COMPUTER AIDED SELECTION OF ROBOT FOR LOADING AND UNLOADING OPERATION BY MULTIPLE ATTRIBUTE DECISION MAKING (MADM) APPROACH

Abhishake Chaudhary[1], Vijayant Maan[2], D.P Singh[3]
[1, 2, 3] Lecturer,Department of Mechanical Engineering,S.C.R.I.E.T,C.C.S University Meerut,Uttar Pradesh,India

ABSTRACT

The selection of a suitable robot is becoming more and more difficult because of the large increase in robot manufacturer, configuration and available option. In this work a methodology and programming algorithm, based on Multiple Attribute Decision Making is developed for such type of selection problem. TOPSIS (Technique for Order Preference by Similarity to the Ideal Solution) method is used to rank the available alternative. Weights are assigned by the decision maker to the different attribute according to their importance. This method will help the decision maker to select a suitable robot according to his requirement. The developed software is analysed by an illustrative example for loading and unloading operation with same data.

***Keywords*:** Attribute, selection of robot, MADM approach, rank according to reqirement .

I. INTRODUCTION

There has been rapid increase in the number of robot systems and robot manufacturers. Robots with vastly different capabilities and specifications are available for a wide range of applications. The selection of the robot to suit a particular application and production environment, from the large number of robots available in the market today has become a difficult task. Various considerations such as availability, management policies, production systems compatibility, and economics need to be considered before a suitable robot can be selected. The complexity of problem can be better appreciated when one realizes that there are over 75 attributes that have to be considered in the selection of robot for particular application. Moreover, many of them are conflicting in nature and have different units, which cannot be unified and compared as they are. The quantification and monitoring of the attribute magnitudes will help the manufacturer to control them closely so that he can fulfil the demand of the user precisely. Moreover, he can find out the market trend by observing the attributes magnitudes. This will help the manufacturer to modify his product to suit the future needs of the robot user. He can use the database to produce optimum robots in the minimum possible time. The robot manufacturer can also use these attributes for the SWOT (Strength–Weakness–Opportunity–Threat) analysis of his product. This identification of the attributes will help the user for the database storage and its retrieval. This will generate the computerized database, which can be used in different formats for different purposes by different people in the organization. It also will help the user to select the best possible robot for the particular application whenever it is required. The user will know exactly what are the physical characteristics and performance parameters of the robot. This will keep the user well informed about the capabilities of the robot while putting it to use.

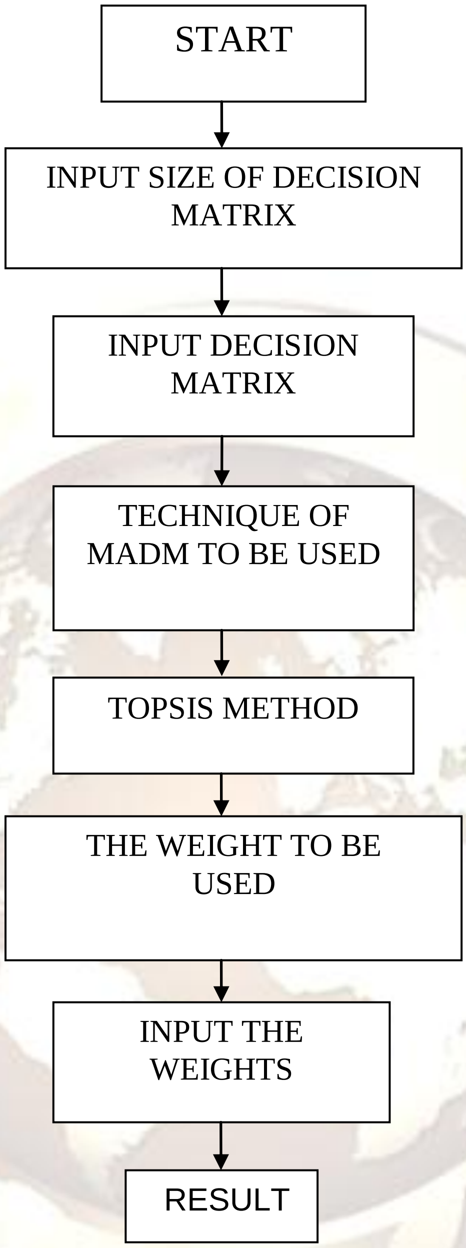
II. LITERATURE REVIEW

Author	Method	Remark
1.Quing wang	Multiple attribute decision making under fuzzy data	Approach deals with attribute weights which are completely unknown is developed by using expected value operator of fuzzy variables
2.R.Venkata Rao and K.K. Padmanabhanb	Digraph and matrix methods	A step by step procedure for evaluation of robot selection index is suggested.
3. A.Y. Odabas	fuzzy multiple attributive group decision making methodology	An attribute based aggregation technique for heterogeneous group of experts is employed and used for dealing with fuzzy opinion aggregation for the subjective attributes of the decision problem

