Construction of Pena's DP2-based ordinal synthetic indicator when partial indicators are rank scores

Mishra, SK

North-Eastern Hill University, Shillong (India)

28 May 2012

Construction of Pena's DP2-Based Ordinal Synthetic Indicator When Partial Indicators are Rank Scores

SK Mishra
Dept. of Economics
North-Eastern Hill University
Shillong, Meghalaya (India)
Contact: mishrasknehu@yahoo.com

I. Introduction: A synthetic indicator (or composite index), $Z[n]$, is oftentimes a linear combination of m partial indicators (or the constituent variables of Z), $X[n,m]$, such that $Z = Xw$, where $w[m]$ is the weight vector. In many cases, w is extraneously determined (e.g. on the basis of expert opinion), but applications abound when w is determined intrinsically from X itself. Factor Analysis or Principal Component Analysis is one of the most popular methods of obtaining Z (which is called the factor scores). However it has been found that the factor scores thus obtained are highly elitist; often ignoring the poorly correlated partial indicators (OECD, 2003; Munda & Nardo, 2005; Somarriba & Pena, 2009; Mishra, 2011).

In response to the limitations of the Principal Component (or Factor) Analysis as a tool to construct the synthetic indicators, a new method based on Pena's DP2 distance (Pena, 1977) was devised (Zarzosa, 1996; Somarriba & Pena, 2009). Pena's DP2 is defined as:

$$DP2_i = \sum_{j=1}^{m}\left[\left(\partial_{ij}\right)\left(1-R^2_{j,j-1,...,1}\right)\right] = \sum\nolimits_{j=1}^{m}\partial_{ij}w_j; \quad i=1,2,...,n \qquad ... \qquad (1)$$

where: $i=1,2,...,n$ are cases (e.g. countries, districts, institutions, individuals, items and so on); m is the number of constituent variables (partial indicators), X, such that $x_{ij} \in X; i=1,2,...,n; j=1,2,...,m;$ $d_{ij} = \left|x_{ij} - x_{\kappa j}\right|; i=1,2,...,n; j=1,2,...,m;$ κ is the reference case pertaining to $\min_i (x_{ij})$; $\partial_{ij} = d_{ij}/\sigma_j$; σ_j is the standard deviation of variable j; $R^2_{j,j-1,...,1}$; $j>1$ is the coefficient of determination in the regression of x_j on $x_{j-1}, x_{j-2},..., x_1$. Moreover, $R_1^2 = 0$. It may be noted that the first variable obtains an absolute weight of unity: $\left(1-R_1^2\right)=1$ for $R_1^2=0$. The subsequent variable $j=2$ obtains a weight $\left(1-R^2_{2,1}\right)$ and in general, the j^{th} variable obtains a weight of $\left(1-R^2_{j,j-1,...,1}\right)$. Thus, the DP2 method of constructing synthetic indicator differs from the more conventional methods regarding the manner it assigns weight to the constituent variables. For example, the Principal Component Analysis determines the weights (w) in $Z = Xw$ such that $\sum\nolimits_{j=1}^{m}[r(Z,x_j)]^2$ is maximized. For the DP2 method, there is no such clearly defined objective function to optimize. It is also obvious that the weights assigned to a variable will depend on its position in the order (Montero, 2010), which makes DP2-based composite (synthetic) indices indeterminate and arbitrary unless some criterion is used for the entry of variables in the formula or some objective function is set to be optimized.

There could be several procedures to resolve the indeterminacy pointed out above. One among them is the following iterative procedure (Montero, 2010; Nayak & Mishra, 2012):

1: Define $\varepsilon = 0.001$, for accuracy. Initialize the weight vector, $w_j = 1$; $j = 1, 2, ..., m.$

2: Obtain $DF_i = \sum_{j=1}^{m} \left[\partial_{ij} w_j \right]; \quad i = 1, 2, ..., n$

3: Compute the coefficient of correlation, $r(DF, \partial_j)$, between DF and ∂_j ; $j = 1, 2, ..., m$

4: Arrange $| r(DF, \partial_j) |$ in a descending order. This is the criterion on which the variables would enter into the DP2 formula and obtain weights.

5: Compute $Z_i = \sum_{j=1}^{m} \left[\partial_{ij} w_j \right]; \quad i = 1, 2, ..., n; \quad w_j = (1 - R^2_{j, j-1, j-2, ..., 1})$ for $j = 2, 3, ..., m$ and $w_1 = 1$.

6: If $\sum_{i=1}^{n} | DF_i - Z_i |^2 \geq \varepsilon$ then: $DF \Leftarrow Z$, go to step-3. Otherwise: stop.

II. The Cases when Partial Indicators are Ordinal (Ranking Scores): When every partial indicator, $x_j \in X$, is ordinal (or ranking scores), then obtaining weights (w) and the construction of ordinal composite index (synthetic indicator), Z, require a special treatment. First, transformation of x_{ij} into ∂_{ij} is neither legitimate nor required. Secondly, regression of x_j on $x_{j-1}, x_{j-2, ...,} x_1$ in the usual manner applicable to cardinally measured variables cannot be carried out. Thirdly, $Z = Xw$ must be ordinal (ranking scores). Fourthly, an appropriate measure of correlation (i.e. non-parametric) has to be used since at every iteration $r(Z, x_j)$ is computed. Fifthly, the condition of convergence would be different since any deviation of $| DF_i - Z_i |$ would be either zero or a non-zero integer, not a small number, ε. Finally, there are several types of ranking (such as ordinal, dense, standard competition, modified competition, etc) and a choice must be made regarding the same.

III. Objectives of the Present Work: The present study aims at formulating a computational scheme (and developing a computer program) that may be appropriate to construct Pena's DP2 (ordinal) synthetic indicator (Z) from the partial indicators (X) all of which are ordinal (ranking scores). Additionally, an attempt will be made to empirically apply the method (and the computer program) to obtain the ordinal synthetic indicator.

IV. The Computational Method: The crux of the computational work lies in obtaining $R^2_{j, j-1, ..., 1.}$ For this, we directly maximize the squared correlation between x_j (which is ordinal) and $\hat{x}_{ij} = ord_i(x_{ij-1}v_{j-1} + x_{ij-2}v_{j-2} + ... + x_{i1}v_1)$ or the expected x_j, which is also ordinal, with $v_1, ..., v_{j-1}$ as the decision variables (in the range -γ to γ, where γ could be unity). The function $ord_i(.)$ (running over cases, $i = 1, 2, ..., n$) converts (.) into the ranking scores. For the purpose of maximization of $R^2_{j, j-1, ..., 1.}$ we use the Differential Evolution method of global optimization, which is well known for its efficiency (Mishra, 2006).

As for computation of correlation coefficient, Karl Pearson's formula for computing correlation may be used to obtain Spearman's rank correlation coefficient. Alternatively, Kendall's tau may be used. For ranking, any one of the five possible schemes (ordinal or 1-2-3-4 rule, dense or 1-2-2-3 rule, standard competition or 1-2-2-4 rule, modified competition or 1-3-3-4 rule, fractional ranking or 1-2.5-2.5-4 rule) may be used. It may be noted, however, that the difference in ranking scores obtained by the different rules arises in case of ties only. For termination of iteration we have used $\varepsilon = 1$. This implies that the current Z (i.e. $Z^{(t)}$) and the previous Z (i.e. $Z^{(t-1)}$) are identical.

V. The Data: For the purpose of empirical testing of the proposed method, we have used the data from Mishra (2010), Example-1, Table-1.1. The data are on 30 items each with ranking score in 7 dimensions (i.e., 7 partial indicators). The data are reproduced here (Table-1) for a ready reference. In the table, Z_0 is the composite ranking score (synthetic indicator) obtained from 7-dimensional partial indicators by applying the Ordinal Principal Component Analysis.

