Helsinki University Of Technology, Systems Analysis Laboratory
Mat-2.108 Independent research projects in applied mathematics (3 cr)

Rank ordering attribute weighting methods
– simulation study

Antti Laukkanen
50116 R
ajlaukka@cc.hut.fi
# Table of Contents

1. Introduction ........................................................................................................... 3
2. Multiattribute decision-making ............................................................................. 4  
   2.1 Multiattribute decision-making problems ...................................................... 4  
   2.2 Lower bounding of weights ............................................................................ 6  
3. Rank ordering weight approximation methods .................................................. 6  
4. Decision rules ....................................................................................................... 8  
   4.1 Central Values Decision rule ........................................................................... 9  
5. Simulation ........................................................................................................... 10  
   5.1 Evaluation measures ....................................................................................... 11  
   5.2 Distribution of scores ..................................................................................... 11  
   5.3 Simulation steps ............................................................................................. 12  
6. Results ................................................................................................................ 15  
7. Conclusions ......................................................................................................... 16  
8. References .......................................................................................................... 17
1 Introduction

In a multiattribute decision-making problem, i.e. problem with several alternatives and attributes, the decision maker (DM) is often assumed to be able to provide information of rank ordering of attributes instead of specific, or “true”, weights of importance of the attributes, as is done in [1]. Such assumption, complete rank ordering being provided by DM, is also made in this study. The “true” weights of attributes remain unknown in practice, because accuracy that is needed for elicitation of exact weights may be actually absent in the mind of the DM [1]. Even if the elicitation of precise weights was possible, it would probably be time-consuming and difficult and therefore impractical. Rank ordering weighting methods provide approximations of “true” weights of attributes when rank ordering information is known, for best outcome of the decision-making process.

Rank ordering weighting methods take into account DM’s information about rank ordering of attribute weights, from which approximations for attribute weights are calculated by using corresponding formula. Rank sum [2], rank order centroid [3] and rank reciprocal [3] are the rank ordering weighting methods benchmarked in this study.

In addition to the rank ordering methods, usage of a decision rule is benchmarked in this study. Decision rules are heuristics that recommend one alternative of the alternatives available. In a multiattribute decision-making problem the available alternatives are evaluated by decision rules, which are used by taking into account the rank ordering information. Usage of Central Values [4] decision-rule is benchmarked in this study.

Simulation of 36 different runs, 5000 rounds each, was done in this study. Different decision-making problems were simulated. Amount of alternatives and attributes in the problem was varied. Triangular- and uniform distributions for alternatives’ scores were applied and effect of lower bounded weights for attributes studied.

This paper is arranged as follows: Multiattribute decision-making problem is discussed and illustrated via example in Section 2. Used weighting methods and the decision rule are introduced in Section 3. Section 4 describes the simulation procedure and Section 5 the results of the simulation. Conclusions are discussed in Section 6.
2 Multiattribute decision-making

2.1 Multiattribute decision-making problems

Many implementations for multiattribute decision-making problems exist in practice. There are many challenges considering the solving of such problems in practice. To make a definitely right decision, attributes’ weights should be exactly known and scores for alternatives evaluated. As discussed in the previous section, defining of exact weights for attributes is not easy for DM, or perhaps even possible, in practice.

Following notations are used throughout this study:

\[ n = \text{number of attributes} \]
\[ m = \text{number of alternatives} \]
\[ W_i = \text{“true” weight of } i:th \text{ attribute} \]

In this study, only possible weights are assumed to exist for alternative. “True” weights for attributes:

\[ W_i \geq 0, \ i = 1, \ldots, n, \]

where \( W_i \) stands for the “real” weight for \( i \)th attribute. In this study, all weights are assumed to be normalized, i.e. their sum equals one:

\[ \sum_{i=1}^{m} W_i = 1. \quad (1) \]

Model for alternatives’ evaluation is assumed to be additive multiattribute value (MAV) of form, [1]:

\[ v(j) = \sum_{i=1}^{n} W_i v_{ij}, \quad j = 1, \ldots, m \]  \quad (2)

where \( v(j) \) is the overall multiattribute value of alternative \( j \), whereas total number of alternatives is \( m \). \( v_{ij} \) stands for the value, \textit{score}, associated to \( j \)th attribute of \( i \)th alternative. Complete rank ordering, that DM is in this study assumed to able to provide, defines the rank ordering of the attribute weights,

\[ r = (r_1, r_2, \ldots, r_n), \]