4. Jian Chen & Song Lin	interactive neural network	To obtain preference information, it is necessary to evaluate many alternatives that are associated with multiple, conflicting and non-commensurate criteria.
5.Jian Ma	multiple attribute decision making	optimization model is constructed to assess attribute weights and then to rank the alternatives or select the most desirable one.
6. Omar F. EI-Gayar	multiple criteria decision making framework	The MCDM model seeks a desirable allocation of resources and activity levels that strikes an acceptable balance among the various development goals
7.Layek Albde-Malek & Lars Johan Resare	analytical algorithm based decision support system	DSS evaluates the design and geometry of the mating parts that are to be processed and assembled by the cell. Accordingly, it recommends the machining centre and robot that maximize the cell’s performance subject to various operational and budget constraints.
8. Haymwantee P. Singh & Wilfred V. Huang	fuzzy set method	Data from both evaluations are finally processed such that a fuzzy set decision making body are integrated

III. METHODOLOGY

3.1 Flow chart for Multiple attribute decision making is given below-



1.2 THE 3-STAGE SELECTION PROCEDURE

1.2.1 Elimination search: -

Though all the attributes have been identified, all of them would not be important while selecting the robot for particular application. There will be few attributes, which will have direct effect on the selection procedure. This small number of attributes may be set-aside as pertinent attributes as necessitated by the particular application and/or the user. The threshold values to these pertinent attributes may be assigned by obtaining information from the user and the group of experts. On the basis of the threshold values of the pertinent attributes, a shortlist of robots is obtained. This may be achieved by scanning the database for pertinent attributes, one at a time, to eliminate the robot alternatives, which have one or more pertinent attribute values that fall short of the minimum required (threshold) values. To facilitate this search procedure an identification system has been made for all the robots in the data base.

3.2.2 Evaluation procedure: -

A mini-database is thus formed which comprises these satisfying solutions i.e., alternatives which have all attributes satisfying the acceptable levels of aspiration. The problem is now one of finding out the optimum or best out of these satisfying solutions. The selection procedure therefore needs to rank these solutions in order of merit. The first step here will be to represent all the information available from the database about these satisfying solutions in the matrix form. Such a matrix is called as decision matrix, **D**. Each row of this matrix is allocated to one candidate robot and each column to one attribute under consideration. Therefore an element d_{ij} of the decision matrix D, gives the value of jth attribute in the row (non-normalized) form and units, for the ith robot. Thus if the number of short-listed robots is m and the number of pertinent attributes is n, the decision matrix is an m * n matrix. This evaluation procedure completes in three steps

Step-1 Normalized specifications:

The next step is construction of the normalized specification matrix, N, from the decision matrix, D. Normalization is used to bring the data within particular range or scale, and moreover, it provides the dimensionless magnitudes. This phenomenon is used to calculate the normalized specification matrix. The normalized specification matrix will have the magnitudes of all the attributes of the robots on the common scale of 0 to 1. It is a sort of value, which indicates the standing of that particular attribute magnitude when compared to the whole range of the magnitudes for all candidate robots.
An element n_{ij} of the normalized matrix N can be calculated as

$$n_{ij} = d_{ij} \Big/ \sqrt{\sum_{i=1}^{m} d_{ij}^2}$$

Where d_{ij} is an element of the decision matrix D.