Table-1. Ranking Scores of Thirty individuals in Seven Dimensions and the Composite Ranking Score obtained by the Ordinal Principal Component Analysis.

SL	x_1	x_2	x_3	x_4	x_5	x_6	x_7	Z_0	SL	x_1	x_2	x_3	x_4	x_5	x_6	x_7	Z_0
1	2	3	1	2	4	12	4	1	16	16	17	11	16	14	20	19	16
2	6	2	2	4	7	3	3	2	17	15	23	16	27	10	8	14	17
3	1	10	3	1	1	6	8	3	18	30	20	23	19	6	11	6	18
4	14	1	5	3	2	1	9	4	19	5	21	26	23	23	9	15	19
5	4	9	12	14	11	5	1	5	20	22	15	21	20	17	19	13	20
6	10	7	6	7	9	22	5	6	21	20	28	8	13	25	21	23	21
7	3	8	13	8	3	13	17	7	22	11	19	18	26	20	23	26	22
8	9	14	17	5	18	2	2	8	23	25	12	22	22	19	30	21	23
9	7	11	4	10	24	4	10	9	24	18	25	30	21	16	18	24	24
10	19	6	19	6	5	10	7	10	25	23	29	15	18	30	17	29	25
11	8	16	9	15	15	27	12	11	26	28	18	20	25	27	15	30	26
12	17	5	7	17	8	26	20	12	27	24	26	27	28	13	29	25	27
13	13	13	10	11	28	7	22	13	28	26	27	28	24	29	28	18	28
14	12	30	14	12	12	14	11	14	29	27	22	24	29	26	25	27	29
15	21	4	25	9	22	16	16	15	30	29	24	29	30	21	24	28	30

Source: Mishra (2010), pp. 175-77. Z_0 = Composite Index by Ordinal Principal Component Analysis.

VI. The Findings: Presented in Table-2, all the composite ranking scores (or ordinal synthetic indicators), $Z_{\alpha\beta}$, are constructed by using the procedure mentioned in section-IV. The subscripts α (α =1 for Perason/Spearman coefficient of correlation and α = 2 for Kendall's tau) and β (β =0 for 1-2-3-4 rule, β =1 for 1-2-2-3 rule, β = 2 for 1-2-2-4 rule, β = 3 for 1-3-3-4 rule and β = 4 for 1-2.5-2.5-4 rule) specify as to the type of correlation and the rule for rank-ordering. In particular, Z_u is the un-weighted (or equally weighted) ordinal synthetic index (rank scores) obtained by $ord_i(\sum_{j=1}^{7} x_{ij})$ and Z_0 is the ordinal composite index obtained by the Ordinal Principal Component Analysis. Presented in Table-3, all relative weights correspond to the synthetic indicators ($Z_{\alpha\beta}$, Z_u and Z_0) in Table-2. The relative weights sum up to unity. The last two columns of Table-3 report the appropriate (α = 1 or 2) coefficient of correlation of $Z_{\alpha\beta}$ with the equally weighted ordinal composite index (Z_u) and the Ordinal Principal Component index.

Table-2. Composite Ranking Score obtained by Different Correlation & Ranking Rules												
SL	Z_{10}	Z_{20}	Z_{11}	Z_{21}	Z_{12}	Z_{22}	Z_{13}	Z_{23}	Z_{14}	Z_{24}	Z_0	Z_u
1	1	2	1	2	1	2	1	2	1	2	1	2
2	3	1	3	1	3	1	3	1	3	1	2	1
3	2	3	2	3	2	3	2	3	2	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4
5	6	5	6	5	6	5	5	5	6	5	5	5
6	8	7	8	7	8	7	8	7	8	7	6	7
7	7	8	7	8	7	8	7	8	7	8	7	6
8	5	6	5	6	5	6	6	6	5	6	8	8
9	10	10	10	10	10	10	10	10	10	10	9	9
10	9	9	9	9	9	9	9	9	9	9	10	10
11	13	13	13	13	13	13	13	13	13	13	11	12
12	15	11	15	11	15	11	15	11	15	11	12	11
13	11	12	11	12	11	12	11	12	11	12	13	13
14	12	14	12	14	12	14	12	14	12	14	14	14
15	14	15	14	15	14	15	14	15	14	15	15	15
16	16	16	16	16	16	16	16	16	16	16	16	16
17	18	18	18	18	18	18	18	18	18	18	17	17
18	17	17	17	17	17	17	17	17	17	17	18	18
19	19	19	19	19	19	19	19	19	19	19	19	19
20	20	20	20	20	20	20	20	20	20	20	20	20
21	21	21	21	21	21	21	21	21	21	21	21	21
22	22	22	22	22	22	22	22	22	22	22	22	22
23	24	23	24	23	24	23	24	23	24	23	23	23
24	23	24	23	24	23	24	23	24	23	24	24	24
25	25	25	25	25	25	25	25	25	25	25	25	25
26	26	26	26	26	26	26	26	26	26	26	26	26
27	27	27	27	27	27	27	27	27	27	27	27	27
28	28	28	28	28	28	28	28	28	28	28	28	28
29	29	29	29	29	29	29	29	29	29	29	29	29
30	30	30	30	30	30	30	30	30	30	30	30	30

.

Table-3. Relative Weights of Partial Indicators obtained under Different Correlations and Ranking Rules									
var	Relative Weights assigned to the Partial Indicators							Correlation Coefficient	
w	x_1	x_2	x_3	x_4	x_5	x_6	x_7	$r_\alpha(Z_u, Z_{\alpha\beta})$	$r_\alpha(Z_0, Z_{\alpha\beta})$
w_{10}	0.13561	0.12508	0.09042	0.26405	0.12457	0.12863	0.13165	0.98888	0.99021
w_{20}	0.12355	0.13477	0.13574	0.19304	0.13477	0.13860	0.13954	0.97241	0.96782
w_{11}	0.13557	0.12247	0.09039	0.26397	0.12657	0.12943	0.13160	0.98563	0.99021
w_{21}	0.12378	0.13405	0.13600	0.19341	0.13503	0.13792	0.13981	0.95402	0.96782
w_{12}	0.13596	0.12283	0.09066	0.26474	0.12369	0.13015	0.13199	0.98671	0.99021
w_{22}	0.12237	0.13562	0.13562	0.19288	0.13465	0.13942	0.13942	0.95402	0.96782
w_{13}	0.13533	0.12482	0.09024	0.26351	0.12652	0.12820	0.13138	0.98820	0.99155
w_{23}	0.12249	0.13381	0.13576	0.19307	0.13576	0.13956	0.13956	0.95402	0.96782
w_{14}	0.13575	0.12367	0.09051	0.26432	0.12487	0.12910	0.13178	0.98742	0.99021
w_{24}	0.12331	0.13451	0.13548	0.19267	0.13548	0.13928	0.13928	0.95402	0.96782
w_0	0.15870	0.13538	0.14931	0.21889	0.11524	0.09786	0.12463	-	-
w_u	0.14286	0.14286	0.14286	0.14286	0.14286	0.14286	0.14286	-	-

.

