A note on the sub-optimality of rank ordering of objects on the basis of the leading principal component factor scores

SK Mishra

North-Eastern Hill University, Shillong (India)

30. December 2008
A note on the sub-optimality of rank ordering of objects on the basis of the leading principal component factor scores

SK Mishra
Department of Economics
North-Eastern Hill University
Shillong (India)

1. The Problem
Suppose we intend to optimally rank order \( n \) objects (candidates) each of which has \( m \) attributes or rank scores awarded by \( m \) evaluators. More explicitly, let \( m \) evaluators award ranking scores to \( n \) \((n > m)\) candidates and let the array of ranking scores be denoted by \( X \) in \( n \) rows and \( m \) columns. The objective is to summarize \( X \) by a single \( n \)-element column vector of ranking scores, \( Z \), such that, in some sense, \( Z \) is the optimal representation of \( X \). Among many possible criteria of representation, the one could follow the principle that the sum of squares of the product moment coefficients of correlation between \( Z \) and \( X \) is maximum, or, stated symbolically, \( \sum_{j=1}^{m} r^2(Z, x_j) \) is maximum.

Conventionally, this problem is solved by the principal component analysis (Kendall and Stuart, 1968). It consists of obtaining \( Z = \mathcal{R}(Y) : Y = Xw; \max \sum_{j=1}^{m} r^2(Y, x_j) \), where the transformation \( \mathcal{R}(Y) \) assigns ranking scores to the elements of \( Y \) by any one of the ranking rules such as (a) standard competition ranking or the 1-2-2-4 rule, (b) modified competition ranking or 1-3-3-4 rule, (c) dense ranking or 1-2-2-3 rule, (d) ordinal ranking or 1-2-3-4 rule, (e) fractional ranking or 1-2.5-2.5-4 rule, etc (Wikipedia, 2008). The crux of the problem is, however, that optimality of \( Y \) does not entail optimality of \( Z \) or, stated differently, \( Z = \mathcal{R}(Y) : Y = Xw; \max \sum_{j=1}^{m} r^2(Y, x_j) \) does not necessarily ensure the optimality of \( Z \) with respect to the constituent variables. If \( Z = \mathcal{R}(Y) : Y = Xw; \max \sum_{j=1}^{m} r^2(Y, x_j) \) then oftentimes \( Z \) turns out to be only a suboptimal representation of \( X \).

The procedure of obtaining the ranking scores \( Z = \mathcal{R}(Y) : Y = Xw; \max \sum_{j=1}^{m} r^2(Y, x_j) \) follows two straightforward steps. First, the Principal component analysis is run on \( X \) to obtain \( Y \), which is the best linear combination, \( Xw \), of \( X \). It lies in (i) computing the inter-correlation matrix, \( R(m, m) \), from \( X \) such that \( r_{ij} \in R; r_{ij} = \text{cov}(x_i, x_j) / \sqrt{\text{var}(x_i) \text{var}(x_j)} \forall i, j \), (ii) finding the normed eigenvector, \( w \), of \( R \) associated with its largest eigenvalue, \( \lambda \), and (iii) using \( w \) to obtain \( Y = X' w \), where \( x_j^* = (x_j - \bar{x}_j) / \sqrt{\text{var}(x_j)} \). Next, \( Z = \mathcal{R}(Y) \) is obtained. It is well known that this procedure maximizes \( \sum_{j=1}^{m} r^2(Y, x_j) \). The direct maximization of \( \sum_{j=1}^{m} r^2(Y, x_j) \) is a rare practice.

However, what we seek is not the \( Z \) that maximizes \( \sum_{j=1}^{m} r^2(Y, x_j) \) but, instead, the \( Z \) that maximizes \( \sum_{j=1}^{m} r^2(Z, x_j) \). This is not guaranteed by the conventional Principal Component Analysis. This is what we demonstrate in this paper.

2. The Objective
Our objective in this paper is to show, by means of a few numerical examples, that the rank ordering of objects (candidates) by \( Z = \mathcal{R}(Y) \), where \( Y = Xw \) obtained by
maximizing $\sum_{j=1}^{m} r^2(Y, x_j)$, is suboptimal in the sense that it may not maximize $\sum_{j=1}^{m} r^2(Z, x_j)$. The rank ordering of objects by direct maximization of $\sum_{j=1}^{m} r^2(Z, x_j)$ is a better alternative. However, the conventional procedure of Principal Component analysis does not provide any scope for the same. There is no ready-to-use software available to maximize $\sum_{j=1}^{m} r^2(Z, x_j)$.

3. Materials and Methods: We simulate ranking score awarded to 30 candidates (objects) by 7 evaluators. The dataset, $X$, may also be viewed as the ranking scores obtained by 30 objects on each of their 7 attributes. First, $X$ is subjected to the Principal Component analysis and the composite score, $Y = Xw$, is obtained. This composite score relates to the first principal component associated with the largest eigenvalue of the correlation matrix of $X$. Therefore, $Y$ maximizes $\sum_{j=1}^{m} r^2(Y, x_j)$. Then $Y$ is rank ordered according to the 1234 ordinal ranking rule to obtain $Z$, the overall ranking scores for the candidates (objects). The inter-correlation matrix, $R$, is computed for $[Z_1, x_1, x_2, x_3, x_4, x_5, x_6, x_7]$ and, consequently, the measure of representativeness of $Z$, namely $F_1 = \sum_{j=1}^{7} r^2(Z_1, x_j)$, is computed. It may be noted that in this scheme $\sum_{j=1}^{m} r^2(Y, x_j)$ rather than $F_1 = \sum_{j=1}^{7} r^2(Z_1, x_j)$ is maximized.

In the second scheme, $F_2 = \sum_{j=1}^{7} r^2(Z_2, x_j)$, an alternative measure of representativeness, is directly maximized, where $Z_2 = R(Y') : Y' = Xv$. In finding $Z_2 = R(Y')$ the 1234 ordinal ranking rule is applied. As before, the inter-correlation matrix, $R$, is computed for $[Z_2, x_1, x_2, x_3, x_4, x_5, x_6, x_7]$. The method of Differential Evolution has been used for maximization of these alternative measures of representativeness, $F_1$ and $F_2$.

It would be pertinent to provide an introduction to the Differential Evolution (DE) method of optimization. The DE is one of the most recently invented methods of global optimization that has been very successful in optimization of extremely difficult multimodal functions. The DE is a population-based stochastic search method of optimization grown out of the Genetic algorithms. The crucial idea behind DE is a scheme for generating trial parameter vectors. Initially, a population of points (in d-dimensional space) is generated and evaluated (i.e. $f(p)$ is obtained) for their fitness. Then for each point ($p_i$) three different points ($p_a$, $p_b$ and $p_c$) are randomly chosen from the population. A new point ($p_i$) is constructed from those three points by adding the weighted difference between two points ($w(p_b-p_c)$) to the third point ($p_a$). Then this new point ($p_i$) is subjected to a crossover with the current point ($p_i$) with a probability of crossover ($c_r$), yielding a candidate point, say $p_u$. This point, $p_u$, is evaluated and if found better than $p_i$ then it replaces $p_i$ else $p_i$ remains. Thus we obtain a new vector in which all points are either better than or as good as the current points. This new vector is used for the next iteration. This process makes the differential evaluation scheme completely self-organizing. Operationally, this method consists of three basic steps: (i) generation of (large enough) population with N individuals [$u = (u_1, u_2, \ldots, u_m)$] in the m-dimensional space, randomly distributed over the entire domain of the function in question and evaluation of the individuals of the so generated by finding $f(u)$; (ii) replacement of this current population by a better fit new population, and (iii) repetition of this replacement until satisfactory results are obtained or certain criteria of termination are met. The crux of the problem lays in replacement of the
current population by a new population that is better fit. In this context, the meaning of ‘better’ is in the Pareto improvement sense. A set \( S_a \) is better than another set \( S_b \) iff: (i) no \( u_i \in S_a \) is inferior to the corresponding member of \( u_i \in S_b \); and (ii) at least one member \( u_k \in S_a \) is better than the corresponding member \( u_k \in S_b \). Thus, every new population is an improvement over the earlier one. To accomplish this, the DE procedure generates a candidate individual to replace each current individual in the population. The candidate individual is obtained by a crossover of the current individual and three other randomly selected individuals from the current population. The crossover itself is probabilistic in nature. Further, if the candidate individual is better fit than the current individual, it takes the place of the current individual, else the current individual stays and passes into the next iteration (Mishra, 2006).

It may further be noted that we have maximized \( \sum_{j=1}^{m} r^2(Y, x_j) \) as well as \( \sum_{j=1}^{7} r^2(Z_2, x_j) \) by DE. We have compared our results of maximization of \( \sum_{j=1}^{m} r^2(Y, x_j) \) with those obtained from STATISTICA (which has a built in facility to find the principal component-based factor scores and other related measures such as the eigenvalues, factor loadings, etc). Our results are essentially identical to those obtained from STATISTICA. This comparison ensures that our efforts in direct optimization have been perfectly successful. However, STATISTICA does not have a provision to maximize \( \sum_{j=1}^{7} r^2(Z_2, x_j) \). But since in the examples shown below \( F_2 \) is larger than \( F_1 \), we conclude that rank ordering of objects by \( Z_1 \) is suboptimal.

