Learning Lexicographic Orders

József Dombi[*], Csanád Imreh, Nándor Vincze

Department of Informatics, University of Szeged, Árpád tér 2, H-6720 Szeged, Hungary,

email addresses: dombi@inf.u-szeged.hu, cimreh@inf.u-szeged.hu, vincze@szef.u-szeged.hu

Abstract

The purpose of this paper is to learn the order of criteria of lexicographic decision under various reasonable assumptions. We give a sample evaluation and an oracle based algorithm. In the worst case analysis we are dealing with the adversarial models. We show that if the distances of the samples are less than 4, then it is not learnable, but 4-distance samples are polynomial learnable.

Keywords: Multiple criteria analysis, Decision support systems, Learning, Lexicographical decision

1 Introduction

In the area of multicriteria decision making a lot of mathematical models exist. It is a paradox now to choose the right model. It could also be a multicriteria problem, which can be solved by a multicriteria method. Nearly all of the methods use parameters given by the users. It is important to find the right values. The parameters should be user friendly, i.e. it is important to give its semantical meaning. The most user friendly way to find the right parameters is the order of some alternatives. From a large amount of alternatives for the user it is very easy to select some alternatives which order is evident. This fact is a good tool to check whether the model - equipped by these parameters - is well based. If not, the user begins to play with the parameters, and during this procedure the user implicitly learns to manipulate the decision model too. The question naturally arises:

[*]corresponding author. Tel:+36 62 546194; Fax:+36 62 546397

are there algorithms to find the right parameters under some order information? For the utility approach the solution is the UTA method developed by Jacquet-Lagrèze (Jacquet-Lagrèze 1982; Jacquet-Lagrèze and Siskos 1982; Jacquet-Lagrèze and Shakun 1984; Jacquet-Lagrèze *et al.*, 1987; Jacquet-Lagrèze 1990; Jacquet-Lagrèze 1995; Jacquet-Lagrèze and Siskos 2001). It is an excellent solution to approximate the additive model. Nowadays we can formulate the above mentioned problem statement as learning the parameter using order information. In this paper we will examine the lexicographic model from the point of view of learning. Here the model parameters are the order of criteria.

The lexicographic decision model is one of the simplest. In the mid-70's Fishburn (Fishburn 1974) wrote a state of the art survey on the method. Although it is very simple, it is the most common and used decision model in everyday life. Even if the decision-makers use another model, they translate it (if it is possible) to lexicographic, because for the verbal communication only this approach is good (see Moshkovich *et al.*, 2005). Lexicographical decisions appear in different research areas. Usually in multicriteria optimization models the criteria are ordered by importance and the optimal solution is defined by the lexicographic order of the feasible solutions. The lexicographical ordering also appears in the area of linear programming, a version of the simplex algorithm where the pivoting element is selected by a lexicographic ordering has been developed for the solution of the problem.

In the lexicographical decision model we have an n element set of criteria and a linear importance order on this set. We say that an alternative precedes lexicographically an another one, if it has a more preferred value in the most important criteria where they are not the same.

From the learning aspects the examination of lexicographical decision making was not as an easy task as it could be supposed from this simple model. Several interesting results and algorithms were discovered.

We use the following mathematical model. There are n criteria, and they are linearly ordered by importance. The decision alternatives are ordered by the lexicographical ordering. Our goal is to learn the importance order of the criteria by decision samples. We suppose that the samples are given by exchange value (EV) vectors which are evaluated by the exchange value evaluation (EVE) function. An EV vector is an arbitrary n-dimensional vector which contains -1, 0 or $+1$ in the coordinates. The $+1$ in the i-th coordinate means that we improve the decision alternative concerning the i-th criterion, the -1 means that we deteriorate the decision alternative concerning the i-th criterion, the 0 means that we do not change it concerning the i-th criterion. The EVE function has value $+1$ on the EV vector if the resulted alternative precedes the original in the lexicographical order, in the opposite case it has value -1.

