Acta Psychologica 80 (1992) 67–89
North-Holland

Evaluating options by combined qualitative and quantitative methods

Marko Bohanec [a], Božo Urh [a] and Vladislav Rajkovič [a,b]

[a] 'Jozef Stefan' Institute, Ljubljana, Slovenia
[b] University of Maribor, Kranj, Slovenia

This paper addresses two problems related to qualitative decision making: option ranking and non-sensitivity to small differences between options. In general, only a partial order of options can be established by a qualitative model, which might be insufficient particularly when the number of options is large. A qualitative model is also incapable of discriminating between slightly different options. In the paper, a solution is proposed that is based on an automatic construction of a quantitative evaluation model from the qualitative one. In addition to a qualitative class, a quantitative utility is obtained for each option, which is used to rank options within classes and to reflect the sensitivity to small differences between options.

Introduction

The traditional approach to multi-attribute decision making (MADM) involves *quantitative* concepts such as probabilities, utilities, scores and weights (Keeney and Raiffa 1976; Chankong and Haimes 1983; French 1986). In contrast, *qualitative* methods deal with descriptive values expressed by ordinal symbols or words. Qualitative methods have been one of the central issues of artificial intelligence and pattern recognition research. Ordinal properties were studied and utilized in statistical clustering (Anderberg 1973). They were also important for the development of fuzzy linguistic variables (Zadeh 1975; Zimmermann 1987). Another related area is qualitative modelling of physical systems (Forbus 1981; Kuipers 1986).

Qualitative methods also influenced the areas of decision support systems (DSS) and MADM, particularly from the practical and ap-

Correspondence to: M. Bohanec, 'Jozef Stefan' Institute, Jamova 39, 61111 Ljubljana, Slovenia. E-mail: marko.bohanec@ijs.si

plied side. During the last decade, various approaches and computer-based systems have been developed that can be classified as qualitative or at least incorporate some qualitative elements. They are based on the integration of DSS or MADM with techniques, developed mainly in the framework of artificial intelligence, most notably expert systems and machine learning (Mitra 1986; Pomerol 1988; Turban 1988; Klein and Methlie 1990; Larichev et al. 1991). The list of such systems is becoming rather extensive, so let us mention only a few that are closely related to MADM: ZAPROS (Moshkovich 1991), ASTRIDA (Berkeley et al. 1990), DECMAK (Bohanec et al. 1983; Bohanec and Rajkovič 1987), DEX (Bohanec and Rajkovič 1990) and the nameless system of Lévine et al. (1990). The related research in psychology is reviewed in Moshkovich (1991).

Practical applications have confirmed that the qualitative knowledge-based approach to decision support is not only feasible, but can also provide some valuable features, such as: explicit and flexible knowledge representation (Fox 1985), transparency (Lévine et al. 1990), natural languague dialogues (Vari and Vecsenyi 1988), explanation of reasoning (Efstathiou and Mamdani 1986), and presentation of knowledge from different viewpoints at different levels of detail (Rajkovič and Bohanec 1991).

Nevertheless, there are some useful features of quantitative decision-making methods that are difficult to obtain with qualitative approaches. These include option ranking and option evaluation which are sensitive to small differences of attribute values. It seems that a reasonable *combination* of the two approaches is needed to provide this type of functionality.

This paper presents an *extension* of a qualitative MADM method DECMAK (Bohanec and Rajkovič 1987; Rajkovič et al. 1988) with some quantitative elements that are used for the *evaluation* of options. In addition to the qualitative evaluation that has been used so far, each option is separately (but consistently) evaluated by a quantitative method. Therefore, two evaluation results are obtained for each option: a qualitative *class* and a quantitative *utility*. The utility is used to rank options within classes and to reflect the sensitivity to small variations of attribute values.

The outline of the paper is as follows. The following section introduces DECMAK. Next, the problems of option ranking and sensitivity are presented. Solutions to these problems are proposed in the following two sections. Two methods for the construction of quantitative evaluation models from rules in the knowledge base are described and illustrated by an application in personnel management. Finally, the possible contributions of this approach are discussed and summarized.

DECMAK: A qualitative approach to MADM

DECMAK closely follows the main idea of MADM, which is based on the decomposition of a decision problem into smaller, less complex problems. Options are decomposed onto different dimensions, usually called *attributes, performance variables, criteria*, etc. These are evaluated independently. The total utilities of options are then obtained by some aggregation procedure, for example a weighted sum. The obtained utilities are finally used to select the best option or to rank options.

In DECMAK, the basic MADM approach is combined with some elements of expert systems and machine learning. Attributes and aggregation procedures are treated as an explicit *knowledge base* that consists of: (1) one or more *trees of attributes*, (2) aggregation procedures (called *utility functions*), and (3) descriptions of *options*. These components are described below.

Tree of attributes

A tree of attributes represents the structure of a decision problem. Attributes are structured according to their interdependence: a higher-level attribute depends on its descendants (sons) in the tree. Leaves of the tree, referred to as *basic attributes*, depend on options. Internal nodes in the tree are called *aggregate attributes*. Their values are determined by utility functions. The root of the tree represents the overall classification of options.

An important difference with more common quantitative approaches is that attributes are *qualitative* in DECMAK. They can take values from discrete and (optionally) ordered value domains. The values are words such as *high* or *acceptable*, or numeric intervals, for example *$100–200*.