Table-4. Spearman's Correlation among Different Ordinal Composite Scores and the Partial Indicators												
$\frac{Z}{Z,x}$	Z_{10}	Z_{20}	Z_{11}	Z_{21}	Z_{12}	Z_{22}	Z_{13}	Z_{23}	Z_{14}	Z_{24}	Z_0	Z_u
Z_{10}	1.0000	0.9924	1.0000	0.9924	1.0000	0.9924	0.9996	0.9924	1.0000	0.9924	0.9902	0.9889
Z_{20}	0.9924	1.0000	0.9924	1.0000	0.9924	1.0000	0.9929	1.0000	0.9924	1.0000	0.9960	0.9969
Z_{11}	1.0000	0.9924	1.0000	0.9924	1.0000	0.9924	0.9996	0.9924	1.0000	0.9924	0.9902	0.9889
Z_{21}	0.9924	1.0000	0.9924	1.0000	0.9924	1.0000	0.9929	1.0000	0.9924	1.0000	0.9960	0.9969
Z_{12}	1.0000	0.9924	1.0000	0.9924	1.0000	0.9924	0.9996	0.9924	1.0000	0.9924	0.9902	0.9889
Z_{22}	0.9924	1.0000	0.9924	1.0000	0.9924	1.0000	0.9929	1.0000	0.9924	1.0000	0.9960	0.9969
Z_{13}	0.9996	0.9929	0.9996	0.9929	0.9996	0.9929	1.0000	0.9929	0.9996	0.9929	0.9915	0.9902
Z_{23}	0.9924	1.0000	0.9924	1.0000	0.9924	1.0000	0.9929	1.0000	0.9924	1.0000	0.9960	0.9969
Z_{14}	1.0000	0.9924	1.0000	0.9924	1.0000	0.9924	0.9996	0.9924	1.0000	0.9924	0.9902	0.9889
Z_{24}	0.9924	1.0000	0.9924	1.0000	0.9924	1.0000	0.9929	1.0000	0.9924	1.0000	0.9960	0.9969
Z_0	0.9902	0.9960	0.9902	0.9960	0.9902	0.9960	0.9915	0.9960	0.9902	0.9960	1.0000	0.9987
Z_u	0.9889	0.9969	0.9889	0.9969	0.9889	0.9969	0.9902	0.9969	0.9889	0.9969	0.9987	1.0000
x_1	0.8011	0.7887	0.8011	0.7887	0.8011	0.7887	0.8033	0.7887	0.8011	0.7887	0.8109	0.8082
x_2	0.7433	0.7811	0.7433	0.7811	0.7433	0.7811	0.7455	0.7811	0.7433	0.7811	0.7766	0.7815
x_3	0.7597	0.7842	0.7597	0.7842	0.7597	0.7842	0.7620	0.7842	0.7597	0.7842	0.8007	0.7980
x_4	0.9181	0.9012	0.9181	0.9012	0.9181	0.9012	0.9141	0.9012	0.9181	0.9012	0.8937	0.8914
x_5	0.6707	0.6845	0.6707	0.6845	0.6707	0.6845	0.6739	0.6845	0.6707	0.6845	0.6890	0.6934
x_6	0.7059	0.6770	0.7059	0.6770	0.7059	0.6770	0.7046	0.6770	0.7059	0.6770	0.6538	0.6623
x_7	0.8323	0.8331	0.8323	0.8331	0.8323	0.8331	0.8327	0.8331	0.8323	0.8331	0.8278	0.8194
SAR	5.4311	5.4498	5.4311	5.4498	5.4311	5.4498	5.4361	5.4498	5.4311	5.4498	5.4525	5.4542
SSR	4.2552	4.2802	4.2552	4.2802	4.2552	4.2802	4.2613	4.2802	4.2552	4.2802	4.2879	4.2862

It is interesting to note that in comparison to the Pearson/Spearman-based $(\alpha=1)$ composite rank scores, the Kendall's tau-based $(\alpha=2)$ composite rank scores have relatively weaker correlation with Z_u and Z_0.

In Table-4 we present the rank correlation (coefficients of) $Z_{\alpha\beta}$, Z_0 and Z_u among themselves and with the partial indicators $x_j \in X$; $j=1,2,...,m$. We also present the SAR (sum of absolute correlation, $\sum_{j=1}^{m}|r(Z,x_j)|$, for the relevant Z) and SSR (sum of squared correlation, $\sum_{j=1}^{m}|r(Z,x_j)|^2$, for the relevant Z). We find that the SAR as well as SSR of Z_0 and Z_u are larger than those of $Z_{\alpha\beta}$. This implies that from the viewpoint of overall representation of the partial indicators (ranking scores) by the composite indicator (composite ranking scores), Z_0 and

Z_u outperform $Z_{\alpha\beta}$ for α and β whatsoever. Nevertheless, the weights associated with $Z_{\alpha\beta}$ are more egalitarian.

V. Concluding Remarks: The findings of this study attract our attention to a number of points. First, it is computationally possible to extend the method of constructing Pena's DP2 synthetic indicators to the cases when all the partial indicators are ordinal (rank scores) and the synthetic rank scores are to be obtained from them. Secondly, in computing such synthetic rank scores, it is possible to use rank correlation (of Spearman or Kendall). It is also possible to use different schemes or rules of ranking. Thirdly, DP2-based composite ranking scores may not have better SAR and SSR than the other composite scores obtained by simple averaging or using the Ordinal Principal Component Analysis (OPCA). Fourthly, DP2-based composite scores have more egalitarian distribution of weights than the OPCA-based composite scores may have.

References:

Mishra, S.K. (2006) "Global Optimization by Differential Evolution and Particle Swarm Methods Evaluation on Some Benchmark Functions", MPRA, Munich. http://mpra.ub.uni-muenchen.de/1005/

Mishra, S.K. (2010) "The Most Representative Composite Rank Ordering of Multi-Attribute Objects by the Particle Swarm Optimization", *Journal of Quantitative Economics*, 8(2): 165-200. http://www.jqe.co.in/journals/JQE_v8_n2_2010_p11.pdf

Mishra, S.K. (2011) "A Comparative Study of Various Inclusive Indices and the Index Constructed by the Principal Component Analysis", *IUP Journal of Computational Mathematics*, 4(2): 7-26.

Mishra, S.K. (2012-a) "A Note on Construction of Heuristically Optimal Pena's Synthetic Indicators by the Particle Swarm Method of Global Optimization", SSRN: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2028395

Mishra, S.K. (2012-b) "A Maximum Entropy Perspective of Pena's Synthetic Indicators", http://mpra.ub.uni-muenchen.de/37797

Montero, J.M., Chasco, C. and Larraz, B. (2010) "Building An Environmental Quality Index for A Big City: A Spatial Interpolation Approach Combined with A Distance Indicator", *Journal of Geographical Systems*, 12(4): 435-459.

Munda, G. and Nardo, M. (2005) "Constructing Consistent Composite Indicators: The Issue of Weights", *EUR 21834 EN*, Institute for the Protection and Security of the citizen, European Commission, Luxembourg.

Nayak, P. and Mishra, S.K. (2012) "Efficiency of Pena's P2 Distance in Construction of Human Development Indices", SSRN: http://ssrn.com/abstract=2066567

OECD (2003) "Composite Indicators of Country Performance: A Critical Assessment", DST/IND(2003) 5, Paris.

Pena, J.B. (1977) *Problemas de la medición del bienestar y conceptos afines* (Una aplicación al caso español). (Madrid: Instituto Nacional de Estadística (INE)).

Somarriba, N. and Pena, B. (2009) "Synthetic Indicators of Quality of Life in Europe", *Social Indicators Research*, 94(1): 115–133.

Zarzosa, P. (1996) *Aproximación a la medición del bienestar social*. Secretariado de Publicaciones e intercambio científico, Universidad de Valladolid.

Appendix

The following FORTRAN 77 Program (ORDP2) computes the composite (synthetic) rank scores from m (>1) given partial rank scores (Z[n,m]), i=1,2, …, n and j=1,2, …, m each of which is a rank score obtained by the individual (case) in a particular dimension. There are n individuals.

Data is fed in an input file (filename.txt) which contains n rows (cases or individuals) in:

(a) m columns – if there is no reference index (refi) to compare. In that case the program sums the given m rank scores, orders them and ranks this total to make refi.

(b) m + 1 columns – if there is an extraneous reference index, (in the last m+1 column)

The program asks to choose certain parameters (regarding type of correlation, ranking rule, random number seed), and information regarding input and output files, etc which are self explanatory.