We present the above notation on the following example. Suppose there are three criteria: A is the most important, B is the second most important and C is the less important, each criterion can get positive integers as value, and in each case the larger values are preferred. Then an EV vector $+1, -1, 0$ belongs to the situation when we increase the value of criterion A and decrease the value of criterion B. Since A is the most important criterion thus the resulted alternative will be worst, therefore the EVE function has value -1 on this vector.

In Section 2, we present an algorithm called Sample Evaluation algorithm which determines the importance order of the criteria by the sample if it is possible. In the remaining part we investigate how long sequence of EV vectors can be necessary to determine the importance order. First in Section 3 we consider the best case, we suppose that we can generate the EV vectors and the EVE function evaluates them, we call this case Oracle model. Later in Section 4 we consider the worst case, we suppose that the evaluated EV vectors are generated by an adversary who has the goal to present as long sequence as possible. We call this model adversarial model.

2 Sample Evaluation algorithm

In this section we present an algorithm which considers a sample containing a sequence L of EV vectors evaluated by the EVE function. The algorithm evaluates the sequence and as a result it determines the importance order of the criteria if it is possible, or it decides whether the sample is insufficient (it could be generated by more importance orders) or inconsistent (it cannot be obtained by the lexicographical ordering).

The algorithm works in n phases. In the i-th phase it determines the i-th criterion in the importance order. If the algorithm is not able to determine the i-th criterion in the i-th phase then it concludes that the sample is insufficient or it concludes that the sample is inconsistent. In the i-th phase the algorithm examines the EV vectors which may contain useful information - the EV vectors which obtain the value by the $i-1$ most important criteria are already eliminated - and uses these vectors to exclude candidates for the position of the i-th criterion in the importance order. If the set of the candidates contains one element at the end of the phase then it is the i-th criterion in the importance order. If it is empty then the sample is inconsistent, if it contains more elements then the sample will be insufficient or inconsistent in some later phase. We give the algorithm in details below:

SamEv Algorithm

Initialization

$S_1 := \{1 \ldots, n\}$

Iteration part

for $i = 1$ to n do
  $S := S_i$,
  for every $p_k \in L$ do
    if $i > 1$ and $p_k(l_{i-1}) \neq 0$ **then**
      delete p_k from L,
    else if $EVE(p_k) = 1$ **then**
      delete each j with $p_k(j) = -1$ from S,
    else if $EVE(p_k) = -1$ **then**
      delete each j with $p_k(j) = 1$ from S,
  endfor
  if $|S| = 0$ **then**
    stop, the sample is inconsistent,
  if $|S| \geq 2$ then
    note that the sample is not sufficient,
    delete arbitrary $|S| - 1$ elements from S.
  Let $S_{i+1} := S_i \setminus S$,
  let l_i be the index which is contained in S,
endfor

Output:

One importance order is $l_1, \ldots, l_n$, and if we noted at some phase that the sample is insufficient then other importance orders are also consistent with the sample.

Example: Suppose that there are four criteria and the following sample is given: $EVE(1, 0, 1, -1) = 1$, $EVE(-1, 1, 0, 1) = 1$, $EVE(1, 0, -1, 0) = -1$, $EVE(-1, 0, 1, 0) = 1$, $EVE(1, 0, 0, -1) = 1$. Then in the first phase it is determined that the second criterion is the most important. In the next phase the second EV vector is deleted (its second component is not 0) and it is determined that the third criterion is the second most important. In the next phase only the last EV-vectors remains and it follows that the first criterion is the third most important, and the fourth criterion is the less important.

The algorithm solves the problem as the following statement shows.