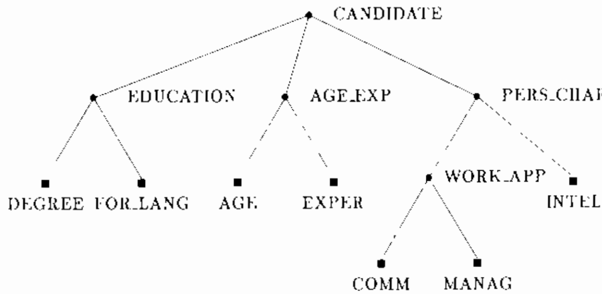


Fig. 1. Tree of attributes for employee selection.

To illustrate these concepts, consider a common problem of personnel management: selecting the best candidate for a particular job. The quality of options, i.e., candidates, can be determined by a tree of attributes that is shown in fig. 1. There are three main groups of attributes: (1) education of the candidate, (2) age and experience, and (3) personal characteristics. The appropriateness of candidates' education depends on the degree of education and mastering of foreign languages. The remaining two groups of attributes are decomposed in a similar way (fig. 2). Note, however, that the tree presented here is an oversimplification of the real problem of employee selection. In one of the real-life applications (Črnivec et al. 1988), the tree consisted of 40 attributes.

Fig. 2 also presents qualitative values that can be assigned to attributes. The majority of values are words like *unacc*eptable, *accept*-able, *poor*, etc. The values of AGE and EXPER are numeric intervals, measured in years. Note that the values of all the aggregate

CANDIDATE	Candidate for employment	*unacc, accept, good, excel*
— EDUCATION	Education of the candidate	*unacc, accept, approp*
—— DEGREE	Degree of education	*under, B.Sc., M.Sc., Ph.D.*
—— FOR_LANG	Foreign languages	*none, passive, active*
— AGE_EXP	Age and experience	*unacc, accept, approp*
—— AGE	Age of the candidate	*18–20, 21–25, 26–40, 41–55, more*
—— EXPER	Experience (in years)	*none, <1, 1–5, 6–10, >10*
— PERS_CHAR	Personal characteristics	*unacc, accept, approp*
—— WORK_APP	Work approach	*less_acc, accept, good*
——— COMM	Communicability	*poor, average, good, excel*
——— MANAG	Management abilities	*poor, average, good*
—— INTEL	Intelligence test	*d, c, b, a*

Fig. 2. Description and value domains of attributes for employee selection.

attributes are ordered from non-preferred ('bad') to highly preferred ('good') values.

Utility functions

Utility functions define the aggregation of lower-level attributes into the corresponding aggregate (higher-level) ones. For each aggregate attribute X, a utility function that maps the sons of X into X should be defined by the decision maker.

In contrast to quantitative MADM, where utility functions are defined by a certain formula (say, a weighted sum), DECMAK uses *decision rules* for this purpose. Consider the WORK_APP attribute in fig. 1 which depends on two other attributes, COMM and MANAG. The aggregation of COMM and MANAG into WORK_APP can be defined by rules as shown in table 1; each line represents a simple *if–then* rule. For example, the fourth line can be interpreted as

if COMM = *excel* **and** MANAG = *poor*
then WORK_APP = *less_acc.*

In table 1, all the 12 possible combinations of values of COMM and MANAG are shown. Note, however, that the decision maker does not need to define all these combinations. Only the most significant rules

Table 1
A table of decision rules.

	COMM	MANAG	WORK_APP
1	*poor*	*poor*	*less_acc*
2	*average*	*poor*	*less_acc*
3	*good*	*poor*	*less_acc*
4	*excel*	*poor*	*less_acc*
5	*poor*	*average*	*less_acc*
6	*poor*	*good*	*less_acc*
7	*average*	*average*	*accept*
8	*good*	*average*	*accept*
9	*average*	*good*	*accept*
10	*excel*	*average*	*good*
11	*good*	*good*	*good*
12	*excel*	*good*	*good*

	description	evaluation
• CANDIDATE		*good*
• EDUCATION		*approp*
• DEGREE	*M.Sc.*	
• FOR_LANG	*passive*	
• AGE_EXP		*accept*
• AGE	*21–25*	
• EXPER	*<1*	
• PERS_CHAR		*approp*
• WORK_APP		*good*
• COMM	*good*	
• MANAG	*good*	
• INTEL	*b*	

Fig. 3. Description and evaluation of a candidate.

should be defined, while the remaining are obtained by an approximation method (Bohanec and Rajkovič 1988). In addition, the acquisition of rules is actively supported by a computer using various man–machine dialogues (Bohanec and Rajkovič 1988; Rajkovič et al. 1988) and knowledge explanation methods (Bohanec et al. 1988; Rajkovič and Bohanec 1991).

Options

In DECMAK, options are treated in three main stages: (1) description of options, (2) evaluation and (3) analysis of options. In the first stage, the decision maker describes options (i.e., candidates in the employee selection example) by assigning values to basic attributes (i.e., leaves of the tree). An example of such an assignment for one candidate is shown in the 'description' column of fig. 3.

In the evaluation stage, a bottom-up procedure is applied in order to obtain the overall classification of options. To illustrate the procedure, consider the candidate in fig. 3 and note that the value *good* was assigned to basic attributes COMM and MANAG. This assignment matches rule 11 in table 1, which assigns *good* to the corresponding aggregate attribute WORK_APP. This process continues until a value is assigned to the root of the tree which represents the overall classification of the candidate. The intermediate and final values are shown in the 'evaluation' column of fig. 3. Thus, the final classification of the candidate is *good*.

The evaluation is followed by a thorough analysis of options where the results are justified and explained. DECMAK provides a number of methods that explain the evaluation process to the decision maker, perform *what–if* analysis, highlight the advantages and disadvantages of each option and selectively compare options. These methods are presented and discussed elsewhere (Bohanec and Rajkovič 1989).

