Solving Multi Criteria Decision Aiding (MCDA) Problems Using Spreadsheets

T. Ganesh*, PRS Reddy

Department of Statistics, S.V. University, Tirupati, India

Abstract In Managerial Decision making, the problem environment will be encircled by a set of alternatives for set of criteria. The main objective is to choose the best alternative under each criterion. In this contest, the Decision Maker (DM) plays an important role in solving the hard/complex problems. This type of scenario gives raise to the concept of MCDA. In this paper, we made an attempt to provide some algorithms which are user-friendly. In this paper, we have provided some algorithms which supports in computing the concordance and discordance indices.

Keywords Multi Criteria, Concordance, Discordance, Outranking Index

1. Introduction

In any environment, the main objective is to provide a set of best alternatives for given criteria. The decision maker provides some necessary and basic information about each criterion and the alternatives that helps in identifying the relation between them. The problems of this kind can be dealt with Multi Criteria Decision Making or Multi Criteria Decision Aid (MCDA) techniques.

The main aim of MCDA is to account for several views and provide some tools for the Decision Maker (DM) in solving complex decision problems. The trade-off between the criteria and DM’s preferences lies in providing compromise solutions. In each and every problem or situation, the DM, Stakeholder and Analyst play an important role.

DM is a person, who has a great impact in evaluating the situation, expressing preferences, considering solutions and approving the final result. Stakeholders are members involved in decision situation and interested in finding a solution for the problem. For the situation considered, the Analyst is responsible in recognizing the consequences and selecting an appropriate decision aiding method/tool for the construction of decision models.

In every MCDA problem environment, each criterion will be embedded with a set of alternatives out of which one alternative will act as the best for that particular criterion. These set of alternatives will be finite if a proper definition about all the members is given, otherwise infinite. If the number and content of alternatives are fixed and cannot be varied during the decision aiding process, then this nature is said to be stable otherwise volatile. At the final stage of the decision aiding process, if we come across a single best alternative which excludes the possibility of choosing any other alternative, it is referred as Comprehensive and if we opt for a combination of alternatives, it is fragmented. In brief, the alternatives are estimated on a set of criteria. The criterion defines the feature and some properties of the set of alternatives.

Notations

- \( x_i \): \( i^{th} \) alternative (\( i=1,\ldots,m \))
- \( X \): Set of alternatives
- \( g_j \): \( j^{th} \) criterion (\( j=1,\ldots,n \))
- \( G \): set of criteria
- \( Q_j \): \( j^{th} \) Indifference thresholds
- \( P_j \): \( j^{th} \) Preference thresholds
- \( W_j \): \( j^{th} \) Weights
- \( V_j \): \( j^{th} \) Veto thresholds
- \( \lambda \): Cutting level
- \( b_q \): \( q^{th} \) boundary alternative (\( q=1,\ldots,s \))
- \( B \): set of boundary alternatives (\( b_1, b_2, \ldots, b_q \))
- \( l_q \): \( q^{th} \) boundary class
- \( C_j (x_i, b_q) \) and \( D_j (x_i, b_q) \): partial concordance and partial discordance of the \( x_i \) and \( b_q \)
- \( C_j (b_q, x_i) \) and \( D_j (b_q, x_i) \): partial concordance and partial discordance of the \( b_q \) and \( x_i \)
- \( C (x_i, b_q) \) and \( C (b_q, x_i) \): overall concordance indices
- \( S_j (x_i, b_q) \): outranking index for \( x_i \) and \( b_q \)
- \( S_j (b_q, x_i) \): outranking index for \( b_q \) and \( x_i \)
- \( C_q \): \( q^{th} \) category
- \( P \): strict preference
- \( Q \): weak preference
- \( I \): indifference
- \( J \): incomparability

The entire MCDA problem will be expressed in terms of
relations existing between the alternatives and criteria. We
brief out each and every relation and the nomenclature for it.

1.1. Relations
- The indifference relation between two alternatives \(x_i\) and
  \(x_j\), denoted as \(x_i|x_j\), means that the two alternatives \(x_i\) and \(x_j\)
  are equally preferable or equally important to the DM. This
  relation is reflexive and symmetric.
- The strict preference is a relation of \(x_i\) over \(x_j\), denoted as
  \(x_i|P|x_j\), which gives the meaning that \(x_i\) is better than the \(x_j\)
  for the DM. It is asymmetric and non-reflexive.
- The weak preference is a relation which hesitates to
  make a specific judgment about the preference or
  indifference between \(x_i\) and \(x_j\), denoted by \(x_i|Q|x_j\). It is also
  asymmetric and non-reflexive.
- If \(x_i\) is not in any of the above mentioned relations with \(x_j\),
  then it is referred to as incomparability relation, denoted by
  \(x_i|I|x_j\). This relation is symmetric and non-reflexive.
- The outranking relation is denoted as \(x_i|x_j\). It defines the
  situation in which the preference (strong- \(x_i|P|x_j\) or weak-
  \(x_i|Q|x_j\)) or indifference relation (\(x_i|x_j\)) is true or not.

In order to observe a specific type of relation between
alternative and criterion, there is a need to compute some
indices such as partial concordance, discordance and
outranking indices. over the years, many methodologies
were developed of which the most familiar method is the
Outranking Methodology. In outranking methodology, we
have considered ELECTRE TRI method and for this we have
developed spreadsheet algorithms, which support the analyst
to analyze and to provide a better decision making. First we
review some literature confining to ELECTRE TRI method
and then a detailed algorithmic approach is given along with
the results.

