METU Studies in Development, 46 (June), 2019, 17-46

An interactive ranking-based multi-criteria choice algorithm with filtering: Applications to university selection

Gülşah Karakaya
Department of Business Administration, Middle East Technical University, Ankara, Turkey
e-mail: kgulsah@metu.edu.tr

Ceren Tuncer Şakar
Department of Industrial Engineering, Hacettepe University, Ankara, Turkey
e-mail: cerents@hacettepe.edu.tr

Abstract

In this study, we develop an interactive algorithm to converge to the most preferred alternative of a decision maker (DM) among a set of discrete alternatives. The algorithm presents a limited number of alternatives to the DM and collects preference ranking of them iteratively. The preferences are modeled by a flexible and realistic preference function. To improve the performance, the alternatives presented are determined by a filtering method. We compare our algorithm with benchmark algorithms on numerous data sets from Quacquarelli Symonds, a higher education marketing company that reports annual rankings of universities under different categories. The results show that our algorithm outperforms the benchmark algorithms.

Key words: Multiple criteria decision making, ranking, interactive method, filtering

1. Introduction

In many real-life decision making problems, the decision maker (DM) has multiple concerns. There are numerous criteria that have to be addressed simultaneously, and these criteria generally conflict with each other. As a result, there is no single optimal solution. In Multiple Criteria Decision Making (MCDM) problems, the focus is on efficient solutions instead. For efficient solutions, the value of a criterion cannot be improved unless at least one other criterion is worsened. Different DMs may have different preferences for the criteria and as a result, each can choose a different efficient solution as her/his best solution. In fact, theoretically, for each efficient solution, there can be a DM who will prefer it above all others. In such problem settings, MCDM methods come forward as appropriate and useful solution procedures.

Among areas that MCDM methods have been applied to, some examples are country rankings, economics, energy, environmental issues, healthcare, sustainability in social issues and education. Munda and Nardo (2009), for example, ranked 146 countries with respect to numerous criteria related to environmental sustainability like pollution, natural resource depletion, and ability to respond to environmental challenges. They employed a nonlinear aggregation technique to obtain scores for the countries. Nuuter et al. (2015) evaluated the sustainability of housing markets with respect to six criteria that reflected economic, social and environmental issues. In their applications, they used a popular MCDM method, COPRAS, to rank European countries. Kadzinski and Tervonen (2013) applied their additive ranking approach to evaluate 20 European countries with respect to the quality of their universities. Criteria were reflecting the number of universities in the top ranks of world rankings, the number of leading universities, and economical position of the country. As an example of studies in the energy sector, Karagiannidis and Perkoulidis (2009) used Electre III, which is one of the most popular MCDM techniques, to assess and rank anaerobic digestion technologies. The criteria used are greenhouse gas emission, energy recovered, material recovered, and operating cost. Thokala and Duenas (2012) explored the use of a variety of multiple criteria techniques in health technology assessment. They reviewed certain MCDM methods and applied some of them to a case study of assessing the performance of different drugs as treatment options. MCDM methods have also been applied for performance evaluation. As an example, Yalcin et al. (2012) used different MCDM methods to assess the financial performance of Turkish manufacturing firms in different industries. They employed fuzzy AHP to determine the weights of criteria used and then applied TOPSIS and VIKOR to obtain a ranking of the firms. They considered traditional accounting-based financial performance measures as well as new value-based financial performance measures. One can see Mardani et al. (2015) for a review of several MCDM techniques and their applications for the years 2000-2014. They reported the most prevalent methods, and indicated that energy, environment and sustainability are the most widely studied areas.

When the MCDM problem involves a large number of efficient solutions that the DM should choose from, it is difficult for the DM to make a decision. In this paper, we propose an interactive algorithm designed to converge to the most preferred solution of the DM among a set of discrete solutions. Our algorithm contributes to the literature by guiding the DM in reaching a final solution with considerably low number of iterations. In addition, the final solution is either the true best solution of the DM or a very close solution to it. We model the preferences of the DM with weighted L_α functions. These functions minimize the weighted L_α distance from the ideal vector of the best values in all criteria. L_α functions are capable of representing different preference structures of DMs with different criteria weight vectors and distance metrics (Karakaya et al., 2018). Through successive iterations, we present the DM small subsets of solutions and ask her/him to provide a ranking of them. Using the preference information gathered, we perform preference parameter estimations and eliminate solutions that cannot be the best for the DM. We apply a filtering approach in the selection of the solutions that will be presented to the DM in order to converge to the most preferred solution faster. We apply our algorithm to several problem sets that consist of different numbers of universities and criteria. We test the performance of our algorithm extensively using numerous preference parameters to simulate DM preferences. We also compare our algorithm with benchmark algorithms.

The paper is organized as follows: Section 2 contains the literature review on preference representation and elicitation in MCDM problems. In Section 3, we present our approach. Section 4 introduces the test problems and reports the results of experiments. In Section 5, we conclude our work with discussions of results and future studies.

2. Preference modeling and elicitation in the literature

Many studies in MCDM literature assume that preferences of the DM can be represented through certain functions. These functions are referred to as preference/value/utility functions. Using quantitative evaluations of solutions with respect to each criterion, these functions serve to obtain an overall preference measure for a given solution. It is assumed that the DM can compare and choose between available solutions using this overall measure. As stated in Roy (2016), decision making problems in multiple criteria analysis can be categorized into three groups: (i) *the choice problem*, where the best solution or a small set of best solutions are sought, (ii) *the sorting problem*, where the solutions are grouped into classes and these classes are preference ordered and (iii) *the ranking problem*, where solutions are listed in the order of preference. In this paper, we work on the choice problem where we aim to find the most preferred solution of the DM.

Linear, additive, quasiconcave, and general monotone functions are among the commonly used preference functions. The additive form is a classical and widely-used preference function; its foundations can be seen in Keeney and Raiffa (1976) and Wakker (1989). The Interactive Weighted-Sums/Filtering Method of Steuer (1986) assumed a weighted additive preference function and reduced the feasible weight space using DM preferences gathered through iterations. Jacquet-Lagrèze and Siskos (1982) proposed the popular UTA method that aims to find additive utility functions compatible with DM preferences. After the UTA method, which has been the start of the ordinal regression paradigm in MCDM, preference functions of different forms have also been studied (a review of these studies is available in Siskos et al., 2016). In traditional ordinal regression based approaches, after utility functions in line with DM preferences are derived, one of these functions is selected according to some decision rules. In robust ordinal regression, however, all functions that obey DM preferences are considered and accounted for (Greco et al., 2008). Angilella et al. (2004) claimed that the assumptions of the UTA method about the utility function can make the problem infeasible and proposed a fuzzy integral framework with nonadditive functions that can take into account any interactions between criteria. Marichal and Roubens (2000) also worked with interacting criteria and fuzzy integrals using partial rankings of criteria and alternatives. Benabbou et al. (2015) argued that nonlinear aggregation functions bring flexibility to the preference elicitation process by allowing for interactions between criteria. They estimated the parameters of these functions with a mini-max regret approach.

Several researchers developed methods to handle uncertainties and incomplete information in the preference elicitation process. Salo and Hämäläinen (2001) accepted partial preference statements of the DM and derived information from these. Sarabando and Dias (2010) considered a setting where only rankings of criteria weights and rankings of alternatives in individual criteria are present. They estimated the parameters of the aggregation model using Monte Carlo simulation and then obtained a ranking of the alternatives. A comparison of studies where only ranking information of the weights is present (as opposed to cardinal values) can be found in Ahn and Park (2008). In some cases, the DM can be indifferent between some alternatives and may not be able to choose among them. Considering this situation, Branke et al. (2015) proposed the use of indifference thresholds in robust ordinal regression. Again using the concept of thresholds, Branke et al. (2017) proposed several heuristics to decrease the level of interaction with the DM to find the preferred solution. One can see Pirlot and Vincke (2013) for a general discussion on the use of indifference thresholds in preference models.

A number of studies claim that preferences of humans can be represented with quasiconcave preference functions when objectives are to be maximized (Silberberg and Suen, 2001; Nicholson and Snyder, 2008). Ulu and Köksalan (2014) provided an example application of these preference functions for a multiple criteria sorting problem. Lokman et al. (2018) assumed three nondecreasing quasiconcave preference functions in their study and aimed to find the preferred solution of the DM with a specified level of accuracy. Özpeynirci et al. (2017) also worked with quasiconcave preference functions in a multiple criteria choice problem. They estimated the likelihoods that one alternative would be preferred to another and used this information to ask the DM fewer questions. When the objectives are of minimization type, quasiconvex preference functions prevail as appropriate and useful representations. In this paper, we assume a weighted L_α preference function which is a member of the set of quasiconvex preference functions. Tuncer Şakar and Karakaya (2018) also worked with a quasiconvex preference function in an interactive choice problem. Working in an iterative manner, they presented the DM sets of alternatives for her/him to select the most preferred one.

In this study, we make our experiments with data from Quacquarelli Symonds (QS), which is a higher education marketing company that provides annual rankings of universities from all over the world in different categories. Ranking of academic programs is a popular issue and the periodical lists are awaited with interest. Besides QS, publications such as Times Higher Education, US News & World Report and Financial Times also provide rankings of universities and MBA programs. In these rankings, criteria that are considered to be important in the evaluation of academic programs and their weights are predetermined. When determining the criteria, various factors such as education quality, the quality and the quantity of publications, international diversity of students and academic staff, their female ratios, the occupations and income of the alumni are considered. One can refer to Millot (2015) for a comparison of the ranking methods applied to universities and other higher education systems. Hazelkorn (2015) provided detailed and general information about these rankings. Some researchers developed their own ranking methods using published criteria values of academic institutions (Köksalan and Tuncer, 2009; Köksalan et al., 2010). However, these studies are not related to preference elicitation and guiding the DM towards preferred solutions. Our approach takes the universities in QS rankings as alternatives and uses their scores in the criteria considered as the data set. It does not use the aggregation methodology or the specific criteria weights of QS.