The program (codes/exe files) and the input data file must be in the same folder/directory. Output file also is stored in the same directory. The FORCE Fortran 77 compiler may be used to compile the codes. FORCE is free downloadable Fortran 77 compiler.

```
      PROGRAM ORDP2
      !-------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NX=1000,MX=50, EPS=1.D0)
      CHARACTER *40 INFIL,OUTFIL
      INTEGER IU, IV
      COMMON /RNDM/IU,IV ! RANDOM NUMBER GENERATION (IU = 4-DIGIT SEED)
      COMMON /PREGRESS/YP(NX),XP(NX,MX),NCASE,MVAR,RSQ
      COMMON /CORDAT/CDAT(NX,MX),QIND(NX),R(MX),NCOR
      COMMON /CORLNO/ZF(NX,2)
      COMMON /RANKS/NRL
      DIMENSION Z(NX,MX),Y(NX),X(NX,MX),CIND(NX),ZB(MX),SD(MX),RZ(MX)
      DIMENSION ZMIN(MX),ZMAX(MX),DF(NX),DFF(NX),YH(NX),IR(MX),W(MX)
      DIMENSION REFI(NX),DFZ(NX)
      ! REFI IS THE REFERENCE INDEX, IF ANY. IF THERE IS NONE, THEN THIS
      ! PROGRAM GENERATES IT AS RANK SCORE OF TOTAL PARTIAL RANKS-SCORES
      !=============================================================
      WRITE(*,*)'=========== DP2 SYNTHETIC INDICES OF RANK DATA ====='
      WRITE(*,*)'---ALGORITHM AND PROGRAM BY PROF.SK MISHRA (NEHU) ===='
      WRITE(*,*)
      WRITE(*,*)'NO. OF CASES AND VARIABLES ?'
      READ(*,*) N,M
      NCASE=N
      MVAR=M
      WRITE(*,*)'CORRELATION TYPE ?'
      WRITE(*,*)'CORRELATION : PEARSON/SPEARMAN RHO (1), KENDALL TAU(2)'
      READ(*,*) NCOR
C     NRL=0 FOR ORDINAL RANKING (1-2-3-4 RULE);
C     NRL=1 FOR DENSE RANKING (1-2-2-3 RULE);
C     NRL=2 FOR STANDARD COMPETITION RANKING (1-2-2-4 RULE);
C     NRL=3 FOR MODIFIED COMPETITION RANKING (1-3-3-4 RULE);
C     NRL=4 FOR FRACTIONAL RANKING (1-2.5-2.5-4 RULE);
      WRITE(*,*)'WHAT WILL BE THE RANKING RULE ?'
      WRITE(*,*)'NRL =0 (1,2,3,4); NRL=1 (1,2,2,3); NRL=2 (1,2,2,4)'
      WRITE(*,*)'NRL =3 (1,3,3,4); NRL=4 (1,2.5,2.5,4)'
      READ(*,*) NRL

      WRITE(*,*)'A FOUR-DIGIT POSITIVE ODD INTEGER, SAY, 1171'
      READ(*,*) IU
      WRITE(*,*)'INPUT AND OUTPUT FILES ?'
      READ(*,*) INFIL,OUTFIL
      !==============================================================
      WRITE(*,*)'DOES THE DATA SET HAVE THE REFERENCE COMPOSITE INDEX?'
      WRITE(*,*)'IF NO, FEED 0 ELSE FEED ANY NON-ZERO NUMBER'
      READ(*,*) IREFI
      OPEN(7, FILE=INFIL)
      DO I=1,N
      IF(IREFI.EQ.0) THEN
      READ(7,*)(Z(I,J),J=1,M)
      ELSE
      READ(7,*)(Z(I,J),J=1,M),REFI(I)
      ENDIF
      ENDDO
      CLOSE(7)

      IF(IREFI.EQ.0) THEN ! GENERATE REFI FROM  M GIVEN RANK SCORES
```

```
DO I=1,N
REFI(I)=0
DO J=1,M
REFI(I)=REFI(I)+Z(I,J)
ENDDO
ENDDO
CALL DORANK(REFI,N) ! RANKING OF REFI
ENDIF

DO I=1,N
WRITE(*,3)I,(Z(I,J),J=1,M),REFI(I)
ENDDO


3 FORMAT(I4,20F4.0)
WRITE(*,*)
WRITE(*,*)'=========================================================='
WRITE(*,*)
! FIND MEAN AND STANDARD DEVIATION OF ALL VARIABLES
DO J=1,M
ZB(J)=0.D0
SD(J)=0.D0
DO I=1,N
ZB(J)=ZB(J)+Z(I,J)
SD(J)=SD(J)+Z(I,J)**2
ENDDO
SD(J)=DSQRT((N*SD(J)-ZB(J)**2))/N
ZB(J)=ZB(J)/N
ENDDO
WRITE(*,*)'MEAN =',(ZB(J),J=1,M)
WRITE(*,*)'SD =',(SD(J),J=1,M)
! FIND MINIMUM AND MAXIMUM OF EACH VARIABLE
DO J=1,M
ZMIN(J)=Z(1,J)
ZMAX(J)=Z(1,J)
DO I=2,N
IF(ZMIN(J).GT.Z(I,J)) ZMIN(J)=Z(I,J)
IF(ZMAX(J).LT.Z(I,J)) ZMAX(J)=Z(I,J)
ENDDO
ENDDO
! STANDARDIZATION OF VARIABLES
!DO J=1,M
!DO I=1,N
!Z(I,J)=(Z(I,J)-ZMIN(J))/SD(J)
! Z(I,J)=(Z(I,J)-ZMIN(J))/(ZMAX(J)-ZMIN(J))
!ENDDO
!ENDDO
OPEN(8,FILE='ZDAT.TXT')
DO I=1,N
WRITE(8,2)(Z(I,J),J=1,M)
ENDDO
CLOSE(8)
2  FORMAT(10F8.5)
! PAUSE
! MAKE THE INITIAL COMPOSITE INDEX (DF)
```

```
DO I=1,N
DF(I)=0.D0
DO J=1,M
DF(I)=DF(I)+Z(I,J)
ENDDO
ENDDO
!--------------------------------------------------------------
CALL DORANK(DF,N)  ! RANKING OF DF
!--------------------------------------------------------------
DO I=1,N
DFZ(I)=DF(I) ! PRESERVE THE INITIAL DF
ENDDO

! ------------------------------------------------------------
! FIND CORRELATION BETWEEN DF AND THE CONSTITUENT VARIABLES (Z)
ITERATION=0
DELTA=1.0D30
DO WHILE (DELTA.GT.EPS) ! DO WHILE LOOP BEGINS
ITERATION=ITERATION+1

WRITE(*,*)
WRITE(*,*)'**************** ITERATION',ITERATION,'****************
************************'
WRITE(*,*)
DO J=1,M
DO I=1,N
ZF(I,1)=Z(I,J)
ZF(I,2)=DF(I)
ENDDO

IF(NCOR.EQ.1) CALL CORLN(RHO)
IF(NCOR.EQ.2) CALL TAU(RHO)

RZ(J)=DABS(RHO)
IR(J)=J ! INDEX OF RZ
ENDDO
WRITE(*,*)'CORRELATIONS WITH SYNTHETIC INDEX'
WRITE(*,*) (', [',IR(J),'] ',RZ(J),J=1,M)
! PAUSE
! ORDER THE VARIABLES ACCORDING TO MAGNITUDE OF RZ(J) THAT IS,  R
DO J=1,M-1
DO JJ=J+1,M
IF(RZ(J).LT.RZ(JJ)) THEN
T=RZ(J)
RZ(J)=RZ(JJ)
RZ(JJ)=T
IT=IR(J)
IR(J)=IR(JJ)
IR(JJ)=IT
ENDIF
ENDDO
ENDDO
WRITE(*,*)'ORDERED CORRELATIONS WITH SYNTHETIC INDEX'
WRITE(*,*) (', {',IR(J),'} ',RZ(J),J=1,M)
!PAUSE
! COMPUTATION OF WEIGHTS
```

```
W(1)=1.D0
DO J=2,M
! MAKE Y
  DO I=1,N
  YP(I)=Z(I,IR(J))
  ENDDO
! MAKE X
  DO JJ=1,J-1
  DO I=1,N
  XP(I,JJ)=Z(I,IR(JJ))
  ENDDO
  ENDDO

  MVAR=J
 DO I=1,N
  XP(I,MVAR)=1.D0
  ENDDO
  NCASE=N
!-----------------------------------------------------------
  CALL DE(RSQUARE) ! DIFFERENTIAL EVOLUTION OPTIMIZER