Step-2 Method for Assigning Weights:

Many methods for MADM problems require information about the relative importance of each attribute. It is usually given by a set of weights which is normalized to sum to 1. In case of n attributes, a set of weights is-

$$W^T = (W_1, W_2, W_3, \ldots\ldots\ldots\ldots, W_n)$$

$$\sum_{j=1}^{n} W_j = 1$$

Step-3 Weighted normalized specification: -

The weights obtained from the relative importance matrix have to be applied to the normalized specifications since all attributes have different importance while selecting the robot for particular application. The matrix, which combines the relative weights and normalized specification of the candidates, is weighted normalized matrix, V. It will give the true comparable values of the attributes. This can be obtained as follows:

V =

$$\begin{pmatrix} w_1 n_{1,1} & w_2 n_{1,2} & \ldots w_n n_{1,n} \\ w_1 n_{2,1} & \ddots & \vdots \\ w_1 n_{m,1} & w_2 n_{m,2} & \ldots w_n n_{m,n} \end{pmatrix} = \begin{pmatrix} v_{1,1} & v_{1,2} & \cdots v_{1,n} \\ v_{2,1} & \ddots & \vdots \\ v_{m1} & v_{m,2} & \cdots v_{m,n} \end{pmatrix}$$

1.3 TOPSIS (Technique for Order Preference by Similarity to Ideal Solution)

The weighted normalized matrix V is used to obtain the +ve and -ve benchmark robots, where the both benchmark robots are hypothetical robots, which supposed to have best and worst possible attribute magnitudes. Hwang and Yoon developed TOPSIS based upon the concept that the chosen option (optimum) should have the shortest distance from the +ve benchmark robot (best possible robot) and be farthest from the -ve benchmark robot (worst possible robot). The measure ensures that the top ranked robot is closest to +ve benchmark robot and farthest from -ve benchmark robot. Here, we calculate separation measures from +ve and -ve benchmark robots, respectively, as S_i^* and S_i^- as follows.

The separation from the +ve benchmark robot is given by

$$S_i^* = \left[\sum_{j=1}^{n} (v_{ij} - v_1^*)^2\right]^{1/2} \qquad (i = 1, 2, \ldots, m)$$

and separation from the -ve benchmark robot is given by

$$S_i^- = \left[\sum_{j=1}^{n} (v_{ij} - v_1^-)^2\right]^{1/2} \qquad (i = 1, 2, \ldots, m)$$

Then the relative closeness to the +ve benchmark robot, C*, which is a measure of the suitability of the robot for the chosen application on the basis of attributes considered, is calculated. A robot with the largest C* is preferable.

$$C^* = S_i^- / (S_i^* + S_i^-)$$

Ranking of the candidate robots in accordance with the decreasing values of indices C* indicating the most preferred and the least preferred feasible optional solutions is done.

IV. Selection of ROBOT using Multiple Attribute Decision Making

We take the example of robot selection for welding operation using MADM approach. The minimum requirement for this application is as follows Table 1:

Table.1

1. Load capacity	minimum 2 kg
2. Repeatability	0.5 mm
3. Maximum tip speed	at least 255 mm/s
4. Type of drives (actuators)	electrical only
5. Memory capacity	At least 250 points/steps
6. Manipulator reach	500 mm
7. Degree of freedom	at least 5

From the database generated, after 'elimination search' we can find out manageable number of candidate robots and their pertinent attributes.

Candidate robots are listed below: -

ASEA-IRB60/2 - **(A_1)**

Cybotech v 15 Electric Robot - **(A_2)**

Hitachi America Process Robot - **(A_3)**

Unimation Puma500/600 - **(A_4)**

Kuka Robotics India Pvt. Ltd. (KR 360 L150-2P) - **(A_5)**

Precision Automation & Robotics India Pvt.Ltd - **(A_6)**

Rhythmsoft Robotics & Automation Pvt. Ltd (KR 500 570 – 2PA) - **(A_7)**

Pertinent attributes are listed below: -

Load Capacity (kg) - (X_1)
Repeatability (mm) - (X_2)
Reach (mm) - (X_3)
Max. Tip Speed (mm/sec) - (X_4)
Memory Capacity (Points or Steps) - (X_5)
Price (Rs.) - (X_6)
Degree of freedom - (X_7)

Attributes for the short-listed candidate robots is show in table 2:

Table.2

Here repeatability and cost are the type of attribute

Att-Alt	X_1	X_2	X_3	X_4	X_5	X_6	X_7
(A_1)	60	0.40	990	2540	500	175000	6
(A_2)	6.8	0.1	1676	1727	1500	200000	7
(A_3)	10	0.2	965	1000	2000	325000	6
(A_4)	2.5	0.1	915	560	500	150000	6
(A_5)	150	0.15	3500	1400	1200	900000	6
(A_6)	300	0.5	1500	1600	1950	875000	7
(A_7)	570	0.15	2826	1550	1600	1072500	6

of which is the minimum magnitude is preferable and hence the reciprocal of the values in column representing repeatability should be used to form the decision matrix, **D**.