(3)
where \( r_i \) is the rank ordering of the \( i \)th attribute, \( r=1 \) represents the most important attribute and \( r=n \) the least important. For instance, in a problem of three attributes, for a rank ordering \( r = (r_1=1, r_2=3, r_3=2) \) or in more compact form, \( r = (1,3,2) \), the real weights must fulfill \( W_1 \geq W_3 \geq W_2 \). In addition, no equally important ranks are assumed to exist, and therefore rank ordering is always defined. More generally,

\[
  r_i < r_j \rightarrow W_i \geq W_j, \quad \forall \ i, j \in \{1,2,\ldots,n\} \land i \neq j. \quad (4)
\]

A multiattribute decision-making problem is illustrated for instance by taking a look at a family buying a new car. There are plenty of alternatives (different car brands, different models), and various attributes (price, comfort, looks, speed, economy, space, brand attractiveness, etc.) involved. Due to limited testing effort of the family, number of alternatives is reduced to seven by defining which are among the best seven alternatives. Similarly number of attributes to be taken into account is limited by naming the ones belonging to top five. Thus, the family has ended up to a decision problem with seven alternatives and five attributes. It is easily realized that precise weights for the five attributes of a car are certainly hard for the family to define.

With rank ordering (3) of the attributes known, rank ordering weighting methods can be used to achieve approximations \( w_i \) for assumed “real” attribute weights behind the rank ordering in the minds of the DMs:

\[
  w_i \sim W_i. \quad (5)
\]

Naturally, also approximations must fulfill the rank ordering:

\[
  r_i < r_j \land w_i \geq w_j, \quad i, j \in \{1,2,\ldots,n\} \land i \neq j. \quad (6)
\]

In addition, scores of alternatives must be evaluated, which may not be an easy task in practice either. Verbal evaluation may give considerable aid for evaluation of scores for attributes. After evaluating the scores, family can calculate values (2) for their attributes, using the calculated weight approximations and scores of alternatives, and define the best alternative.

Another approach would be usage of decision rules [4], i.e. heuristics to choose one alternative of the available ones. The decision-rule approach is quite different compared to the weight approximation. No attribute weights are approximated, but instead possible rank orderings for attribute weights are
defined from provided rank ordering information. In the case with complete rank ordering information provided, approach of decision rules is straightforward. Scores for alternatives must be evaluated. Based on the possibilities for weights provided by rank ordering information, possible score ranges for alternatives can be calculated, and therefore decision-rule applied to the problem. Thus, the use of decision rules accepts the uncertainty of attribute weights all the way and assumes nothing about the exact weights.

### 2.2 Lower bounding of weights

It may be realistic to assume that all the weights are at some reasonable level, so that there is no extremely small value for any attribute. This assumption is done by lower bounding the weights relatively to the number of attributes n, [6]:

\[ W_i \geq \frac{1}{3n}, \quad i = 1, \ldots, n. \]  

(7)

### 3 Rank ordering weight approximation methods

As it was discussed in Sections 1 and 2, values for weights can be approximated via rank ordering weight approximation methods. The idea of the methods is following; as the complete rank ordering information of attribute weights is known, approximations for weights are calculated by using appropriate formula, and by using the approximated weights the values for alternatives are calculated and the alternative with the highest value chosen. Used formulas are in force only when sum of the weights equals one (1). Following three formulas for approximated weights, \( w_i \), are used in this study:

*Rank-order centroid* weights, [2]:

\[ w_i^* (ROC) = \frac{1}{n} \sum_{j=1}^{n} \frac{1}{j}, \quad i = 1, \ldots, n, \]  

(8)

where \( w_i^* \) is the approximated weight for ith important attribute. Approximations for attribute weights must correspond to the rank ordering. Approximated weights \( w_i \) are achieved following way:

\[ w_i = w_j^*, \quad r_i = j, \quad i = 1, \ldots, n. \]  

(9)
Similarly, weights corresponding to the ranks, *rank sum* [3]:

\[
 w_i^r (RS) = \frac{n + 1 - i}{\sum_{j=1}^{n} j} = \frac{2(n + 1 - i)}{n(n-1)}, \quad i = 1, ..., n
\]  

(10)

and an approach based on the reciprocal of the ranks, *rank reciprocal* [3]:

\[
 w_i^r (RR) = \frac{i}{\sum_{j=1}^{n} \frac{1}{j}}, \quad i = 1, ..., n
\]  

(11)

Weights of RS are reduced linearly lower from the most important to the least important. RREC weights descend aggressively after the most important, but in the least important end the ROC weights are the lowest ones. Differences of the weighting methods are illustrated in the following example of five attributes (Figure 1).