3. An interactive ranking-based algorithm with filtering

To structure and develop our algorithm, we modify the interactive choice algorithm of Tuncer Şakar and Karakaya (2018). That algorithm is also developed to aid the DM in arriving at a single preferred solution among a set of discrete solutions, and the preferences of the DM are assumed to be consistent with a weighted L_α preference function. The DM is again presented with a set of solutions determined with filtering in each iteration, but instead of ranking them, she/he selects the most preferred one. That algorithm was shown to produce good results in finding the most preferred solution(s) of the DM. Specifically, the advantages of determining the presented solutions with a filtering method rather than random selection were extensively shown. In addition, it provided more favorable convergence speed performance as opposed to another choice algorithm from the literature, the algorithm by Korhonen et al. (1984). Here in this study, we focus on developing a more effective way of eliciting preference information from the DM after the solutions to be presented are determined. By asking the DM to provide a preference ranking, we aim to obtain better convergence to the most preferred solution of the DM. In addition, we anticipate to obtain more accurate parameter estimates for the preference function. Instead of just guiding the DM towards a final solution, we plan to obtain good preference parameter estimates that can be used in similar problems for the DM. If our algorithm estimates the distance metric parameter α and the criteria weights vector $\boldsymbol{w}$ well, these can be used directly for other similar problems without running the algorithm.

We compare our algorithm with two benchmark algorithms from the literature. The first one is the interactive choice algorithm of Tuncer Şakar and Karakaya (2018), which was shown to outperform the widely-used algorithm by Korhonen et al. (1984). The second benchmark algorithm is the Interactive Weighted-Sums/Filtering Method of Steuer (1986). Similar to our algorithm, this algorithm guides the DM to reach a final solution through successive iterations of collecting preference information. It also uses filtering to determine the set of solutions to present the DM. It starts with a set of random weight vectors and they are filtered to obtain representative ones. For representative weight vectors, the corresponding best solutions are found with weighted sum linear programs and these solutions are also filtered to further reduce the search region. The resulting solutions are presented to the DM and the DM selects the best one. Then, the algorithm proceeds with new random weight vectors generated around the weight vector of the best solution and continues for a predetermined number of iterations. At each iteration, the ranges for the random weight vectors are tightened. The best solution of the last iteration is the final solution.

3.1. Preliminaries

Filtering refers to the process of choosing a small representative subset from a large but limited number of points. It is desired that the resulting filtered subset contains points that are different from each other so that they will be able to represent the whole set well. The filtering method that we use is *the method of first point outside the neighborhoods* (Steuer, 1986, pp. 314-318). Given a list of data points, this method selects the first point and puts it in the filtered set that will contain the representative ones. Then, starting with the second point and continuing down as necessary, it looks for the first point that is significantly different from the point in the filtered set. This point is also put into the filtered set. Proceeding down the list one by one, the next point to be retained must be significantly different from both of the points in the filtered set. Continuing in this manner, each point retained must be significantly different from all the points in the filtered set.

The method of first point outside the neighborhoods is given in Algorithm 1. X is the set of solutions to be filtered, F is the set of solutions retained by the filter, d is the distance parameter used to test significant difference, and p is the desired number of filtered solutions to obtain. At the start of Algorithm 1, there is no fixed rule to order the points in set X, the points can be ordered randomly. However, the first point in the list will necessarily be filtered. Similarly, there is no fixed d value. One can try different values for d until the desired value of p is reached. The reader is referred to Steuer (1986, pp. 316-318) for a systematic approach to shorten this trial-and-error process.

Algorithm 1
Filtering

procedure Filter
  list all points in X in an order as $x_1, x_2, \ldots, x_k$ where $k = |X|$;
  $p \leftarrow$ initial value, $d \leftarrow$ initial value, $F = \{\ \}$;
  while $|F| \neq p$ **do**
  $F = \{\ \}$, $t = 1$, $F \leftarrow F \cup \{x_t\}$ and $t \leftarrow t+1$;
    while $t \leq k$ **do**
      for each $x_i \in F$ **do**
        calculate d_{ti}, the distance of x_t from x_i;
      end for
      if $\min_{i \ni x_i \in F} d_{ti} \geq d$ **then**
        $F \leftarrow F \cup \{x_t\}$;
      end if
      $t \leftarrow t+1$;
    end while
    if $|F| < p$ **then**
      decrease the value of d;
    else if $|F| > p$ **then**
      increase the value of d;
    end if
  end while
  return F;
end procedure

For the weighted L_α preference function, when the weight vector is denoted as $\boldsymbol{w}$ ($w_j > 0$, $\sum_j w_j = 1$), the representation becomes $L_\alpha^{\boldsymbol{w}}$. In $L_\alpha^{\boldsymbol{w}}$ functions, weighted L_α

distance of a point from the ideal point is calculated. The ideal point, $\boldsymbol{z}^* \in R^n$ is defined as the point that has the best value in each criterion. It is a utopian point that does not really exist but is used as a reference point. The weighted L_α distance of vector $\boldsymbol{z} \in R^n$ to $\boldsymbol{z}^*$ in the presence of n criteria is calculated as in Eqn. 1:

$$L_\alpha^{\boldsymbol{w}}(|\boldsymbol{z}^*\text{-}\boldsymbol{z}|) = \begin{cases} \left(\sum_{j=1}^{n} \left(w_j |z_j^*\text{-}z_j|\right)^\alpha\right)^{1/\alpha} & \text{if } 1 \leqslant \alpha < \infty \\ \max_{j=1,\dots,n} \{w_j |z_j^*\text{-}z_j|\} & \text{if } \alpha = \infty \end{cases} \quad (1)$$

When a DM with this preference function is presented with the solutions to be ranked, she/he places the solution with the smallest $L_\alpha^{\boldsymbol{w}}$ value in the first position, the solution with the next smallest $L_\alpha^{\boldsymbol{w}}$ value in the second position and so on. $L_\alpha^{\boldsymbol{w}}$ function is a flexible and realistic preference function that can simulate the behavior of various DMs with different choices for the values of α and $\boldsymbol{w}$.

3.2. The proposed algorithm

Let Z be the set of available efficient solutions, $\boldsymbol{z}^{inc}$ be the incumbent solution (the most preferred solution found by the algorithm so far), k be the number of solutions presented to the DM in each iteration, r be the iteration counter, F^r be the set of filtered solutions in iteration r, S^r be the set of solutions presented to the DM in iteration r, PS be the preference set formed by all selections of the DM so far, $PS=\{(\boldsymbol{z_m},\boldsymbol{z_p}):u(\boldsymbol{z_m})<u(\boldsymbol{z_p})\}$ where $u(\cdot)$ represents the true preference function value of the DM. Finally, let α^r and $\boldsymbol{w}^r$ be the estimated parameters of the preference function in iteration r. The pseudocode of our algorithm is shown in Algorithm 2 as follows:

Algorithm 2
The proposed algorithm

```
procedure Proposed algorithm
    list all points in Z in an order as z_1, z_2, ..., z_|Z|;
    r = 1, S^r = { }, PS = { };
    select k solutions from Z with Algorithm 1, and place them in set S^r;
    while |Z| ≥ k do
        ask the DM to rank the k solutions in S^r in order of preference;
        label the first solution in the order as z^inc;
        update PS, Z←Z \ S^r;
        if termination condition is not satisfied then
            r←r+1;
            using all preference information gathered so far, PS, update α^r and w^r;
            find the inferior solutions in Z, eliminate them, and update Z;
            if |Z| < k then
                S^r= Z ∪{z^inc};
                ask the DM to rank the solutions in S^r in order of preference;
                update z^inc as the first solution in the order;
            else
                find the best solution in Z \{z^inc} according to the estimated
                preference function;
                put this solution in the first order and make a list of the points in Z;
                select k-1 solutions from Z with Algorithm 1 and put them in F^r;
                S^r= F^r ∪{z^inc};
            end if
        end if
    end while
    return z^inc;
end procedure
```

To initialize Algorithm 2, we list the solutions in Z as stated in Section 3.1 and select filtered solutions. At each iteration, we present S^r to the DM and ask her/him to rank the solutions in order of preference. The most preferred solution among these becomes the incumbent solution. The other solutions evaluated are removed from Z and they are not presented to the DM again in later iterations. Using all past preferences of the DM, we estimate the parameters of the $L_\alpha^{\boldsymbol{w}}$ function with the mathematical model defined by Eqn. 2-6 (for details, see Karakaya et al., 2018). In this model, α is the estimated parameter of the $L_\alpha^{\boldsymbol{w}}$ function, z_{ij} is the j^{th} criterion value of solution i and z_j* is the ideal value of criterion j. Decision variables w_j and ε stand for the weight of criterion j and the minimum difference between the preference function values of the solutions that the DM provides a ranking of, respectively.

$$\text{Max } \varepsilon \tag{2}$$

$$\left[\sum_{j=1}^{n}\left(w_j\left(z_j^*-z_{pj}\right)\right)^{\alpha}\right]^{1/\alpha} \geq \left[\sum_{j=1}^{n}\left(w_j\left(z_j^*-z_{mj}\right)\right)^{\alpha}\right]^{1/\alpha} + \varepsilon \quad \forall \left(\boldsymbol{z_m},\boldsymbol{z_p}\right) \in PS \tag{3}$$

$$w_j \geq \varepsilon \quad \forall j \tag{4}$$

$$\sum_{j=1}^{n} w_j = 1 \tag{5}$$

$$\varepsilon \geq 0 \tag{6}$$

We incorporate all preference information provided by the DM into the model in Eqn. 3 using the estimated preference function. At each iteration, the DM evaluates k solutions and ranks them in order of preference. To enter this information to the model, we use as few constraints as possible. We write a constraint to ensure that the first ranked solution is better than the second in terms of preference value by at least ε. We write a similar constraint for solutions ranked second and third, and continue until the rank list is exhausted. We keep writing such constraints so long as new preference information is collected at each iteration. The aim of objective function (Eqn. 2) is to find w_j values that suit the preferences of the DM for the estimated α value with the largest ε. Eqn. 4 ensures that the w_j values found are *centralized* weights in the sense that they are as far as possible from the closest constraint. This helps to restrict the weight region in a reasonable way without concentrating on extreme regions (Köksalan, 1984). Eqn. 5 normalizes w_j weights so that their sum is 1. Eqn. 6 restricts ε to be nonnegative.