Applications and experience

DECMAK has been used in about forty complex decision-making problems in industrial, governmental, educational and research institutions in Yugoslavia, Italy and Peru. The main application areas were the following:

(1) evaluation of computer hardware and software (Bohanec et al. 1983; Bohanec and Rajkovič 1988),
(2) personnel management (Črnivec et al. 1988; Olave et al. 1989),
(3) performance evaluation of enterprises (Barrera and Bohanec 1987; Bohanec and Rajkovič 1990).

DECMAK performed particularly well in complex decision problems in which decisions depended on qualitative judgment and expert rules rather than exact mathematical models. In problems of this type, some important advantages with respect to conventional approaches were observed, most notably:

(1) qualitative knowledge representation and reasoning,
(2) powerful and flexible tools for knowledge acquisition,
(3) transparency and explainability of knowledge and evaluation results.

On the other hand, the qualitative nature of DECMAK raised two practical problems which do not occur in quantitative MADM: option ranking and non-sensitivity to small variation of basic attribute values. These problems are presented in the following section.

Two problems of qualitative MADM

Option ranking

There is an important difference between quantitative and qualitative MADM related to option ranking. Quantitative evaluation yields a numeric utility for each option. In principle, the total ranking of options can be established on this basis. On the other hand, qualitative evaluation determines only a partial order of options. Options are classified into a relatively small number of distinct classes, such as *unacc*, *accept*, *good* and *excel* in the employee selection example. The model does not discriminate between the options *within* the same class. This may occur even in the case of strict domination, as shown by the following example.

Consider the candidate (A) presented in fig. 3. Suppose another candidate (B) whose basic attribute values are equal to the A's except that his communicability (COMM) is *excel*lent rather than *good*. Rules 11 and 12 in table 1 assign WORK_APP = *good* for both candidates. Since there is no other difference between A and B, both candidates achieve the same final classification, *good*. In other words, the final class does not discriminate between the candidates, although B is better than A.

Additional analysis is, therefore, required to identify the best option within a particular class or to make the total rank of options. In DECMAK, this is supported by a semi-automatic procedure called *selective comparison of options* (Bohanec and Rajkovič 1989). The automatic part performs a *pairwise* comparison of options. For each selected pair, the most important differences between the options are found. They are presented to the decision maker who is then responsible to determine which of the two options is better. In the above example, the only difference concerns the COMM attribute (*good* vs. *excel*), so it is easy to see that B dominates A.

Although highly subjective, the above method performs well in practice, provided that the number of similar options is relatively small, say, about five. Otherwise, the number of pairwise comparisons increases considerably and requires a lot of time and effort from the decision maker. In personnel management decisions, for example, it is not uncommon to deal with a substantial number of candidates. When they are evaluated qualitatively using, say, only four classes, it might be extremely difficult to rank them by the pairwise comparison procedure. Therefore, a more effective method is needed for option ranking in such cases.

There are two possible solutions. The first is to increase the number of classes. This is a limited solution which may help in some situations. In general, however, it does not guarantee that the total order of options will be established nor that the number of required comparisons will be significantly reduced. In addition, a more detailed decision model is needed to cope with more classes. This requires the modification of existing and the addition of new decision rules, which might be a demanding task.

The second solution is to extend the qualitative evaluation model with some kind of quantitative evaluation. The idea is that, in addition to a qualitative class, an additional numeric utility is provided for each option, which can be used to rank the options within classes. Such an approach is presented later in this paper.

Sensitivity

Qualitative MADM models are less sensitive to small variations of attribute values than the quantitative ones. When the difference between two options is small enough, both options are described with equal qualitative attribute values; the decision model can not discriminate between them whatsoever. To see this, take again the candidate A from fig. 3 and suppose another candidate, C. Let C be equal to A except that his communicability (COMM) is *slightly* worse. Suppose that this difference is insufficient to make the *qualitative* distinction between A and C in terms of COMM, i.e., to assign COMM = *good* for A and COMM = *average* for C. Therefore, COMM = *good* is set for both candidates. This makes them totally equivalent within the model, although it is apparent that A is slightly better than C. In order to discriminate between A and C, a more sensitive decision model is needed.

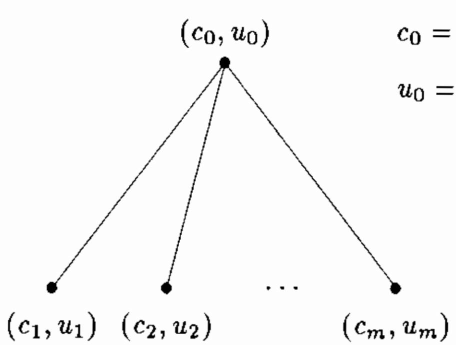
Similarly to option ranking, there are two solutions to increase sensitivity. First, the number of qualitative attribute values can be increased. For example, the four-valued scale of COMM (fig. 2) can be extended to six values, say, *poor, accept, average, good, very good* and *excel*. This possibly allows the discrimination between A and C in the above case. However, the disadvantage of this approach is that it may cause a combinatorial explosion, requiring extremely large utility functions in terms of decision rules. The second solution, which seems more feasible, is again a combination of qualitative and quantitative evaluation models. This will be discussed after the next section, in which we focus on combined methods for ranking options.

A combined evaluation method for option ranking

A combined qualitative and quantitative evaluation method is presented in this section that was designed according to the following goals:

(1) in addition to a qualitative *class*, the method should provide a numeric *utility* for each option, whose role is to rank options within the class;
(2) utilities should be *consistent* with classes: they should not violate the partial order of options established by the qualitative model;
(3) utilities should reflect the *domination* of options and *importance* of attributes;
(4) the quantitative evaluation model must be *constructed automatically* from the available qualitative model.