2. Outranking Methodology

In MCDA, the outranking methodology comes under the
framework of classification problems. Basing on the same
criterion, the methodology allows comparing the pairs of
alternatives by considering indifference, preference and veto
thresholds. This helps in determining the indifference,
preference to one over the other and incomparable relation
between alternatives. The seminal work on this methodology
was proposed by B. Roy (1965). He developed some
mathematical structures about the ELECTRE family which
help in choosing the best alternative from the set of
alternatives. In recent years, many state of art surveys were
conducted and reported on the development of the MCDA
methodologies by M. Bruen and L. Maystre (2000), B. Roy

B. Roy (1977, 1981) proposed the Trichotomic
segmentation outranking based classification method for
sorting problems with three classes. Later, this method was
extended to an arbitrary number of classes in N-TOMIC by R.
Massaglia et al. (1991) and few ELECTRE methods by V.

2.1. ELECTRE TRI Method

ELECTRE method helps to identify the outranking
relations between pairs of alternatives for each criterion. In
classification problems, a given set of alternatives \(X\) with a
set of criteria \(G\) are to be assigned into a set of ordered
classes \(L\) by the predefined set of boundary alternatives \(B\).
Each class is considered by two (upper and lower) boundary
alternatives. The upper bound \(b_u\) of the class \(l_q\) is the lower
bound of the class \(l_q\) \((q=1,...,s)\). Changing the least one
criterion moves the boundary alternative to the neighbouring
class.

For solving the classification problem the method
estimates the outranking relation for each alternative \(x_i\in X\)
\((i=1,...,m)\) which is to be classified and each boundary
alternative \(b_q\) between classes \(l_{q-1}\) and \(l_q\) by calculating the
outranking index \(I\). If \(l_q\) is preferred to the lower boundary
alternative \(l_{q-1}\) of the class, we assign the alternative \(x_i\) to the
class \(l_q\) and the upper boundary alternative \(b_q\) of the class is
preferred to this alternative.

For calculating the outranking index, the DM should give
the information about
(i) the set of alternatives to be classified
(ii) the set of criteria on which alternatives are evaluated
with a scale of quantitative values for each criterion.
(iii) the number of classes as well as their order according
to preference.
(iv) the upper and lower boundary alternatives for each
class \(l_q\).

For each criterion \(g_j\) \((j=1,...,n)\), the ELECTRE TRI
method requires to define the preference \(p_j(\cdot)\), indifference
\(q_j(\cdot)\), veto \(v_j(\cdot)\) thresholds as well as weights \(w_j\) and cutting
level \(\lambda\) (should lie between 0.5 and 1).

(a) the preference \(p_j(\cdot)\) threshold indicates the smallest
difference between two alternatives on the criterion \(g_j\), that is
one alternative is preferred to the other.
(b) the indifference \(q_j(\cdot)\) threshold indicates the largest
difference between two alternatives on the criterion \(g_j\).
(c) the veto \(v_j(\cdot)\) threshold indicates the smallest difference
between the alternatives on the criterion \(g_j\), that says
incomparability of these two alternatives.
(d) All the above three thresholds should satisfy the
constraint, \(v_j(\cdot) > p_j(\cdot) > q_j(\cdot)\)
(e) the weight \(w_j\) indicates the relative importance of
criterion when compare to the other criterion in terms of
votes.
(f) the cutting level \(\lambda\) shows the smallest value of the
outranking index, which is sufficient for considering an
outranking situation between two alternatives.

The outranking relation is verified by two conditions;
concordance and discordance, with respect to the thresholds,
weights and cutting level \(\lambda\). Concordance requires preference
of the alternative \(x_i\) over the boundary alternative \(b_q\) on the
majority of criteria. Discordance demands the absence of
strong opposition to the first condition in the majority of
criteria. We need to compute two partial indices for each criterion, that is partial concordance $C_j(x_i, b_q)$ and partial discordance $D_j(x_i, b_q)$ and partial concordance $D_j(b_q, x_i)$ for each boundary alternative $b_q$. The above partial indices help in computing the outranking indices $S_j(x_i, b_q)$ and $S_j(b_q, x_i)$. Using a specific cutting level $\lambda$, a comparison of outranking indices is possible and turns to two types of assignment procedures namely pessimistic and optimistic.

The pessimistic procedure starts with the comparison of an alternative to the lower bound of the highest class and the optimistic procedure starts with the comparison of an alternative to the upper bound to the lowest class. In section 3, we describe the mathematical structures of outranking indices and assignment procedures.

### 3. Algorithm of the ELECTRE TRI Method

The ELECTRE TRI method has been divided into two parts; part I is to compute the outranking indices and to identify the relations between the alternatives and criteria and in part II, using the obtained outranking relation and cutting level $\lambda$, we provide the final result for the MCDA problem.

**Part I:** To construct the outranking relation $x_i S b_q$ for each alternative $x_i$ to be classified and each boundary alternative $b_q$.