Theorem 1 *If algorithm SamEv does not stop with inconsistent sample then it results in an importance order which is consistent with the sample. Furthermore it determines correctly if the sample is inconsistent, and also determines if the sample is insufficient.*

Proof: First observe that if a criterion is eliminated in the i-th phase from set S then this criterion cannot be the most important one among the criteria in set S_i. If a criterion is the most important in the set S_i then for each EV vector which contains 0 in the coordinates of the criteria not contained in the set S_i and contain $+1$ or -1 in the coordinate of the criterion considered the EVE function evaluates the vector with the same value as the criterion considered. We exclude a criterion if we find a vector which does not satisfies the above condition. Therefore if at the i-th phase the algorithm stopes with the result that the sample is inconsistent then we obtain that none of the criteria in set S_i can be the most important in this set, and this yields that the sample is indeed inconsistent.

Now suppose that the algorithm does not stop with inconsistent sample. We show that the lexicographical decision based on the importance order obtained by the algorithm gives the same evaluation as the EVE function. Consider an arbitrary EV vector from the sample. Let the i-th criterion be the most important one with nonzero coordinate in the vector. Suppose that this nonzero coordinate is $+1$, we can handle the opposite case in the same way. In this case the lexicographical decision will give the value $+1$ to the vector. On the other hand during algorithm $SamEv$ in the first i phases this vector was not eliminated from set L, therefore we examine it in the i-th phase. If the EVE function evaluates it with value -1 then the algorithm excludes the i-th criterion from set S, and this is a contradiction. Therefore it also evaluates the vector with value $+1$ and this proves the statement.

Now we prove that if the algorithm determines an insufficient sample, then there are more importance orders which are consistent with the EVE function. Consider the phase where we noted that the sample is insufficient. Denote by i the criterion which was chosen by the algorithm from the set S.

Suppose that a modified algorithm chooses an another criterion from this set S and it is continued in the same way as $SamEv$. We state that this modified algorithm does not determine inconsistent sample, and thus it produces an another importance order which is consistent with the EVE function. We prove this by contradiction. Suppose that at the j-th phase the modified algorithm eliminates every elements from set S. Let x be the criterion from the modified S_j set which is the most important in the importance order given by the original algorithm. This means that in the phase where x was chosen by the original algorithm all of the input vectors were investigated which are investigated in the j-th phase of the modified algorithm. On the other hand in the original algorithm we did not exclude x from the set S of candidates and this yields the contradiction.

3 Oracle model

In this section we examine how long sequence of EV vectors can be necessary to determine the importance order in the best case. We use the following model, which we call the Oracle model. We suppose that we can use the EVE function as an oracle, i.e. we can ask to evaluate a sequence of EV vectors generated by us. We want to find a short sequence of EV vectors which determines the importance order of the criteria. We use the following algorithm to generate the sequence. In the i-th phase we determine the i-th most important criterion by performing a binary search on the set S of the possible candidates. During a phase in each step we half the set C of candidates and we determine which half may contain the desired criterion. At the beginning of the algorithm S is the set of the criteria. We use the following notation. For any two sets C_1, C_2 of criteria the vector $EV(C_1, C_2)$ denotes the EV vector which is $+1$ in the coordinates contained in C_1, -1 in the coordinates contained in C_2, and 0 in the other coordinates.

***EV* sequence generating algorithm**

for $i := 1$ **to** n
  $C := S$
  while $|C| > 1$ do
    let C_1 be the set of first $\lfloor |C|/2 \rfloor$ elements of C
    let C_2 be the set of the other elements of C
    let $V = EV(C_1, C_2)$

if $EVE(V) = +1$ **then** $C := C_1$
else $C := C_2$
endwhile
C contains the i-th criterion in the importance order
delete the element of C from S
endfor

Example: Suppose that there are four criteria. Then we generate the first EV vector which is $(1, 1, -1, -1)$. Suppose that it is evaluated as 1. Then we generate $(1, -1, 0, 0)$. Suppose that it is evaluated as -1. Then we conclude that the second is the most important criteria, and we generate $(1, 0, -1, -1)$. Suppose that it is evaluated as -1. Then we generate $(0, 0, 1, -1)$. Suppose that it is evaluated as -1. Then we conclude that the forth is the second most important criteria, and we generate $(1, 0, -1, 0)$. Suppose that it is evaluated as 1, then we conclude that the first is the third most important criterion, and the third is the less important.