We assume α to take integer values and at the beginning of Algorithm 2, we set the estimated value of α=1. We solve the above model for the current value of estimated α. In case the model is not feasible for the current α value, we increase its value iteratively, in unit increments, until the model becomes feasible. The main purpose is to find the smallest α value that supports the preferences of the DM so that we decrease computational complexity and refrain from dealing with rounding errors due to high α values. During Algorithm 2, we eliminate the inferior solutions using the approach of Karakaya et al. (2018). This enables us to speed up the algorithm by eliminating redundant questions to the DM.

If the number of solutions remaining in the solution set drops below the number of solutions presented to the DM in any iteration (k), all the solutions in the set and the incumbent solution are presented to the DM for the final time. The most preferred solution among these becomes the final solution and Algorithm 2 stops. On the other hand, if the cardinality of the solution set is above k, past preference information of the DM is utilized and the best solution in this set with respect to the estimated preference function is found. The inclusion of this solution amongst the set to be presented to the DM in the next iteration is expected to improve the algorithm. Therefore, we place it at the top of the filtering list. The solutions that we present to the DM at each iteration include the incumbent solution, so Algorithm 2 keeps the best solution found so far.

There are different termination conditions that can be applied in Algorithm 2. For example, the DM can choose to end the process with the current incumbent solution without exhausting the whole solution set or the DM and/or the analyst can determine the maximum number of iterations for the algorithm before the start. However, these termination conditions are not appropriate for simulation without real DMs. As a result, to be able to apply a consistent and objective rule, we choose to end Algorithm 2 when the improvements become insubstantial. We stop it whenever in two consecutive iterations the incumbent solution and the estimated α value do not change, and the maximum difference of the weight values in the estimated $\boldsymbol{w}$ vector is below some threshold. Even though we implement this termination condition, the progress of Algorithm 2 after the condition is also studied and reported in Section 4.

When the DM is to provide a ranking of solutions, we allow for indifference. It is possible that the DM will not be able to give a complete ranking; i.e., some solutions could have equal rank. In this case, we do not force the DM and use only the information that she/he can provide. Eqn. 3 is written only for the pairs that the DM is comfortable with. In experiments, we simulate this situation through the use of an indifference threshold. If the preference function values of two solutions do not differ by at least this threshold, we assume that the DM is indifferent and do not force a constraint. If there is a tie between the solutions in the first rank, one of them can randomly be selected as the incumbent solution. As another special case, inconsistent DMs can be observed. The preferences of a DM may not be in line with her/his preferences in previous iterations. This situation may result in infeasibility when the preference function parameters are to be estimated. To handle this issue, approaches explained in Chinnek (2008) to deal with infeasibility can be applied. For instance, the oldest DM responses that cause infeasibility can be deleted or the constraints that cause the maximum level of infeasibility may be removed.

4. Experimental studies

In this section, first we introduce the data set that we use in our experiments. Subsequently we present and discuss our applications and results.

4.1. Test problems

QS marketing company publishes annual rankings of universities from all over the world under various categories like World university rankings, Asia university rankings, Global MBA rankings and so on (QS Top Universities, 2018). For different categories, QS uses different criteria to evaluate the contenders. From 2018 rankings, we have selected several data sets from different categories to apply our method. Firstly, we studied the list of World universities that are evaluated with respect to 6 criteria, all of which are to be maximized. These criteria are *academic reputation, citations per faculty, employer reputation, faculty-student ratio, international faculty ratio* and *international student ratio*. Out of this list, we were able to identify 50 efficient universities. We name this set W50-6 for our experiments. Then, we aimed to obtain a larger set of efficient solutions and found 100 such solutions from the Engineering and Technology list. The 4 criteria for this list are *academic reputation, employer reputation, citations per paper* and *h-index citations*. We name this set E100-4. As we were not able to obtain other efficient sets with 100 solutions or higher, we studied other categories for efficient sets with cardinalities between 50 and 100. We identified 70 efficient solutions from Life Sciences and Medicine (L70-4) and Arts and Humanities (A70-4) lists that are evaluated with respect to the same criteria as E100-4. Lastly, using a different region of the Engineering and Technology list (different from E100-4), we found another 70 efficient solutions (E70-4). As a result, we have 5 data sets in total: A70-4, E70-4, L70-4, E100-4, and W50-6. For all these sets, we take the criteria values of the corresponding universities published by QS. We do not take the QS weights or the rankings into account.

The parameters of the preference function we use in this study are α and $\boldsymbol{w}$. To be able to test the algorithm for different underlying preference functions, we assume various values for these parameters. In our experiments, we use 1, 2, 3, 4, 5, 6 and infinity for α. In this way, we can model a wide variety of preference functions including linear, quadratic, and Tchebycheff forms. We simulate DM responses consistent with the assumed preference functions. When determining $\boldsymbol{w}$ vectors, we aimed for sets that represent the entire weight space well. For this purpose, we used the 50-50 strategy explained in set discretization in Steuer (1986, pp. 326-330). As illustrated in this reference, random weight vectors selected from the uniform distribution tend to gather around in the center region of the weight space, which means the weights are close to each other. On the other hand, weight vectors selected from the Weibull distribution tend to have high values for some weights and low values for the others. Selecting 50% of the weights from the uniform and the other 50% from the Weibull distribution results in a well-spread set of weights. As in Steuer (1986, pp. 326-330), we select the scale parameter of Weibull distribution as 0.1 and the shape parameter as 0.3; these parameters produce well-dispersed results for our experiments. Furthermore, we eliminate any weight vector that has a component less than 0.01 to be more realistic. Since we are experimenting with 4- and 6-criteria sets, we need two weight sets. For both cases, we generated 500 random weight vectors from uniform and 500 random weight vectors from Weibull distributions. We normalized each vector by its L_1 norm so that the summation of criteria weights is 1. Randomly listing the 1,000 weight vectors for both cases, we applied the filtering method of first point outside the neighborhoods that was shown in Algorithm 1 in Section 3.1. The resulting 50 weight vectors for 4-criteria and 6-criteria cases are provided in Tables A1 and A2 in the Appendix, respectively.

4.2. Computational results

In our experiments, we present 5 solutions to the DM in each iteration. This parameter allows for adequate preference information elicitation without imposing too much cognitive burden on the DM. As proposed in Karakaya et al. (2018), when the estimated α value is larger than 4, we accept it as infinity. This is done to avoid computing complications and round off errors in execution. The stopping condition is applied as follows: in two consecutive iterations, if the incumbent solution and the estimated α are the same, and the largest absolute difference between the corresponding weights in estimated weight vectors is less than 10^{-3}, we stop. The indifference threshold is set to 10^{-8}. Euclidean distance is used for the filtering method. We define the ideal point as the point that has the best value in each criterion. We then scale the values of the alternatives between 0 and 1.

The first benchmark algorithm, the interactive choice algorithm of Tuncer Şakar and Karakaya (2018) is also executed with the same parameter settings and stopping condition. We refer to this benchmark algorithm as the Selection-based Algorithm (SA) and our algorithm as the Ranking-based Algorithm (RA). We refer to the second benchmark algorithm, the Interactive Weighted-Sums/Filtering Method, as IWS. Following the guideline in Steuer (1986, pp. 394-396), we set the number of iterations and the number of solutions presented to the DM at each iteration to 6 in data set W50-6, and to 5 in all other data sets. Algorithms are coded with C++ programming language; all experiments are run on a computer with Intel(R) Core(TM) i7-6700 CPU @ 3.40 GHz, 8 GB RAM, 64-bit Microsoft Windows 10 operating system. The mathematical models are solved with GAMS 23.9.