1. Calculate the partial concordance indices $C_j(x_i, b_q)$ and $C_j(b_q, x_i)$ for each criteria $g_j$ according to the increasing direction of preferences. The partial concordance index $C_j(x_i, b_q)$ is as follows:

$$C_j(x_i, b_q) = \begin{cases} 
0, & \text{if } g_j(b_q) - g_j(x_i) \geq p_j(b_q) \\
1, & \text{if } g_j(b_q) - g_j(x_i) < q_j(b_q) \\
\frac{p_j(b_q) - g_j(b_q) + g_j(x_i)}{p_j(b_q) - q_j(b_q)}, & \text{if } g_j(b_q) - p_j(b_q) < g_j(x_i) \leq g_j(b_q) - q_j(b_q) 
\end{cases}$$

The partial concordance index $C_j(b_q, x_i)$ is as follows:

$$C_j(b_q, x_i) = \begin{cases} 
0, & \text{if } g_j(x_i) - g_j(b_q) \geq p_j(x_i) \\
1, & \text{if } g_j(x_i) - g_j(b_q) < q_j(x_i) \\
\frac{p_j(x_i) - g_j(x_i) + g_j(b_q)}{p_j(x_i) - q_j(x_i)}, & \text{if } g_j(x_i) - p_j(x_i) < g_j(b_q) \leq g_j(x_i) - q_j(x_i) 
\end{cases}$$

2. To find the overall concordance indices $C(x_i, b_q)$ and $C(b_q, x_i)$ as an aggregation of partial concordance indices.

$$C(x_i, b_q) = \frac{\sum_{j=1}^{n} W_j C_j(x_i, b_q)}{\sum_{j=1}^{n} W_j}$$

$$C(b_q, x_i) = \frac{\sum_{j=1}^{n} W_j C_j(b_q, x_i)}{\sum_{j=1}^{n} W_j}$$

3. Calculate partial discordance indices $D_j(x_i, b_q)$ and $D_j(b_q, x_i)$ for each criteria $g_j$. We compute the partial discordance index $D_j(x_i, b_q)$ according to the increasing direction of preference.
3.1. The Pessimistic Procedure

In this procedure the comparison will start from alternative \( x_i \) to the lower bound \( b_{q,l} \) of the highest class \( l_q \) \((q = s, s-1, \ldots, 1)\) and continues in decreasing order until, a lower bound \( b_{q,l} \) is found, that is \( x_i \in C_{b_{q,l}} \), and for estimating the outranking relation we calculate \( S(x_i, b_{q,l}) \). Once the outranking relation is obtained, we calculate outranking index between \( x_i \) and \( b_{q,l} \). We assign the alternative \( x_i \) to the \( l_q \) if \( S(x_i, b_{q,l}) \geq \lambda \) and \( S(x_i, b_{q,l}) < \lambda \).

1. Compare \( x_i \) successively to \( b_{q,l} \) for \( q = s, s-1, \ldots, 0 \)
2. \( b_{q,l} \) being the first bound such that \( x_i \in C_{b_{q,l}} \), assign \( x_i \) to category \( C_{q+1} \) \((x_i \not\in C_{q+1})\) otherwise.

In other words, the above procedure can also be expressed as follows; \( b_{q,l} \) and \( b_{q,l-1} \) are the upper and lower bound of category \( C_{q} \), the pessimistic procedure assigns alternative \( x_i \) to the highest category \( C_{q} \) such that \( x_i \in C_{b_{q,l-1}} \). When using this procedure with \( \lambda = 1 \), an alternative \( x_i \) can be assign to category \( C_q \) only if \( g_j(x_i) \) equals or exceeds \( g_j(b_{q,l-1}) \) for each criterion. When \( \lambda \) decreases the pessimistic characters of this rule is weakened.

3.2. The Optimistic Procedure

Here, we begin to compare the alternative \( x_i \) to the upper bound \( b_{q,u} \) of the lowest class \( l_q \) \((q = s, s-1, \ldots, 1)\) and proceed in increasing order until we find such a upper bound \( b_{q,l} \) that has strict preferences over the alternatives \( x_i \), then we calculate \( S(x_i, b_{q,u}) \) and assign that alternative to the class \( l_q \) if \( S(x_i, b_{q,u}) \geq \lambda \) and \( S(x_i, b_{q,u}) < \lambda \).

1. Compare \( x_i \) successively to \( b_{q,u} \) for \( q = s, s-1, \ldots, 0 \)
2. \( b_{q,u} \) being the first bound such that \( b_{q,u} P x_i \), assign \( x_i \) to \( C_q \) \((x_i \not\in C_q)\) otherwise.

The optimistic procedure assign to \( x_i \) to the lowest
category \( C_q \) for which the upper bound \( b_q \) is preferred to \( x_i \). When using this procedure with \( \lambda = 1 \), an alternative \( x_i \) can be assigned to category \( C_q \) when \( g/b_q \) exceeds \( g(x_i) \) at least for one criterion. When \( \lambda \) decreases the optimistic character of this rule is weakened.

3.3. Comparison of Two Assignment Procedures

Let us suppose that an alternative \( x_i \) is assigned to \( C_q \) and \( C_r \) by the pessimistic and optimistic procedures, if the following conditions holds good

- \( C_q \) is lower or equal to \( C_r \) \((q \leq r)\)
- \( C_q > C_r \), when \( x_i bh_F \) for every \( F, r \leq q \).

More specifically when the evaluation of an alternative \( x_i \) and \( b_q \) and \( x_i \) divergence exists among the results of the two assignment procedures only when an alternative is incomparable to one or several \( b_q \), in such case the pessimistic rule assigns the alternative to lower category than the optimistic.

Here, we demonstrate a spreadsheet algorithm for the ELECTRE TRI method using a numerical illustration. We have programmed two algorithms, of which the first one helps in finding the values of partial concordance and discordance along with the outranking index between \( x_i \) and \( b_q \) and the second algorithm provides solution for \( C(b_q, x_i) \) and \( C(x_i, b_q) \).