**Example-1**: The simulated dataset \((X)\) on ranking scores of 30 candidates awarded by 7 evaluators, the results obtained by running the principal component algorithm and the overall rankings based on the same \((Y\) and \(Z_1\)) and the results of rank order optimization exercise based on our method \((Y'\) and \(Z_1\)) are presented in Table-1.1. In table-1.2 are presented the inter-correlation matrix, \( R_1 \), for the variables \([Z_1, x_1, x_2, x_3, x_4, x_5, x_6]\). The last two rows of Table-1.2 are the weight \((w)\) vector used to obtain \( Y = Xw \) and factor loadings, that is, \( r(Y, x_j) \). The sum of squares of factor loadings \((S_1) = 4.352171\) and the measure of representativeness of \( Z_1 \) that is \( F_1 = \sum_{j=1}^{7} r^2(Z_1, x_j) = 4.287558 \).

In Table-1.3 we have presented the inter-correlation matrix, \( R_2 \), for variables \([Z_2, x_1, x_2, x_3, x_4, x_5, x_6]\), weights and the factor loadings when the same dataset (as mentioned above) is subjected to the direct maximization of \( \sum_{j=1}^{7} r^2(Z_2, x_j) \). The weights and the factor loadings relate to \( Y' = Xv \) and \( r(Z_2, x_j) \). The sum of square of factor loadings \((S_2) = 4.287902\) and the measure of representativeness of \( Z_2 \) that is \( F_2 = \sum_{j=1}^{7} r^2(Z_2, x_j) \) also is 4.287902. Since \( F_2 > F_1 \), the sub-optimality of the PC-based \( F_1 \) for this dataset is demonstrated. Notably, the candidates \#8, #20, #21 and #26 are rank ordered differently by the two methods.

**Example-2**: The simulated data and \( Y, Y', Z_1 \) and \( Z_2 \) for this dataset are presented in Table-2.1. The inter-correlation matrices, \( R_3 \) and \( R_5 \) and the associated weights and factor loadings also are presented in Tables-2.2 and 2.3. The values of \( F_1 \) and \( F_2 \) for this dataset are 3.120505 and
3.124149 respectively. This also shows the sub-optimality of the PC-based $F_1$. The candidates #8, #16, #18, #21, #23, #27, #28 and #29 are rank ordered differently by the two methods.

**Example-3:** One more simulated dataset and $Y$, $Y'$, $Z_1$ and $Z_2$ for this dataset are presented in Table-3.1. The inter-correlation matrices, $R_1$ and $R_2$ and the associated weights and factor loadings also are presented in Tables-3.2 and 3.3. The values of $F_1$ and $F_2$ for this dataset are 2.424195 and 2.426101 respectively. Once again, it is demonstrated that the PC-based $F_1$ is sub-optimal. The candidates #8, #13, #15, #19, and #26 are rank ordered differently by the two methods.

**4. Conclusion:** The three numerical examples suffice to demonstrate that the Principal Component based rankings scores, $Z_1$, obtained by maximization of $\sum_{j=1}^{m} r^2(Y_j, x_j): Y = Xw$ are not the best rank order scores and do not optimally represent the dataset from which they have been derived. Its alternative is to obtain $Z_2$ by the direct optimization of $\sum_{j=1}^{m} r^2(Z_j, x_j) = \sum_{j=1}^{m} r^2(\Re(Y'_j), x_j): Y'_i = Xv$. Although this direct optimization problem is extremely nonlinear and complicated, it can be accomplished by the use of advanced methods such as the Differential Evaluation method of global optimization.

**References**


Note: The Fortran Computer program for this exercise may also be obtained from mishrasknehu@yahoo.com
Table-1.1: Dataset Relating to Example-1 Showing Sub-optimality of PC-based Rank-ordering of Objects

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Ranking Scores of 30 candidates awarded by Seven Evaluators</th>
<th>Composite Score (Y) Optimized Results</th>
<th>Rank-Order (Z) Optimized Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X₁</td>
<td>X₂</td>
<td>X₃</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>28</td>
<td>18</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>23</td>
<td>29</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>19</td>
<td>18</td>
</tr>
<tr>
<td>6</td>
<td>26</td>
<td>27</td>
<td>28</td>
</tr>
<tr>
<td>7</td>
<td>18</td>
<td>25</td>
<td>30</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>9</td>
<td>15</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>21</td>
<td>26</td>
</tr>
<tr>
<td>10</td>
<td>16</td>
<td>17</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>22</td>
<td>15</td>
<td>17</td>
</tr>
<tr>
<td>12</td>
<td>25</td>
<td>12</td>
<td>22</td>
</tr>
<tr>
<td>13</td>
<td>15</td>
<td>23</td>
<td>16</td>
</tr>
<tr>
<td>14</td>
<td>21</td>
<td>4</td>
<td>25</td>
</tr>
<tr>
<td>15</td>
<td>24</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>16</td>
<td>29</td>
<td>24</td>
<td>29</td>
</tr>
<tr>
<td>17</td>
<td>3</td>
<td>8</td>
<td>13</td>
</tr>
<tr>
<td>18</td>
<td>12</td>
<td>30</td>
<td>14</td>
</tr>
<tr>
<td>19</td>
<td>14</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>20</td>
<td>17</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>21</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>22</td>
<td>20</td>
<td>28</td>
<td>8</td>
</tr>
<tr>
<td>23</td>
<td>10</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>24</td>
<td>9</td>
<td>14</td>
<td>17</td>
</tr>
<tr>
<td>25</td>
<td>30</td>
<td>20</td>
<td>23</td>
</tr>
<tr>
<td>26</td>
<td>6</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>27</td>
<td>13</td>
<td>13</td>
<td>10</td>
</tr>
<tr>
<td>28</td>
<td>19</td>
<td>6</td>
<td>19</td>
</tr>
<tr>
<td>29</td>
<td>27</td>
<td>22</td>
<td>24</td>
</tr>
<tr>
<td>30</td>
<td>7</td>
<td>11</td>
<td>4</td>
</tr>
</tbody>
</table>

Table-1.2: Inter-Correlation Matrix, Weights and Factor Loadings of Composite Score Optimized Overall Ranking Scores for the Dataset in Example-1
(F₁=4.287558; S₁= 4.352171)

<table>
<thead>
<tr>
<th>Z₁</th>
<th>X₁</th>
<th>X₂</th>
<th>X₃</th>
<th>X₄</th>
<th>X₅</th>
<th>X₆</th>
<th>X₇</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z₁</td>
<td>1.000000</td>
<td>0.805117</td>
<td>0.781980</td>
<td>0.801112</td>
<td>0.891880</td>
<td>0.690768</td>
<td>0.658287</td>
</tr>
<tr>
<td>X₁</td>
<td>0.805117</td>
<td>1.000000</td>
<td>0.474082</td>
<td>0.666741</td>
<td>0.645384</td>
<td>0.451390</td>
<td>0.537709</td>
</tr>
<tr>
<td>X₂</td>
<td>0.781980</td>
<td>0.474082</td>
<td>1.000000</td>
<td>0.554616</td>
<td>0.688543</td>
<td>0.552392</td>
<td>0.409121</td>
</tr>
<tr>
<td>X₃</td>
<td>0.801112</td>
<td>0.666741</td>
<td>0.554616</td>
<td>1.000000</td>
<td>0.731257</td>
<td>0.438487</td>
<td>0.426919</td>
</tr>
<tr>
<td>X₄</td>
<td>0.891880</td>
<td>0.645384</td>
<td>0.688543</td>
<td>0.731257</td>
<td>1.000000</td>
<td>0.526140</td>
<td>0.606229</td>
</tr>
<tr>
<td>X₅</td>
<td>0.690768</td>
<td>0.451390</td>
<td>0.552392</td>
<td>0.438487</td>
<td>0.526140</td>
<td>1.000000</td>
<td>0.324583</td>
</tr>
<tr>
<td>X₆</td>
<td>0.658287</td>
<td>0.537709</td>
<td>0.688543</td>
<td>0.688543</td>
<td>0.731257</td>
<td>0.526140</td>
<td>1.000000</td>
</tr>
<tr>
<td>X₇</td>
<td>0.824694</td>
<td>0.597330</td>
<td>0.569299</td>
<td>0.491880</td>
<td>0.606229</td>
<td>0.324583</td>
<td>0.608009</td>
</tr>
<tr>
<td>Weights</td>
<td>0.381563</td>
<td>0.369979</td>
<td>0.377237</td>
<td>0.430743</td>
<td>0.337890</td>
<td>0.337585</td>
<td>0.401926</td>
</tr>
<tr>
<td>Loadings</td>
<td>0.796012</td>
<td>0.771845</td>
<td>0.786986</td>
<td>0.898611</td>
<td>0.704264</td>
<td>0.704901</td>
<td>0.838493</td>
</tr>
</tbody>
</table>
Table-1.3: Inter-Correlation Matrix, Weights and Factor Loadings of
Rank Order Optimized Overall Ranking Scores for the Dataset in Example-1
\((F_2=4.287902; S^2=4.287902)\)