We have determined several measures to compare RA against SA and IWS. For each problem setting, we run the algorithms with 50 weight vectors and report the overall performances. *Average absolute deviation* calculates the average difference between the true preference function values of the solutions found with the algorithms and the true best solution for 50 weight vectors. For *average proportional deviation*, we calculate the ratio of absolute deviation measure to the preference function value difference of true best and worst solutions for each weight vector and report the average values. This metric gives rate information about the error in the algorithms' final solution values. In each problem setting with 50 different weight vectors, we report the number of times the algorithms fail to find the true best solution (at the termination condition) with *number best not found*. If the algorithms find the true best solution until the termination condition, this metric takes a value of 0. If for example, they cannot find the true best solution with 4 weight vectors, it becomes 4. *Average iteration* reports the average number of iterations until the termination condition is met. *Average best found* gives the average number of iterations it takes for the algorithms to find the true best solution. RA and SA guarantee to find the true best solution if the whole solution set is exhausted; so for these two algorithms, this measure reports the average performance in all 50 weight vectors. IWS, on the other hand, cannot find the true best solution for some weight vectors. For IWS, we report the average number of iterations to reach the true best solution among the weight vectors where the true best solution could be found. Lastly*, average eliminated* represents the average number of solutions eliminated throughout RA and SA because they are in dominated regions defined by DM responses. Since IWS does not eliminate solutions throughout its execution, we do not report this measure for IWS. Table 1 contains the comparison of RA, SA and IWS with respect to these measures.

In Table 1, the first column shows that in 28 cases out of 35, RA has lower average absolute deviation than SA. In 5 cases SA has lower values and in 2 cases these two algorithms have equal values. Comparing RA and IWS in that column, we see that in all 35 cases, RA has lower values than IWS. RA obtained low deviations close to 0 in most of the cases. When we study average proportional deviation, we see that for 28 cases out of 35, RA has lower scores than SA, for 5 cases SA has lower scores, and for 2 cases they are equal. Again in all cases, RA has lower values than IWS. Looking at the number of times that the algorithms failed to find the true best solution of the DM, we observe that in only 2 cases SA has lower values than RA, and in both cases the difference between the algorithms is just 1. In 31 cases RA has lower values and in 2 cases SA and RA are equal. For all cases, RA has lower values than IWS. For many problem settings, RA is able to find the best solution until the termination condition. If we compare the algorithms with respect to the average number of iterations until the termination condition, for most cases SA and IWS have slightly lower values than RA. However, termination of an algorithm at earlier stages is not a clear indicator of superiority as an algorithm may stop without generating a good enough solution. In terms of the average number of iterations it takes to find the true best solution, RA has lower values than SA for all cases. For RA, the average number of iterations to find the best solution ranges from 2.06 to 3.84, considerably low values. IWS has lower values than RA in this measure; however, we should note that the values for IWS are calculated by taking the average of the cases where it was able to find the best solution. So, the performance in the average best found measure should be considered together with the number best not found measure. In the last column, we see the average number of solutions eliminated from the solution set as a result of DM responses. It can be observed that RA is able to eliminate considerably higher number of solutions than SA. This shows that RA can obtain more useful information from the DM and thus can reduce the solution space more effectively.

We also conduct statistical tests to compare the performance of RA against SA and IWS, we use paired *t*-test for this purpose. To apply paired *t*-test, normality assumption should be satisfied for the sampling distributions. However, by the Central Limit Theorem, for sample sizes of at least 30, the sampling distribution is approximately normal regardless of the probability distribution of the sampled population (Mendelhall and Sincich, 1996, pp. 36, 60). Since we carry out our comparisons with 50 weight vectors, it is appropriate to use paired *t*-test. We calculate the paired differences of absolute deviation, proportional deviation and number best not found, and build 95% confidence intervals for these differences. SA/IWS value - RA value for 35 test cases are given in Table A3 in Appendix. In absolute deviation, RA is superior to SA in 10 cases and superior to IWS in 31 cases. In no cases the benchmarks are superior to RA. In proportional deviation, RA outperforms SA and IWS in 6 and 31 cases, respectively. Again the benchmarks never outperform RA. In number best not found, RA is better than SA and IWS in 15 and 35 cases, respectively; the benchmarks are never better than RA. In iteration measure, however, in 9 cases SA has lower values than RA. Between RA and IWS, IWS has lower values in 7 cases and RA in 6 cases. Considering Table 1 and A3 together, we can conclude that the overall performance of RA is superior to the benchmarks, especially IWS.

Table 1
Comparison of the performances of RA, SA and IWS

α	Set	Algorithm	Average absolute deviation	Average proportional deviation	Number best not found	Average iteration	Average best found	Average eliminated
1	A70-4	RA	0.00000	0.00000	0	4.80	2.24	29.78
		SA	0.00337	0.00582	6	4.94	3.22	23.34
		IWS	0.00182	0.00369	10	5.00	1.98	-
	E70-4	RA	0.00000	0.00000	0	4.72	2.24	29.80
		SA	0.00000	0.00000	0	4.72	2.54	23.38
		IWS	0.00428	0.00887	8	5.00	1.65	-
	L70-4	RA	0.00000	0.00000	0	4.68	2.08	31.84
		SA	0.00000	0.00000	0	4.62	2.46	25.00
		IWS	0.00381	0.01166	8	5.00	1.69	-
	E100-4	RA	0.00000	0.00000	0	5.08	2.30	51.28
		SA	0.00014	0.00043	1	4.90	2.62	42.50
		IWS	0.00172	0.00340	7	5.00	1.53	-
	W50-6	RA	0.00000	0.00000	0	5.28	2.06	9.14
		SA	0.00003	0.00003	1	5.44	2.22	5.92
		IWS	0.00130	0.00205	6	6.00	1.24	-
2	A70-4	RA	0.00026	0.00064	4	5.12	2.62	34.04
		SA	0.00138	0.00314	8	5.06	3.48	22.66
		IWS	0.00226	0.00792	12	5.00	2.05	-
	E70-4	RA	0.00000	0.00000	0	5.48	2.78	36.16
		SA	0.00050	0.00174	4	4.86	2.94	22.54
		IWS	0.00496	0.01042	17	5.00	1.62	-
	L70-4	RA	0.00014	0.00048	2	5.14	2.10	35.08
		SA	0.00007	0.00019	1	4.90	2.36	23.34
		IWS	0.00622	0.02302	18	5.00	2.11	-
	E100-4	RA	0.00000	0.00000	0	5.86	2.94	58.84
		SA	0.00117	0.00404	8	5.12	3.54	43.20
		IWS	0.00750	0.02307	24	5.00	2.00	-
	W50-6	RA	0.00072	0.00252	2	5.18	2.60	12.08
		SA	0.00141	0.00477	5	5.04	3.18	6.68
		IWS	0.00497	0.01440	17	6.00	1.71	-
3	A70-4	RA	0.00144	0.00486	5	4.98	2.90	35.28
		SA	0.00176	0.00434	7	5.02	3.24	22.34
		IWS	0.00291	0.00973	12	5.00	2.05	-

Table 1 (cont'd)

	E70-4	RA	0.00000	0.00000	0	5.46	2.92	37.74
		SA	0.00078	0.00322	4	4.92	2.94	23.78
		IWS	0.00539	0.01188	19	5.00	1.68	-
	L70-4	RA	0.00034	0.00148	2	5.20	2.20	36.00
		SA	0.00036	0.00153	3	4.88	2.72	23.24
		IWS	0.01066	0.03820	21	5.00	2.17	-
	E100-4	RA	0.00000	0.00000	0	5.96	3.24	60.42
		SA	0.00194	0.00597	9	5.32	3.88	43.14
		IWS	0.01034	0.03156	25	5.00	2.34	-
	W50-6	RA	0.00123	0.00434	4	5.08	2.70	13.56
		SA	0.00119	0.00460	3	5.08	3.00	7.30
		IWS	0.00642	0.01906	21	6.00	1.88	-
4	A70-4	RA	0.00262	0.00794	6	4.88	3.04	36.66
		SA	0.00204	0.00510	9	5.06	3.48	23.10
		IWS	0.00386	0.01281	14	5.00	2.00	-
	E70-4	RA	0.00000	0.00000	0	5.18	2.94	39.74
		SA	0.00071	0.00323	3	5.10	3.16	24.48
		IWS	0.00640	0.01434	22	5.00	1.81	-
	L70-4	RA	0.00017	0.00062	2	5.42	2.40	37.18
		SA	0.00074	0.00216	6	4.88	3.00	23.56
		IWS	0.01108	0.03913	23	5.00	2.15	-
	E100-4	RA	0.00000	0.00000	0	6.32	3.16	61.18
		SA	0.00231	0.00752	9	5.42	4.00	43.46
		IWS	0.01291	0.03984	27	5.00	2.42	-
	W50-6	RA	0.00133	0.00478	4	4.96	2.80	14.36
		SA	0.00143	0.00550	5	5.04	3.06	7.18
		IWS	0.00737	0.02229	21	6.00	1.76	-
5	A70-4	RA	0.00269	0.00831	5	5.10	3.12	37.72
		SA	0.00242	0.00590	11	5.02	3.56	24.18
		IWS	0.00457	0.01503	17	5.00	2.03	-
	E70-4	RA	0.00000	0.00000	0	5.42	3.16	41.06
		SA	0.00093	0.00399	6	5.10	3.36	25.06
		IWS	0.00710	0.01605	22	5.00	1.87	-
	L70-4	RA	0.00000	0.00000	0	5.54	2.30	37.72
		SA	0.00067	0.00179	5	4.94	2.94	22.94
		IWS	0.01225	0.04230	25	5.00	2.06	-
	E100-4	RA	0.00077	0.00271	2	6.12	3.30	62.48
		SA	0.00210	0.00742	8	5.48	4.04	43.88

Table 1 (cont'd)

