weighting (SAW). This approach is based on calculating a weighted score for each alternative $x \in X$ by some function $\sum_{i=1}^{q} w_i f_i(x)$. In the example, we use the weights: 0.2, 0.3, and 0.5. Thus, the common weight of the 'power' criteria c1 and c2 equals the weight of the price.
2. Conjunctive Levels (CL). This approach is based on the consideration of minimum requirements (aspiration levels) for each criterion. Only alternatives which reach the levels for all criteria are considered as 'feasible'. In the example, we use an aspiration level for the criterion c4 of 0.66, i.e. the price of the notebook is required to be at least 66 percent better than the worst evaluation or, in absolute terms, it should be lower or equal than 1400 Euro. The levels for the other criteria are assumed to be 0. (If, for instance, all criteria would be assumed to be 0.3 then no alternative would be feasible considering these aspiration levels.)
3. A reference point (RP) approach. This method is based on the computation of the alternative's distances to a reference point $\hat{y} \in R^q$ with respect to an l_2 metrics, i.e. $(\sum_{i=1}^{q} w_i(\hat{y}_i - f_i(x))^2)^{0.5}$. The reference point $\hat{y}$ is assumed to be (1.0, 1.0, 1.0), i.e. the utopia point of the MADM problem consisting of the component-wise best criteria evaluations. The metrics weights all criteria equally, i.e. $w_i = 1/3$ for $i \in \{1, \ldots, 3\}$.

Table 3
Results of 3 MCDM methods

	SAW	CL	RP
n1	0.5	1	0.18
n2	0.53	1	0.39
n3	0.58	0	0.51
n4	0.5	0	0.38
n5	0.5	0	0.42

Details on these methods can be found in Hwang and Yoon (1981). The results obtained by the considered MCDM methods are presented in Table 3. Each column represents the results of one of the applied MCDM methods. The *i*th component of a column vector represents the evaluation of the alternative x_i according to the respective MCDM method. For the CL method, a value of 1 indicates that an alternative is feasible according to the aspiration levels, a value of 0 implies that it is not feasible. The RP approach returns for each alternative the distance to the ideal solution. Since this value d is desired to be as small as possible, we convert it into a maximization criterion by a transformation to $(1 - d)$.

In Fig. 2 the proceeding of applying the three MCDM methods in parallel is illustrated with an MCDM network. This result can be interpreted as a level 2 MADM problem with the 3 method results as criteria, thus the decision matrix shown in Table 3.

Note that, although the original decision problem is purified from nonessential objectives, the meta problem (Hanne, 2001) is affected by nonessential criteria, the result of the SAW and the RP method, because the efficient set, n2 and n3, does not change when either the criterion 'SAW' or the criterion 'RP' is dropped. Thus, redundancy

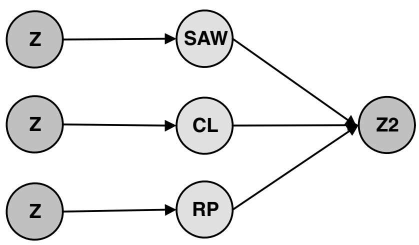

Fig. 2. Representation of the application of 3 MCDM methods by an MCDM network.

may result within the MCDM network. Such redundancy can, of course, be eliminated by dropping nonessential objective functions from the decision matrix which may, for instance, be processed furthermore by another (second level) MCDM method. It is, however, questionable whether such repetitive applications of methods for redundancy elimination (and parameter adaptations) pay off with respect to computer time etc. compared to the processing of MADM problems including nonessential criteria.

5. Conclusions

In this paper, we have reconsidered the problem of nonessential objectives in MCDM problems, i.e. those which do not change the efficient set when they are removed. Results obtained here and in earlier work show that the dropping of nonessential objectives often requires more or less sophisticated adaptations of the parameters of MCDM methods. It is often not possible at all to perform adaptations of the parameters of an MCDM method such that its application with or without dropping some nonessential objective functions leads to the same results.

On the other hand, we have seen that, for instance, network structures such as neural networks and MCDM networks inherently work with redundant information and are capable of handling this. As nature suggests, there are also good reasons for maintaining such redundancies, last but not least because of the computational costs (required time) for removing nonessential objectives. A double counting as often assumed in literature on decision making and decision analysis is not that obvious in each case, especially since the DM's attitude towards nonessential criteria is unclear. However, these subjects are frequently neglected in decision theory and practice. We, therefore, hope to stimulate the discussion on redundancies in practical decision problems.

References

Bouyssou, D., 1992. Building criteria: A prerequisite for MCDA. In: Bana e Costa, C.A. (Ed.), Readings in Multiple Criteria Decision Aid. Springer, Berlin, pp. 58–80.