It can be seen easily that in the i-th phase we indeed obtain the i-th most important criterion, we exclude every other candidate with the generated EV vectors. Consider now the length of the sequence of the EV vectors. Each phase of the algorithm can be characterized with a binary tree. We can represent each vertex of the tree with a subset of S. The root of the tree is represented with the actual S and for every ν point we divide the C_ν subset belonging to the point into two subsets for which the number of elements are $\lceil C_\nu/2 \rceil$ and $\lfloor C_\nu/2 \rfloor$. These are the children of the C_ν point in the graph. When $|C_\nu| = 1$, then it contains the i-th most important criterion. Using this representation it is clear that the number of the used EV vectors is at most $\lceil \log_2 |S| \rceil$. Therefore the total number of the used EV vectors is at most

$$\lceil \log_2 n \rceil + \lceil \log_2 (n-1) \rceil + ... + \lceil \log_2 2 \rceil = n \lceil \log_2 n \rceil - 2^{\lceil \log_2 n \rceil} + 1.$$

Remark: We can reduce this problem to the classical problem of sorting a set. If we consider an EV vector which has -1 in the i-th component and $+1$ in the j-th component, then this vector gives the importance order of the i-th and j-th criteria. Therefore any of the well-known sorting algorithms (heapsort, quicksort) which are based on comparisons can be transformed to an EV vector generating algorithm, where the number of the used EV vectors is the number of the necessary comparisons of the algorithm. On the other hand these algorithms require at least $\Omega(n \log n)$ comparisons in the worst case (see Knuth 1973) so this approach cannot improve asymptotically our above bound.

4 Adversarial models

In contrast with the oracle model in this section we investigate the worst case situation. Here we suppose that the list of the EV vectors is generated by an adversary who has the goal to present as long list as possible. Such kind of worst-case analysis based on adversarial sequences are used in different areas of the theoretical computer science, one can find several applications in the area of the competitive analysis of on-line algorithms or in the area of the analysis of queuing strategies (see Borodin and El-Yaniv (1998) and Borodin *et al.* (2001) for details).

We can see easily that the adversary can use exponential length sequences, as the following sequence shows. The adversary can start the sequence with the the EV vectors started by 0, the number of such possible nonzero EV vectors is $3^{n-1} - 1$. Using the evaluation of these EV vectors we cannot conclude anything about the importance position of the criterion belonging to the first coordinate. In the problems where the adversary is too strong it is an often used idea to decrease the strength of the adversary by different restrictions, one can find several examples for such restricted adversaries in the area of on-line algorithms in the paper Fiat and Woeginger (1998). In this problem there are some natural ways to restrict the adversary, in the remaining part of the section we investigate these restrictions. We present two cases where even the restricted adversaries can generate exponential length EV vector sequences. We close the section with a restriction (strict distance restriction) where we can give $O(n^2)$ bound on the possible length of the EV vector sequences.

4.1 Exclusion of the redundant EV vectors

If we consider all of the possible EV vectors then many redundant information is given. The first restriction which is examined is that we forbid the adversary to generate some types of redundant EV vectors. We use the following rules to decrease redundancy

- If the EVE function takes the value $+1$ on an EV vector, then we know that it evaluates the reverse of the vector (we change each $+1$ to -1 and each -1 to $+1$) to -1. Therefore it is a redundant information to consider both a vector and its reverse. So we suppose that the adversary is allowed only to generate vectors which are evaluated by the EVE function to $+1$.

- Since each nonzero EV vector which does not contain -1 is evaluated to $+1$, we may suppose that the adversary generates only such EV vectors which contain -1.