Algorithm 3.1

Step 1: Enter the criteria values along with alternatives in ‘\( \text{max} \)’ design.

Step 2: Enter threshold values in a separate row below to the ‘\( \text{max} \)’ design.

Step 3: To compute the partial concordance between \( i^{th} \) criteria and \( j^{th} \) alternative \( C(j,b_q, x_i) \) the following ‘\( \text{NESTED IF (\( \lambda \))} \)’ condition has been used

\[
\text{IF}((B6+B10)<B2,0,IF((B6+B9)>B2,1,((B6-B2+B10)/(B10-B9))))
\]

Step 4: Repeat Step 3 for finding the left out concordance values.

Step 5: The overall concordance of two alternatives \( C(b_q, x_i) \) can be obtained using

\[ \text{\text{SUMPRODUCT}(H2:L2,B11:F11)/SUM(B11:F11))} \]

Step 6: To compute the partial discordance between \( i^{th} \) criteria and \( j^{th} \) alternative \( D(j,b_q, x_i) \) the following ‘\( \text{NESTED IF (\( \lambda \))} \)’ condition has been used

\[
\text{IF}((B20-B24)<B24,0,IF((B20-B9)>B2,1,((B20-B24)/(B26-B24))))
\]

Step 7: To compute the out ranking index between \( i^{th} \) criteria and \( j^{th} \) alternative \( S(b_q, x_i) \) the following ‘\( \text{IF (\( \lambda \))} \)’ condition has been used

\[
\text{IF}((H16>$S$2,($S$2*(1-H16))/(1-$S$2)),$S$2)
\]

Algorithm 3.2

Step 1: Enter the criteria values along with alternatives in ‘\( \text{max} \)’ design.

Step 2: Enter threshold values in a separate row below to the ‘\( \text{max} \)’ design.

4. Numerical Illustrations

Let us consider an MCDA problem which has five criteria and three alternatives for each criterion. The table below gives the boundary alternatives \( b_1 \) and \( b_2 \) and various thresholds given by the decision maker (DM).

<table>
<thead>
<tr>
<th>Alternatives</th>
<th>Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>( g_1 ) ( g_2 ) ( g_3 ) ( g_4 ) ( g_5 )</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>75       67   85   82   90</td>
</tr>
<tr>
<td>( x_3 )</td>
<td>28       35   70   90   95</td>
</tr>
</tbody>
</table>

Step 3: To compute the partial concordance between \( i^{th} \) criteria and \( j^{th} \) alternative \( C(j,b_q, x_i) \) the following ‘\( \text{NESTED IF (\( \lambda \))} \)’ condition has been used

\[
\text{IF}((B6+B10)<B2,0,IF((B6+B9)>B2,1,((B6-B2+B10)/(B10-B9))))
\]

Step 4: Repeat Step 3 for finding the left out concordance values.

Step 5: The overall concordance of two alternatives \( C(b_q, x_i) \) can be obtained using

\[ \text{\text{SUMPRODUCT}(H2:L2,B11:F11)/SUM(B11:F11))} \]

Step 6: To compute the partial discordance between \( i^{th} \) criteria and \( j^{th} \) alternative \( D(j,b_q, x_i) \) the following ‘\( \text{NESTED IF (\( \lambda \))} \)’ condition has been used

\[
\text{IF}((B20-B24)<B24,0,IF((B20-B9)>B2,1,((B20-B24)/(B26-B24))))
\]

Step 7: To compute the out ranking index between \( i^{th} \) criteria and \( j^{th} \) alternative \( S(b_q, x_i) \) the following ‘\( \text{IF (\( \lambda \))} \)’ condition has been used

\[
\text{IF}((H16>$S$2,($S$2*(1-H16))/(1-$S$2)),$S$2)
\]

Now, using the algorithm 3.1 and 3.2, the following values are computed. Along with the partial concordance and discordance, the overall concordance is also reported in the tables 1, 2, 3 and 4.

**Table 1.** Partial concordance of \( C(x_i, b_q) \)

<table>
<thead>
<tr>
<th>Alternatives</th>
<th>Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>( g_1 )</td>
<td>( g_2 ) ( g_3 ) ( g_4 ) ( g_5 )</td>
</tr>
<tr>
<td>( x_1 )</td>
<td>1        1    1    1    1</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>0        0    1    1    1</td>
</tr>
<tr>
<td>( x_3 )</td>
<td>1        1    1    1    1</td>
</tr>
<tr>
<td>( x_4 )</td>
<td>0        0    1    1    1</td>
</tr>
<tr>
<td>( x_5 )</td>
<td>0        0    0    0.6 0</td>
</tr>
</tbody>
</table>
On the basis of the above four tables, we have calculated the outranking indices for both $S(b_q, x_i)$ and $S(x_i, b_q)$.