Therefore, a qualitative model is an input to an automatic procedure that constructs a consistent quantitative model. The required result is illustrated in fig. 4. An aggregate attribute is shown that depends on m lower-level (basic or aggregate) attributes. When evaluating an option, each lower-level attribute is assigned a *pair* of values (c_i, u_i), where c_i is a discrete class and u_i a numeric utility. The class c_0 of the aggregate attribute is determined by evaluating F, a utility function represented by decision rules. The evaluation is performed by a simple table search as described earlier. The utility u_0 is obtained by f, a function constructed automatically from F.



$$c_0 = F(c_1, c_2, \ldots, c_m)$$

$$u_0 = f(u_1, u_2, \ldots, u_m)$$

Fig. 4. A subtree of attributes with assigned discrete-numeric pairs of values.

The construction algorithm is performed separately for each aggregate attribute. It consists of three main steps. First, qualitative attribute values are transformed into numeric ones. In the second step, the importance of attributes is estimated from decision rules. Finally, function f is constructed taking into account the obtained importances and satisfying the consistency requirement. A detailed description of these steps is given below.

Step 1: Transformation of value domains

The role of this step is to bridge the gap between qualitative and quantitative attributes. Linguistic values, such as *good* and *active*, are transformed into numbers. In the qualitative model, only the order of linguistic values is known for each attribute. Therefore, it is a natural choice to replace the qualitative values by their ordinal numbers. For example, the values *poor, average, good* and *excel* of the COMM attribute (fig. 2) are represented by ordinals 1 to 4.

Reordering of values. The above transformation is meaningful only when linguistic values are ordered preferentially, i.e., when a higher ordinal number represents a better option value. In DECMAK, it is assumed that all aggregate attributes are preferentially ordered. This is not required for basic attributes. For example, attribute AGE in fig. 2 is preferentially ordered only when an old candidate is preferred for that job. Otherwise, the values should be reordered to reflect the preference, for example: *more* (worst), *18–20, 21–25, 41–55, 26–40* (best).

The preference order of basic attribute-values can be found by an algorithm. The algorithm assumes that aggregate attributes are preferentially ordered and takes into account the corresponding qualitative utility functions. Basically, the task is to find such an order of values for each basic attribute that makes F monotone with respect to that attribute. In other words, if c_a and c_b are two classes ordered such that $c_a < c_b$, then the value of $F(\ldots,c_a,\ldots)$ should increase or remain unchanged when c_a is replaced by c_b. Otherwise, the order of c_a and c_b is probably wrong and should be reversed.

Procedurally, the reordering algorithm is applied on each basic attribute and each pair c_a and c_b of its values ($c_a < c_b$). The algorithm inspects the corresponding pairs of decision rules of F, determining the relation between $F_a = F(\ldots,c_a,\ldots)$ and $F_b = F(\ldots,c_b,\ldots)$ for each pair. If $F_a > F_b$, i.e., monotonicity violation, occurs more frequently than $F_a > F_b$, the values c_a and c_b are interchanged.

This algorithm was tested on 534 utility functions that were developed so far in practical applications of DECMAK (Bohanec 1990). There were only 1.7% of functions where the monotonicity was not achieved, mainly due to inconsistent decision rules. Therefore, the reordering algorithm can be considered fairly reliable for real-life utility functions.

Hereafter we assume that *all* attribute value domains are preferentially ordered and represented with ordinal numbers. The letter c will be used to denote an ordinal number instead of a qualitative class.

Generalization to continuous values. The quantitative evaluation model extends ordinal numbers (denoted c in fig. 4) to continuous values (u). They should be consistent with each other; when options are ranked according to u, the partial order determined by c should not be violated. This is achieved by the following requirement: for each ordinal number c, the corresponding u should be in the interval $I_c = [c - 0.5, c + 0.5]$.

For example, if the ordinal value of the class *good* is 3, then the corresponding utility can take values in the range [2.5, 3.5]. When there is a better class, say *excel* (ordinal 4), the corresponding range is $I_4 = [3.5, 4.5]$. Therefore, all options evaluated as *excel* have a higher rank than the *good* ones, regardless on which value, c or u, is used; the only exception is at the margin $u = 3.5$ where it is assumed that c prevails. In other words, u is used to rank the options *within* the class c. When $c = 3$, $u = 2.5$ and $u = 3.5$ represent, respectively, the worst and best option within that class.

Step 2: Estimating weights

After the values of attributes have been reordered, replaced by ordinal numbers and generalized to continuous values, decision rules can be interpreted as points in a multidimensional space. For example, rules from table 1 are points in a three-dimensional space determined by attributes MANAG, COMM and WORK_APP. They are shown as dots (•) in fig. 5.

For the construction of a quantitative evaluation function f, the importances (weights) of attributes should be estimated. Linear regression is used for this purpose which constructs the function

$$g(u_1,\dots,u_m) = w_0 + \sum_{i=1}^{m} w_i u_i.$$

Here, w_i represent weights that are set so as to minimize

$$\sum_{c \in C} (F(c) - g(c))^2.$$

C denotes the set of all possible vectors of the form $[c_1, c_2,\dots,c_m]$.

Step 3: Construction of f

Functions F and f should be consistent with each other. When F evaluates an option with class c, then f should return a value within the I_c interval. This is not generally true for the above function g. Therefore, function f is constructed from g in this step so as to satisfy consistency. For each class c, g is linearly transformed into

$$f(u_1, u_2,\dots,u_m) = k_c g(u_1, u_2,\dots,u_m) + n_c,$$

where k_c and n_c are set so as to guarantee that $f(u_1,u_2,\dots,u_m) \in I_c$ and, moreover, the maximum and minimum of f for that class are $c - 0.5$ and $c + 0.5$, respectively.