		IWS	0.01498	0.04692	28	5.00	2.48	-
	W50-6	RA	0.00043	0.00105	3	5.34	2.80	14.72
		SA	0.00189	0.00619	6	4.96	3.12	7.44
		IWS	0.00760	0.02277	21	6.00	1.76	-
6	A70-4	RA	0.00346	0.01031	7	5.24	3.31	37.88
		SA	0.00266	0.00649	11	5.12	3.76	24.16
		IWS	0.00441	0.01333	19	5.00	2.09	-
	E70-4	RA	0.00000	0.00000	0	5.42	3.32	41.42
		SA	0.00114	0.00466	7	5.08	3.54	25.04
		IWS	0.00765	0.01739	22	5.00	1.87	-
	L70-4	RA	0.00000	0.00000	1	5.48	2.50	38.20
		SA	0.00076	0.00204	7	4.94	3.12	23.62
		IWS	0.01339	0.04639	28	5.00	2.14	-
	E100-4	RA	0.00084	0.00291	3	6.30	3.50	63.14
		SA	0.00232	0.00833	9	5.50	4.20	44.00
		IWS	0.01656	0.05182	30	5.00	2.52	-
	W50-6	RA	0.00042	0.00101	3	5.32	2.88	15.84
		SA	0.00092	0.00230	5	4.96	3.06	8.16
		IWS	0.00778	0.02308	22	6.00	1.78	-
∞	A70-4	RA	0.00273	0.00683	7	5.00	3.58	39.38
		SA	0.00320	0.00835	11	5.22	4.12	25.90
		IWS	0.00824	0.02516	24	5.00	2.11	-
	E70-4	RA	0.00050	0.00084	2	5.28	3.46	42.88
		SA	0.00277	0.00946	10	5.00	3.70	26.92
		IWS	0.01076	0.02468	24	5.00	1.78	-
	L70-4	RA	0.00011	0.00039	1	5.38	2.70	40.20
		SA	0.00244	0.00708	10	4.92	3.50	25.24
		IWS	0.01662	0.05425	28	5.00	2.00	-
	E100-4	RA	0.00348	0.00620	5	5.82	3.84	65.28
		SA	0.00380	0.01388	10	5.52	4.40	44.74
		IWS	0.02120	0.06396	32	5.00	2.33	-
	W50-6	RA	0.00018	0.00036	1	5.34	2.92	17.38
		SA	0.00091	0.00245	6	5.00	3.18	8.24
		IWS	0.00844	0.02464	23	6.00	1.62	-

The primary goal of RA (and also SA and IWS) is to guide the DM to a good, if not the best, final solution among a set of alternatives. However, since the algorithm proceeds by estimating the parameters of the preference function using DM responses, the final parameter estimates can be useful information as well. If good estimates can be obtained, these can be utilized to rank the solutions. So besides a highly-preferred solution, DM can also be presented with a ranking of all solutions. Furthermore, parameter estimates can be used for similar problems of the DM in the future. If preferences of the DM can be modeled beforehand, it will not be necessary to consider the whole solution set iteratively. Therefore, we also tested the precision of the parameter estimates of RA. Since RA works with the smallest possible α value for reasons explained before, it is not appropriate to directly consider the deviations of the estimated weights from the true weights of the DM. As the estimated α can be lower than the true α, we do not expect full agreement between the estimated weights and true weights. Consequently, we choose to employ a metric that will show how close the ranking of solutions formed by estimated parameters is to that formed by true parameters. Kendall rank correlation coefficient, also known as Kendall's Tau coefficient, is a widely used metric for this purpose (see Daniel, 1990). Kendall's Tau coefficient measures the similarity between two orderings of the same data. If it is 1, two orderings are exactly the same; if it is -1, two orderings are perfect opposites of each other. A coefficient of 0 shows that there is no relationship between the orderings. We calculate Kendall's Tau coefficients for each algorithm using the true rankings of the problem settings. To compare RA and SA fairly, we use the final parameter estimates of these algorithms after all solutions are exhausted. Since the average number of iterations until the termination condition for RA is higher than SA, we do not want to favor RA unfairly by using more iterations from it. In addition, if the parameter estimates are going to be used for similar future problems, it is reasonable to try to obtain as precise parameters as possible. Therefore, for RA and SA, Kendall's Tau coefficient is calculated with parameters estimated after a single solution remains in the set. Unlike RA and SA, IWS does not eliminate solutions from the feasible set as it progresses. So for Kendall's Tau coefficient of IWS, we run it for the average number of iterations it takes for RA and SA to have a single solution left, which is 15. The results are summarized in Table 2; for each case, we report the average of the results of the 50 weight vectors. It can be seen that all Tau coefficients for RA are high values close to 1. On the other hand, Tau coefficients range between 0.5180 and 0.8121 for SA, and between 0.3588 and 0.7220 for IWS. Moreover, the standard deviations for the Tau coefficients of RA are substantially lower than those of SA and IWS. These low standard deviations show that the high average coefficients that we report for RA are reliable. Based on the good performance of RA in Table 2, we can make this conclusion: after RA is run with a DM, the resulting preference parameter estimates can be used to assist that DM in similar problems. For example, if the DM is confronted with another set of universities evaluated with the same criteria, previously estimated parameters will directly provide a good ranking of the new universities.

Table 2

Comparison of RA, SA and IWS with respect to Kendall's Tau coefficient

α	Set	Algorithm	Average Tau	Std. dev. of Tau	α	Set	Algorithm	Average Tau	Std. dev. of Tau
1	A70-4	RA	0.9694	0.0219	5	A70-4	RA	0.9357	0.0376
		SA	0.7978	0.1745			SA	0.6912	0.2277
		IWS	0.7220	0.1724			IWS	0.6097	0.1846
	E70-4	RA	0.9582	0.0265		E70-4	RA	0.9347	0.0498
		SA	0.7925	0.1638			SA	0.7035	0.1543
		IWS	0.6402	0.2054			IWS	0.5417	0.1947
	L70-4	RA	0.9539	0.0578		L70-4	RA	0.9398	0.0497
		SA	0.7725	0.1324			SA	0.6049	0.2553
		IWS	0.6882	0.1571			IWS	0.5326	0.1996
	E100-4	RA	0.9765	0.0141		E100-4	RA	0.9391	0.0471
		SA	0.8121	0.1651			SA	0.6762	0.2265
		IWS	0.6954	0.2074			IWS	0.5376	0.1990
	W50-6	RA	0.9502	0.0259		W50-6	RA	0.9119	0.0550
		SA	0.6072	0.2394			SA	0.5420	0.2082
		IWS	0.5238	0.1920			IWS	0.3769	0.1701
2	A70-4	RA	0.9577	0.0459	6	A70-4	RA	0.9347	0.0426
		SA	0.7606	0.1920			SA	0.6869	0.2270
		IWS	0.6605	0.2083			IWS	0.5882	0.2046
	E70-4	RA	0.9681	0.0278		E70-4	RA	0.9324	0.0469
		SA	0.7363	0.1431			SA	0.7026	0.1581
		IWS	0.5670	0.2042			IWS	0.5339	0.1937
	L70-4	RA	0.9601	0.0407		L70-4	RA	0.9264	0.0569
		SA	0.6944	0.2056			SA	0.6247	0.2572
		IWS	0.6362	0.1708			IWS	0.5211	0.2023
	E100-4	RA	0.9775	0.0160		E100-4	RA	0.9363	0.0419
		SA	0.7452	0.1831			SA	0.6671	0.2236
		IWS	0.6293	0.1838			IWS	0.5281	0.2010

Table 2 (cont'd)

	W50-6	RA	0.9368	0.0456		W50-6	RA	0.9169	0.0383
		SA	0.6013	0.2414			SA	0.5369	0.2078
		IWS	0.4441	0.1786			IWS	0.3724	0.1693
3	A70-4	RA	0.9528	0.0318	∞	A70-4	RA	0.9370	0.0454
		SA	0.7125	0.2243			SA	0.7015	0.2405
		IWS	0.6397	0.1710			IWS	0.5938	0.1881
	E70-4	RA	0.9494	0.0369		E70-4	RA	0.9377	0.0619
		SA	0.7122	0.1486			SA	0.7080	0.1641
		IWS	0.5503	0.1952			IWS	0.5100	0.2064
	L70-4	RA	0.9436	0.0499		L70-4	RA	0.9501	0.0589
		SA	0.6475	0.2370			SA	0.6401	0.2396
		IWS	0.5546	0.2076			IWS	0.5044	0.2006
	E100-4	RA	0.9579	0.0331		E100-4	RA	0.9556	0.0479
		SA	0.7024	0.2019			SA	0.6745	0.2321
		IWS	0.5811	0.1862			IWS	0.5039	0.2198
	W50-6	RA	0.9236	0.0408		W50-6	RA	0.9242	0.0486
		SA	0.5609	0.2034			SA	0.5180	0.2090
		IWS	0.4038	0.1675			IWS	0.3588	0.1915
4	A70-4	RA	0.9410	0.0394					
		SA	0.7050	0.2261					
		IWS	0.6190	0.1830					
	E70-4	RA	0.9390	0.0481					
		SA	0.7032	0.1621					
		IWS	0.5487	0.1941					
	L70-4	RA	0.9462	0.0501					
		SA	0.6387	0.2513					
		IWS	0.5448	0.1988					
	E100-4	RA	0.9524	0.0360					
		SA	0.6860	0.2030					
		IWS	0.5553	0.1813					
	W50-6	RA	0.9169	0.0496					
		SA	0.5410	0.1977					
		IWS	0.3878	0.1677					

5. Conclusions

Many decision making problems involve numerous solutions and these solutions are typically evaluated with multiple conflicting criteria. It is not easy and straightforward for a DM to select the best among numerous solutions. In this study, we propose an interactive algorithm, RA, to guide the DM to a preferred solution. Through successive iterations, we collect preference information from the DM and use this information to reduce the solution space as well as to guide the search. At each iteration, the DM is asked to provide a preference ranking of a small number of solutions. These solutions are determined with the help of a filtering routine to increase the efficiency of the algorithm. Using the rankings of the DM, we estimate the parameters of the DM's preference function. The DM is assumed to have an underlying L_α^w function, a flexible function suited to model the behavior of real DMs. RA continues until a predetermined termination condition is satisfied and a highly preferable solution is presented to the DM in the end. As a side benefit of RA, the estimated parameters can also be used to obtain a ranking of solutions, either for the problem on hand or similar future problems.