### Table 5. Outranking indices for $S(x_i, b_q)$

<table>
<thead>
<tr>
<th></th>
<th>$g_1$</th>
<th>$g_2$</th>
<th>$g_3$</th>
<th>$g_4$</th>
<th>$g_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S(x_1, b_1)$</td>
<td>0.67</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>$S(x_2, b_1)$</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
</tr>
<tr>
<td>$S(x_3, b_1)$</td>
<td>0.92</td>
<td>0.92</td>
<td>0.92</td>
<td>0.92</td>
<td>0.92</td>
</tr>
<tr>
<td>$S(x_1, b_2)$</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
</tr>
<tr>
<td>$S(x_2, b_2)$</td>
<td>0.1</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
</tr>
</tbody>
</table>

### Table 6. Outranking indices for $S(b_q, x_i)$

<table>
<thead>
<tr>
<th></th>
<th>$g_1$</th>
<th>$g_2$</th>
<th>$g_3$</th>
<th>$g_4$</th>
<th>$g_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S(b_1, x_1)$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$S(b_1, x_2)$</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
</tr>
<tr>
<td>$S(b_1, x_3)$</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
</tr>
<tr>
<td>$S(b_2, x_1)$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$S(b_2, x_3)$</td>
<td>0.034</td>
<td>0.102</td>
<td>0.034</td>
<td>0.12</td>
<td>0.034</td>
</tr>
</tbody>
</table>

The table 7 gives a picture about the outranking relation between the criteria and alternatives.
After obtaining the Outranking indices, the decision maker will decide the cutting level \( \lambda \). Using this, the comparison will be done between the alternatives and criteria. Here, the cutting level \( \lambda \) is taken as 0.75. In this problem, we have defined two boundary alternatives that is \( b_1 \) and \( b_2 \). First let us consider the boundary alternative \( b_1 \) with three alternatives for \( g_1 \). The values of the indices \( S (x, b) \) and \( S (b, x) \) hold the relation \( P \) (strictly preference), since \( S (x, b) > \lambda \) and \( S (b, x) < \lambda \). In similar fashion, if we compare \( S (x, b) \) and \( S (b, x) \), then the outranking relation is of weak preference (Q). So here, we made an attempt to demonstrate all sorts of relations between the criteria and boundary alternatives using an MCDA problem. Further, let us consider another boundary alternative \( b_2 \) for three alternatives to explain and observe what sort of relations exists between them. It is observed that \( S (x_1, b) \) and \( S (b, x_1) > \lambda \), then the outranking relation is Incomparable (I). Similarly, if we compare \( S (x_2, b) \) and \( S (b, x_2) \) with \( \lambda \), the two relations are less than \( \lambda \) indicating that outranking relation is Indifference (I). Again on comparing \( S (x_3, b) \) and \( S (b, x_3) \), it is observed that \( S (x_3, b) < \lambda \) and \( S (b, x_3) > \lambda \), the outranking relation is weak preference (Q). Once the outranking relations are identified, the DM will choose any one of the assignment procedures. Here, we have briefly discussed both the procedures for the same problem.

Results of ELECTRE TRI Pessimistic procedure:
- \( x_1 \) is assigned to \( C_1 \) because \( x_1 S b_2 \) does not hold but \( x_2 S b_2 \) holds.
- \( x_2 \) is assigned to \( C_2 \) because \( x_2 S b_3, x_2 S b_2 \) and \( x_2 S b_1 \) do not hold but \( x_2 S b_0 \) holds.
- \( x_3 \) is assigned to \( C_3 \) because \( x_3 S b_3 \) and \( x_3 S b_2 \) do not hold but \( x_3 S b_0 \) holds.

Results of ELECTRE TRI Optimistic procedure:
- \( x_1 \) is assigned to \( C_1 \) because \( b_1 P x_1 \) do not holds but \( b_2 P x_1 \) holds.
- \( x_2 \) is assigned to \( C_2 \) because \( b_2 P x_2, b_1 P x_2 \) and \( b_2 P x_2 \) do not holds but \( b_2 P x_2 \) holds.
- \( x_3 \) is assigned to \( C_3 \) because \( b_3 P x_3, b_1 P x_3 \) does not holds but \( b_3 P x_3 \) holds.

It is observed that \( x_2 \) is assigned to \( C_2 \) by the optimistic procedure and \( C_1 \) by the pessimistic procedure. This shows that, \( x_2 \) is incomparable to both the boundary alternatives \( b_1 \) and \( b_2 \) which in turn gives the meaning that in spite of different priorities, \( x_2 \) a alternative is the preferable one in each and every criterion. Similar kind of interpretation can be given for the remaining criteria, \( g_2, g_3, g_4 \) and \( g_5 \).