<table>
<thead>
<tr>
<th></th>
<th>Z2</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z2</td>
<td>1.000000</td>
<td>0.810901</td>
<td>0.776641</td>
<td>0.800667</td>
<td>0.893660</td>
<td>0.688988</td>
<td>0.653838</td>
<td>0.827809</td>
</tr>
<tr>
<td>X1</td>
<td>0.810901</td>
<td>1.000000</td>
<td>0.474082</td>
<td>0.666741</td>
<td>0.645384</td>
<td>0.451390</td>
<td>0.573709</td>
<td>0.597330</td>
</tr>
<tr>
<td>X2</td>
<td>0.776641</td>
<td>0.474082</td>
<td>1.000000</td>
<td>0.554616</td>
<td>0.688543</td>
<td>0.552392</td>
<td>0.409121</td>
<td>0.569299</td>
</tr>
<tr>
<td>X3</td>
<td>0.800667</td>
<td>0.666741</td>
<td>0.554616</td>
<td>1.000000</td>
<td>0.731257</td>
<td>0.438487</td>
<td>0.426919</td>
<td>0.491880</td>
</tr>
<tr>
<td>X4</td>
<td>0.893660</td>
<td>0.645384</td>
<td>0.731257</td>
<td>0.554616</td>
<td>1.000000</td>
<td>0.526140</td>
<td>0.708120</td>
<td>0.712680</td>
</tr>
<tr>
<td>X5</td>
<td>0.688988</td>
<td>0.451390</td>
<td>0.438487</td>
<td>0.688543</td>
<td>0.526140</td>
<td>1.000000</td>
<td>0.324583</td>
<td>0.630256</td>
</tr>
<tr>
<td>X6</td>
<td>0.653838</td>
<td>0.573709</td>
<td>0.426919</td>
<td>0.645384</td>
<td>0.708120</td>
<td>0.324583</td>
<td>1.000000</td>
<td>0.608009</td>
</tr>
<tr>
<td>X7</td>
<td>0.827809</td>
<td>0.597330</td>
<td>0.491880</td>
<td>0.731257</td>
<td>0.712680</td>
<td>0.630256</td>
<td>0.608009</td>
<td>1.000000</td>
</tr>
</tbody>
</table>

Weights

0.441363 0.418945 0.360646 0.411243 0.274724 0.203625 0.462126

Loadings

0.810901 0.776641 0.800667 0.893660 0.688988 0.653838 0.827809

Table-2.1: Dataset Relating to Example-2 Showing Sub-optimality of PC-based Rank-ordering of Objects

<table>
<thead>
<tr>
<th>Sl No.</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>Y</th>
<th>Z1</th>
<th>Y'</th>
<th>Z2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>5</td>
<td>7</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>27.96401</td>
<td>9</td>
<td>27.66608</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>23</td>
<td>25</td>
<td>30</td>
<td>24</td>
<td>22</td>
<td>61.09793</td>
<td>26</td>
<td>60.93414</td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>4</td>
<td>6</td>
<td>14</td>
<td>13</td>
<td>20.05915</td>
<td>3</td>
<td>19.25318</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>29</td>
<td>7</td>
<td>30</td>
<td>28</td>
<td>21</td>
<td>65.03329</td>
<td>27</td>
<td>64.65080</td>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>13</td>
<td>2</td>
<td>6</td>
<td>14</td>
<td>27.3433</td>
<td>8</td>
<td>27.55713</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>1</td>
<td>11</td>
<td>15</td>
<td>1</td>
<td>18.47398</td>
<td>2</td>
<td>18.61302</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>17</td>
<td>27</td>
<td>18</td>
<td>14</td>
<td>25</td>
<td>39.62596</td>
<td>13</td>
<td>40.64115</td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>10</td>
<td>18</td>
<td>21</td>
<td>10</td>
<td>47.14993</td>
<td>20</td>
<td>47.27035</td>
<td>19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>24</td>
<td>19</td>
<td>12</td>
<td>23</td>
<td>6</td>
<td>43.75459</td>
<td>17</td>
<td>43.78237</td>
<td>17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>22</td>
<td>21</td>
<td>19</td>
<td>30</td>
<td>27</td>
<td>66.20587</td>
<td>28</td>
<td>66.35746</td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>13</td>
<td>12</td>
<td>16</td>
<td>24</td>
<td>30</td>
<td>44.98232</td>
<td>18</td>
<td>45.39889</td>
<td>18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>21</td>
<td>24</td>
<td>7</td>
<td>10</td>
<td>29</td>
<td>48.43576</td>
<td>24</td>
<td>49.56098</td>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>2</td>
<td>16</td>
<td>15</td>
<td>2</td>
<td>25.42514</td>
<td>7</td>
<td>25.15853</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>27</td>
<td>18</td>
<td>21</td>
<td>18</td>
<td>16</td>
<td>54.18113</td>
<td>25</td>
<td>54.42185</td>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>23</td>
<td>11</td>
<td>4</td>
<td>7</td>
<td>24.68224</td>
<td>6</td>
<td>24.91456</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>15</td>
<td>28</td>
<td>6</td>
<td>20</td>
<td>41.87725</td>
<td>16</td>
<td>41.81565</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>4</td>
<td>16</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>11.99989</td>
<td>1</td>
<td>12.74124</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>6</td>
<td>26</td>
<td>25</td>
<td>14</td>
<td>13</td>
<td>47.83244</td>
<td>21</td>
<td>48.71014</td>
<td>22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>25</td>
<td>9</td>
<td>8</td>
<td>11</td>
<td>23</td>
<td>32.41507</td>
<td>11</td>
<td>32.13677</td>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>18</td>
<td>11</td>
<td>10</td>
<td>21</td>
<td>10</td>
<td>34.62240</td>
<td>12</td>
<td>35.12062</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>7</td>
<td>30</td>
<td>20</td>
<td>15</td>
<td>27</td>
<td>47.94087</td>
<td>22</td>
<td>49.26167</td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>30</td>
<td>22</td>
<td>29</td>
<td>26</td>
<td>29</td>
<td>69.86155</td>
<td>30</td>
<td>69.59827</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>26</td>
<td>8</td>
<td>14</td>
<td>19</td>
<td>17</td>
<td>41.57229</td>
<td>15</td>
<td>41.00634</td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>9</td>
<td>20</td>
<td>8</td>
<td>26</td>
<td>13</td>
<td>29.02329</td>
<td>10</td>
<td>30.49082</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>6</td>
<td>13</td>
<td>25</td>
<td>12</td>
<td>23.40574</td>
<td>5</td>
<td>24.09491</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>28</td>
<td>17</td>
<td>23</td>
<td>27</td>
<td>28</td>
<td>66.86960</td>
<td>29</td>
<td>66.99722</td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>14</td>
<td>29</td>
<td>22</td>
<td>19</td>
<td>8</td>
<td>46.79227</td>
<td>19</td>
<td>47.55844</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>8</td>
<td>28</td>
<td>17</td>
<td>24</td>
<td>7</td>
<td>41.44998</td>
<td>14</td>
<td>42.63972</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>19</td>
<td>14</td>
<td>22</td>
<td>17</td>
<td>22</td>
<td>48.27078</td>
<td>23</td>
<td>48.05095</td>
<td>21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>8</td>
<td>22</td>
<td>20.66025</td>
<td>4</td>
<td>21.15313</td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table-2.2: Inter-Correlation Matrix, Weights and Factor Loadings of Composite Score Optimized Overall Ranking Scores for the Dataset in Example-2

\((F_1=3.120505; S_1=3.215147)\)

<table>
<thead>
<tr>
<th></th>
<th>Z₁</th>
<th>X₁</th>
<th>X₂</th>
<th>X₃</th>
<th>X₄</th>
<th>X₅</th>
<th>X₆</th>
<th>X₇</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z₁</td>
<td>1.000000</td>
<td>0.726363</td>
<td>0.499889</td>
<td>0.739266</td>
<td>0.636040</td>
<td>0.696997</td>
<td>0.650723</td>
<td>0.694772</td>
</tr>
<tr>
<td>X₁</td>
<td>0.726363</td>
<td>1.000000</td>
<td>0.189766</td>
<td>0.419355</td>
<td>0.315684</td>
<td>0.393548</td>
<td>0.554616</td>
<td>0.568409</td>
</tr>
<tr>
<td>X₂</td>
<td>0.499889</td>
<td>0.189766</td>
<td>1.000000</td>
<td>0.388654</td>
<td>0.295217</td>
<td>0.125695</td>
<td>0.126585</td>
<td>0.353949</td>
</tr>
<tr>
<td>X₃</td>
<td>0.739266</td>
<td>0.419355</td>
<td>0.388654</td>
<td>1.000000</td>
<td>0.439377</td>
<td>0.617798</td>
<td>0.426919</td>
<td>0.276085</td>
</tr>
<tr>
<td>X₄</td>
<td>0.636040</td>
<td>0.315684</td>
<td>0.295217</td>
<td>0.439377</td>
<td>1.000000</td>
<td>0.417575</td>
<td>0.347949</td>
<td>0.245384</td>
</tr>
<tr>
<td>X₅</td>
<td>0.696997</td>
<td>0.393548</td>
<td>0.125695</td>
<td>0.617798</td>
<td>0.417575</td>
<td>1.000000</td>
<td>0.348165</td>
<td>0.370857</td>
</tr>
<tr>
<td>X₆</td>
<td>0.650723</td>
<td>0.554616</td>
<td>0.126585</td>
<td>0.426919</td>
<td>0.347949</td>
<td>0.348165</td>
<td>1.000000</td>
<td>0.409566</td>
</tr>
<tr>
<td>X₇</td>
<td>0.694772</td>
<td>0.568409</td>
<td>0.353949</td>
<td>0.276085</td>
<td>0.348165</td>
<td>0.409566</td>
<td>1.000000</td>
<td>0.698999</td>
</tr>
</tbody>
</table>