Table 3

Att. Alter	(X_1)	(X_2)	(X_3)	(X_4)	(X_5)	(X_6)	(X_7)
(A_1)	60	2.5	990	2540	500	.00000571	6
(A_2)	6.8	10	1676	1727.2	1500	.000005	7
(A_3)	10	5	965	1000	2000	.00000307	6
(A_4)	2.5	10	915	560	500	.00000666	6
(A_5)	150	6.66	3500	1400	1200	.00000111	6
(A_6)	300	2	1500	1600	1950	.00000114	7
(A_7)	570	6.66	2826	1550	1600	.00000093	6

Data obtained from the above table is use in the procedure of selection of robot which is as follows:

Step 1

Formation of decision matrix, '**D**', i.e., the matrix which will contain all the magnitudes of specifications. The rows of the matrix are the candidate robots, with their attribute values listed in columns.

$$\begin{pmatrix} 60 & 2.5 & 990 & 2540 & 500 & .00000571 & 6 \\ 6.8 & 10 & 1676 & 1727.2 & 1500 & .000005 & 7 \\ 10 & 5 & 965 & 1000 & 2000 & .00000307 & 6 \\ 2.5 & 10 & 915 & 560 & 500 & .00000666 & 6 \\ 150 & 6.66 & 3500 & 1400 & 1200 & .00000111 & 6 \\ 300 & 2 & 1500 & 1600 & 1950 & .00000114 & 7 \\ 570 & 6.66 & 2826 & 1550 & 1600 & .00000092 & 6 \end{pmatrix}$$

Step 2: -

Calculating the normalized specification matrix. This normalization helps to provide the dimensionless elements of the matrix.

$$\begin{pmatrix} 60 & 2.5 & 990 & 2540 & 500 & .00000571 & 6 \\ 6.8 & 10 & 1676 & 1727.2 & 1500 & .000005 & 7 \\ 10 & 5 & 965 & 1000 & 2000 & .00000307 & 6 \\ 2.5 & 10 & 915 & 560 & 500 & .00000666 & 6 \\ 150 & 6.66 & 3500 & 1400 & 1200 & .00000111 & 6 \\ 300 & 2 & 1500 & 1600 & 1950 & .00000114 & 7 \\ 570 & 6.66 & 2826 & 1550 & 1600 & .00000092 & 6 \end{pmatrix}$$

Calculating the normalized specification matrix. This normalization helps to provide the dimensionless elements of the matrix.

$$r_{ij} = d_{ij} \Big/ \left(\sum_{i=1}^{m} d_{ij}^2 \right)^{1/2}$$

$$\begin{pmatrix} 0.090335 & 0.138897 & 0.18694 & 0.604233 & 0.131069 & 0.405223 & 0.359856 \\ 0.010238 & 0.555589 & 0.316476 & 0.410878 & 0.393208 & 0.354836 & 0.419832 \\ 0.015056 & 0.277794 & 0.182219 & 0.237887 & 0.524277 & 0.21787 & 0.359856 \\ 0.003764 & 0.555589 & 0.172967 & 0.133217 & 0.131069 & 0.472642 & 0.359856 \\ 0.225838 & 0.370022 & 0.660899 & 0.333042 & 0.314566 & 0.078774 & 0.359856 \\ 0.451676 & 0.111118 & 0.283243 & 0.380616 & 0.511171 & 0.080903 & 0.419832 \\ 0.858184 & 0.370022 & 0.533629 & 0.368725 & 0.419422 & 0.652899 & 0.359856 \end{pmatrix}$$

Step 3: -

Assign weights for each attribute such that their sum will be equal to one.

$$\sum_{i=1}^{n} w_i = 1$$

$$w_1+w_2+w_3+w_4+w_5=1$$

w_1= 0.21 w_2 = 0.12 w_3= 0.15 w_4= 0.10
w_5= 0.10 w_6= 0.25 w_7=0.08

Calculating the weighted normalized specification matrix. Here we incorporate the relative importance of the attributes with their normalized value to create unique parameter for the candidate robot.