The algorithm for determining k_c and n_c for one function F is thus the following:

for each class c of the aggregate attribute (fig. 4) **do**
  for all vectors $\boldsymbol{c} = [c_1, c_2,\dots,c_m]$ such that $F(\boldsymbol{c}) = c$ **do**
    begin
      $\mathrm{MIN}_c := \min_{\boldsymbol{c}} g(c_1 \pm 0.5,\dots,c_m \pm 0.5)$;
      $\mathrm{MAX}_c := \max_{\boldsymbol{c}} g(c_1 \pm 0.5,\dots,c_m \pm 0.5)$;
      $k_c := 1/(\mathrm{MAX}_c - \mathrm{MIN}_c)$;
      $n_c := c + 0.5 - k_c \mathrm{MAX}_c$;
    end.

Table 2
Calculations needed to construct f for WORK_APP.

Class c		MIN_c	MAX_c	k_c	n_c
less_acc	1	$g(1-0.5, 1-0.5) = -0.054$	$g(4+0.5, 1+0.5) = 2.304$	0.424	0.523
accept	2	$g(2-0.5, 2-0.5) = 1.004$	$g(3+0.5, 2+0.5) = 2.688$	0.594	0.904
good	3	$g(3-0.5, 3-0.5) = 1.871$	$g(4+0.5, 3+0.5) = 3.554$	0.594	1.389

Example

To illustrate the construction of f, take for F the WORK_APP utility function (table 1). This function is already monotone, so no reordering of values of COMM and MANAG takes place in Step 1. The resubstitution of linguistic variables with ordinal numbers yields a three-dimensional decision space as shown in fig. 5. Rules from table 1 are represented by dots ($\cdot$).

Step 2 a is linear regression which estimates weights, resulting in the following g function:

$$g = 0.433 \times \text{COMM} + 0.625 \times \text{MANAG} - 0.583.$$

The calculations of Step 3, in which f is finally constructed, are

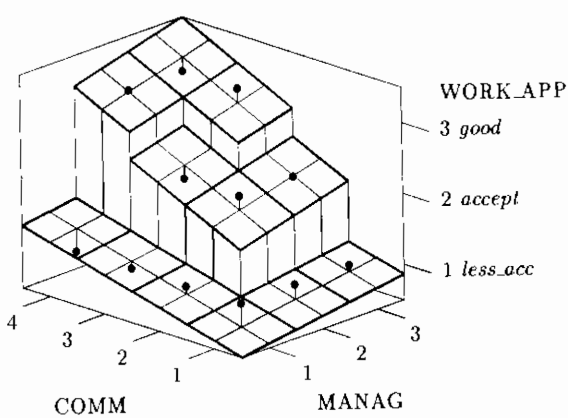


Fig. 5. Graphic representation of f for WORK_APP.

	Candidate A		Candidate B	
	class	utility	class	utility
CANDIDATE	*good*	2.795	*good*	2.819
EDUCATION	*approp*	3	*approp*	3
DEGREE	*M.Sc.*	3	*M.Sc.*	3
FOR_LANG	*passive*	2	*passive*	2
AGE_EXP	*accept*	1.741	*accept*	1.741
AGE	*21–25*	2	*21–25*	2
EXPER	*<1*	2	*<1*	2
PERS_CHAR	*approp*	2.884	*approp*	2.978
WORK_APP	*good*	2.928	*good*	3.186
COMM	*good*	3	*excel*	4
MANAG	*good*	3	*good*	3
INTEL	*b*	3	*b*	3

Fig. 6. Qualitative (F) and quantitative (f) evaluation of two candidates.

presented in table 2. For each class c of WORK_APP, k_c and n_c are determined, resulting in

$$f = \begin{cases} 0.424 \times g + 0.523 & \text{for class } \textit{less\_acc} \quad (c=1) \\ 0.594 \times g + 0.904 & \text{for class } \textit{accept} \quad (c=2) \\ 0.594 \times g + 1.389 & \text{for class } \textit{good} \quad (c=3). \end{cases}$$

This function is presented graphically in fig. 5. Note a separate plane for each of the three classes of WORK_APP. Each plane effectively discriminates among the points within that class taking into account the ordering of values of COMM and MANAG: the larger the value, the larger the difference between f (i.e., the plane) and F (the corresponding dot •). However, due to the required consistency of F and f, the absolute difference never exceeds 0.5.

Functions f that correspond to the remaining aggregate attributes in fig. 1, i.e., EDUCATION, AGE_EXP and PERS_CHAR, are constructed similarly. By this, the quantitative model is established and options can be evaluated both qualitatively and quantitatively. An example is shown in fig. 6. Two candidates, A and B, are evaluated. Regarding basic attributes, they differ only in the qualitative value of COMM. The qualitative evaluation assigns the same classes for both candidates. However, the quantitative model identifies the difference, which occurs in the path from the root of the tree (CANDIDATE) to the WORK_APP attribute.

A combined evaluation method for improving sensitivity

The combined method presented in the previous section is already quite sensitive to small variations of basic attribute-values. In order to discriminate between the candidates A and C, one can set the qualitative value of COMM to *good* for both candidates, but use different utilities, say 3 and 2.7. When f is evaluated, this difference is propagated through the tree, thus discriminating between the candidates. Therefore, the model is generally sensitive to such small variations.