RA is tested with university data taken from QS rankings. We compose various settings with different sets of universities and criteria, different α values and weight vectors for L_α^w preference function. The results show that RA successfully converges to the most preferred solution of the DM and produces good estimates of the preference function parameters. RA is tested against two benchmark algorithms, SA and IWS, with respect to several measures. Compared to SA and IWS, RA is able to find the most preferred solution for most of the cases considered, and the average deviation from the preference value of the true best solution of the DM is lower for RA. Furthermore, the resulting estimates of the preference function are more accurate for RA than the benchmarks. In all three algorithms, the DM is presented with a small set of solutions at each iteration and these solutions are generated with the help of a filtering procedure. The DM selects the best solution in SA and IWS whereas she/he ranks them in RA. The rankings provide more preference information and lead to RA having better convergence performance than SA and IWS. RA is observed to be particularly superior to IWS, because IWS does not update the assumed form of the preference function as the DM answers more questions. RA, on the other hand, uses a flexible preference function whose mathematical form can be updated as iterations continue. This update is also done with the preference rankings of the DM without the need for additional information. However, it should be noted that ranking the solutions is a more difficult task for the DM than selecting the best one.

As future studies, different procedures can be applied in the preference elicitation part of the algorithm to increase the convergence performance of the algorithm so that the DM will be required to answer fewer questions. The solutions presented to the DM can be determined with different heuristics to reduce the number of iterations. In addition, the DM may be asked to provide preference information in formats other than ranking.

References

AHN, B. S. and PARK, K. S. (2008), "Comparing methods for multiattribute decision making with ordinal weights", *Computers & Operations Research,* 35(5), 1660-1670.

ANGILELLA, S., GRECO, S., LAMANTIA, F. and MATARAZZO, B. (2004), "Assessing non-additive utility for multicriteria decision aid", *European Journal of Operational Research,* 158(3), 734-744.

QS Top Universities (2018), *QS University Rankings*, available at: https://www.topuniversities.com/university-rankings, accessed: 1 May 2018.

BENABBOU, N., GONZALES, C., PERNY, P. and VIAPPIANI, P. (2015), "Minimax regret approaches for preference elicitation with rank-dependent aggregators", *EURO Journal on Decision Processes,* 3(1-2), 29-64.

BRANKE, J., CORRENTE, S., GRECO, S. and GUTJAHR, W. J. (2015), "Using indifference information in robust ordinal regression", *International Conference on Evolutionary Multi-Criterion Optimization.* 205-217.

BRANKE, J., CORRENTE, S., GRECO, S. and GUTJAHR, W. (2017), "Efficient pairwise preference elicitation allowing for indifference", *Computers & Operations Research,* 88, 175-186.

CHINNEK, J. W. (2008), *Feasibility and infeasibility in optimization: Algorithms and computational methods,* New York: Springer, 89-210.

DANIEL, W. W. (1990), *Applied nonparametric statistics,* Boston: The Duxbury advanced series in statistics and decision sciences.

GRECO, S., MOUSSEAU, V. and SŁOWINSKI, R. (2008), "Ordinal regression revisited: multiple criteria ranking using a set of additive value functions", *European Journal of Operational Research,* 191(2), 416-436.

HAZELKORN, E. (2015), *Rankings and the Reshaping of Higher Education: The Battle for World-Class Excellence,* New York: Palgrave Macmillan.

JACQUET-LAGRÈZE, E. and SISKOS, Y. (1982), "Assessing a set of additive utility functions for multicriteria decision-making, the UTA method", *European Journal of Operational Research,* 10(2), 151-164.

KADZINSKI, M. and TERVONEN, T. (2013), "Robust multi-criteria ranking with additive value models and holistic pair-wise preference statements", *European Journal of Operational Research,* 228(1), 169-180.

KARAGIANNIDIS, A. and PERKOULIDIS, G. (2009), "A multi-criteria ranking of different technologies for the anaerobic digestion for energy recovery of the organic fraction of municipal solid wastes", *Bioresource Technology,* 100(8), 2355-2360.

KARAKAYA, G., KÖKSALAN, M. and AHİPAŞAOĞLU, S. D. (2018), "Interactive algorithms for a broad underlying family of preference functions", *European Journal of Operational Research,* 265, 248-262.

KEENEY, R. L. and RAIFFA, H. (1976), *Decisions with Multiple Objectives: Preferences and Value Tradeoffs.* New York: Wiley.

KORHONEN P., WALLENIUS J. and ZIONTS S. (1984), "Solving the Discrete Multiple Criteria Problem using Convex Cones", *Management Science*, 30(11), 1336-1345.

KÖKSALAN, M. and TUNCER, C. (2009), "A DEA-Based Approach to Ranking Multi-Criteria Alternatives", *International Journal of Information Technology & Decision Making,* 8, 29-54.

KÖKSALAN, M. (1984), *Multiple criteria decision making with discrete alternatives,* Ph.D. Dissertation, Buffalo: State University New York.

KÖKSALAN, M., BÜYÜKBAŞARAN, T., ÖZPEYNİRCİ, Ö. and WALLENIUS, J. (2010), "A Flexible Approach to Ranking with an Application to MBA Programs", *European Journal of Operational Research,* 201, 470-476.

LOKMAN, B., KÖKSALAN, M., KORHONEN, P. J. and WALLENIUS, J. (2018), "An interactive approximation algorithm for multi-objective integer programs", *Computers and Operations Research,* 96, 80-90.

MARDANI, A., JUSOH, A., Nor, K. MD, KHALIFAH, Z., ZAKWAN, N. And VALIPOUR, A. (2015), "Multiple criteria decision-making techniques and their applications – a review of the literature from 2000 to 2014", *Economic Research,* 28(1), 516-571.

MARICHAL, J. L. and ROUBENS, M. (2000), "Determination of weights of interacting criteria from a reference set", *European Journal of Operational Research,* 124(3), 641-650.

MENDELHALL, W. and SINCICH, T. (1996), *A Second Course in Statistics: Regression Analysis*, New Jersey: Prentice-Hall.

MILLOT, B. (2015), "International rankings: Universities vs. higher education systems", *International Journal of Educational Development,* 40, 156-165.

MUNDA, G. and NARDO, M. (2009), "Noncompensatory/nonlinear composite indicators for ranking countries: a defensible setting", *Applied Economics,* 41, 1513-1523.

NICHOLSON, W. and SNYDER, C. (2008), *Microeconomic theory: Basic Principles and extensions,* Ohio:Thomson/South-Western.

NUUTER, T., LILL, I. and TUPENAITE, L. (2015), "Comparison of housing market sustainability in European countries based on multiple criteria assessment", *Land Use Policy,* 42, 642-651.

ÖZPEYNİRCİ, Ö., ÖZPEYNİRCİ, S. and KAYA, A. (2017), "An interactive approach for multiple criteria selection problem", *Computers and Operations Research,* 78, 154-162.

PIRLOT, M. and VINCKE P. (2013), Semiorders: Properties, Representations, Applications, eBook: Springer Science & Business Media.

ROY, B. (2016), *Paradigm and challenges, Multiple Criteria Decision Analysis: State of the Art Surveys,* Berlin: Springer, 19-39.

SALO, A. A. and HÄMÄLÄINEN, R. P. (2001), "Preference ratios in multiattribute evaluation (prime)-elicitation and decision procedures under incomplete information", *IEEE Transactions on Systems, Man, and Cybernetics - Part A,* 31(6), 533-545.

SARABANDO, P. and DIAS, L. C. (2010), "Simple procedures of choice in multicriteria problems without precise information about the alternatives' values", *Computers & Operations Research,* 37(12), 2239-2247.

SILBERBERG, E. and SUEN, W. C. (2001), *The structure of economics: A mathematical anal-ysis,* New York:McGraw-Hill.

SISKOS, Y., GRIGOROUDIS, E. and MATSATSINIS, N. F. (2016), "UTA methods", M*ultiple Criteria Decision Analysis: State of the Art Surveys,* Berlin: Springer, 315-362.

STEUER, R. E. (1986), *Multiple Criteria Optimization: Theory, Computation and Application*, New York: John Wiley & Sons.

THOKALA, P. and DUENAS, A. (2012), "Multiple Criteria Decision Analysis for Health Technology Assessment", *Value in Health,* 15(8), 1172-1181.

TUNCER ŞAKAR, C. and KARAKAYA, G. (2019), "An Interactive Approach with Filtering to Find a Preferred Solution of a Decision Maker", *Journal of the Faculty of Engineering and Architecture of Gazi University*, 34(4), 1987-2005.

ULU, C. and KÖKSALAN, M. (2014), "An interactive approach to multicriteria sorting for quasiconcave value functions", *Naval Research Logistics,* 61(6), 447-457.

WAKKER, P. P. (1989), *Additive Representations of Preferences: A New Foundation of Decision Analysis,* Dordrecht: Springer.