Weights
0.417835 0.262744 0.427355 0.351103 0.398451 0.384622 0.379179

Loadings
0.749214 0.471122 0.766283 0.629557 0.714566 0.689659 0.679899

Table-2.3: Inter-Correlation Matrix, Weights and Factor Loadings of Rank Order Optimized Overall Ranking Scores for the Dataset in Example-2

\((F_2=3.124149; S_2=3.124149)\)

<table>
<thead>
<tr>
<th></th>
<th>Z₂</th>
<th>X₁</th>
<th>X₂</th>
<th>X₃</th>
<th>X₄</th>
<th>X₅</th>
<th>X₆</th>
<th>X₇</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z₂</td>
<td>1.000000</td>
<td>0.701001</td>
<td>0.535484</td>
<td>0.740156</td>
<td>0.646719</td>
<td>0.695217</td>
<td>0.640934</td>
<td>0.696997</td>
</tr>
<tr>
<td>X₁</td>
<td>0.701001</td>
<td>1.000000</td>
<td>0.189766</td>
<td>0.419355</td>
<td>0.315684</td>
<td>0.393548</td>
<td>0.554616</td>
<td>0.568409</td>
</tr>
<tr>
<td>X₂</td>
<td>0.535484</td>
<td>0.189766</td>
<td>1.000000</td>
<td>0.388654</td>
<td>0.295217</td>
<td>0.125695</td>
<td>0.126585</td>
<td>0.353949</td>
</tr>
<tr>
<td>X₃</td>
<td>0.740156</td>
<td>0.419355</td>
<td>0.388654</td>
<td>1.000000</td>
<td>0.439377</td>
<td>0.617798</td>
<td>0.426919</td>
<td>0.276085</td>
</tr>
<tr>
<td>X₄</td>
<td>0.646719</td>
<td>0.315684</td>
<td>0.295217</td>
<td>0.439377</td>
<td>1.000000</td>
<td>0.417575</td>
<td>0.347949</td>
<td>0.245384</td>
</tr>
<tr>
<td>X₅</td>
<td>0.695217</td>
<td>0.393548</td>
<td>0.125695</td>
<td>0.617798</td>
<td>0.417575</td>
<td>1.000000</td>
<td>0.348165</td>
<td>0.370857</td>
</tr>
<tr>
<td>X₆</td>
<td>0.640934</td>
<td>0.554616</td>
<td>0.126585</td>
<td>0.426919</td>
<td>0.347949</td>
<td>0.348165</td>
<td>1.000000</td>
<td>0.409566</td>
</tr>
<tr>
<td>X₇</td>
<td>0.696997</td>
<td>0.568409</td>
<td>0.353949</td>
<td>0.276085</td>
<td>0.348165</td>
<td>0.409566</td>
<td>1.000000</td>
<td>0.698999</td>
</tr>
</tbody>
</table>

Weights
0.383664 0.31504 0.378173 0.385789 0.414933 0.379452 0.381511

Loadings
0.701001 0.535484 0.740156 0.646719 0.695217 0.640934 0.696997

Table-3.1: Dataset Relating to Example-3 Showing Sub-optimality of PC-based Rank-ordering of Objects

<table>
<thead>
<tr>
<th>Sl No.</th>
<th>X₁</th>
<th>X₂</th>
<th>X₃</th>
<th>X₄</th>
<th>X₅</th>
<th>X₆</th>
<th>X₇</th>
<th>Y</th>
<th>Z₁</th>
<th>Y'</th>
<th>Z₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>5</td>
<td>27</td>
<td>18</td>
<td>2</td>
<td>20</td>
<td>11</td>
<td>32.10776</td>
<td>10</td>
<td>31.59814</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>28</td>
<td>30</td>
<td>29</td>
<td>17</td>
<td>29</td>
<td>13</td>
<td>59.10295</td>
<td>26</td>
<td>58.52745</td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>22</td>
<td>13</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>22.97097</td>
<td>3</td>
<td>21.15144</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>24</td>
<td>23</td>
<td>9</td>
<td>26</td>
<td>30</td>
<td>25</td>
<td>28</td>
<td>62.42732</td>
<td>28</td>
<td>61.62047</td>
<td>28</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>25</td>
<td>22</td>
<td>5</td>
<td>22</td>
<td>6</td>
<td>1</td>
<td>39.50937</td>
<td>18</td>
<td>40.81833</td>
<td>18</td>
</tr>
<tr>
<td>6</td>
<td>27</td>
<td>30</td>
<td>16</td>
<td>24</td>
<td>28</td>
<td>27</td>
<td>30</td>
<td>68.43141</td>
<td>30</td>
<td>67.46975</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>22</td>
<td>2</td>
<td>19</td>
<td>12</td>
<td>16</td>
<td>4</td>
<td>16</td>
<td>32.23785</td>
<td>11</td>
<td>32.42260</td>
<td>11</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>8</td>
<td>20</td>
<td>2</td>
<td>9</td>
<td>21</td>
<td>18</td>
<td>30.81457</td>
<td>6</td>
<td>30.37723</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>18</td>
<td>19</td>
<td>8</td>
<td>21</td>
<td>8</td>
<td>14</td>
<td>7</td>
<td>35.11395</td>
<td>14</td>
<td>34.68187</td>
<td>14</td>
</tr>
<tr>
<td>10</td>
<td>14</td>
<td>6</td>
<td>15</td>
<td>4</td>
<td>15</td>
<td>3</td>
<td>29</td>
<td>30.98154</td>
<td>7</td>
<td>29.98579</td>
<td>7</td>
</tr>
<tr>
<td>11</td>
<td>26</td>
<td>4</td>
<td>14</td>
<td>7</td>
<td>4</td>
<td>7</td>
<td>3</td>
<td>17.61105</td>
<td>1</td>
<td>18.44842</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>21</td>
<td>7</td>
<td>21</td>
<td>14</td>
<td>4</td>
<td>22</td>
<td>10</td>
<td>35.41392</td>
<td>15</td>
<td>35.50598</td>
<td>15</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>20</td>
<td>7</td>
<td>16</td>
<td>5</td>
<td>13</td>
<td>21</td>
<td>31.73604</td>
<td>9</td>
<td>29.76353</td>
<td>6</td>
</tr>
</tbody>
</table>
### Table-3.2: Inter-Correlation Matrix, Weights and Factor Loadings of Composite Score Optimized Overall Ranking Scores for the Dataset in Example-3  
\((F_1=2.424195; S_1=2.521362)\)

<table>
<thead>
<tr>
<th></th>
<th>Z_1</th>
<th>X_1</th>
<th>X_2</th>
<th>X_3</th>
<th>X_4</th>
<th>X_5</th>
<th>X_6</th>
<th>X_7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z_1</td>
<td>1</td>
<td>0.473637</td>
<td>0.612013</td>
<td>0.566185</td>
<td>0.562180</td>
<td>0.741046</td>
<td>0.654283</td>
<td>0.459844</td>
</tr>
<tr>
<td>X_1</td>
<td>0.473637</td>
<td>1</td>
<td>0.108788</td>
<td>0.187542</td>
<td>0.132814</td>
<td>0.470078</td>
<td>-0.006007</td>
<td>0.119911</td>
</tr>
<tr>
<td>X_2</td>
<td>0.612013</td>
<td>0.108788</td>
<td>1</td>
<td>0.163070</td>
<td>0.360623</td>
<td>0.303226</td>
<td>0.402447</td>
<td>0.309010</td>
</tr>
<tr>
<td>X_3</td>
<td>0.566185</td>
<td>0.187542</td>
<td>0.163070</td>
<td>1</td>
<td>0.180868</td>
<td>0.296552</td>
<td>0.389544</td>
<td>0.126585</td>
</tr>
<tr>
<td>X_4</td>
<td>0.562180</td>
<td>0.132814</td>
<td>0.360623</td>
<td>0.180868</td>
<td>1</td>
<td>0.200000</td>
<td>0.284983</td>
<td>0.289433</td>
</tr>
<tr>
<td>X_5</td>
<td>0.741046</td>
<td>0.470078</td>
<td>0.303226</td>
<td>0.296552</td>
<td>0.200000</td>
<td>1</td>
<td>0.384650</td>
<td>0.315684</td>
</tr>
<tr>
<td>X_6</td>
<td>0.654283</td>
<td>-0.006007</td>
<td>0.402447</td>
<td>0.389544</td>
<td>0.284983</td>
<td>0.384650</td>
<td>1</td>
<td>0.156396</td>
</tr>
</tbody>
</table>
| X_7   | 0.459844| 0.119911| 0.309010| 0.126585| 0.284983| 0.315684| 0.156396| 1

| Weights| 0.271198| 0.414408| 0.345937| 0.366429| 0.458344| 0.416812| 0.341998|
| Loadings| 0.430630| 0.658031| 0.549306| 0.581846| 0.543052|