### 4.2. EXAMPLE 2

<table>
<thead>
<tr>
<th>Alternatives</th>
<th>( g_1 )</th>
<th>( g_2 )</th>
<th>( g_3 )</th>
<th>( g_4 )</th>
<th>( g_5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>P</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>P</td>
</tr>
<tr>
<td>( x_2 )</td>
<td></td>
<td></td>
<td>J</td>
<td>J</td>
<td>J</td>
</tr>
<tr>
<td>( x_3 )</td>
<td>P</td>
<td>Q</td>
<td>P</td>
<td>Q</td>
<td>P</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alternatives</th>
<th>( g_1 )</th>
<th>( g_2 )</th>
<th>( g_3 )</th>
<th>( g_4 )</th>
<th>( g_5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>75</td>
<td>67</td>
<td>85</td>
<td>82</td>
<td>90</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>28</td>
<td>35</td>
<td>70</td>
<td>90</td>
<td>95</td>
</tr>
<tr>
<td>( x_3 )</td>
<td>45</td>
<td>60</td>
<td>55</td>
<td>68</td>
<td>60</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Boundary Alternative (b)</th>
<th>70</th>
<th>75</th>
<th>80</th>
<th>75</th>
<th>85</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Q ) (Indifference)</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>( P ) (Preference)</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>( W ) (Weights)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>( V ) (Veto)</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Partial Concordance for ( C(b, x_3) )</th>
<th>Overall Concordance</th>
</tr>
</thead>
<tbody>
<tr>
<td>( g_1 )</td>
<td>( C(x_1, b) )</td>
</tr>
<tr>
<td>( C(b, x_1) )</td>
<td>1.00</td>
</tr>
<tr>
<td>( C(b, x_2) )</td>
<td>1.00</td>
</tr>
<tr>
<td>( C(b, x_3) )</td>
<td>0.92</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Partial discordance for ( D(b, x_4) )</th>
<th>Overall Concordance</th>
</tr>
</thead>
<tbody>
<tr>
<td>( g_1 )</td>
<td>( C(x_1, b) )</td>
</tr>
<tr>
<td>( D(b, x_1) )</td>
<td>0.00</td>
</tr>
<tr>
<td>( D(b, x_2) )</td>
<td>0.00</td>
</tr>
<tr>
<td>( D(b, x_3) )</td>
<td>0.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Outranking indices for ( S(x, b) )</th>
<th>Overall Concordance</th>
</tr>
</thead>
<tbody>
<tr>
<td>( g_1 )</td>
<td>( C(x_1, b) )</td>
</tr>
<tr>
<td>( S(x_1, b) )</td>
<td>0.88</td>
</tr>
<tr>
<td>( S(x_2, b) )</td>
<td>0.88</td>
</tr>
<tr>
<td>( S(x_3, b) )</td>
<td>0.88</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Outranking indices for ( S(b, x) )</th>
<th>Overall Concordance</th>
</tr>
</thead>
<tbody>
<tr>
<td>( g_1 )</td>
<td>( C(x_1, b) )</td>
</tr>
<tr>
<td>( S(b, x_1) )</td>
<td>0.92</td>
</tr>
<tr>
<td>( S(b, x_2) )</td>
<td>0.92</td>
</tr>
<tr>
<td>( S(b, x_3) )</td>
<td>0.92</td>
</tr>
</tbody>
</table>

| \( S(b, x_2) \)                     | 0.92                |
| \( S(b, x_3) \)                     | 0.92                |

| \( S(b, x_2) \)                     | 0.92                |
| \( S(b, x_3) \)                     | 0.92                |

| \( S(b, x_2) \)                     | 0.92                |
| \( S(b, x_3) \)                     | 0.92                |

| \( S(b, x_2) \)                     | 0.92                |
| \( S(b, x_3) \)                     | 0.92                |
5. Conclusions

In MCDA problem, the outranking methodology of ELECTRE TRI method provides a compromise solution. In this paper, we have focused on the usage of spreadsheet procedures for the MCDA problem with ELECTRE TRI method. Further, we have considered two boundary alternatives and highlighted the importance of them. Finally, with the help of the outranking indices and relations, we have interpreted that the alternative $x_2$ is considered to be the best among three alternatives for every criterion. We have considered an MCDA problem which explains all sorts of outranking relations between the boundary alternatives and criteria. The algorithms are user friendly and flexible in handling the MCDA problem with ‘n’ boundary alternatives. The algorithm proposed is a user friendly one and allows user to handle the complex dimensioned MCDA problems very simply using the defined macro. Even though, separate software exists for ELECTRE TRI method, but it is not that easy to access and understand. However, this macro allow user to define the preferences, weights and thresholds. This macro is so handy and with a limited nested – if functions one can easily understand the anatomy of the ELECTRE TRI method.

ACKNOWLEDGEMENTS

The first author would like to acknowledge UGC-BSR for their financial support.
ActiveWindow.ScrollColumn = 3
ActiveWindow.ScrollColumn = 2
ActiveWindow.ScrollColumn = 1

ActiveCell.FormulaR1C1 = __

Range("N4").Select

ActiveWindow.ScrollColumn = 9
ActiveWindow.ScrollColumn = 8
ActiveWindow.ScrollColumn = 7
ActiveWindow.ScrollColumn = 6
ActiveWindow.ScrollColumn = 5
ActiveWindow.ScrollColumn = 4
ActiveWindow.ScrollColumn = 3
ActiveWindow.ScrollColumn = 2
ActiveWindow.ScrollColumn = 1

ActiveWindow.ScrollColumn = 8
ActiveWindow.ScrollColumn = 7
ActiveWindow.ScrollColumn = 6
ActiveWindow.ScrollColumn = 5
ActiveWindow.ScrollColumn = 4
ActiveWindow.ScrollColumn = 3
ActiveWindow.ScrollColumn = 2
ActiveWindow.ScrollColumn = 1

ActiveWindow.ScrollColumn = 9
ActiveWindow.ScrollColumn = 8
ActiveWindow.ScrollColumn = 7
ActiveWindow.ScrollColumn = 6
ActiveWindow.ScrollColumn = 5
ActiveWindow.ScrollColumn = 4
ActiveWindow.ScrollColumn = 3
ActiveWindow.ScrollColumn = 2
ActiveWindow.ScrollColumn = 1

ActiveCell.FormulaR1C1 = __

Range("N4").Select

ActiveWindow.ScrollColumn = 9
ActiveWindow.ScrollColumn = 8
ActiveWindow.ScrollColumn = 7
ActiveWindow.ScrollColumn = 6
ActiveWindow.ScrollColumn = 5
ActiveWindow.ScrollColumn = 4
ActiveWindow.ScrollColumn = 3
ActiveWindow.ScrollColumn = 2
ActiveWindow.ScrollColumn = 1

ActiveWindow.ScrollColumn = 8
ActiveWindow.ScrollColumn = 7
ActiveWindow.ScrollColumn = 6
ActiveWindow.ScrollColumn = 5
ActiveWindow.ScrollColumn = 4
ActiveWindow.ScrollColumn = 3
ActiveWindow.ScrollColumn = 2
ActiveWindow.ScrollColumn = 1