![Figure 1](image_url)

**Figure 1** Approximations for attribute weights given by used formulas, in case of five attributes
4 Decision rules

The use of decision-rules in a multiattribute decision-making problem is based on defining of a feasible region for attribute weights, and therefore for alternatives’ scores as well. Use of decision rules under incomplete information is discussed in [4], and their topology and related definitions in more detail in [6].

Extreme points are the boundary points for feasible region, which fulfils the rank ordering. When complete rank ordering of attribute weights is known, there are n extreme points and their defining is quite straightforward. If the weights are not lower bounded, for kth extreme point:

\[
\begin{align*}
  w_i &= \frac{1}{i}, \quad r(i) \leq k \\
  w_i &= 0, \quad r(i) > k, \quad i, k = 1, ..., n
\end{align*}
\]  

(12)

where \( w_i \) is the value of attribute \( i \) weight in the extreme point.

For instance if there are three attributes and their rank ordering is \( r = (1,2,3) \), corresponding extreme points are \((1,0,0)\), \((0.5, 0.5, 0)\) and \((0.33, 0.33, 0.33)\). Feasible region for weights lies between the extreme points, which can be seen from illustration of the example’s extreme points and resulting feasible region in Figure 2 below:

![figure 2 illustration of extreme points in case of three attributes](image-url)
If the weights are lower bounded, the point calculation changes somewhat as every attribute must have at least the weight of the lower bound. Therefore (11) must be modified, as there is lesser amount of “free” weight as n-1 times the lower bound is reserved for the other weights. Extreme value calculation in case of lower bounded weights becomes therefore:

\[ w_i = \frac{1 - nb}{r_i}, \quad r(i) \leq k \]

\[ w_i = 1, \quad r(i) > k, \quad i, k = 1, \ldots, n \]  

(13)

where b is the lower bound of the weight, 1 / 3n in this study (7). For instance, extreme points for similar three-attribute problem, in case of lower-bounded weights are (0.77, 0.11, 0.11), (0.44, 0.44, 0.11) and (0.33, 0.33, 0.33), Figure 3.

![Figure 3 Illustration of extreme points in case of lower bounded weights and three attributes](image_url)
4.1 Central Values Decision rule

Central values [4] decision rule uses following heuristic: Choose the alternative $x$ for which the mid-point of the feasible value interval is the greatest, i.e.

$$\left[\overline{v}(x) + \underline{v}(x)\right] \geq \left[\overline{v}(x') + \underline{v}(x')\right] \forall x, x' \in X,$$

(14)

where $\overline{v}(x)$ is the best value of the alternative $x$ in feasible region and $\underline{v}(x)$ is the worst.

Considering the calculation, worst and best values are found in extreme points, as the valuation function (2) is linear. In a case with complete rank ordering of attribute weights and scores of alternatives known, as it is the case in this study, it can be said that usage of decision rules is quite straightforward. First the extreme points are identified and therefore feasible values for alternatives calculated. By applying this information to the chosen decision rule, minimum and maximum values in case of central values (14), best alternative is chosen.

5 Simulation

As the scope of this study is to compare the three weighting methods (8),(10),(11), and also benchmark the central values decision rule (14), exact scores for alternatives’ attributes are assumed to be known as well as complete attribute rank ordering information. The “true” weights of attributes are assumed to exist in simulation, so that “real” correct choices can be defined and therefore success of the used methods evaluated and benchmarked. Simulation is done by simulating 5000 cases with randomized attribute weights and scores for alternatives’ attributes.

Simulations were run by varying number of alternatives (5, 7 or 10) and attributes (5, 7 or 10). Lower bounding of weights (7) was applied to half of the runs. Uniform and triangular distributions for scores were used. All combinations were run, and $3 \times 3 \times 2 \times 2 = 36$ simulations were therefore done. Simulation was carried out using Matlab (http://www.mathworks.com) in a UNIX environment. The simulations took approximately an hour.
5.1 Evaluation measures

Success of the different methods is evaluated by two measures, expected loss of value and percentage of correct choices. The latter one is calculated simply by counting the number of correctly chosen cases and dividing it by total number of cases. Expected loss of value (ELV) is the average loss in overall value when the alternative that has been chosen is compared to the correct choice [5]. ELV is calculated as follows:

\[ ELV = \sum_{i=1}^{n} w_i \left( v_i(x_i^*) - v_i(x_i) \right), \]  

(15)

where \( w_i \) is the real weight of attribute \( i \), \( x_i^* \) is the correct choice, i.e. best alternative, and \( x_i \) is the alternative chosen by using decision rule.