### Table-3.3: Inter-Correlation Matrix, Weights and Factor Loadings of Rank Order Optimized Overall Ranking Scores for the Dataset in Example-3  
\((F_2=2.426101; S_2=2.426101)\)

<table>
<thead>
<tr>
<th></th>
<th>Z_2</th>
<th>X_1</th>
<th>X_2</th>
<th>X_3</th>
<th>X_4</th>
<th>X_5</th>
<th>X_6</th>
<th>X_7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z_2</td>
<td>1</td>
<td>0.479867</td>
<td>0.599555</td>
<td>0.576418</td>
<td>0.543493</td>
<td>0.751279</td>
<td>0.666741</td>
<td>0.446941</td>
</tr>
<tr>
<td>X_1</td>
<td>0.479867</td>
<td>1</td>
<td>0.108788</td>
<td>0.187542</td>
<td>0.132814</td>
<td>0.470078</td>
<td>-0.006007</td>
<td>0.119911</td>
</tr>
<tr>
<td>X_2</td>
<td>0.599555</td>
<td>0.108788</td>
<td>1</td>
<td>0.163070</td>
<td>0.360623</td>
<td>0.303226</td>
<td>0.402447</td>
<td>0.309010</td>
</tr>
<tr>
<td>X_3</td>
<td>0.576418</td>
<td>0.187542</td>
<td>0.163070</td>
<td>1</td>
<td>0.180868</td>
<td>0.296552</td>
<td>0.389544</td>
<td>0.126585</td>
</tr>
<tr>
<td>X_4</td>
<td>0.543493</td>
<td>0.132814</td>
<td>0.360623</td>
<td>0.180868</td>
<td>1</td>
<td>0.200000</td>
<td>0.284983</td>
<td>0.289433</td>
</tr>
<tr>
<td>X_5</td>
<td>0.751279</td>
<td>0.470078</td>
<td>0.303226</td>
<td>0.296552</td>
<td>0.200000</td>
<td>1</td>
<td>0.384650</td>
<td>0.315684</td>
</tr>
<tr>
<td>X_6</td>
<td>0.666741</td>
<td>-0.006007</td>
<td>0.402447</td>
<td>0.389544</td>
<td>0.284983</td>
<td>0.384650</td>
<td>1</td>
<td>0.156396</td>
</tr>
</tbody>
</table>
| X_7   | 0.446941| 0.119911| 0.309010| 0.126585| 0.284983| 0.315684| 0.156396| 1

| Weights| 0.297447| 0.386881| 0.363218| 0.338013| 0.508040| 0.430645| 0.268531|
| Loadings| 0.430630| 0.658031| 0.549306| 0.581846| 0.727796| 0.661847| 0.543052|
MAIN PROGRAM : PROVIDES TO USE DIFFERENTIAL EVOLUTION METHOD TO
COMPUTE COMPOSITE INDEX INDICES
BY MAXIMIZING SUM OF (SQUARES, OR ABSOLUTES, OR MINIMUM) OF
CORRELATION OF THE INDEX WITH THE CONSTITUENT VARIABLES. THE MAX
SUM OF SQUARES IS THE PRINCIPAL COMPONENT INDEX. IT ALSO PRIVIDES
TO OBTAIN MAXIMUM ENTROPY ABSOLUTE CORRELATION INDICES.
PRODUCT MOMENT AS WELL AS ABSOLUTE CORRELATION (BRADLEY, 1985) MAY
BE USED. PROGRAM BY SK MISHRA, DEPT. OF ECONOMICS, NORTH-EASTERN
HILL UNIVERSITY, SHILLONG (INDIA)

--- WARNING ---
ADJUST THE PARAMETERS SUITABLY IN SUBROUTINES DE
WHEN THE PROGRAM ASKS FOR PARAMETERS, FEED THEM SUITABLY

PROGRAM RANKOPT
PARAMETER(NOB=30, MVAR=7) !CHANGE THE PARAMETERS HERE AS NEEDED.
NOB=NO. OF CASES AND MVAR=NO. OF VARIABLES
TO BE ADJUSTED IN SUBROUTINE CORD(M,X,F) ALSO: STATEMENT 931
IMPLICIT DOUBLE PRECISION (A-H, O-Z)
COMMON /KFF,KF,NCALL,FTIT ! FUNCTION CODE, NO. OF CALLS & TITLE
CHARACTER *30 METHOD(1)
CHARACTER *70 FTIT
CHARACTER *40 INFILE,OUTFILE
COMMON /CORDAT/CORDAT(NOB,MVAR),QIND(NOB),R(MVAR),ENTROPY,NORM,NCOR
COMMON /XBAS/XBAS
COMMON /RNDM/IV,IU,IV ! RANDOM NUMBER GENERATION (IU = 4-DIGIT SEED)
COMMON /GETRANK/MRNK
INTEGER IV,IU,IV
DIMENSION XX(3,50),KMF(3),MM(3),FMNN(3),XBAS(1000,50)
DIMENSION ZDAT(NOB,MVAR+1),FRANK(NOB),RMAT(MVAR+1,MVAR+1)
DIMENSION X(50) ! X IS THE DECISION VARIABLE X IN F(X) TO MINIMIZE
EMIN IS THE MIN VALUE OF F(X) OBTAINED FROM DE OR RPS
WRITE(*,*)'--------------------------------------- WARNING ---------------------'
WRITE(*,*)'ADJUST PARAMETERS IN SUBROUTINES DE IF NEEDED '
NOPT=1 ! OPTIMIZATION BY DE METHOD
WRITE(*,*)'--------------------------------------- DIFFERENTIAL EVOLUTION'
METHOD(1)=0 : DIFFERENTIAL EVOLUTION'
WRITE(*,*)'--------------------------------------- INITIALIZE THE POPULATION.'
WRITE(*,*)' ' WRITE(*,*)'FEED RANDOM NUMBER SEED,NORM,ENTROPY,NCOR'
WRITE(*,*)'SEED[ANY 4-DIGIT NUMBER]; NORM[1,2,3]; ENTROPY[0,1]; &
&NCOR[0,1]'
WRITE(*,*)' ' WRITE(*,*)'NORM(1)=ABSOLUTE;NORM(2)=PCA-EUCLIDEAN;NORM(3)=MAXIMIN'
WRITE(*,*)' ' WRITE(*,*)'ENTROPY(0)=MAXIMIZES NORM;ENTROPY(1)=MAXIMIZES ENTROPY'
WRITE(*,*)' ' WRITE(*,*)'NCOR(0)=PRODUCT MOMENT; NCOR(1)=ABSOLUTE CORRELATION'
WRITE(*,*)' ' WRITE(*,*)'IU,NORM,ENTROPY,NCOR
WRITE(*,*)'WANT RANK SCORE OPTIMIZATION? YES(1); NO(OTHER THAN 1)'
READ(*,*) MRNK
WRITE(*,*)'INPUT FILE TO READ DATA: YOUR DATA MUST BE IN THIS FILE'
WRITE(*,*)'CASES (NOB) IN ROWS ; VARIABLES (MVAR) IN COLUMNS'
READ(*,*) INFILE
WRITE(*,*)'SPECIFY THE OUTPUT FILE TO STORE THE RESULTS'
READ(*,*) OUTFILE
OPEN(9, FILE=OUTFILE)
OPEN(7, FILE=INFILE)
DO I=1,NOB
READ(7,*) CDAT, (CDAT(I,J),J=1,MVAR)
ENDDO
CLOSE(7)
DO I=1,NOB
DO J=1,MVAR
ZDAT(I,J+1)=CDAT(I,J)
ENDDO
ENDDO
!CALLS DE AND RETURNS OPTIMAL X AND FMIN

WRITE(9,*) 'DIFFERENTIAL EVALUATION OPTIMIZATION RESULTS'
RSUM1=0.D0
RSUM2=0.D0
DO J=1,MVAR
RSUM1=RSUM1+DABS(R(J))
RSUM2=RSUM2+DABS(R(J))**2
ENDDO
WRITE(9,*) 'CORRELATION OF INDEX WITH CONSTITUENT VARIABLES'
WRITE(9,*) (R(J),J=1,MVAR)
WRITE(9,*) 'SUM OF ABS (R)=' ,RSUM1,'; SUM OF SQUARE(R)=' ,RSUM2
WRITE(9,*) 'THE INDEX OR SCORE OF DIFFERENT CASES'
DO I=1,NOPT
IF(I.EQ.1) THEN
WRITE(9,*) '--- WELCOME TO DE/RPS PROGRAM FOR INDEX CONSTRUCTION'
CALL DE(M,X,FMINDE,Q0,Q1) !CALLS DE AND RETURNS OPTIMAL X AND FMIN
ENDIF
WRITE(9,*) 'DIFFERENTIAL EVALUATION OPTIMIZATION RESULTS'
ENDDO
WRITE(9,*) QIND(QJ)
ENDDO
END