- If we obtain a vector which is evaluated to $+1$, then any vector which is larger than it (at least as large in every component as the vector considered) is evaluated to $+1$, so these vectors contain only redundant information. Therefore we suppose that the adversary is allowed to generate only such EV vectors which are not larger than any of the vectors already generated.

Unfortunately these restrictions are not enough to force the adversary to use polynomial sequence. It can use the following sequence. Consider all of the EV vectors which are started by 0 in the first and -1 in the second component and contain $\lceil \frac{n-2}{2} \rceil$ times $+1$ and $\lfloor \frac{n-2}{2} \rfloor$ times 0 in the remaining components. Suppose they are all evaluated to $+1$. It follows immediately by the definition that this list satisfies the assumptions given above. Furthermore we cannot determine the place of the first criterion in the importance order, therefore we need a longer list to determine the importance order. On the other hand this sequence contains $\binom{n}{\lfloor \frac{n-2}{2} \rfloor}$ vectors which is an exponential length in n.

4.2 Weakly distance restricted adversary

In this part we examine the distance restricted adversaries. We use a metric on the set of the EV vectors. We consider the generalized version of the Hamming distance used in the area of error detecting codes (Hammer, 1950). The distance of two EV vectors is the number of the different components. In this part we suppose that in each step the adversary is only allowed to use such EV vector which has distance 1 from the previous vector. Such sequences are called weakly 1-distance restricted sequences.

We show that the adversary can present an exponential length 1-distance restricted EV sequence which is not enough to determine the importance order of the criteria. First observe that we can list all of the EV vectors in a weakly 1-distance restricted sequence. Let S_i be the weakly 1-distance restricted sequence of the i dimensional EV vectors. We can define these sequences recursively. $S_1 = (1, 0, -1)$. We can obtain S_{i+1} in the order $((1, S_i), (0, (S_i)^{-1}), (-1, S_i))$ where $(S_i)^{-1}$ is the opposite order of S_i, and (c, S_i) is the sequence which contains the sequence of EV vectors which are started by c and in the remaining components have the sequence S_i.

Now let us consider the sequence where the vectors are started by 0 and followed by S_{n-1}. We cannot determine the importance order, because of the first criterion. Furthermore, this is a weakly 1-distance restricted adversarial sequence and it has exponential length.

4.3 Strongly distance restricted adversary

In this part we consider strongly distance restricted sequences. A sequence is called strongly k-distance restricted if none of the vectors has larger distance from each other than k. It seems that this is a very strong restriction, but as the following statement show there exist strongly distance restricted sequences which are enough to determine the importance order.

Theorem 2 *If $n > 3$ then there is not such strongly 1-distance restricted sequence which contains enough information to determine the importance order.*

If $n \geq 6$ then there is not such strongly 2-distance restricted sequence which contains enough information to determine the importance order.

If $n \geq 8$ then there is not such strongly 3-distance restricted sequence which contains enough information to determine the importance order.

For arbitrary n there exist strongly 4-distance restricted sequences which contain enough information to determine the importance order.

Proof: If we consider a strongly 1-distance restricted sequence, then we have a vector and a second one which differs only in one component (denote this component by i) from it. If we choose a third vector which has different value in some other component than the first two vectors, then it also differs from one of them in the i-th component, which is a contradiction. Therefore the strongly 1-distance restricted sequences contain such vectors which are the same in $n-1$ component. If $n-1>2$ then it is impossible to determine the importance order of these criteria, thus the first statement of the theorem follows.

Now suppose that $n \geq 6$ and we have a strongly 2-distance restricted sequence which can be used to determine the importance order. Consider the 4 most important criteria. We can suppose that these criteria belong to the first 4 coordinates since changing the order of the coordinates keeps the distance restricted property. Since we are able to determine the importance order of the other criteria as well, the sequence must contain a vector which is started by $0,0,0,0$. On the other hand we can distinguish in the importance order the first and second criteria therefore the sequence most contain a vector which is started by $1,-1$ or by $-1,1$. We can suppose that it is started with $1,-1$ the other case is completely similar. This vector is continued with $0,0$ because it is at most 2 distance away from the vector started by $0,0,0,0$. Moreover we are able to determine the importance relation of the third and forth criteria, so the sequence contains a vector which is started by $0,0$ and continued with 1 and -1 in some order. But this sequence has distance 4 from the vector which is started by $1,-1,0,0$ and this leads to a contradiction which proves the second statement of the theorem.