!-----------------------------------------------------------
WRITE(*,*)'RSQUARE =', RSQUARE
W(J)=1.D0-RSQUARE
ENDDO
! WITH THIS WEIGHT MAKE NEW DF
DO I=1,N
DFF(I)=0.D0
DO J=1,M
DFF(I)=DFF(I)+Z(I,IR(J))*W(J)
ENDDO
ENDDO
!--------------------------------------------------------------
CALL DORANK(DFF,N) ! RANKING OF DFF
!--------------------------------------------------------------
! FIND THE DISTANCE BETWEEN DFF AND DF
DELTA=0.D0
DO I=1,N
DELTA=DELTA+ DABS(DF(I)-DFF(I))**2
ENDDO
WRITE(*,*)'DELTA =',DELTA
! READ(*,*) IGO

DO I=1,N
DF(I)=DFF(I)
ENDDO
WRITE(*,*)'WEIGHTS=',(W(J),J=1,M)
ENDDO ! DO WHILE LOOP ENDS
! ----------------------------------------------------------
DO J=1,M-1
DO JJ=J+1,M
IF(IR(J).GT.IR(JJ)) THEN
T=W(J)
W(J)=W(JJ)
W(JJ)=T
IT=IR(J)
IR(J)=IR(JJ)
```

```
IR(JJ)=IT
ENDIF
ENDDO
ENDDO

OPEN(7,FILE=OUTFIL)
WRITE(*,*)'.....................................................'
WRITE(7,*)'.....................................................'



WRITE(*,*)'SERIALIZED WEIGHTS ARE AS FOLLOWS='
WRITE(*,*)(W(J),J=1,M)
WRITE(7,*)'SERIALIZED WEIGHTS ARE AS FOLLOWS='
WRITE(7,*)(W(J),J=1,M)
SW=0.D0
DO J=1,M
SW=SW+W(J)
ENDDO
WRITE(*,*)'RELATIVE WEIGHTS =',(W(J)/SW,J=1,M)
WRITE(7,*)'RELATIVE WEIGHTS =',(W(J)/SW,J=1,M)
DO I=1,N
CIND(I)=DFF(I) ! THE PENA-DP2 INDEX
ENDDO

DO J=1,M
DO I=1,N
ZF(I,1)=CIND(I)
ZF(I,2)=Z(I,J)
ENDDO
IF(NCOR.EQ.1) CALL CORLN(RHO)
IF(NCOR.EQ.2) CALL TAU(RHO)
RZ(J)=RHO
ENDDO

WRITE(*,*)'RHO(Z,X) =',(RZ(J),J=1,M)
WRITE(7,*)'RHO(Z,X) =',(RZ(J),J=1,M)
SAR=0.D0
SSR=0.D0
DO J=1,M
SAR=SAR+DABS(RZ(J))
SSR=SSR+RZ(J)**2
ENDDO

DO I=1,N
X(I,1)=CIND(I)
X(I,2)=REFI(I)
DO J=3,M+2
X(I,J)=Z(I,J-2)
ENDDO
ENDDO
!CALL STANDARD(X,N,M+2)
WRITE(*,*) 'COMPOSITE SCORES AND PARTIAL SCORES'
 WRITE(7,*) 'COMPOSITE SCORES AND PARTIAL SCORES'
DO I=1,N
WRITE(7,1) (X(I,J),J=1,M+2)
```

```
      ENDDO
      WRITE(*,*)'NOTE:(1). COMPUTED COMPOSITE SCORES; (2). REF COMPOSITE
     * SCORES,'
      WRITE(7,*)'NOTE:(1). COMPUTED COMPOSITE SCORES; (2). REF COMPOSITE
     * SCORES,'
      WRITE(*,*)'(3, 4, ...). PARTIAL RANK SCORES'
      WRITE(7,*)'(3, 4, ...). PARTIAL RANK SCORES'

    1 FORMAT(20F4.0)

      DO I=1,N
      ZF(I,1)=DFZ(I)
      ZF(I,2)=CIND(I)
      ENDDO
      IF(NCOR.EQ.1) CALL CORLN(R1)
      IF(NCOR.EQ.2) CALL TAU(R1)

      DO I=1,N
      ZF(I,1)=CIND(I)
      ZF(I,2)=REFI(I)
      ENDDO

      IF(NCOR.EQ.1) CALL CORLN(R2)
      IF(NCOR.EQ.2) CALL TAU(R2)
      WRITE(*,*)'SUM OF ABS CORRELATION & SQUARED CORRELATION',SAR,SSR
      WRITE(7,*)'SUM OF ABS CORRELATION & SQUARED CORRELATION',SAR,SSR

      WRITE(*,*)'CORRELATION BETWEEN (DP2,DF) & (DP2,REFI)',R1,R2
      WRITE(7,*)'CORRELATION BETWEEN (DP2,DF) & (DP2,REFI)',R1,R2
      WRITE(*,*)'REFI IS ANY GIVEN EXTRENEOUS COMPOSITE INDEX, OR'
      WRITE(7,*)'REFI IS ANY GIVEN EXTRENEOUS COMPOSITE INDEX, OR'
      WRITE(*,*)'REFI IS THE RANK SCORE OF THE TOTAL OF PARTIAL SCORES'
      WRITE(7,*)'REFI IS THE RANK SCORE OF THE TOTAL OF PARTIAL SCORES'

      CLOSE(7)
      WRITE(*,*)'OVER'
      PAUSE
      END
C     -----------------------------------------------------------------

      SUBROUTINE DE(RSQUARE)
C     PROGRAM: "DIFFERENTIAL EVOLUTION ALGORITHM" OF GLOBAL OPTIMIZATION
C     THIS METHOD WAS PROPOSED BY R. STORN AND K. PRICE IN 1995. REF --
C     "DIFFERENTIAL EVOLUTION - A SIMPLE AND EFFICIENT ADAPTIVE SCHEME
C     FOR GLOBAL OPTIMIZATION OVER CONTINUOUS SPACES" : TECHNICAL REPORT
C     INTERNATIONAL COMPUTER SCIENCE INSTITUTE, BERKLEY, 1995.
C     PROGRAM BY SK MISHRA, DEPT. OF ECONOMICS, NEHU, SHILLONG (INDIA)
C     -----------------------------------------------------------------
C     PROGRAM DE
      IMPLICIT DOUBLE PRECISION (A-H, O-Z) ! TYPE DECLARATION
      PARAMETER(NMAX=500,MMAX=50) ! MAXIMUM DIMENSION PARAMETERS
      PARAMETER(NX=1000,MX=50)
      PARAMETER (RX1=0.D0, RX2=0.D0) ! TO BE ADJUSTED SUITABLY, IF NEEDED
C     RX1 AND RX2 CONTROL THE SCHEME OF CROSSOVER. (0 <= RX1 <= RX2) <=1
C     RX1 DETERMINES THE UPPER LIMIT OF SCHEME 1 (AND LOWER LIMIT OF
C     SCHEME 2; RX2 IS THE UPPER LIMIT OF SCHEME 2 AND LOWER LIMIT OF
```

```
C    SCHEME 3. THUS RX1 = .2 AND RX2 = .8 MEANS 0-20% SCHEME1, 20 TO 80
C    PERCENT SCHEME 2 AND THE REST (80 TO 100 %) SCHEME 3.
C    PARAMETER(NCROSS=2) ! CROSS-OVER SCHEME (NCROSS <=0 OR =1 OR =>2)
     PARAMETER(IPRINT=100,EPS=1.D-08)!FOR WATCHING INTERMEDIATE RESULTS
C    IT PRINTS THE INTERMEDIATE RESULTS AFTER EACH IPRINT ITERATION AND
C    EPS DETERMINES ACCURACY FOR TERMINATION. IF EPS= 0, ALL ITERATIONS
C    WOULD BE UNDERGONE EVEN IF NO IMPROVEMENT IN RESULTS IS THERE.
C    ULTIMATELY "DID NOT CONVERGE" IS REOPORTED.
     COMMON /RNDM/IU,IV ! RANDOM NUMBER GENERATION (IU = 4-DIGIT SEED)
     COMMON /KFF/NFCALL,FTIT ! FUNCTION CODE, NO. OF CALLS * TITLE
     COMMON /XBASE/XBAS
     COMMON /PREGRESS/YP(NX),XP(NX,MX),NCASE,MVAR,RSQ
     INTEGER IU,IV     ! FOR RANDOM NUMBER GENERATION
     CHARACTER *70 FTIT ! TITLE OF THE FUNCTION
C    -----------------------------------------------------------------