DO J=1,M
XX(I,J)=X(J)
ENDDO
ENDDO
ENDDO
ENDDO
ENDDO
ENDDO
ENDDO
ENDDO
ENDIF
DO J=1,MVAR
I1=I+1
J1=J+1
IF(I1.EQ.J1) THEN
ENDIF
WRITE(9,*) 'THE INDEX OR SCORE OF DIFFERENT CASES'
DO II=1,NOB
IF((II.EQ.1)) THEN
DO J=1,MVAR
I1=I+1
J1=J+1
IF(I1.EQ.J1) THEN
ENDIF
WRITE(9,*) QIND(QJ)
ENDDO
ENDDO
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDED
WRITE(*,*) 'THE JOB IS OVER'
END
SUBROUTINE DE(M,A,FBEST,G0,G1)
PROGRAM: "DIFFERENTIAL EVOLUTION ALGORITHM" OF GLOBAL OPTIMIZATION
THIS METHOD WAS PROPOSED BY R. STORN AND K. PRICE IN 1995. REF --
"DIFFERENTIAL EVOLUTION - A SIMPLE AND EFFICIENT ADAPTIVE SCHEME
FOR GLOBAL OPTIMIZATION OVER CONTINUOUS SPACES" : TECHNICAL REPORT
PROGRAM BY SK MISHRA, DEPT. OF ECONOMICS, NEHU, SHILLONG (INDIA)

IMPLICIT DOUBLE PRECISION (A-H, O-Z) ! TYPE DECLARATION
PARAMETER(NMAX=500,MMAX=50) ! MAXIMUM DIMENSION PARAMETERS
PARAMETER(RX1=1.0D0, RX2=1.0D0) ! TO BE ADJUSTED SUITABLY, IF NEEDED
RX1 AND RX2 CONTROL THE SCHEME OF CROSSOVER. (0 <= RX1 <= RX2) <=1
RX1 DETERMINES THE UPPER LIMIT OF SCHEME 1 (AND LOWER LIMIT OF
SCHEME 2); RX2 IS THE UPPER LIMIT OF SCHEME 2 AND LOWER LIMIT OF
SCHEME 3. THUS RX1 = .2 AND RX2 = .8 MEANS 0-20% SCHEME1, 20 TO 80
PERCENT SCHEME 2 AND THE REST (80 TO 100 %) SCHEME 3.
PARAMETER(NCROSS=2) ! CROSS-OVER SCHEME (NCROSS <=0 OR =1 OR =>2)
PARAMETER(IPRINT=100, EPS=1.D-10) ! FOR WATCHING INTERMEDIATE RESULTS
IT Printer THE INTERMEDIATE RESULTS AFTER EACH IPRINT ITERATION AND
EPS DETERMINES ACCURACY FOR TERMINATION. IF EPS= 0, ALL ITERATIONS
WOULD BE UNIRED EVEN IF NO IMPROVEMENT IN RESULTS IS THERE.
ULTIMATELY "DID NOT CONVERGE" IS REPORTED.
COMMON /RNDM/IU,IV ! RANDOM NUMBER GENERATION (IU = 4-DIGIT SEED)
INTEGER IU,IV ! FOR RANDOM NUMBER GENERATION
COMMON /KF/KF,NFCALL,FTIT ! FUNCTION CODE, NO. OF CALLS * TITLE
COMMON /XBASE/XBAS

CHARACTER *70 FTIT ! TITLE OF THE FUNCTION

THE PROGRAM REQUIRES INPUTS FROM THE USER ON THE FOLLOWING -------
(1) FUNCTION CODE (KF), (2) NO. OF VARIABLES IN THE FUNCTION (M);
(3) N=POPULATION SIZE (SUGGESTED 10 TIMES OF NO. OF VARIABLES, M,
FOR SMALLER PROBLEMS N=100 WORKS VERY WELL);
(4) PCROS = PROB. OF CROSS-OVER (SUGGESTED : ABOUT 0.85 TO .99);
(5) FACT = SCALE (SUGGESTED 0.5 TO .95 OR 1, ETC);
(6) ITER = MAXIMUM NUMBER OF ITERATIONS PERMITTED (5000 OR MORE)
(7) RANDOM NUMBER SEED (4 DIGITS INTEGER)

DIMENSION X(NMAX,MMAX),Y(NMAX,MMAX),A(MMAX),FV(NMAX)
DIMENSION IR(3),XBASE(1000,50)

--------- SELECT THE FUNCTION TO MINIMIZE AND ITS DIMENSION -------
CALL FSSELECT(KF,M,FTIT)

SPECIFY OTHER PARAMETERS --------------------------------------
WRITE(*,*)'POPULATION SIZE [N] AND NO. OF ITERATIONS [ITER] ?'
WRITE(*,*)'SUGGESTED : N = 100 OR =10.M; ITER 10000 OR SO'
READ(*,*) N,ITER
WRITE(*,*)'CROSSOVER PROBABILITY [PCROS] AND SCALE [FACT] ?'
WRITE(*,*)'SUGGESTED : PCROS ABOUT 0.9; FACT=.5 OR LARGER BUT <=1'
READ(*,*) PCROS,FACT
WRITE(*,*)'RANDOM NUMBER SEED ?'
WRITE(*,*)'A FOUR-DIGIT POSITIVE ODD INTEGER, SAY, 1171'
READ(*,*) IU
NFCALL=0 ! INITIALIZE COUNTER FOR FUNCTION CALLS
GBEST=1.D30 ! TO BE USED FOR TERMINATION CRITERION
C
INITIALIZATION : GENERATE X(N,M) RANDOMLY
DO I=1,N
DO J=1,M
CALL RANDOM(RAND) ! GENERATES INITIATION X WITHIN
X(I,J)=(RAND-.5D0)*2000 ! GENERATES INITIATION X WITHIN
RANDOM NUMBERS BETWEEN -RANGE AND +RANGE (BOTH EXCLUSIVE)
X(I,J)=XBASE(I,J) ! TAKES THESE NUMBERS FROM THE MAIN PROGRAM
ENDDO
ENDDO
WRITE(*,*)'COMPUTING --- PLEASE WAIT '
IPCOUNT=0
DO 100 ITR=1,ITER  ! ITERATION BEGINS

---------------------------------------------------------------------

DO I=1,N
    DO J=1,M
        A(J)=X(I,J)
    ENDDO

CALL FUNC(A,M,F)

---------------------------------------------------------------------

DO 100 ITR=1,ITER  ! ITERATION BEGINS

---------------------------------------------------------------------

CALL GINI(FV,N,G0)

---------------------------------------------------------------------

FBEST=FV(1)

KB=1

DO IB=2,N
    IF (FV(IB).LT.FBEST) THEN
        FBEST=FV(IB)
        KB=IB
    ENDIF
ENDDO

BEST FITNESS VALUE = FBEST : INDIVIDUAL X(KB)

---------------------------------------------------------------------

DO I=1,N  ! I LOOP BEGINS

INITIALIZE CHILDREN IDENTICAL TO PARENTS; THEY WILL CHANGE LATER
    DO J=1,M
        Y(I,J)=X(I,J)
    ENDDO

---------------------------------------------------------------------

CALL RANDOM(RAND)

IRJ=INT(RAND*N)+1

DO 100 IRI=1,3  ! IRI LOOP BEGINS
    IR(IRI)=0
ENDDO

---------------------------------------------------------------------

DO IX=1,M
    IF (IR(IX).LE.0) THEN
        GOTO 20  ! IF NOT THEN REGENERATE
   ENDIF
ENDDO

THREE RANDOMLY CHOSEN INDIVIDUALS DIFFERENT FROM I AND DIFFERENT
FROM EACH OTHER ARE IR(1),IR(2) AND IR(3)

---------------------- SCHEME 1 ----------------------------------

NO CROSS OVER, ONLY REPLACEMENT THAT IS PROBABILISTIC

IF(NCROSS.LE.0) THEN
    DO J=1,M  ! J LOOP BEGINS
        IF(NCROSS.LE.0) THEN
            CALL RANDOM(RAND)
        endif
    ENDDO
C                     BETTER CHILDREN

C     PERSONAL LETTER TO THE AUTHOR (DATED OCTOBER 18, 2006)
C     CROSSOVER SCHEME (NEW) SUGGESTED BY KENNETH PRICE IN HIS
C     ESPECIALLY SUITABLE TO NON-DECOMPOSABLE (NON-SEPERABLE) FUNCTIONS
C     ------------------------ SCHEME 3 --------------------------------