YALCIN, N., BAYRAKDAROGLU, A. and KAHRAMAN, C. (2012), "Application of fuzzy multi-criteria decision making methods for financial performance evaluation of Turkish manufacturing industries", *Expert Systems with Applications,* 39(1), 350-364.

Appendix

Table A1

Weight vectors used in the experiments for 4-criteria problems

#	w_1	w_2	w_3	w_4	#	w_1	w_2	w_3	w_4
1	0.4000	0.3000	0.1500	0.1500	**26**	0.1955	0.3737	0.2599	0.1708
2	0.0104	0.1279	0.0957	0.7661	**27**	0.2620	0.0136	0.1255	0.5988
3	0.7505	0.1021	0.0234	0.1240	**28**	0.2672	0.1215	0.5482	0.0631
4	0.4272	0.5304	0.0158	0.0266	**29**	0.4154	0.2544	0.0671	0.2630
5	0.2972	0.2230	0.2897	0.1901	**30**	0.6734	0.0468	0.2526	0.0273
6	0.4714	0.3272	0.1128	0.0886	**31**	0.1842	0.6517	0.0832	0.0809
7	0.0851	0.5473	0.0384	0.3293	**32**	0.0689	0.0552	0.2509	0.6249
8	0.0123	0.3164	0.6455	0.0259	**33**	0.0356	0.4345	0.3454	0.1845
9	0.1705	0.0241	0.4302	0.3752	**34**	0.1925	0.3810	0.1293	0.2973
10	0.3565	0.4145	0.0160	0.2130	**35**	0.0170	0.3421	0.0274	0.6135
11	0.2842	0.2037	0.0377	0.4744	**36**	0.2749	0.4794	0.1307	0.1150
12	0.0696	0.0931	0.7753	0.0620	**37**	0.0172	0.0223	0.6581	0.3024
13	0.9318	0.0201	0.0369	0.0112	**38**	0.3204	0.0361	0.4432	0.2003
14	0.1223	0.5379	0.2095	0.1303	**39**	0.1774	0.2807	0.4678	0.0741
15	0.1568	0.0242	0.0545	0.7645	**40**	0.5909	0.0287	0.0737	0.3068
16	0.4982	0.0922	0.3114	0.0982	**41**	0.0210	0.4409	0.1876	0.3505
17	0.3534	0.0643	0.2920	0.2902	**42**	0.2222	0.0156	0.7494	0.0129
18	0.0309	0.0178	0.0266	0.9247	**43**	0.6273	0.3123	0.0438	0.0166
19	0.0931	0.2498	0.3856	0.2715	**44**	0.0460	0.4234	0.5027	0.0278
20	0.0724	0.8769	0.0309	0.0199	**45**	0.0182	0.7500	0.1869	0.0448
21	0.0289	0.2939	0.2112	0.4660	**46**	0.0142	0.5703	0.3834	0.0320
22	0.0419	0.2076	0.5721	0.1784	**47**	0.3500	0.2685	0.3522	0.0294
23	0.1894	0.1477	0.2777	0.3853	**48**	0.0149	0.7696	0.0316	0.1840
24	0.4419	0.0185	0.1035	0.4361	**49**	0.4173	0.0122	0.5153	0.0552
25	0.0164	0.0540	0.4222	0.5073	**50**	0.1316	0.3737	0.0200	0.4747

Table A2

Weight Vectors Used in the Experiments for 6-Criteria Problems

#	w_1	w_2	w_3	w_4	w_5	w_6	#	w_1	w_2	w_3	w_4	w_5	w_6
1	0.4000	0.2000	0.1000	0.2000	0.0500	0.0500	**26**	0.3068	0.0439	0.1614	0.4043	0.0633	0.0204
2	0.1668	0.1917	0.0750	0.0633	0.3070	0.1962	**27**	0.0713	0.3343	0.0342	0.2392	0.2270	0.0939
3	0.3408	0.0274	0.1474	0.1238	0.3049	0.0556	**28**	0.0334	0.0157	0.5232	0.1054	0.0343	0.2879
4	0.2687	0.1623	0.0533	0.2480	0.1705	0.0972	**29**	0.1200	0.0357	0.1525	0.3478	0.0850	0.2590
5	0.0366	0.0701	0.1510	0.2418	0.3453	0.1552	**30**	0.0224	0.2719	0.3036	0.2546	0.1210	0.0264
6	0.0146	0.3188	0.2438	0.0643	0.0941	0.2644	**31**	0.0164	0.0107	0.0129	0.0269	0.0121	0.9210
7	0.4445	0.2344	0.0117	0.1652	0.0177	0.1265	**32**	0.0219	0.2720	0.0230	0.3108	0.0190	0.3533
8	0.5328	0.0277	0.1039	0.0376	0.1036	0.1943	**33**	0.3169	0.4706	0.1086	0.0413	0.0435	0.0191
9	0.2285	0.3040	0.0876	0.0513	0.0697	0.2589	**34**	0.3037	0.3075	0.3477	0.0107	0.0189	0.0116
10	0.2234	0.0128	0.0984	0.0849	0.0129	0.5677	**35**	0.8234	0.0228	0.0145	0.0632	0.0297	0.0466
11	0.0603	0.0137	0.0264	0.7078	0.0122	0.1796	**36**	0.0338	0.6090	0.2497	0.0163	0.0425	0.0487
12	0.0238	0.4155	0.0123	0.4731	0.0198	0.0555	**37**	0.5930	0.0259	0.3085	0.0139	0.0415	0.0172
13	0.0111	0.0115	0.8771	0.0442	0.0416	0.0144	**38**	0.1132	0.0258	0.3275	0.4509	0.0200	0.0627
14	0.1886	0.0858	0.3111	0.1160	0.1205	0.1780	**39**	0.3192	0.0625	0.0639	0.1777	0.0452	0.3316
15	0.0552	0.1005	0.5381	0.0893	0.1837	0.0333	**40**	0.4280	0.0220	0.0108	0.0182	0.5097	0.0114
16	0.0411	0.3905	0.0695	0.0399	0.4042	0.0549	**41**	0.0504	0.4848	0.0347	0.0441	0.1738	0.2123
17	0.0296	0.0150	0.2497	0.0955	0.1470	0.4632	**42**	0.0421	0.7898	0.0285	0.0769	0.0274	0.0353
18	0.0999	0.0239	0.0203	0.0129	0.3188	0.5242	**43**	0.0755	0.0289	0.4117	0.0113	0.4159	0.0567
19	0.0764	0.1698	0.0200	0.4907	0.2046	0.0385	**44**	0.0259	0.0668	0.0101	0.2578	0.2000	0.4393
20	0.0265	0.0109	0.0429	0.0141	0.8883	0.0173	**45**	0.0113	0.0126	0.0505	0.8986	0.0163	0.0107
21	0.1474	0.0228	0.0410	0.0198	0.4647	0.3043	**46**	0.0151	0.1765	0.1080	0.0128	0.0797	0.6079
22	0.0145	0.3840	0.4518	0.0233	0.1130	0.0134	**47**	0.3351	0.2606	0.0173	0.0368	0.3336	0.0166
23	0.1096	0.1022	0.1712	0.0618	0.4965	0.0586	**48**	0.0105	0.0393	0.0472	0.3017	0.5904	0.0108
24	0.0106	0.0140	0.0344	0.2409	0.0317	0.6684	**49**	0.2636	0.0451	0.6005	0.0291	0.0408	0.0210
25	0.1083	0.5403	0.0449	0.1846	0.0956	0.0264	**50**	0.5472	0.0399	0.0117	0.3080	0.0116	0.0816

Table A3
95% Confidence Intervals for the Performance of RA Against SA and IWS

α	Set	Average absolute deviation		Average proportional deviation		Number best not found		Average iteration	
		SA - RA	IWS - RA	SA - RA	IWS - RA	SA - RA	IWS - RA	SA - RA	IWS - RA
1	A70-4	(0.0000; 0.0067)	(0.0006; 0.0030)	(-0.0008; 0.0125)	(0.0013; 0.0061)	(0.0267; 0.2133)	(0.0852; 0.3148)	(-0.3400; 0.6200)	(-0.1490; 0.5490)
	E70-4	(0.0000; 0.0000)	(0.0013; 0.0073)	(0.0000; 0.0000)	(0.0018; 0.0160)	(0.0000; 0.0000)	(0.0548; 0.2652)	(-0.4770; 0.4770)	(-0.1610; 0.7210)
	L70-4	(0.0000; 0.0000)	(0.0009; 0.0067)	(0.0000; 0.0000)	(0.0025; 0.0209)	(0.0000; 0.0000)	(0.0548; 0.2652)	(-0.5420; 0.4220)	(-0.1480; 0.7880)
	E100-4	(-0.0001; 0.0004)	(0.0000; 0.0034)	(-0.0004; 0.0013)	(0.0006; 0.0062)	(-0.0202; 0.0602)	(0.0404; 0.2396)	(-0.6310; 0.2710)	(-0.5170; 0.3570)
	W50-6	(0.0000; 0.0001)	(0.0000; 0.0026)	(0.0000; 0.0001)	(0.0001; 0.0040)	(-0.0202; 0.0602)	(0.0267; 0.2133)	(-0.3610; 0.6810)	(0.2710; 1.1690)
2	A70-4	(-0.0001; 0.0023)	(0.0003; 0.0037)	(-0.0002; 0.0052)	(0.0010; 0.0135)	(-0.0168; 0.1768)	(0.0271; 0.2929)	(-0.5940; 0.4740)	(-0.5670; 0.3270)
	E70-4	(-0.0001; 0.0011)	(0.0021; 0.0079)	(-0.0001; 0.0035)	(0.0047; 0.0161)	(0.0021; 0.1579)	(0.2040; 0.4760)	(-1.1520; -0.0880)	(-1.0010; 0.0410)
	L70-4	(-0.0004; 0.0002)	(0.0029; 0.0093)	(-0.0013; 0.0008)	(0.0104; 0.0347)	(-0.0901; 0.0501)	(0.1861; 0.4539)	(-0.7860; 0.3060)	(-0.6170; 0.3370)
	E100-4	(0.0000; 0.0023)	(0.0042; 0.0108)	(0.0003; 0.0078)	(0.0118; 0.0344)	(0.0548; 0.2652)	(0.3366; 0.6234)	(-1.1690; -0.3110)	(-1.2820; -0.4380)
	W50-6	(-0.0003; 0.0017)	(0.0017; 0.0069)	(-0.0010; 0.0054)	(0.0041; 0.0196)	(-0.0082; 0.1282)	(0.1684; 0.4316)	(-0.6820; 0.4020)	(0.3650; 1.2750)