5.2 Distribution of scores

Scores are generated either from an uniform distribution or a triangular distribution [5], to have values over [0,1]. Uniformly distributed scores are given simply by in simulation generated random number \( u \) over [0,1], i.e. \( s_{ij} = u \) (2). Therefore it is assumemd that alternatives do not have any common level of importance as levels of attributes can vary very much. Triangular distribution, in contrast, reflects a situation where the alternatives score quite similar values as the density of the distribution is at highest level around value of 0.5, which is also distribution’s mean. The density of triangular distribution is in other hand diminishingly small in its extremes, 0 and 1.

Tri-distributed scores \( s \) are achieved from following function of \( u \), [5]:

\[ s = \frac{\sqrt{u}}{2}, \text{ when } u \leq 0.5 \text{ and } \]

\[ s = 1 - \frac{\sqrt{(1-u)}}{2}, \text{ when } u > 0.5. \]  

(15)
5.3 Distribution of weights

The weights are normalized, so that their sum equals one (1). The weights of the attributes are assumed to be randomly distributed from flat-distribution, and have values over [0,1] in not lower bounded cases. Lower bounding (7) of weights is applied to the half of the simulation runs. Lower bounding of weights guarantees that the weights have no extreme deviation as all are above certain level, and therefore reflect the real situation better. The weights in lower bounded cases have values over [b,1-nb].

5.4 Simulation steps

The simulation principals are illustrated in the figure 4 below:
The simulation steps in more detail:

0. Define simulation parameters:

   number of alternatives, \(k\) (5/7/10)
   number of attributes, \(n\) (5/7/10)
   lowerBounded (true/false)
   triDistributed (true/false)
   rounds (5000)

Then, repeat \(rounds\) times steps 1-3:

1. Simulate decision problem

   1.1 Randomize values for attributes of alternatives
   If \(\text{triDistributed}\) case then randomize from tri-distribution, otherwise randomize from uniform-distribution. Result is \((k,n)\)-matrix.

   1.2 Randomize real weights for attributes
   If \(\text{lowerBounded}\) case then limit attributes' lower bounds

   1.3 Normalize scores for each attribute

   Each score must have value from \([0,1]\)

   1.4 Calculate true values for alternatives, and choose the correct choice, CC

2. Choose best alternatives given by the methods

   Rank ordering attribute weighting methods:

   2.1 Calculate weights for RROC, RSUM, RREC

   2.2 Using approximated attribute weights, calculate values for every alternative and choose the best, RROC\_best, RSUM\_best, RREC\_best.
**Decision rule:**

2.3 Calculate extreme points for central values-method

2.4 For each alternative calculate central values using extreme points, and choose the best, \( CV_{best} \)

3. Calculate and write down results

3.1. For each method compare the best chosen alternative and the correct choice CC. If the alternative chosen is different than the the CC, write down incorrect, otherwise correct choice.

3.2. For each incorrect choice made, calculate loss of value, ie. difference of values of real and heuristics best alternatives.