C     PERSONAL LETTER TO THE AUTHOR (DATED SEPTEMBER 29, 2006)
C     CROSSOVER SCHEME (EXPONENTIAL) SUGGESTED BY KENNETH PRICE IN HIS
C     THE STANDARD CROSSOVER SCHEME
C     ----------------------- SCHEME 2 ---------------------------------

RANKOPT.f 6/12

336:     IF (RAND .LE. PCROS) THEN  ! REPLACE IF RAND < PCROS
337:         A(J) = X(IR(1),J) + (X(IR(2),J) - X(IR(3),J)) * FACT  ! CANDIDATE CHILD
338:     ENDIF
339:     ENDDO  ! J LOOP ENDS
340:     ENDIF
341:     C     ------------------------ SCHEME 2 --------------------------------
342:     C     THE STANDARD CROSSOVER SCHEME
343:     C     CROSSOVER SCHEME (NEW) SUGGESTED BY KENNETH PRICE IN HIS
344:     C     PERSONAL LETTER TO THE AUTHOR (DATED SEPTEMBER 29, 2006)
345:     C     ------------------------ SCHEME 3 --------------------------------
346:     IF (NCROSS .EQ. 1) THEN
347:         CALL RANDOM(RAND)
348:             JR = INT (RAND * M) + 1
349:             J = JR + 1
350:         IF (J .GT. M) J = 1
351:         IF (J .EQ. JR) GOTO 10
352:         CALL RANDOM(RAND)
353:         IF (PCROS .LE. RAND) GOTO 2
354:             6 A(J) = X(I,J)
355:             J = J + 1
356:         IF (J .GT. M) J = 1
357:         IF (J .EQ. JR) GOTO 10
358:     GOTO 6
359:     10 CONTINUE
360:     ENDIF
361:     C     ------------------------ SCHEME 3 --------------------------------
362:     C     ESPECIALLY SUITABLE TO NON-DECOMPOSABLE (NON-SEPERABLE) FUNCTIONS
363:     C     CROSSOVER SCHEME (NEW) SUGGESTED BY KENNETH PRICE IN HIS
364:     C     PERSONAL LETTER TO THE AUTHOR (DATED OCTOBER 18, 2006)
365:     C     ------------------------ SCHEME 3 --------------------------------
366:     IF (NCROSS .GE. 2) THEN
367:         CALL RANDOM(RAND)
368:             IF (RAND .LE. PCROS) THEN
369:                 CALL NORMAL(RN)
370:                     DO J = 1, M
371:                         A(J) = X(I,J) + (X(IR(1),J) + X(IR(2),J) - 2 * X(I,J)) * RN
372:                     ENDDO
373:                 ELSE
374:                     DO J = 1, M
375:                         A(J) = X(I,J) + (X(IR(1),J) + X(IR(2),J)) * FACT ASSUMED TO BE 1
376:                     ENDDO
377:             ENDDO
378:     ENDIF
379:     ENDIF
380:     C     ------------------------ SCHEME 3 --------------------------------
381:     C     EVALUATE THE OFFSPRING
382:     IF (F .LT. FV(I)) THEN  ! IF BETTER, REPLACE PARENTS BY THE CHILD
383:         FV(I) = F
384:         Y(I,J) = A(J)
385:     ENDDO
386:     ENDIF
387:     ENDDO  ! I LOOP ENDS
388:     DO I = 1, N
389:         DO J = 1, M
390:             X(I,J) = Y(I,J)  ! NEW GENERATION IS A MIX OF BETTER PARENTS AND
391:                     BETTER CHILDREN
392:     ENDDO
393:     ENDDO
394:     DO J = 1, M
395:         A(J) = X(KB,J)
396:     ENDDO
397:     IPCOUNT = IPCOUNT + 1
398:     IF (IPCOUNT .EQ. IPRINT) THEN
399:         WRITE (*) (X(KB,J), J = 1, M), ' FBEST UPTO NOW = ', FBEST
400:         WRITE (*) 'TOTAL NUMBER OF FUNCTION CALLS = ', NFCALL
401:         IF (DABS (FBEST - GBEST) .LT. EPS) THEN
403: WRITE(*,*)'FTIT
404: WRITE(*,*)'COMPUTATION OVER'
405: GO TO 999
406: ELSE
407: GBEST=FBEST
408: ENDIF
409: IPCOUNT=0
410: ENDIF
411: C 100 ENDDO ! ITERATION ENDS : GO FOR NEXT ITERATION, IF APPLICABLE
412: C-----------------------------------------------------------------
413: WRITE(*,*)'DID NOT CONVERGE. REDUCE EPS OR RAISE ITER OR DO BOTH'
414: WRITE(*,*)'INCREASE N, PCROS, OR SCALE FACTOR (FACT)'
415: 999 CALL FUNC(A,M,FBEST)
416: CALL GINI(FV,N,G1)
417: RETURN
418: END
419: C-----------------------------------------------------------------
420: SUBROUTINE NORMAL(R)
421: C PROGRAM TO GENERATE N(0,1) FROM RECTANGULAR RANDOM NUMBERS
422: C IT USES BOX-MULLER VARIATE TRANSFORMATION FOR THIS PURPOSE.
423: C-----------------------------------------------------------------
424: C ----- BOX-MULLER METHOD BY GEP BOX AND ME MULLER (1958) --------
425: C BOX, G. E. P. AND MULLER, M. E. "A NOTE ON THE GENERATION OF
427: C IF U1 AND U2 ARE UNIFORMLY DISTRIBUTED RANDOM NUMBERS (0,1),
428: C THEN X=([-2*LN(U1)]**.5)*(COS(2*PI*U2) IS N(0,1)
429: C ALSO, X=([-2*LN(U1)]**.5)*(SIN(2*PI*U2) IS N(0,1)
430: C PI = 4*ARCTAN(1.0)= 3.1415926535897932384626433832795
431: C 2*PI = 6.283185307179586476925286766559
432: C-----------------------------------------------------------------
433: IMPLICIT DOUBLE PRECISION (A-H,O-Z)
434: COMMON /RNDM/IU,IV
435: INTEGER IU,IV
436: C-----------------------------------------------------------------
437: CALL RANDOM(RAND) ! INVOKES RANDOM TO GENERATE UNIFORM RAND [0, 1]
438: CALL RANDOM(RAND) ! INVOKES RANDOM TO GENERATE UNIFORM RAND [0, 1]
439: CALL RANDOM(RAND) ! INVOKES RANDOM TO GENERATE UNIFORM RAND [0, 1]
440: CALL RANDOM(RAND) ! INVOKES RANDOM TO GENERATE UNIFORM RAND [0, 1]
441: CALL RANDOM(RAND) ! INVOKES RANDOM TO GENERATE UNIFORM RAND [0, 1]
442: R=DSQRT(-2.0*DLNG(U1))
443: R=R*DCOS(U2*6.283185307179586476925286766559)
444: R=R*DCOS(U2*6.283185307179586476925286766559)
445: RETURN
446: END
447: C-----------------------------------------------------------------
448: SUBROUTINE GINI(F,N,G)
449: PARAMETER (K=1) ! K=1 GINI COEFFICIENT; K=2 COEFFICIENT OF VARIATION
450: C THIS PROGRAM COMPUTES MEASURE OF INEQUALITY
451: C IF K =1 GET THE GINI COEFFICIENT. IF K=2 GET COEFF OF VARIATION
452: IMPLICIT DOUBLE PRECISION (A-H,O-Z)
453: DIMENSION F(*)
C     NOB=NO. OF OBSERVATIONS (CASES) & MVAR= NO. OF VARIABLES
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
C     ==chr==
C     ==chr==
C     TEST FUNCTIONS FOR GLOBAL OPTIMIZATION PROGRAM
C     -----------------------------------------------------------------
C     KF IS THE CODE OF THE TEST FUNCTION
C     TEST FUNCTIONS FOR GLOBAL OPTIMIZATION PROGRAM
C     -----------------------------------------------------------------
C     (1) FUNCTION CODE (KF), (2) NO. OF VARIABLES IN THE FUNCTION (M);
C     THE PROGRAM REQUIRES INPUTS FROM THE USER ON THE FOLLOWING -----
C     -----------------------------------------------------------------
C     G=H/DABS(S) !IF S NOT ZERO, K=1 THEN G=GINI, K=2 G=COEFF VARIATION
C                             FOR K=2 H IS STANDARD DEVIATION

RANKOPT.f 8/12

DIMENSION CHARACTER*70 FTIT(100),FTIT

SUBROUTINE FSELECT(KF,M,FTIT)

THE PROGRAM REQUIRES INPUTS FROM THE USER ON THE FOLLOWING ------

(1) FUNCTION CODE (KF), (2) NO. OF VARIABLES IN THE FUNCTION (M);

CHARACTER *70 TIT(100),FTIT

DATA TIT(1)/'CONSTRUCTION OF INDEX FROM M VARIABLES '/

SUBROUTINE FUNC(X,M,F)

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

COMMON /RNDM/IV,IU

COMMON /KFF/KF,NFCALL,FTIT

INTEGER IU,IV

DIMENSION X(*)

CHARACTER *70 FTIT

NFCALL=NFCALL+1 ! INCREMENT TO NUMBER OF FUNCTION CALLS

IF(KF.EQ.1) THEN

CALL CORD(M,X,F)

RETURN

ENDIF

PRINT *,'FUNCTION NOT DEFINED. PROGRAM ABORTED'

STOP

END

SUBROUTINE CORD(M,X,F)

PARAMETER (NOB=30,MVAR=7) ! CHANGE THE PARAMETERS HERE AS NEEDED.