C    THE PROGRAM REQUIRES INPUTS FROM THE USER ON THE FOLLOWING ------
C    (1) NO. OF VARIABLES IN THE FUNCTION (M);
C    (2) N=POPULATION SIZE (SUGGESTED 10 TIMES OF NO. OF VARIABLES, M,
C       FOR SMALLER PROBLEMS N=100 WORKS VERY WELL);
C    (3) PCROS = PROB. OF CROSS-OVER (SUGGESTED : ABOUT 0.85 TO .99);
C    (4) FACT = SCALE (SUGGESTED 0.5 TO .95 OR  1, ETC);
C    (5) ITER = MAXIMUM NUMBER OF ITERATIONS PERMITTED (5000 OR MORE)
C    (6) RANDOM NUMBER SEED (4 DIGITS INTEGER)
C    ----------------------------------------------------------------
     DIMENSION X(NMAX,MMAX),Y(NMAX,MMAX),A(MMAX),FV(NMAX)
     DIMENSION IR(3),XBAS(1000,50)
C    ----------------------------------------------------------------
C    ------- SELECT THE FUNCTION TO MINIMIZE AND ITS DIMENSION -------
     FTIT='CONSTRUCTION OF INDEX FROM M VARIABLES '
C    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
     M=MVAR
     N=100
     ITER=100000
     !WRITE(*,*)'CROSSOVER PROBABILITY [PCROS] AND SCALE [FACT] ?'
     !WRITE(*,*)'SUGGESTED : PCROS ABOUT 0.9; FACT=.5 OR LARGER BUT <=1'
     !READ(*,*) PCROS,FACT
     PCROSS=0.9
     FACT=0.5
     !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
C     CALL RANDOM(RAND) ! GENERATES INITION X WITHIN
C     X(I,J)=(RAND-.5D00)*100 ! GENERATES INITION X WITHIN
C    RANDOM NUMBERS BETWEEN -RRANGE AND +RRANGE (BOTH EXCLUSIVE)
     CALL RANDOM(RAND)
     XBAS(I,J)=(RAND-0.5)
     X(I,J)=XBAS(I,J)! TAKES THESE NUMBERS FROM THE MAIN PROGRAM
     ENDDO
```

```
     !WRITE(*,*)(X(I,J),J=1,MVAR),'GENERATED DECISION VARIABLES'
     ENDDO
     WRITE(*,*)'======================================================='
     IPCOUNT=0
     DO 100 ITR=1,ITER  ! ITERATION BEGINS
C    ---------------------------------------------------------------
C    EVALUATE ALL X FOR THE GIVEN FUNCTION
     DO I=1,N
     DO J=1,M
     A(J)=X(I,J)
     ENDDO
     CALL FUNC(A,M,F)
C    STORE FUNCTION VALUES IN FV VECTOR
     FV(I)=F
     ENDDO

C    ---------------------------------------------------------------
C    FIND THE FITTEST (BEST) INDIVIDUAL AT THIS ITERATION
           FBEST=FV(1)
           KB=1
           DO IB=2,N
              IF(FV(IB).LT.FBEST) THEN
              FBEST=FV(IB)
              KB=IB
              ENDIF
           ENDDO
C    BEST FITNESS VALUE = FBEST : INDIVIDUAL X(KB)
C    ---------------------------------------------------------------
C    GENERATE OFFSPRINGS
     DO I=1,N    ! I LOOP BEGINS
C    INITIALIZE CHILDREN IDENTICAL TO PARENTS; THEY WILL CHANGE LATER
        DO J=1,M
        Y(I,J)=X(I,J)
        ENDDO
C    SELECT RANDOMLY THREE OTHER INDIVIDUALS
  20    DO IRI=1,3  ! IRI LOOP BEGINS
       IR(IRI)=0

       CALL RANDOM(RAND)
        IRJ=INT(RAND*N)+1
C    CHECK THAT THESE THREE INDIVIDUALS ARE DISTICT AND OTHER THAN I
        IF(IRI.EQ.1.AND.IRJ.NE.I) THEN
        IR(IRI)=IRJ
        ENDIF
        IF(IRI.EQ.2.AND.IRJ.NE.I.AND.IRJ.NE.IR(1)) THEN
        IR(IRI)=IRJ
        ENDIF
      IF(IRI.EQ.3.AND.IRJ.NE.I.AND.IRJ.NE.IR(1).AND.IRJ.NE.IR(2)) THEN
         IR(IRI)=IRJ
         ENDIF
         ENDDO   ! IRI LOOP ENDS
C    CHECK IF ALL THE THREE IR ARE POSITIVE (INTEGERS)
       DO IX=1,3
       IF(IR(IX).LE.0) THEN
       GOTO 20  ! IF NOT THEN REGENERATE
       ENDIF
```

```
      ENDDO
C    THREE RANDOMLY CHOSEN INDIVIDUALS DIFFERENT FROM I AND DIFFERENT
C    FROM EACH OTHER ARE IR(1),IR(2) AND IR(3)
C    ==================== RANDOMIZATION OF NCROSS ===================
C    RANDOMIZES NCROSS
     NCROSS=0
     CALL RANDOM(RAND)
     IF(RAND.GT.RX1) NCROSS=1 ! IF RX1=>1, SCHEME 2 NEVER IMPLEMENTED
     IF(RAND.GT.RX2) NCROSS=2 ! IF RX2=>1, SCHEME 3 NEVER IMPLEMENTED

C    ---------------------- SCHEME 1 ---------------------------------
C     NO CROSS OVER, ONLY REPLACEMENT THAT IS PROBABILISTIC
      IF(NCROSS.LE.0) THEN
      DO J=1,M      ! J LOOP BEGINS
      CALL RANDOM(RAND)
      IF(RAND.LE.PCROSS) THEN ! REPLACE IF RAND < PCROS
      A(J)=X(IR(1),J)+(X(IR(2),J)-X(IR(3),J))*FACT ! CANDIDATE CHILD
      ENDIF
      ENDDO  ! J LOOP ENDS
      ENDIF
C    ----------------------- SCHEME 2 --------------------------------
C    THE STANDARD CROSSOVER SCHEME
C    CROSSOVER SCHEME (EXPONENTIAL) SUGGESTED BY KENNETH PRICE IN HIS
C    PERSONAL LETTER TO THE AUTHOR (DATED SEPTEMBER 29, 2006)
     IF(NCROSS.EQ.1) THEN
      CALL RANDOM(RAND)
      JR=INT(RAND*M)+1
      J=JR
   2   A(J)=X(IR(1),J)+FACT*(X(IR(2),J)-X(IR(3),J))
      J=J+1
      IF(J.GT.M) J=1
      IF(J.EQ.JR) GOTO 10
      CALL RANDOM(RAND)
      IF(PCROS.LE.RAND) GOTO 2
   6   A(J)=X(I,J)
      J=J+1
      IF(J.GT.M) J=1
      IF (J.EQ.JR) GOTO 10
      GOTO 6
  10   CONTINUE
     ENDIF
C    ------------------------ SCHEME 3 -------------------------------
C    ESPECIALLY SUITABLE TO NON-DECOMPOSABLE (NON-SEPERABLE) FUNCTIONS
C    CROSSOVER SCHEME (NEW) SUGGESTED BY KENNETH PRICE IN HIS
C    PERSONAL LETTER TO THE AUTHOR (DATED OCTOBER 18, 2006)
     IF(NCROSS.GE.2) THEN
       CALL RANDOM(RAND)
       IF(RAND.LE.PCROS) THEN
         CALL NORMAL(RN)