However, the above approach was designed to discriminate between options that qualitatively differ in basic attributes, but obtain the same overall class. In other words, the method discriminates between the rules of F that belong to the same class (such as rules 10, 11 and 12 in table 1). This is reflected in the fact that in general $F(c_1, \ldots, c_m)$ is different to $f(c_1, \ldots, c_m)$ for any set of ordinal arguments c_i. In such a model, the effects of small variations and ordering of rules are intermixed; it is difficult to analyze only the effects of the former. Also, the interpretation of such utilities might be rather difficult for the user.

For this reason, an alternative approach was designed in which only small variations of values are taken into account; no ordering of rules takes place. A different function, h, is constructed which, in contrast to the previous approach, preserves the relation $F(c_1, \ldots, c_m) = h(c_1, \ldots, c_m)$ for all qualitative c_i.

Construction of h

The construction of h is performed in two steps. The first one is exactly the same as in Step 1, discussed in the previous section. Linguistic values are transformed into ordinal numbers and the decision space is extended to continuous values. The consistency requirement holds as well, so for each class c, the corresponding utility is limited to the $I_c = [c - 0.5, c + 0.5]$ interval.

In the second step, h is constructed. The construction first satisfies the requirement that $F = h$ for all ordinal arguments. Then, h is generalized to continuous values of its arguments. It is assumed that linearity holds in a small neighborhood of each discrete point of F. Therefore, small hyperplanes are constructed around each discrete point so as to best fit F. The slopes of a hyperplane are determined only on the basis of the nearest neighbors of that point (fig. 7).

More formally, for each vector $\boldsymbol{c} = [c_1, \dots, c_m]$ and for each $i = 1, 2, \dots, m$, the trend of $F(c_1, \dots, c_m)$ in the positive and negative direction of the i-th attribute is identified:

$$d_i^+(c_1, \dots, c_m) = F(c_1, \dots, c_{i-1}, c_i + 1, c_{i+1}, \dots, c_m) - F(c_1, \dots, c_m),$$

$$d_i^-(c_1, \dots, c_m) = F(c_1, \dots, c_{i-1}, c_i - 1, c_{i+1}, \dots, c_m) - F(c_1, \dots, c_m).$$

Note that only the nearest neighbors of $\boldsymbol{c}$ are considered. When there is no such neighbor, i.e., when the argument $c_i \pm 1$ of F exceeds the valid range of ordinal values, the corresponding d_i is deliberately set to 0.

For each vector $\boldsymbol{c} = [c_1, \dots, c_m]$, h is then generalized to continuous arguments $\boldsymbol{u} = [u_1, \dots, u_m]$. Since $\boldsymbol{u}$ is consistent with $\boldsymbol{c}$, each $u_i \in [c_i - 0.5,\ c_i + 0.5]$. In other words, $\boldsymbol{u}$ is near to $\boldsymbol{c}$. For such a subspace, hyperplanes are constructed that are all based in $F(\boldsymbol{c})$ and whose slopes are determined according to d_i:

$$h(u_1, \dots, u_m) = F(c_1, \dots, c_m) + 1/m \sum_{i=1}^{m} u_i d_i^*(c_1, \dots, c_m).$$

Here, d_i^* stands for d_i^+ when $u_i \geqslant 0$, and for d_i^- otherwise.

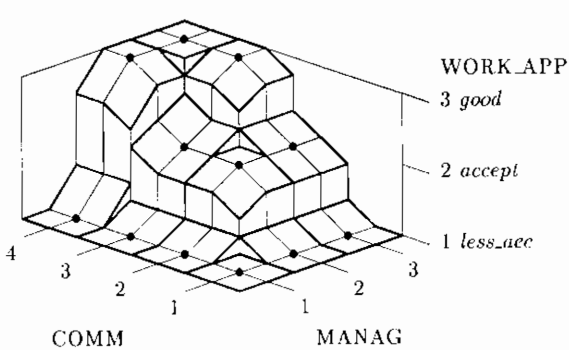


Fig. 7. Graphic representation of h for WORK_APP.

	Candidate A		Candidate C	
	class	utility	class	utility
CANDIDATE	*good*	3	*good*	2.975
EDUCATION	*approp*	3	*approp*	3
DEGREE	*M.Sc.*	3	*M.Sc.*	3
FOR_LANG	*passive*	2	*passive*	2
AGE_EXP	*accept*	2	*accept*	2
AGE	*21–25*	2	*21–25*	2
EXPER	*<1*	2	*<1*	2
PERS_CHAR	*approp*	3	*approp*	2.925
WORK_APP	*good*	3	*good*	2.850
COMM	*good*	3	*good*	2.7
MANAG	*good*	3	*good*	3
INTEL	*b*	3	*b*	3

Fig. 8. Qualitative (F) and quantitative (h) evaluation of two candidates.

Example

When applied on the WORK_APP function (table 1), the above construction yields h as shown in fig. 7. It is composed of small planes that fit the discrete values of F. Each plane reflects the trend of F from the basic point in the direction of its two nearest neighbors.

Take again the candidates A and C from section 3.2 and evaluate them by h. Since C is slightly worse than A in terms of COMM, assign 2.7 to COMM for C and leave 3 for A. The candidates are then evaluated as shown in fig. 8. Only ordinal values are assigned to A's basic attributes. In this case, the requirement for $F = h$ applies, so utilities exactly match the corresponding classes. The overall utility of A is 3. On the other hand, a small decrease of COMM in C's case is propagated towards the root of the tree, resulting in a slightly lower utility, 2.975.

Discussion

The motivation for the approach presented in this paper came from practice: to solve the problems of qualitative decision support related to option ranking and sensitivity. The proposed methods provide a solution that is based on a supplementary utilization of qualitative and quantitative modelling. Their strong points, in our opinion, are the following:

- structured approach to the classification of options,
- automatic construction of the quantitative model,
- consistency of the two models.