$$V_{ij} = N_{ij}\, W_i$$

$$V_{ij=}$$

$$\begin{pmatrix} 0.090335 & 0.138897 & 0.18694 & 0.604233 & 0.131069 & 0.405223 & 0.359856 \\ 0.010238 & 0.555589 & 0.316476 & 0.410878 & 0.393208 & 0.354836 & 0.419832 \\ 0.015056 & 0.277794 & 0.182219 & 0.237887 & 0.524277 & 0.21787 & 0.359856 \\ 0.003764 & 0.555589 & 0.172967 & 0.133217 & 0.131069 & 0.472642 & 0.359856 \\ 0.225838 & 0.370022 & 0.660899 & 0.333042 & 0.314566 & 0.078774 & 0.359856 \\ 0.451676 & 0.111118 & 0.283243 & 0.380616 & 0.511171 & 0.080903 & 0.419832 \\ 0.858184 & 0.370022 & 0.533629 & 0.368725 & 0.419422 & 0.652899 & 0.359856 \end{pmatrix} \begin{pmatrix} 0.20 \\ 0.12 \\ 0.15 \\ 0.10 \\ 0.10 \\ 0.25 \\ 0.08 \end{pmatrix}$$

V_{ij} =

$$\begin{pmatrix} 0.018067 & 0.016668 & 0.028041 & 0.060423 & 0.013107 & 0.101306 & 0.028788 \\ 0.002048 & 0.066671 & 0.047471 & 0.041088 & 0.039321 & 0.088709 & 0.033587 \\ 0.003011 & 0.03335 & 0.027333 & 0.023789 & 0.052428 & 0.054467 & 0.028788 \\ 0.000753 & 0.066671 & 0.025945 & 0.013322 & 0.013107 & 0.118161 & 0.028788 \\ 0.045168 & 0.044403 & 0.099135 & 0.033304 & 0.031457 & 0.019693 & 0.028788 \\ 0.090335 & 0.013334 & 0.042486 & 0.038062 & 0.051117 & 0.020226 & 0.033587 \\ 0.171637 & 0.044403 & 0.080044 & 0.036872 & 0.041942 & 0.163225 & 0.028788 \end{pmatrix}$$

4.1 TOPSIS method for ranking

This is the fifth step of the selection procedure. The weighted normalized attributes for the +ve and −ve benchmark robots can be obtained as:

v^* = (0.0171637 0.066671 0.099135 0.060423 0.0524428 0.163225 0.033587)

v^- = (0.000753 0.013334 0.025945 0.013322 0.013107 0.019693 0.028788)

Separation of the alternatives from the ideal and negative ideal solution is as follows:

$$S_1^+ = 0.197381 \qquad S_1^- = 0.096048$$
$$S_2^+ = 0.201186 \qquad S_2^- = 0.097753$$
$$S_3^+ = 0.22541 \qquad S_3^- = 0.057207$$
$$S_4^+ = 0.208265 \qquad S_4^- = 0.111985$$
$$S_5^+ = 0.199823 \qquad S_5^- = 0.096111$$
$$S_6^+ = 0.185181 \qquad S_6^- = 0.105841$$
$$S_7^+ = 0.039344 \qquad S_7^- = 0.240992$$

A programme is developed for verifying the result which is given below-

4.2 Software for selection of ROBOT using Multiple Attribute Decision Making

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
#define ROWS 6
#define COLS 6
class Robo
{
private:
double D[ROWS][COLS],N[ROWS][COLS],
V[ROWS][COLS];
double Result_Mat[ROWS][COLS],R[ROWS];
double S_Positive[COLS],S_Negative[COLS],
C[COLS];
double V_Positive[COLS],V_Negative[COLS];
float W[COLS];
static int index_positive;
static int index_negative;

void squareMatrix_SQRT(double M[][COLS],int
r,int c)
{
        double sum=0;
```

```
for(int i=0;i<r;i++)
{
sum=sum+(M[i][c]*M[i][c]);
}

R[c]=sqrt(sum);
cout<<"sqrt: "<<sqrt(sum);