COMMON /CORDAT/CDAT(NOB,MVAR),QIND(NOB),R(MVAR),ENTROPY,NORM,NCOR

COMMON /GETRANK/MRnk

INTEGER IU,IV

DIMENSION X(*),Z(NOB,2)

SUBROUTINE END
DO I=1,M
   IF(X(I).LT.-1.0D0.OR.X(I).GT.1.0D0) THEN
      CALL RANDOM(RAND)
      X(I)=(RAND-0.5D0)*2
   ENDIF
ENDDO
XNORM=0.D0
DO J=1,M
   XNORM=XNORM+X(J)**2
ENDDO
XNORM=DSQRT(XNORM)
DO J=1,NOB
   X(J)=X(J)/XNORM
ENDDO
C     CONSTRUCT INDEX
DO I=1,NOB
   QIND(I)=0.D0
   DO J=1,M
      QIND(I)=QIND(I)+CDAT(I,J)*X(J)
   ENDDO
ENDDO
C     ------------------------------------------------------------------
!FIND THE RANK OF QIND
IF(MRNK.EQ.1) CALL DORANK(QIND,NOB)
C     ------------------------------------------------------------------
C     COMPUTE CORRELATIONS
DO I=1,NOB
   Z(I,1)=QIND(I)
ENDDO
DO J=1,M
   DO I=1,NOB
      Z(I,2)=CDAT(I,J)
   ENDDO
IF(NCOR.EQ.0) THEN
   CALL CORLN(Z,NOB,RHO)
ELSE
   CALL CORA(Z,NOB,RHO)
ENDIF
R(J)=RHO
ENDDO
C     MAXIMIN SOLUTION
IF(ENTROPY.EQ.0.D0) THEN
   ENT=0.0D0
   DO J=1,M
      ENT=ENT+DABS(R(J))
   ENDDO
   F=ENT*DLOG(ENT)
   DO J=1,M
      FX=DABS(R(J))
      F=F+FX/(FX/ENT)*DLOG(FX/ENT)
   ENDDO
ELSE
   IF(ENTROPY.LE.0.D0) THEN
      F=0.D0
      DO J=1,M
         F=F+DABS(R(J))**NORM
      ENDDO
      ENDIF
   ELSE
      IF(ENTROPY.NE.0.D0) THEN
         ENT=0.0D0
         DO J=1,M
            ENT=ENT+DABS(R(J))
         ENDDO
         F=ENT*DLOG(ENT)
         DO J=1,M
            FX=DABS(R(J))
            F=F+(FX/ENT)*DLOG(FX/ENT)
         ENDDO
      ENDIF
C     SUBTRACT RESPECTIVE MEDIANS FROM X AND Y AND FIND ABS DEVIATIONS
C     FIND MEDIAN
C     ARRANGE X ANY IN AN ASCENDING ORDER
C     COMPUTING ABSOLUTE CORRELATION MATRIX
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C     DIMENSION Z(NOB,2),AV(2),SD(2)
DO J=1,2
   AV(J)=0.D0
   SD(J)=0.D0
   DO I=1,NOB
      AV(J)=AV(J)+Z(I,J)
   END DO
   SD(J)=SD(J)+Z(I,J)**2
END DO
WRITE(*,*)'AV AND SD ',AV(1),AV(2),SD(1),SD(2)
RHO=0.D0
DO I=1,NOB
   RHO=RHO+(Z(I,1)-AV(1))*(Z(I,2)-AV(2))
END DO
RHO=(RHO/NOB)/(SD(1)*SD(2))
RETURN
END

C     SUBROUTINE CORA(Z,N,R)
C     COMPUTING ABSOLUTE CORRELATION MATRIX
C     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C     DIMENSION Z(N,2),X(N),Y(N)
C     PUT Z INTO X AND Y
DO I=1,N
   X(I)=Z(I,1)
   Y(I)=Z(I,2)
END DO
C     ARRANGE X ANY IN AN ASCENDING ORDER
DO I=1,N-1
   DO II=I+1,N
      IF(X(I).GT.X(II)) THEN
         TEMP=X(I)
         X(I)=X(II)
         X(II)=TEMP
      END IF
   END DO
   IF(Y(I).GT.Y(II)) THEN
      TEMP=Y(I)
      Y(I)=Y(II)
      Y(II)=TEMP
   END IF
END DO
C     FIND MEDIAN
IF(INT(N/2).EQ.N/2.D0) THEN
   XMED=(X(N/2)+X(N/2+1))/2.D0
ENDIF
IF(INT(N/2).NE.N/2.D0) THEN
   XMED=X(N/2+1)
ENDIF
YMED=Y(N/2+1)
C     SUBTRACT RESPECTIVE MEDIAN FROM X AND Y AND FIND ABS DEVIATIONS
671: VX=0.D0
672: VY=0.D0
673: DO I=1,N
674: X(I)=X(I)-XMED
675: Y(I)=Y(I)-YMED
676: VX=VX+DABS(X(I))
677: VY=VY+DABS(Y(I))
678: ENDDO
679: C SCALE THE VARIABLES X AND Y SUCH THAT VX=VY
680: IF(VX.EQ.0.D0.OR.VY.EQ.0.D0) THEN
681: R=0.D0
682: RETURN
683: ENDIF
684: DO I=1,N
685: X(I)=X(I)*VY/VX
686: ENDDO
687: C COMPUTE CORRELATION COEFFICIENT
688: VZ=0.D0
689: R=0.D0
690: DO I=1,N-1
691: DO II=I+1,N
692: IF(X(I).GT.X(II)) THEN ! ARRANGE ACCORDING TO ASCENDING ORDER OF X
693: T=X(I)
694: X(I)=X(II)
695: X(II)=T
696: IT=ID(I)
697: ID(I)=ID(II)
698: ID(II)=IT
699: ENDF
700: ENDDO
701: ENDDO
702: DO I=1,N
703: X(I)=I+0.D0
704: ENDDO
705: DO I=1,N-1
706: DO II=I+1,N
707: IF(ID(I).GT.ID(II)) THEN ! ARRANGE ACCORDING TO ASCENDING ORDER OF ID
708: T=X(I)
709: X(I)=X(II)
710: X(II)=T
711: IT=ID(I)
712: ID(I)=ID(II)
713: ID(II)=IT
714: ENDF
715: ENDDO
716: ENDDO
717: C ------------------------------------------------------------------
718: SUBROUTINE DORANK(X,N) ! N IS THE NUMBER OF OBSERVATIONS
719: PARAMETER (MXD=1000) ! MXD IS MAX DIMENSION FOR TEMPORARY VARIABLES
720: ! THAT ARE LOCAL AND DO NOT GO TO THE INVOKING PROGRAM
721: ! X IS THE VARIABLE TO BE SUBSTITUTED BY ITS RANK VALUES
722: IMPLICIT DOUBLE PRECISION (A-H,O-Z)
723: DIMENSION X(N),ID(MXD)
724: ! GENERATE ID
725: DO I=1,N
726: ID(I)=I
727: ENDDO
728: DO I=1,N-1
729: DO II=I+1,N
730: IF(X(I).GT.X(II)) THEN ! ARRANGE ACCORDING TO ASCENDING ORDER OF X
731: T=X(I)
732: X(I)=X(II)
733: X(II)=T
734: IT=ID(I)
735: ID(I)=ID(II)
736: ID(II)=IT
737: ENDF
738: ENDDO
739: ENDDO
SUBROUTINE CORREL(X,N,M,RMAT)

PARAMETER (NMX=30) !DO NOT CHANGE UNLESS NO. OF VARIABLES EXCEED 30

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

DIMENSION X(N,M),RMAT(M,M),AV(NMX),SD(NMX)

DO J=1,M
AV(J)=0.D0
SD(J)=0.D0
DO I=1,N
AV(J)=AV(J)+X(I,J)
SD(J)=SD(J)+X(I,J)**2
ENDDO
AV(J)=AV(J)/N
SD(J)=DSQRT(SD(J)/N-AV(J)**2)
ENDDO
DO J=1,M
DO JJ=1,M
RMAT(J,JJ)=0.D0
DO I=1,N
RMAT(J,JJ)=RMAT(J,JJ)+X(I,J)*X(I,JJ)
ENDDO
ENDDO
DO J=1,M
DO JJ=1,M
RMAT(J,JJ)=RMAT(J,JJ)/N-AV(J)*AV(JJ)
RMAT(J,JJ)=RMAT(J,JJ)/(SD(J)*SD(JJ))
ENDDO
ENDDO
RETURN
END