ActiveWindow.ScrollColumn = 9
ActiveWindow.ScrollColumn = 8
ActiveWindow.ScrollColumn = 7
ActiveWindow.ScrollColumn = 6
ActiveWindow.ScrollColumn = 5
ActiveWindow.ScrollColumn = 4
ActiveWindow.ScrollColumn = 3
ActiveWindow.ScrollColumn = 2
ActiveWindow.ScrollColumn = 1

ActiveWindow.ScrollColumn = 8
ActiveWindow.ScrollColumn = 7
ActiveWindow.ScrollColumn = 6
ActiveWindow.ScrollColumn = 5
ActiveWindow.ScrollColumn = 4
ActiveWindow.ScrollColumn = 3
ActiveWindow.ScrollColumn = 2
ActiveWindow.ScrollColumn = 1

ActiveWindow.ScrollColumn = 9
ActiveWindow.ScrollColumn = 8
ActiveWindow.ScrollColumn = 7
ActiveWindow.ScrollColumn = 6
ActiveWindow.ScrollColumn = 5
ActiveWindow.ScrollColumn = 4
ActiveWindow.ScrollColumn = 3
ActiveWindow.ScrollColumn = 2
ActiveWindow.ScrollColumn = 1

ActiveWindow.ScrollColumn = 8
ActiveWindow.ScrollColumn = 7
ActiveWindow.ScrollColumn = 6
ActiveWindow.ScrollColumn = 5
ActiveWindow.ScrollColumn = 4
ActiveWindow.ScrollColumn = 3
ActiveWindow.ScrollColumn = 2
ActiveWindow.ScrollColumn = 1

ActiveWindow.ScrollColumn = 9
ActiveWindow.ScrollColumn = 8
ActiveWindow.ScrollColumn = 7
ActiveWindow.ScrollColumn = 6
ActiveWindow.ScrollColumn = 5
ActiveWindow.ScrollColumn = 4
ActiveWindow.ScrollColumn = 3
ActiveWindow.ScrollColumn = 2
ActiveWindow.ScrollColumn = 1

ActiveWindow.ScrollColumn = 8
ActiveWindow.ScrollColumn = 7
ActiveWindow.ScrollColumn = 6
ActiveWindow.ScrollColumn = 5
ActiveWindow.ScrollColumn = 4
ActiveWindow.ScrollColumn = 3
ActiveWindow.ScrollColumn = 2
ActiveWindow.ScrollColumn = 1

ActiveWindow.ScrollColumn = 9
ActiveWindow.ScrollColumn = 8
ActiveWindow.ScrollColumn = 7
ActiveWindow.ScrollColumn = 6
ActiveWindow.ScrollColumn = 5
ActiveWindow.ScrollColumn = 4
ActiveWindow.ScrollColumn = 3
ActiveWindow.ScrollColumn = 2
ActiveWindow.ScrollColumn = 1

ActiveWindow.ScrollColumn = 8
ActiveWindow.ScrollColumn = 7
ActiveWindow.ScrollColumn = 6
ActiveWindow.ScrollColumn = 5
ActiveWindow.ScrollColumn = 4
ActiveWindow.ScrollColumn = 3
ActiveWindow.ScrollColumn = 2
ActiveWindow.ScrollColumn = 1
ActiveWindow.ScrollColumn = 3
ActiveWindow.ScrollColumn = 4
ActiveWindow.ScrollColumn = 5
Range("N19") Select
Selection.AutoFill Destination:=Range("N19:R19"),
Type:=xlFillDefault
Range("N19:R19").Select
Range("N20").Select
ActiveWindow.ScrollColumn = 4
ActiveWindow.ScrollColumn = 3
ActiveWindow.ScrollColumn = 2
ActiveWindow.ScrollColumn = 1
ActiveCell.FormulaR1C1 = _
Range("N21") Select
ActiveWindow.ScrollColumn = 2
ActiveWindow.ScrollColumn = 3
ActiveWindow.ScrollColumn = 4
ActiveWindow.ScrollColumn = 5
ActiveWindow.ScrollColumn = 6
ActiveWindow.ScrollColumn = 7
ActiveWindow.ScrollColumn = 8
Range("N20") Select
Selection.AutoFill Destination:=Range("N20:R20"),
Type:=xlFillDefault
Range("N20:R20").Select
Range("N21").Select
ActiveWindow.ScrollColumn = 7
ActiveWindow.ScrollColumn = 6
ActiveWindow.ScrollColumn = 5
ActiveWindow.ScrollColumn = 4
ActiveWindow.ScrollColumn = 3
ActiveWindow.ScrollColumn = 2
ActiveWindow.ScrollColumn = 1
ActiveCell.FormulaR1C1 = _
Range("N22") Select
ActiveWindow.ScrollColumn = 2
ActiveWindow.ScrollColumn = 3
ActiveWindow.ScrollColumn = 4
ActiveWindow.ScrollColumn = 5
ActiveWindow.ScrollColumn = 6
ActiveWindow.ScrollColumn = 7
ActiveWindow.ScrollColumn = 8
ActiveWindow.ScrollColumn = 9
ActiveWindow.ScrollColumn = 10
ActiveWindow.ScrollColumn = 11
Range("N21") Select
Selection.AutoFill Destination:=Range("N21:R21"),
Type:=xlFillDefault
Range("N21:R21").Select
"=IF(RC[-12]>R4C19,(R4C19*(1-RC[-12])/(1-R4C19)),R4C19)"
  Range("T19").Select
  ActiveWindow.SmallScroll ToRight:=5
  Range("T18").Select
  Selection.AutoFill Destination:=Range("T18:X18"),
  Type:=xlFillDefault
  Range("T18:X18").Select
  Range("T19").Select
  ActiveWindow.ScrollColumn = 10
  ActiveWindow.ScrollColumn = 9
  ActiveWindow.ScrollColumn = 8
  ActiveWindow.ScrollColumn = 7
  ActiveWindow.ScrollColumn = 6
  ActiveWindow.ScrollColumn = 5
  ActiveWindow.ScrollColumn = 4
  ActiveWindow.ScrollColumn = 3
  ActiveWindow.ScrollColumn = 2
  ActiveWindow.ScrollColumn = 1
  ActiveWindow.ScrollColumn = 0
  ActiveWindow.ScrollColumn = 11
  ActiveWindow.ScrollColumn = 10
  ActiveWindow.ScrollColumn = 9
  ActiveWindow.ScrollColumn = 8
  ActiveWindow.ScrollColumn = 7
  ActiveCell.FormulaR1C1 = _