            }
 void call_weitageMatrix(double N[][COLS])
{

for(int ncol=0;ncol<COLS;ncol++)
{
for(int nrow=0;nrow<ROWS;nrow++)
{
V[nrow][ncol]=(N[nrow][ncol])*(W[ncol]);
         }
       }
}
void insert_TempArray(double V[][COLS],int r,int c)
            {
double tempArray[ROWS];
for(int i=0;i<r;i++)
{
tempArray[i]=V[i][c];
                        }
                        /*
cout<<endl;
for(i=0;i<r;i++)
{
cout<<" "<<tempArray[i];
}
*/
sortArray(tempArray,r);
}
/*
function to sort elements in Temp Array
*/
void sortArray(double TA[],int n)
            {
double temp=0;
for(int i=0;i<n;i++)
{
for(int j=i+1;j<n;j++)
{
if(TA[i]<TA[j])
{
temp=TA[i];
TA[i]=TA[j];
TA[j]=temp;
            }
        }
}
V_Positive[index_positive]=TA[0];
V_Negative[index_negative]=TA[n-1];
index_positive++;
index_negative++;
}

 void display_DecessionMatrix()
            {
cout<<"DECESSION MATRIX: \n";
for(int i=0;i<ROWS;i++)
{
for(int j=0;j<COLS;j++)
{
cout<<D[i][j]<<"\t";
}
cout<<endl;
    }
}
void display_normalizedMatrix()
            {
cout<<"\nNORMALIZED MATRIX:\n";
for(int i=0;i<ROWS;i++)
{
for(int j=0;j<COLS;j++)
{
cout<<N[i][j]<<"   ";
}
cout<<endl;
  }
}

void display_Weitage()
{
double sum=0;
cout<<endl<<endl;
for(int i=0;i<COLS;i++)
{
cout<<" W"<<i+1<<":"<<W[i]<<"\t";
sum=sum+W[i];
}
cout<<"\nW1+W2+W3+W4+W5+W6+W7: "<<sum;
            }

void display_weitageMatrix()
            {
cout<<"\n\nWEITAGE MATRIX:\n";
for(int i=0;i<ROWS;i++)
{
for(int j=0;j<COLS;j++)
{
cout<<V[i][j]<<"     ";
}
cout<<endl;
   }
}
double SPositive(double V[][COLS],double V_PosValue[],int nrow)
            {
double sum=0;
```

```
for(int ncol=0;ncol<COLS;ncol++)
{
  //cout<<V[nrow][ncol]<<"--
"<<V_Positive[ncol]<<endl;
sum=sum+pow((V[nrow][ncol]-
V_PosValue[ncol]),2);
        }
return sqrt(sum);
}
double SNegative(double V[][COLS],double
V_NegValue[],int nrow)
        {
double sum=0;
for(int ncol=0;ncol<COLS;ncol++)
{
sum=sum+pow((V[nrow][ncol]-
V_NegValue[ncol]),2);
}
return sqrt(sum);
}
void display_SPos_SNeg()
        {
cout<<"\nS_POSITIVE: ";
for(int i=0;i<ROWS;i++)
{
cout<<S_Positive[i]<<"  ";
        }
cout<<"\nS_NEGATIVE: ";
for(i=0;i<ROWS;i++)
{
cout<<S_Negative[i]<<"  ";
                }
        }
void display_CArray()
{
cout<<"\nC= ";
for(int i=0;i<COLS;i++)
     {
cout<<C[i]<<"  ";
    }
}
public:

Robo()
{
double r=0;

for(int i=0;i<COLS;i++)
{
squareMatrix_SQRT(D,ROWS,i);
}
for(i=0;i<ROWS;i++)
{
for(int j=0;j<COLS;j++)
{
N[i][j]=D[i][j]/R[j];
                        }
                }
        }
void create_Weitage()
{
double sum=0;
cout<<endl<<"\nInsert Weitge for Each Attribute:
";
for(int i=0;i<COLS;i++)
{
cin>>W[i];
}
for(i=0;i<COLS;i++)
{
sum=sum+W[i];
}
if((int)sum!=1)
{
cout<<"\nAddition of Weitage is not equal to 1
Please check: \n";
create_Weitage();
                }
        }
void weitageMatrix()
        {
call_weitageMatrix(N);
}

void maxWeitage_V_Matrix()
{
/*
double R[ROWS][COLS]={
1,2,3,4,5,6,7,
2,3,4,5,6,7,8,
9,4,5,6,7,8,9,
1,2,3,4,5,6,7,
2,3,4,5,6,7,8,
3,4,5,6,7,8,9,
4,5,6,7,8,9,1,
};
*/
for(int ncol=0;ncol<COLS;ncol++)
{
insert_TempArray(V,ROWS,ncol);
                }
        }
void SPos_SNeg()
{
//      cout<<"\ncheck\n";
for(int nrow=0;nrow<ROWS;nrow++)
{