Now suppose that $n \geq 8$ and we have a strongly 3-distance restricted sequence which can be used to determine the importance order. We can obtain a contradiction in a similar way as in the previous case. Consider the 6 most important criteria. We can suppose that these criteria belong to the first 6 coordinates since changing the order of the coordinates keeps the distance restricted property. Since we are able to determine the importance order of the other criteria as well, the sequence must contain a vector which is started by $0,0,0,0,0,0$. On the other hand we can distinguish in the importance order the first and second criteria therefore the sequence most contain a vector which is started by $1,-1$ or by $-1,1$. We can suppose that it is started with $1,-1$ the other case is completely similar. The other components of this vector contains at least three 0 because it is at most 3 distance away from the vector started by $0,0,0,0,0,0$. Among these three 0 there must be two which belong to neighboring coordinates. The only possibility to distinguish the importance of these coordinates is a vector which contain $0,0$ in the first two coordinates and $1,-1$ or $-1,1$ in the coordinates considered. On the other hand this vector is 4 distance away from the vector which is started by $1,-1$ and contains $0,0$ in the coordinates considered, and this leads to a contradiction which proves the third statement of the theorem.

To see that we can define strongly 4-distance restricted sequences which determine the importance order, recall that any EV vectors which contain 1 in the i-th coordinate and -1 in the j-th coordinate gives the importance relation of the i-th and j-th criteria. On the other hand any set of such EV vectors is strongly 4-distance restricted, so we can use any sorting algorithm which is based on comparisons.

In the rest of this section we investigate strongly 4-distance restricted sequences. Since the number of vectors which are at most 4-distance away from a fixed vector is $O(n^4)$, thus it is straightforward from the definition that the number of the vectors in a strongly 4-distance sequence is at most $O(n^4)$. On the other hand we can use better bound as the following statement shows.

Theorem 3 *The number of the EV vectors in a strongly 4-distance restricted sequence is at most $O(n^2)$.*

Proof: Consider a strongly 4-distance sequence. First suppose that there are two vectors a and b which have distance 4 from each other. We can suppose that they differ in the first 4 coordinates. Denote by A the set of vectors which differ in at most 2 coordinates from the last $n-4$ coordinates of a. Denote by B the set of vectors which differ in at most 2 coordinates from the last $n-4$ coordinates of b. Observe that every vector from the sequence is in A or B otherwise it would have larger distance than 4 from a or b. But the number of the possible elements in A and also in B is $O(n^2)$, and thus the theorem in this case follows.

Suppose that there are not two vectors which have distance 4 from each other. Then the sequence is strongly 3-distance restricted. Then suppose that there are two vectors a and b which have distance 3 from each other. We can suppose that they differ in the first 3 coordinates. Denote by A the set of vectors which differ in at most 2 coordinates from the last $n-3$ coordinates of a. Denote by B the set of vectors which differ in at most 2 coordinates from the last $n-3$ coordinates of b. Observe that every vector from the sequence is in A or B otherwise it would have larger distance than 3 from a or b. But the number of the possible elements in A and also in B is $O(n^2)$, and thus the theorem in this case follows.

Suppose that there are not two vectors which have distance 4 or 3 from each other. Then the sequence is strongly 2-distance restricted. On the other hand the number of EV vectors which are at most distance 2 from a fixed vectors is $O(n^2)$ and this yields the statement of the theorem in this case.

Using the above theorem we receive the following result about the strongly distance restricted adversary.