Table A3 (cont'd)

3	A70-4	(-0.0017; 0.0024)	(-0.0011; 0.0040)	(-0.0071; 0.0061)	(-0.0046; 0.0143)	(-0.0743; 0.1543)	(0.0115; 0.2685)	(-0.4570; 0.5370)	(-0.4340; 0.4740)
	E70-4	(0.0000; 0.0016)	(0.0021; 0.00866)	(-0.0003; 0.0067)	(0.0057; 0.0181)	(0.0021; 0.1579)	(0.2407; 0.5193)	(-1.0380; -0.0420)	(-0.9840; 0.0640)
	L70-4	(0.0000; 0.0001)	(0.0052; 0.0154)	(-0.0001; 0.0002)	(0.0173; 0.0562)	(-0.0202; 0.0602)	(0.2407; 0.5193)	(-0.7840; 0.1440)	(-0.6180; 0.2180)
	E100-4	(0.0002; 0.0037)	(0.0060; 0.0147)	(0.0008; 0.0112)	(0.0178; 0.0454)	(0.0697; 0.2903)	(0.3565; 0.6435)	(-1.1560; -0.1240)	(-1.4980; -0.4220)
	W50-6	(-0.0010; 0.00089)	(0.0022; 0.0082)	(-0.0026; 0.0032)	(0.0059; 0.0235)	(-0.1106; 0.0706)	(0.1924; 0.4876)	(-0.5940; 0.5940)	(0.3940; 1.4460)
4	A70-4	(-0.0033; 0.0022)	(-0.0021; 0.0046)	(-0.0111; 0.0054)	(-0.0067; 0.0164)	(-0.0606; 0.1806)	(0.0271; 0.2929)	(-0.4020; 0.7620)	(-0.4050; 0.6450)
	E70-4	(-0.0001; 0.0015)	(0.0028; 0.0101)	(-0.0007; 0.0071)	(0.0073; 0.0214)	(-0.0082; 0.1282)	(0.2975; 0.5825)	(-0.5870; 0.4270)	(-0.6200; 0.2600)
	L70-4	(-0.0002; 0.0013)	(0.0061; 0.0157)	(-0.0005; 0.0036)	(0.0191; 0.0580)	(0.0021; 0.1579)	(0.2783; 0.5617)	(-1.1370; 0.0570)	(-0.9560; 0.1160)
	E100-4	(0.0005; 0.0041)	(0.0079; 0.0179)	(0.0018; 0.0132)	(0.02376; 0.05592)	(0.0697; 0.2903)	(0.3969; 0.6831)	(-1.3980; -0.4020)	(-1.8350; -0.8050)
	W50-6	(-0.0008; 0.0010)	(0.00275; 0.00933)	(-0.0022; 0.0036)	(0.0073; 0.0278)	(-0.0706; 0.1106)	(0.1924; 0.4876)	(-0.4520; 0.6120)	(0.5950; 1.4850)
5	A70-4	(-0.0032; 0.00262)	(-0.0017; 0.0054)	(-0.0111; 0.0063)	(-0.0057; 0.0191)	(-0.0037; 0.2437)	(0.1046; 0.3754)	(-0.6790; 0.5190)	(-0.6330; 0.4330)
	E70-4	(-0.0001; 0.0019)	(0.0032; 0.0110)	(-0.0005; 0.0085)	(0.0085; 0.0236)	(0.0267; 0.2133)	(0.2975; 0.5825)	(-0.8050; 0.1650)	(-0.9270; 0.0870)
	L70-4	(-0.0002; 0.0015)	(0.0071; 0.0174)	(-0.0003; 0.0039)	(0.0219; 0.0627)	(0.0139; 0.1861)	(0.3565; 0.6435)	(-1.1540; -0.0460)	(-1.0410; -0.0390)

Table A3 (cont'd)

	E100-4	(-0.0002; 0.0028)	(0.0090; 0.0194)	(-0.0006; 0.0100)	(0.0269; 0.0615)	(0.0105; 0.2295)	(0.3655; 0.6745)	(-1.1200; -0.1600)	(-1.6130; -0.6270)
	W50-6	(-0.0008; 0.0037)	(0.0035; 0.0109)	(-0.0032; 0.0135)	(0.0094; 0.0340)	(-0.0460; 0.1660)	(0.2107; 0.5093)	(-0.9510; 0.1910)	(0.1160; 1.2040)
6	A70-4	(-0.0036; 0.0019)	(-0.0020; 0.0039)	(-0.0123; 0.0047)	(-0.0057; 0.0117)	(-0.0463; 0.2063)	(0.1046; 0.3754)	(-0.7510; 0.5110)	(-0.8100; 0.3300)
	E70-4	(0.0000; 0.0022)	(0.00356; 0.01174)	(-0.0003; 0.0096)	(0.0094; 0.0254)	(0.0404; 0.2396)	(0.2975; 0.5825)	(-0.8220; 0.1420)	(-0.9070; 0.0670)
	L70-4	(-0.0001; 0.0016)	(0.0079; 0.0189)	(-0.0002; 0.0042)	(0.0237; 0.0691)	(0.0267; 0.2133)	(0.3969; 0.6831)	(-1.0670; -0.0130)	(-0.9750; 0.0150)
	E100-4	(-0.0001; 0.0031)	(0.0100; 0.0214)	(-0.00057; 0.01140)	(0.0298; 0.0681)	(0.0105; 0.2295)	(0.3858; 0.6942)	(-1.2450; -0.3550)	(-1.8110; -0.7890)
	W50-6	(-0.0007; 0.0017)	(0.0037; 0.0111)	(-0.00191; 0.00449)	(0.0097; 0.0344)	(-0.0588; 0.1388)	(0.2293; 0.5307)	(-0.9130; 0.1930)	(0.1460; 1.2140)
8	A70-4	(-0.0022; 0.0031)	(0.0014; 0.0096)	(-0.00513; 0.00818)	(0.0057; 0.0310)	(-0.0463; 0.2063)	(0.1715; 0.5085)	(-0.3400; 0.7800)	(-0.4660; 0.4660)
	E70-4	(0.0002; 0.0044)	(0.0056; 0.0150)	(0.00133; 0.01592)	(0.0139; 0.0338)	(0.0271; 0.2929)	(0.2975; 0.5825)	(-0.7140; 0.1540)	(-0.7170; 0.1570)
	L70-4	(0.0006; 0.0040)	(0.0105; 0.0225)	(0.00186; 0.01151)	(0.0310; 0.0767)	(0.0697; 0.2903)	(0.3969; 0.6831)	(-0.9580; 0.0380)	(-0.8530; 0.0930)
	E100-4	(-0.0045; 0.0051)	(0.0094; 0.0260)	(-0.00293; 0.01829)	(0.0350; 0.0805)	(-0.0316; 0.2316)	(0.3658; 0.7142)	(-0.7710; 0.1710)	(-1.3160; -0.3240)
	W50-6	(-0.0004; 0.0019)	(0.0042; 0.0123)	(-0.00100; 0.00519)	(0.0111; 0.0375)	(-0.0035; 0.2035)	(0.2975; 0.5825)	(-0.8290; 0.1490)	(0.1910; 1.1290)

Özet

Sıralama tabanlı, filtrelemeli ve etkileşimli bir çok kriterli seçim algoritması: Üniversite seçiminde uygulamalar

Bu çalışmada, karar vericinin (KV) ayrık bir alternatifler kümesi içinde en çok tercih ettiği çözüme yakınsamak için etkileşimli bir yöntem geliştirilmiştir. Yöntem iterasyonlar boyunca KV'ye sınırlı sayıda alternatif sunup bunlar için sıralamalı tercih bilgisi toplamaktadır. KV tercihleri esnek ve gerçekçi bir tercih fonksiyonu ile modellenmiştir. Yöntemin performansını iyileştirmek için sunulan alternatifler bir filtreleme prosedürü ile belirlenmiştir. Önerilen yöntem, üniversitelerin farklı kategorilerde yıllık sıralamalarını yayınlayan pazarlama şirketi Quacquarelli Symonds'tan çeşitli veri setleri kullanılarak karşılaştırma algoritmaları ile kıyaslanmıştır. Sonuçlar önerilen algoritmanın karşılaştırma algoritmalarından üstün olduğunu göstermektedir.

Anahtar kelimeler: Çok kriterli karar verme, sıralama, etkileşimli yöntem, filtreleme