Structuring is one of the key elements of effective human information processing and problem solving. In DECMAK, it occurs at several levels. First, the whole decision problem is decomposed into smaller problems by a tree of attributes. Each attribute is further decomposed into symbolic ordinal values. Another structuring dimension occurs at the stepwise evaluation of options. Options are first classified into distinct ordered classes; the classification is determined by decision rules. After this high-level classification is available, the more detailed quantitative evaluation takes place. These structuring mechanisms are based on concepts which are commonly used by people: tree-structured hierarchies, symbols, rules and tables.

Automatic construction. The proposed methods automatically derive quantitative models from qualitative ones, which is of particular practical significance. No additional effort is required in the initial development of qualitative models, while the overall functionality is substantially extended. The initial development may be even simplified for those subproblems which exhibit high regularity (e.g., linearity of utility functions). In such cases, a detailed evaluation model can be automatically developed from only a few qualitative components, i.e., attribute values and decision rules.

Consistency. The construction algorithms take special care to maintain consistency between the models. The need for consistency was clearly demonstrated in an early attempt to option ranking in DECMAK (Olave et al. 1989). There, two models, a quantitative and qualitative one, were developed independently. Although both performed quite well, inconsistency occurred in some cases. The users accepted this rather dramatically, refusing both methods and asking for interpretation. The conflict was resolved by a thorough investigation of both models, after which a decision was made to use the qualitative model for the primary classification and the quantitative one only for option ranking within the classes. This was, actually, the starting point of the research reported here. This problem is related to the integration of various methods, which is further discussed at the end of this section.

Apart from the advantages, there still exist some weak points of the proposed approach and open questions, which should be further justified from the theoretical and practical viewpoint. First, the methods are heuristic and, as such, they should be used with care. They are based on several assumptions about the problem domain. When these do not hold, the methods still work, possibly constructing inappropriate models. In particular, the following elements of the construction procedures should be verified and, if necessary, adapted to the specific problem:

(1) Representing qualitative values by ordinal numbers: Equidistant ordinal numbers are just a rough approximation of linguistic values. A different representation, such as real numbers, intervals or even distributions of numeric values, might be more appropriate in some cases.
(2) Estimating attribute weights by linear regression: This step assumes that utility functions are linear at least near the discrete points (qualitative rules). Otherwise, the estimation of weights from rules becomes unreliable, so some other estimation method should be used.
(3) Constructing function h: The solution presented here is only one among the possible ones; there are many other ways to fit a qualitative function with small hyperplanes. Some of them may be more appropriate for some, more or less general, classes of utility functions.

Therefore, the automatic construction might be dangerous if it is applied strictly as presented here: a 'black box' procedure, which is not supervised by the decision maker. Apparently, such a control is highly desirable and should be seriously considered in further development and implementation.

From the viewpoint of psychological validity, the use of ± 0.5 utility intervals seems somewhat arbitrary. They are used to maintain consistency in the mathematical sense. However, is a utility difference of 0.3 on a basic attribute psychologically relevant? Are people sufficiently precise and reliable in determining these values? The distinction between utilities is needed basically for two purposes: (1) to discriminate between similar options, and (2) to analyze the possible effects of changes to the evaluation of a single option. In other words, the problem is to determine whether small differences add up and make a difference in the overall judgement or not. For these purposes, the magnitude of utilities is not as important as their relative proportion.

This means that the decision maker, when determinig utilities for a particular basic attribute, is actually required to make a comparison of options with respect to that attribute. In our experiments, people did not express any particular difficulty in performing this task. This is in accordance with research findings related to the comparison of options (Moshkovich 1991). However, the achieved levels of precision and reliability, which are probably not very high, have not been measured in the experiments yet.

Finally, let us mention that there are other feasible ways of combining qualitative and quantitative decision models. Probably the most straightforward one, at least conceptually, is to intermix quantitative and qualitative attributes within the same model; each property is measured by an attribute of the most appropriate type (i.e., quantitative or qualitative). This approach is quite common in expert systems (Klein and Methlie 1990). It seems entirely feasible also in the context of MADM; it should deserve more attention in the future.

Another interesting combination is to use methods separately, instead of deriving one from another. This solution would require additional effort to develop two (or more) models, but would, on the other hand, provide a variety of answers (the so-called 'second opinion'), which might be beneficial for the analysis and justification of the decision. However, the answers that differ too much might confuse unexperienced decision makers, as indicated in the case presented above. The essential problem here is therefore the integration of various methods at the level of final results: a mechanism, implicit or explicit, for the final aggregation of partial results is required. Such mechanisms are one of the central topics of a new dynamic field of artificial intelligence, called multi-strategy learning (Michalski and Tecuci 1991).

Conclusion

The proposed approach demonstrates that the functionality of qualitative multi-attribute decision-making methods can be substantially extended when they are combined with quantitative ones. Tasks that are difficult when approached qualitatively, such as option ranking and discrimination between similar options, become straightforward when the corresponding quantitative model is available. Additionally, it is shown that such a model can be constructed directly from the qualitative one.

Two construction methods were proposed in the paper. The first one is more general; although it was designed primarily for option ranking, it also exhibits sensitivity to small differences between options. The second method is better in terms of sensitivity, but is inappropriate for option ranking. Both methods were practically tested in decisions related to personnel management. Three particularly advantageous features were observed: structured decision modelling, automatic construction of the quantitative model from the qualitative one, and consistency of the two models.

The presented methods, however, are by no means complete or ideal. Some elements and assumptions of the model should be subjected to additional practical and theoretical justification, particularly with respect to psychological validity, and descriptive and behavioral decision research. Nevertheless, we believe that they provide a firm and useful framework for further research in these directions.