         DO J=1,M
         A(J)=X(I,J)+(X(IR(1),J)+ X(IR(2),J)-2*X(I,J))*RN
         ENDDO
        ELSE
        DO J=1,M
        A(J)=X(I,J)+(X(IR(1),J)- X(IR(2),J))! FACT ASSUMED TO BE 1
        ENDDO
```

```
         ENDIF
      ENDIF
C     ------------------------------------------------------------------
        CALL FUNC(A,M,F) ! EVALUATE THE OFFSPRING
        IF(F.LT.FV(I)) THEN ! IF BETTER, REPLACE PARENTS BY THE CHILD
        FV(I)=F
        DO J=1,M
        Y(I,J)=A(J)
        ENDDO
        ENDIF
      ENDDO   ! I LOOP ENDS
      DO I=1,N
      DO J=1,M
      X(I,J)=Y(I,J) ! NEW GENERATION IS A MIX OF BETTER PARENTS AND
C                 BETTER CHILDREN
      ENDDO
      ENDDO
      IPCOUNT=IPCOUNT+1
      IF(IPCOUNT.EQ.IPRINT) THEN
      DO J=1,M
      A(J)=X(KB,J)
      ENDDO
       WRITE(*,*)(X(KB,J),J=1,M),'  FBEST UPTO NOW = ',FBEST
       WRITE(*,*)'TOTAL NUMBER OF FUNCTION CALLS =',NFCALL
         IF(DABS(FBEST-GBEST).LT.EPS) THEN
!     WRITE(*,*) FTIT
         WRITE(*,*)'CONVERGED!'
         GOTO 999
         ELSE
         GBEST=FBEST
         ENDIF
      IPCOUNT=0
      ENDIF
C     ------------------------------------------------------------------
  100 ENDDO   ! ITERATION ENDS : GO FOR NEXT ITERATION, IF APPLICABLE
C     ------------------------------------------------------------------
      !WRITE(*,*)'DID NOT CONVERGE. REDUCE EPS OR RAISE ITER OR DO BOTH'
      !WRITE(*,*)'INCREASE N, PCROS, OR SCALE FACTOR (FACT)'

  999 RSQ=FBEST
      RSQUARE=-RSQ
      RETURN
      END
C     ------------------------------------------------------------------
      SUBROUTINE NORMAL(R)
C     PROGRAM TO GENERATE N(0,1) FROM RECTANGULAR RANDOM NUMBERS
C     IT USES BOX-MULLER VARIATE TRANSFORMATION FOR THIS PURPOSE.
C     ------------------------------------------------------------------
C     ----- BOX-MULLER METHOD BY GEP BOX AND ME MULLER (1958) ---------
C     BOX, G. E. P. AND MULLER, M. E. "A NOTE ON THE GENERATION OF
C     RANDOM NORMAL DEVIATES." ANN. MATH. STAT. 29, 610-611, 1958.
C     IF U1 AND U2 ARE UNIFORMLY DISTRIBUTED RANDOM NUMBERS (0,1),
C     THEN X=[(-2*LN(U1))**.5]*(COS(2*PI*U2) IS N(0,1)
C     ALSO,  X=[(-2*LN(U1))**.5]*(SIN(2*PI*U2) IS N(0,1)
C     PI = 4*ARCTAN(1.0)= 3.1415926535897932384626433832795
C     2*PI = 6.283185307179586476925286766559
```

```
C    ------------------------------------------------------------------
     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
     COMMON /RNDM/IU,IV
     INTEGER IU,IV
C    ------------------------------------------------------------------
     CALL RANDOM(RAND) ! INVOKES RANDOM TO GENERATE UNIFORM RAND [0, 1]
     U1=RAND ! U1 IS UNIFORMLY DISTRIBUTED [0, 1]
     CALL RANDOM(RAND) ! INVOKES RANDOM TO GENERATE UNIFORM RAND [0, 1]
     U2=RAND ! U1 IS UNIFORMLY DISTRIBUTED [0, 1]
     R=DSQRT(-2.D0*DLOG(U1))
     R=R*DCOS(U2*6.28318530717958647692528676655 9D00)
C     R=R*DCOS(U2*6.28318530718D00)
     RETURN
     END
C    ------------------------------------------------------------------
C     RANDOM NUMBER GENERATOR (UNIFORM BETWEEN 0 AND 1 - BOTH EXCLUSIVE)
     SUBROUTINE RANDOM(RAND1)
      DOUBLE PRECISION  RAND1
      COMMON /RNDM/IU,IV
     INTEGER IU,IV
      IV=IU*65539
      IF(IV.LT.0) THEN
      IV=IV+2147483647+1
      ENDIF
      RAND=IV
      IU=IV
      RAND=RAND*0.4656613E-09
      RAND1= DBLE(RAND)
      RETURN
      END
C    ------------------------------------------------------------------
     SUBROUTINE FUNC(X,M,F)
     PARAMETER (NX=1000,MX=50)
C     TEST FUNCTIONS FOR GLOBAL OPTIMIZATION PROGRAM
     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
     COMMON /RNDM/IU,IV
     COMMON /PREGRESS/YP(NX),XP(NX,MX),NCASE,MVAR,RSQ
     COMMON /KFF/NFCALL,FTIT
     INTEGER IU,IV
     DIMENSION X(*)
     CHARACTER *70 FTIT
     NFCALL=NFCALL+1 ! INCREMENT TO NUMBER OF FUNCTION CALLS
C    ---------------------------------------------------------------
     !WRITE(*,*)(X(I),I=1,MVAR),' FUNC DE DECISION VARIABLES'
     CALL CORDINATE(M,X,F)
     RETURN
     END
C    ------------------------------------------------------------------
     SUBROUTINE CORDINATE(M,X,F)
     !PARAMETER (MVAR=6)! CHANGE THE PARAMETERS HERE AS NEEDED.
     PARAMETER(NX=1000,MX=50)
C    ------------------------------------------------------------------
C     NOB=NO. OF OBSERVATIONS (CASES) & MVAR= NO. OF VARIABLES
     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
     COMMON /RNDM/IU,IV
     COMMON /CORDAT/CDAT(NX,MX),QIND(NX),R(MX),NCOR
```

```
      COMMON /GETRANK/MRNK
      COMMON /PREGRESS/YP(NX),XP(NX,MX),NCASE,MVAR,RSQ
      COMMON /CORLNO/ZF(NX,2)
      INTEGER IU,IV
      DIMENSION X(*)
      !WRITE(*,*)(X(I),I=1,MVAR),' DE DECISION VARIABLES'
      DO I=1,MVAR
      IF(X(I).LT.-1.0D0.OR.X(I).GT.1.0D0) THEN
      CALL RANDOM(RAND)
      X(I)=(RAND-0.5D0)*2
      ENDIF

      ENDDO
      DO I=1,NCASE
      !WRITE(*,*)'XP =',(XP(I,J),J=1,MVAR)
      ENDDO

C     CONSTRUCT EXPECTED VARIABLE
      DO I=1,NCASE
      QIND(I)=0.D0
      DO J=1,MVAR