"=IF(RC[-12]>R5C19,(R5C19*(1-RC[-12])/(1-R5C19)),R5C19)"
  Range("T20").Select
  ActiveWindow.ScrollColumn = 8
  ActiveWindow.ScrollColumn = 7
  ActiveWindow.ScrollColumn = 6
  ActiveWindow.ScrollColumn = 5
  ActiveWindow.ScrollColumn = 4
  ActiveWindow.ScrollColumn = 3
  ActiveWindow.ScrollColumn = 2
  ActiveWindow.ScrollColumn = 1
  ActiveWindow.ScrollColumn = 0
  ActiveWindow.ScrollColumn = 11
  ActiveWindow.ScrollColumn = 10
  ActiveWindow.ScrollColumn = 9
  ActiveWindow.ScrollColumn = 8
  ActiveWindow.ScrollColumn = 7
  ActiveWindow.ScrollColumn = 6
  ActiveWindow.ScrollColumn = 5
  ActiveWindow.ScrollColumn = 4
  ActiveWindow.ScrollColumn = 3
  ActiveWindow.ScrollColumn = 2
  ActiveWindow.ScrollColumn = 1
  ActiveWindow.ScrollColumn = 0

"=IF(RC[-12]>R6C19,(R6C19*(1-RC[-12])/(1-R6C19)),R6C19)"
  Range("T21").Select
  ActiveWindow.SmallScroll ToRight:=3
  Range("T20").Select
  Selection.AutoFill Destination:=Range("T20:X20"),
  Type:=xlFillDefault
  Range("T20:X20").Select
  Range("T21").Select
  ActiveWindow.SmallScroll ToRight:=2
  ActiveCell.FormulaR1C1 = _

"=IF(RC[-12]>R7C19,(R7C19*(1-RC[-12])/(1-R7C19)),R7C19)"
  Range("T22").Select
  ActiveWindow.SmallScroll ToRight:=2
  Range("T21").Select
  Selection.AutoFill Destination:=Range("T21:X21"),
  Type:=xlFillDefault
  Range("T21:X21").Select
  ActiveWindow.SmallScroll ToRight:=6
  Range("Z16").Select
  ActiveCell.FormulaR1C1 = _

"=IF(RC[-18]>R2C20,(R2C20*(1-RC[-18])/(1-R2C20)),R2C20)"
  Range("Z16").Select
  Selection.AutoFill Destination:=Range("Z16:AD16"),
  Type:=xlFillDefault
  Range("Z16:AD16").Select
  Range("Z17").Select
  ActiveCell.FormulaR1C1 = _

"=IF(RC[-18]>R3C20,(R3C20*(1-RC[-18])/(1-R3C20)),R3C20)"
  Range("Z17").Select
  Selection.AutoFill Destination:=Range("Z17:AD17"),
  Type:=xlFillDefault
  Range("Z17:AD17").Select
  Range("Z18").Select
  ActiveCell.FormulaR1C1 = _

"=IF(RC[-18]>R4C20,(R4C20*(1-RC[-18])/(1-R4C20)),R4C20)"
  Range("Z18").Select
  Selection.AutoFill Destination:=Range("Z18:AD18"),
  Type:=xlFillDefault
  Range("Z18:AD18").Select
  Range("Z19").Select
  ActiveCell.FormulaR1C1 = _

"=IF(RC[-18]>R5C20,(R5C20*(1-RC[-18])/(1-R5C20)),R5C20)"
  Range("Z19").Select
  Selection.AutoFill Destination:=Range("Z19:AD19"),
  Type:=xlFillDefault
  Range("Z19:AD19").Select
  Range("Z20").Select
  ActiveCell.FormulaR1C1 = _

"=IF(RC[-18]>R6C20,(R6C20*(1-RC[-18])/(1-R6C20)),R6C20)"
  Range("Z20").Select
  Selection.AutoFill Destination:=Range("Z20:AD20"),
  Type:=xlFillDefault
  Range("Z20:AD20").Select
  Range("Z21").Select
  ActiveCell.FormulaR1C1 = _

"=IF(RC[-18]>R7C20,(R7C20*(1-RC[-18])/(1-R7C20)),R7C20)"
  Range("Z21").Select
  Selection.AutoFill Destination:=Range("Z21:AD21"),
  Type:=xlFillDefault
REFERENCES