References

Anderberg, M.R., 1973. Cluster analysis for applications. New York: Academic Press.

Barrera, A. and M. Bohanec, 1987. An experiment with DECMAK in the domain of performanee evaluation of publie enterprises. Report DP-4723, Institut Jozef Stefan, Ljubljana.

Berkeley, D., P. Humphreys, O. Larichev and M. Moshkovich, 1990. 'Aiding strategic decision making: Derivation and development of ASTRIDA'. In: J. Vecsenyi and H.G. Sol (eds.), Proceedings of the IFIP TC-8.3 Conference, Budapest.

Bohanec, M., 1990. Metode za vrednotenje variant in razlago znanja v večparametrskem odločanju [Methods for knowledge explanation and evaluation of options in multi-attribute decision making]. Ph.D. thesis, University of Ljubljana, Faculty of Electrical and Computer Engineering.

Bohanec, M., I. Bratko and V. Rajkovič, 1983. 'An expert system for decision making'. In: H.G. Sol (ed.), Processes and tools for decision support. Amsterdam: North-Holland.

Bohanec, M. and V. Rajkovič, 1987. 'An expert system approach to multi-attribnte decision making'. In: M.H. Hamza (ed.), Proceedings of IASTED conference on expert systems. Anaheim, CA: Acta Press.

Bohanec, M. and V. Rajkovič, 1988. 'Knowledge acquisition and explanation for multi-attribute decision making'. Proceedings of the eighth international workshop on expert systems and their applications, Vol. 1, Avignon.

Bohanec, M. and V. Rajkovič, 1989. Knowledge-based explanation in multi-attribute decision making. Produktivnost 30, 182–189.

Bohanec, M. and V. Rajkovič, 1990. DEX: An expert system shell for decision support. Sistemica 1, 145–157.

Bohanec, M., V. Rajkovič and N. Lavrač, 1988. 'Knowledge explanation in expert systems: A decision support system and machine learning view'. In: M.H. Hamza (ed.), Proceedings of IASTED conference on expert systems. Anaheim, CA: Acta Press.

Chankong, V. and Y.Y. Haimes, 1983. Multiobjective decision making: Theory and methodology. Amsterdam: North-Holland.

Črnivec, Ž., V. Rajkovič, M. Bohanec and B. Skuber, 1988. Računarski kadrovsko-obrazovni sistem i odlučivanje o razvojnom putu kadrova [Computerized personnel and educational expert system and decisions on personnel development]. Andragogija 34, 331–348.

Efstathiou, J. and E.H. Mamdani, 1986. 'Expert systems and how they are applied to industrial decision making'. In: G. Mitra (ed.), Computer assisted decision making. Amsterdam: North-Holland.

Forhus, K.D., 1981. 'Qualitative reasoning about physical processes'. In: Proceedings of IJCAI-81, Vaneouver.

Fox, M.S., 1985. 'Knowledge representation for decision support'. In: L.B. Methlie and R.H. Sprague (eds.), Knowledge representation for decision support systems. Amsterdam: North-Holland.

French, S., 1986. Decision theory: An introduction to the mathematics of rationality. New York: Wiley.

Keeney, L.R. and H. Raiffa, 1976. Decisions with multiple ohjectives: preferences and value tradeoffs. New York: Wilcy.

Klein, M. and L.B. Methlie, 1990. Expert systems: A decision support approach. Reading, MA: Addison-Wesley.

Kuipers, B., 1986. Qualitative simulation. Artificial Intelligence 29, 289–338.

Larichev, O.I, H.M. Moshkovich, E.M. Furems, A.I. Mechitov and V.K. Morgoev, 1991. Knowledge acquisition for the construction of full and contradiction free knowledge bases. Groningen: iec ProGAMMA.

Lévine, P., M.-J. Pomerol and R. Saneh, 1990. Rules integrate data in a multicriteria decision support system. IEEE Trans. on Systems, Man, and Cybernetics 20, 678–686.

Michalski, R.S. and G. Tecuci (eds.), 1991. Proceedings of the first international workshop on multi-strategy learning. Fairfax, VA: George Mason University.

Mitra, G., 1986. Computer assisted decision making. Amsterdam: North-Holland.

Moshkovich, H.M., 1991. Qualitative methods in multiattribute decision making. Paper presented at the 13th SPUDM conference, Fribourg, Switzerland.

Olave, M., V. Rajkovič and M. Bohanec, 1989. 'An application for admission in public school systems'. In: I.Th.M. Snellen, W.B.H.J. van de Donk and J.-P. Baquiast (eds.), Expert systems in public administration. Amsterdam: Elsevier.

Pomerol, J.-Ch., 1988. MCDM and AI: Heuristics, expert and multi-expert systems. EURO VIII, Paris.

Rajkovič, V. and M. Bohanec, 1991 'Decision support by knowledge explanation'. In: H.G. Sol and J. Vecsenyi (eds.), Environments for supporting decision processes. Amsterdam: North-Holland.

Rajkovič, V., M. Bohanec and V. Batagelj, 1988. Knowledge engineering techniques for utility identification. Acta Psychologica 68, 271–286.

Turban, E., 1988. Decision support and expert systems. New York: Macmillian.

Vari, A. and J. Vecsenyi, 1988. Concepts and tools of artificial intelligence for human decision making. Acta Psychologica 68, 217–236.

Zadeh, L.A., 1975. The concept of a linguistic variable and its application to approximate reasoning. Information Science 8, 301–351.

Zimmermann, H.J., 1987. Fuzzy sets, decision making and expert systems. Boston, MA: Kluwer.