4. After all the rounds have passed, calculate

- Expected loss of value, ELV (15)
- Percentage of correct choices
### 6 Results

<table>
<thead>
<tr>
<th>Simulation parameters</th>
<th>percentage of correct choices</th>
<th>expected loss of value</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of alternatives</td>
<td>rroc</td>
<td>rsum</td>
</tr>
<tr>
<td>7</td>
<td>0.829</td>
<td>0.804</td>
</tr>
<tr>
<td>5</td>
<td>0.832</td>
<td>0.794</td>
</tr>
<tr>
<td>yes</td>
<td>0.822</td>
<td>0.796</td>
</tr>
<tr>
<td>5</td>
<td>0.825</td>
<td>0.790</td>
</tr>
<tr>
<td>5</td>
<td>0.816</td>
<td>0.801</td>
</tr>
<tr>
<td>no</td>
<td>0.837</td>
<td>0.776</td>
</tr>
<tr>
<td>flat</td>
<td>0.826</td>
<td>0.807</td>
</tr>
<tr>
<td>5</td>
<td>0.837</td>
<td>0.776</td>
</tr>
<tr>
<td>7</td>
<td>0.836</td>
<td>0.792</td>
</tr>
<tr>
<td>no</td>
<td>0.841</td>
<td>0.763</td>
</tr>
<tr>
<td>flat</td>
<td>0.838</td>
<td>0.803</td>
</tr>
<tr>
<td>7</td>
<td>0.849</td>
<td>0.759</td>
</tr>
<tr>
<td>5</td>
<td>0.799</td>
<td>0.771</td>
</tr>
<tr>
<td>5</td>
<td>0.809</td>
<td>0.756</td>
</tr>
<tr>
<td>yes</td>
<td>0.796</td>
<td>0.777</td>
</tr>
<tr>
<td>10</td>
<td>0.811</td>
<td>0.783</td>
</tr>
<tr>
<td>no</td>
<td>0.800</td>
<td>0.773</td>
</tr>
<tr>
<td>flat</td>
<td>0.816</td>
<td>0.752</td>
</tr>
<tr>
<td>7</td>
<td>0.811</td>
<td>0.775</td>
</tr>
<tr>
<td>yes</td>
<td>0.818</td>
<td>0.752</td>
</tr>
<tr>
<td>10</td>
<td>0.803</td>
<td>0.766</td>
</tr>
<tr>
<td>10</td>
<td>0.830</td>
<td>0.740</td>
</tr>
<tr>
<td>no</td>
<td>0.810</td>
<td>0.776</td>
</tr>
<tr>
<td>flat</td>
<td>0.827</td>
<td>0.753</td>
</tr>
<tr>
<td>10</td>
<td>0.840</td>
<td>0.819</td>
</tr>
<tr>
<td>yes</td>
<td>0.860</td>
<td>0.820</td>
</tr>
<tr>
<td>5</td>
<td>0.840</td>
<td>0.818</td>
</tr>
<tr>
<td>no</td>
<td>0.857</td>
<td>0.813</td>
</tr>
<tr>
<td>5</td>
<td>0.847</td>
<td>0.822</td>
</tr>
<tr>
<td>yes</td>
<td>0.867</td>
<td>0.811</td>
</tr>
<tr>
<td>5</td>
<td>0.855</td>
<td>0.823</td>
</tr>
<tr>
<td>no</td>
<td>0.859</td>
<td>0.817</td>
</tr>
<tr>
<td>5</td>
<td>0.852</td>
<td>0.812</td>
</tr>
<tr>
<td>yes</td>
<td>0.863</td>
<td>0.801</td>
</tr>
<tr>
<td>10</td>
<td>0.856</td>
<td>0.824</td>
</tr>
<tr>
<td>5</td>
<td>0.873</td>
<td>0.803</td>
</tr>
</tbody>
</table>

The results are quite similar in all the cases simulated, however some differences and interesting issues were found. Common for all the runs is the successfullness of the methods. The RROC is the most successful method in all cases. Decision rule central values gives most wrong decisions. RREC is often better than RSUM, but not in all cases. The order is same in terms of ELV, except when k=7, n=5 and k=n=5, as the rank sum has the worst and central values the second worst ELV.
An interesting issue is the effect of lower-bounding of weights. The RSUM and CV succeed better when the weights are lower bounded. In contrast, RROC and RREC succeed worse when the weights are lower bounded. As it can be seen from the Figure 1, weights of RS are more uniformly distributed than RREC and RROC. The lower bounding of the weights reduces their mutual differences, and therefore RS fits closer to the real situation. In addition, lower bounding of weights seems to improve the success of central values, especially when there are many attributes.

The distribution of scores does not seem to have a clear impact to the percentage of correct choices. However, expected loss of value seems to get somewhat bigger for all the methods when the scores come from uniform-distribution. This is a result of bigger deviation of scores and therefore also differences of alternatives’ values increase.

7 Conclusions

Usage of three rank ordering attribute weighting methods, rank order centroid (8), rank sum (10), rank reciprocal (11), and central values decision rule (14), were discussed and simulated in this study. Simulation of total 36 different runs was made, 5000 rounds each. Different decision-making problems were simulated. Amount of alternatives and attributes was varied, as well as distribution of scores of alternatives and weights of attributes.

The simulation results were quite similar in all the cases, whereas some interesting issues were found. Rank order centroid was the most successful method in terms of both used criteria, percentage of correct choices and expected loss of value. The central values decision rule was the worst method. Rank reciprocal was mostly better than rank sum. Lower bounding of weights has an improving effect to the success of rank sum method and central values, whereas rank order centroid and rank reciprocal score worse when weights are lower bounded. Expected loss of value increases when uniform distribution of weights is used instead of triangular distribution.
8 References