Corollary 1 *The size of the longest strongly 4-distance restricted sequence of EV vectors which is necessary to determine the importance order of the criteria is at most* $O(n^2)$.

5 Summary and open questions

In this paper we consider the problem of learning the importance order of criteria by decision samples. We present an algorithm which can determine the importance order. Furthermore we study the possible necessary length of the sample. We use two models, in the best case where we are allowed to generate the EV vectors we show a method which uses at most $n\lceil\log_2 n\rceil - 2^{\lceil\log_2 n\rceil}+1$. EV vectors to determine the importance order. In the adversarial model where the list of EV vectors is generated by an adversary we consider different types of adversary. We show a restriction the strongly 4-distance restriction which forces the adversary to use $O(n^2)$ length sequences.

Concerning this problem some further question arise. In the adversarial model it could be interesting to investigate further restrictions. From the theoretical point of view it would be interesting to characterize the maximal insufficient EV sequences which are not enough to determine the importance order. We can call an EV sequence maximal insufficient if any further EV vector is enough to determine the importance order or to conclude that the sample cannot be generated by the lexicographical decision.

References

[1] Borodin, A., Kleinberg, J, Raghavan, P., and Sudan, M. Adversarial Queueing Theory. Journal of the ACM 2001; 48; 13–38.

[2] Borodin, A El-Yaniv, R. Online Computation and Competitive Analysis. Cambridge University Press. 1998.

[3] Fiat, A, Woeginger G. J. 1998. Competitive Odds and Ends. In: Fiat A, Woeginger G. J. (eds) Online algorithms: The State of the Art, Vol. 1442 of Lecture Notes in Computer Science, Springer-Verlag Berlin; 1990 p. 385–394.

[4] Hamming, R. W. Error detecting and error correcting codes. Bell System Technical Journal 1950; 29; 147–160.

[5] Knuth, D. E. (1973) Sorting and searching, The Art of Computer Programming III. Addison Wesley. 1973

[6] Fishburn P. C. Lexicographic Orders, Utilities and Decision Rules: A Survey. Managenment Science 1974; 20; 1442–1471.

[7] Jacquet-Lagrèze E, and Siskos J. Assessing a Set of Additive Utility Functions for Multicriteria Decision-Making, the UTA Method. European Journal of Operational Research 1982; 10; 151–164.

[8] Jacquet-Lagrèze E, Meziani R., and Slowinski R. MOLP with an Interactive Assessment of a Piecewise Linear Utility Function. European Journal of Operational Research 1987; 31; 350–357.

[9] Jacquet-Lagrèze E. Binary Preference Indices: A New Look on Multicriteria Aggregation Procedures. European Journal of Operational Research 1982; 10; 26–32.

[10] Jacquet-Lagrèze E. 1990. Interactive Assessment of Preferences Using Holistic Judgments: The PREFCALC System. In: Bana e Costa CA. (Eds), Readings in Multiple Criteria Decision Aid. Springer-Verlag: Berlin; 1990. p. 225–250.

[11] Jacquet-Lagrèze E. 1995. An Application of the UTA Discriminant Model for the Evaluation of R&D Projects. In: ardalos PM, Siskos Y, Zopounidis C. (Eds), Advances in Multicriteria Analysis, Kluwer Academic Publishers: Dordrecht; 1995. p. 203–211.

[12] Jacquet-Lagrèze E, and Shakun M. F. Decision support systems for semistructured buying decisions. European Journal of Operational Research 1984; 16; 48–56.

[13] Jacquet-Lagrèze E., and Siskos Y. Preference disaggregation: 20 years of MCDA expreience. European Journal of Operational Research 2001; 130; 233–245.

[14] Moshkovich H, Mechitov A, and Olson D. 2005. Verbal Decision Analysis. In: Figueira J, Greco S, Ehrgott M. (Eds), Multiple Criteria Decision Analysis Springer-Verlag: Berlin; 2005. p. 609–634.