Carlsson, C., Fullér, R., 1994. Interdependence in fuzzy multiple objective programming. Fuzzy Sets and Systems 65, 19–29.

Carlsson, C., Fullér, R., 1995. Multiple criteria decision making. The case for interdependence. Computers & Operations Research 22 (3), 251–260.

Decision Analysis Society, 2002. A Lexicon of Decision Making. Available from: <http://faculty.fuqua.duke.edu/daweb/lexicon.htm>.

Gal, T., 1980. A note on size reduction of the objective function matrix in vector maximum problems. In: Fandel, G., Gal, T. (Eds.), Multiple Criteria Decision Making: Theory and Application. Springer, Berlin, pp. 74–84.

Gal, T., 1995. Postoptimal Analyses, Parametric Programming, and Related Topics. Degeneracy, Multicriteria Decision Making, Redundancy, second ed. Walter de Gruyter, Berlin.

Gal, T., Hanne, T., 1999. Consequences of dropping nonessential objectives for the application of MCDM methods. European Journal of Operational Research 119, 373–378.

Gal, T., Leberling, H., 1977. Redundant objective functions in linear vector maximum problems and their determination. European Journal of Operational Research 1, 176–184.

Hanne, T., 1995. An application of different MCDM methods to bank balance sheet analysis. In: Derigs, U., Bachem, A., Drexl, A. (Eds.), Operations Research Proceedings 1994. Springer, Berlin, pp. 506–511.

Hanne, T., 2001. Intelligent Strategies for Meta Multiple Criteria Decision Making. Kluwer, Boston.

Hecht-Nielsen, R., 1989. Neurocomputing. Addison-Wesley, Reading.

Hwang, C.-L., Yoon, K., 1981. Multiple Attribute Decision Making. Methods and Applications. Springer, Berlin.

Johnson, G.V., Baker, D.J., Sorenson, K.B., Bocke, S.G., 2002. Guidance tools for use in nuclear material management decision making. Paper presented at the WM'02 Conference, 24–28 February 2002, Tucson, AZ. Available from: <emi-web.inel.gov/NMFA/WM02%20Papers/WM2002-GuidanceTools.pdf>.

Karwan, M.H., Lotfi, V., Telgen, J., Zionts, S. (Eds.), 1983. Redundancy in Mathematical Programming: A State-of-the-Art Survey. Springer, New York.

Keeney, R.L., 1992. Value-Focussed Thinking. Harvard University Press, Cambridge, MA.

Keeney, R.L., Raiffa, H., 1976. Decisions with Multiple Objectives: Preferences and Value Tradeoff. Wiley, New York.

Leberling, H., 1977. Zur Theorie der linearen Vektormaximumprobleme, Dissertation, RWTH Aachen.

Malakooti, B., Zhou, Y.Q., 1994. Feed forward artificial neural networks for solving discrete multiple criteria decision making problems. Management Science 40 (11), 1542–1561.

Malakooti, B., Raman, V., 2000. Clustering and selection of multiple criteria alternatives using unsupervised and supervised neural networks. Journal of Intelligent Manufacturing 11, 435–453.

Malinowska, A., 2002. Changes of the set of efficient solutions by extending the number of objectives. Cybernetics and Control 31 (4), 965–974.

Roy, B., 1977. A conceptional framework for a prescriptive theory of "decision-aid". In: Starr, M.K., Zeleny, M. (Eds.), Multiple Criteria Decision Making. TIMS Studies in the Management Science 6, 179–210.

Roy, B., Vincke, P., 1984. Relational systems of preference with one or more pseudo-criteria: Some new concepts and results. Management Science 30 (11), 1323–1335.

Sun, M., Stam, A., Steuer, R.E., 1996. Solving multiple objective programming problems using feed-forward artificial neural networks: The interactive FFANN procedure. Management Science 42 (6), 835–849.

Torgo, L., 1993. Controlled redundancy in incremental rule learning. In: Proceedings of the European Conference on Machine Learning (ECML-93). In: Brazdil, P. (Ed.), Lecture Notes in Artificial Intelligence, vol. 667. Springer, Berlin, pp. 185–195.

Wang, J., Malakooti, B., 1992. A feedforward neural network for multiple criteria decision making. Computers & Operations Research 19 (2), 151–167.

Yu, P.L., 1974. Cone convexity, cone extreme points, and nondominated solutions in decision problems with multiobjectives. Journal of Optimization Theory and Applications 14 (3), 319–377.

Zurada, J.M., Malinowski, A., Usui, S., 1997. Perturbation method for deleting redundant inputs of perceptron networks. Neurocomputing 14, 177–193.