      QIND(I)=QIND(I)+XP(I,J)*X(J)
      ENDDO
      !WRITE(*,*)I,QIND(I),' QIND'
      ENDDO
C     -----------------------------------------------------------------
      !FIND THE RANK OF QIND
      CALL DORANK(QIND,NCASE)
C     -----------------------------------------------------------------
C     COMPUTE CORRELATIONS
      DO I=1,NCASE
      ZF(I,1)=QIND(I)
      ZF(I,2)=YP(I)
      ENDDO
      DO I=1,NCASE
      !WRITE(*,*)I, ZF(I,1),ZF(I,2),' : Z1 AND Z2 BEFORE CORLN CALLED'
      ENDDO

      IF(NCOR.EQ.1) CALL CORLN(RHO)
      IF(NCOR.EQ.2) CALL TAU(RHO)

      !-----------------------------------------------------------------
      F=-DABS(RHO)**2
      RETURN
      END
C     -----------------------------------------------------------------
      SUBROUTINE CORLN(RHO)
      PARAMETER (NX=1000, MX=50)
C     NOB = NO. OF CASES
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /PREGRESS/YP(NX),XP(NX,MX),NCASE,MVAR,RSQ
      COMMON /CORLNO/ZF(NX,2)
      DIMENSION AV(2),SD(2)
      NOB=NCASE
      DO I=1,NOB
      !WRITE(*,*)I,ZF(I,1),ZF(I,2),  'IT IS IN CORRELATION PROGRAM'
```

```
      ENDDO
      DO J=1,2
      AV(J)=0.D0
      SD(J)=0.D0
      DO I=1,NOB
      AV(J)=AV(J)+ZF(I,J)
      SD(J)=SD(J)+ZF(I,J)**2
      ENDDO
      ENDDO
      DO J=1,2
      AV(J)=AV(J)/NOB
      SD(J)=DSQRT(SD(J)/NOB-AV(J)**2)
      ENDDO
C       WRITE(*,*)'AV AND SD ', AV(1),AV(2),SD(1),SD(2)
      RHO=0.D0
      DO I=1,NOB
      RHO=RHO+(ZF(I,1)-AV(1))*(ZF(I,2)-AV(2))
      ENDDO
      RHO=(RHO/NOB)/(SD(1)*SD(2))
      ! WRITE(*,*)'RHO =',RHO
      ! PAUSE
      RETURN
      END
C     -----------------------------------------------------------------
      SUBROUTINE TAU(RHO)
      PARAMETER (NX=1000, MX=50)
C     NOB = NO. OF CASES
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /PREGRESS/YP(NX),XP(NX,MX),NCASE,MVAR,RSQ
      COMMON /CORLNO/ZF(NX,2)
      RHO=0.D0
      DO I=1,NCASE
      DO J=1,NCASE
      IF((ZF(I,1).GT.ZF(J,1)).AND.(ZF(I,2).GT.ZF(J,2))) RHO=RHO+1.D0
      ENDDO
      ENDDO
      RHO=4.D0*RHO/(NCASE**2-NCASE)-1.D0
      ! WRITE(*,*)'RHO =',RHO
      ! PAUSE
      RETURN
      END
C     -------------------------------------------------------------------
      SUBROUTINE DORANK(X,N)! N IS THE NUMBER OF OBSERVATIONS
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C     !THIS PROGRAM RETURNS RANK-ORDER OF A GIVEN VECTOR
      PARAMETER (MXD=1000)! MXD IS MAX DIMENSION FOR TEMPORARY VARIABLES
      PARAMETER (NX=1000)
      ! THAT ARE LOCAL AND DO NOT GO TO THE INVOKING PROGRAM
      ! X IS THE VARIABLE TO BE SUBSTITUTED BY ITS RANK VALUES
      COMMON /RANKS/NRL
      ! THIS VALUE IS TO BE SET BY THE USER
C                   !THE VALUE OF NRL DECIDES THE SCHEME OF RANKINGS
C
C     NRULE=0 FOR ORDINAL RANKING (1-2-3-4 RULE);
C     NRULE=1 FOR DENSE RANKING (1-2-2-3 RULE);
C     NRULE=2 FOR STANDARD COMPETITION RANKING (1-2-2-4 RULE);
```

```
C     NRULE=3 FOR MODIFIED COMPETITION RANKING (1-3-3-4 RULE);
C     NRULE=4 FOR FRACTIONAL RANKING (1-2.5-2.5-4 RULE);
      DIMENSION X(NX),NF(MXD),NCF(MXD),RANK(MXD),ID(MXD),XX(MXD)
C     GENERATE ID(I),I=1,N
      DO I=1,N
      ID(I)=I
      NF(I)=0
      ENDDO
C     ARRANGE DATA (X) AND THE IDS IN ASCENDING ORDER
      DO I=1,N-1
      DO II=I,N
      IF(X(II).LT.X(I)) THEN
      TEMP=X(I)
      X(I)=X(II)
      X(II)=TEMP
      ITEMP=ID(I)
      ID(I)=ID(II)
      ID(II)=ITEMP
      ENDIF
      ENDDO
      ENDDO
C     MAKE DISCRETE UNGROUPED FREQUENCY TABLE
      K=0
      J=1
    1 K=K+1
      XX(K)=X(J)
      NF(K)=0
      DO I=J,N
      IF(XX(K).EQ.X(I)) THEN
      NF(K)=NF(K)+1
      ELSE
      J=I
      IF(J.LE.N) THEN
      GOTO 1
      ELSE
      GOTO 2
      ENDIF
      ENDIF
      ENDDO
    2 KK=K
      DO K=1,KK
      IF(K.EQ.1) THEN
      NCF(K)=NF(K)
      ELSE
      NCF(K)=NCF(K-1)+NF(K)
      ENDIF
      ENDDO
      DO I=1,N
      RANK(I)=1.D0
      ENDDO

      IF(NRL.GT.4) THEN
      WRITE(*,*)'RANKING RULE CODE GREATER THAN 4 NOT PERMITTED',NRL
      STOP
      ENDIF
```

```
      IF(NRL.LT.0) THEN
      WRITE(*,*)'RANKING RULE CODE LESS THAN 0 NOT PERMITTED',NRL
      STOP
      ENDIF

      IF(NRL.EQ.0) THEN
      DO I=1,N
      RANK(I)=I
      ENDDO
      ENDIF
C     ----------------------------------------------------------------
      IF(NRL.GT.0) THEN
       DO K=1,KK
       IF(K.EQ.1) THEN
       K1=1
       ELSE
       K1=NCF(K-1)+1
       ENDIF
       K2=NCF(K)
       DO I=K1,K2
       SUM=0.D0
       DO II=K1,K2
       SUM=SUM+II
       ENDDO
       KX=(K2-K1+1)
       IF(NRL.EQ.1)RANK(I)=K ! DENSE RANKING (1-2-2-3 RULE)
       IF(NRL.EQ.2)RANK(I)=K1!STANDARD COMPETITION RANKING(1-2-2-4 RULE)
       IF(NRL.EQ.3)RANK(I)=K2!MODIFIED COMPETITION RANKING(1-3-3-4 RULE)
       IF(NRL.EQ.4)RANK(I)=SUM/KX !FRACTIONAL RANKING (1-2.5-2.5-4 RULE)
       ENDDO
       ENDDO
      ENDIF
C     ----------------------------------------------------------------
      DO I=1,N
      X(ID(I))=RANK(I) ! BRINGS THE DATA TO ORIGINAL SEQUENCE
      ENDDO
      RETURN
      END
```