S_Positive[nrow]=SPositive(V,V_Positive,nrow);

S_Negative[nrow]=SNegative(V,V_Negative,nrow
);
                }
        }
void TOPSIS_Rank()
        {
```

```
for(int i=0;i<COLS;i++)
{
C[i]=S_Negative[i]/(S_Positive[i]+S_Negative[i]);
}
//sortArray(C,COLS);
}
void display_TOPSIS_Sorting()
{
double temp[COLS];
double tvalue;

for(int i=0;i<COLS;i++)
{
temp[i]=C[i];
//          cout<<C[i]<<"   ";
}
for(i=0;i<COLS;i++)
{
for(int j=i+1;j<COLS;j++)
{
if(temp[i]<=temp[j])
{
tvalue=temp[i];
temp[i]=temp[j];
temp[j]=tvalue;
        }
    }
}
for(i=0;i<COLS;i++)
{
if(temp[0]==C[i])
{
cout<<"\n Result is "<<i+1;
                              }
                    }
          }

void displayResult()
          {
clrscr();
display_DecessionMatrix();
display_normalizedMatrix();
display_Weitage();
display_weitageMatrix();

cout<<"\nV_Positive Array: ";
for(int i=0;i<COLS;i++)
{
cout<<" "<<V_Positive[i];
}
cout<<"\nV_Negative Array: ";
for(i=0;i<COLS;i++)
{
cout<<" "<<V_Negative[i];
}
display_SPos_SNeg();
display_CArray();
display_TOPSIS_Sorting();
          }
};
int Robo::index_positive=0;
int Robo::index_negative=0;
void main()
{
clrscr();
Robo r;
r.create_DecessionMatrix();
r.normalizedMatrix();
r.create_Weitage();
r.weitageMatrix();
r.maxWeitage_V_Matrix();
r.SPos_SNeg();
r.TOPSIS_Rank();
r.displayResult();
getch();
}
```

V. RESULT AND DISCUSSION

Relative closeness to the ideal solution obtained as:

$$C_1^+ = 0.327331$$
$$C_2^+ = 0.326999$$
$$C_3^+ = 0.202418$$
$$C_4^+ = 0.349679$$
$$C_5^+ = 0.324773$$
$$C_6^+ = 0.363687$$
$$C_7^+ = 0.859654$$

The software is developed for robot selection procedure based on Multiple Attribute Decision Making (MADM) approach. Analysis of the software model was done with the help of example. Result obtained with this model is shown below.

Table 3

Sr. No.	**Alternatives**	**TOPSIS—closeness to the +ve benchmark robot *C****	**Rank based on *C****
1	ASEA-IRB 60/2	0.327331	4
2	Cybotech V 15 Electric Robot	0.326999	5
3	Hitachi America Process Robot	0.202418	7
4	Unimation Puma 500/600	0.349679	3

Sr. No.	Alternatives	TOPSIS—closeness to the +ve benchmark robot *C**	Rank based on *C**
5	Kuka Robotics India Pvt. Ltd. (KR 360 L150-2P)	0.324773	6
6	Precision Automation & Robotics India Pvt.Ltd	0.363687	2
7	Rhythmsoft Robotics & Automation Pvt. Ltd (KR 500 570 – 2PA)	0.859654	1 ←

Arrow shows the best alternative according to user requirement.

Thus the robots are ranked in order of preference based on the attributes selected. For the purchase of a new robot, the management can use the above ranking effectively to select the robot, which will be best suitable for the application and is based on this set together with other considerations

The result calculated mathematically is same as that of what we have got from the output of the computer program within permissible limit of error. Hence, it can be safely conclude that the software developed serves its purpose of selecting best robot from different alternatives according user requirement.

REFERENCES

[1] Zeshui Xu, "a method for multiple attribute decision making with incomplete weight information in linguistic setting", Elsevier Science Publishers, Vol.-20,Issue 8, 719-725

[2] R. Venkata Rao and K.K. Padmanabhanb, "Selection, identification and comparison of

[3] industrial robots using diagraph and matrix methods", Year 2006, Vol. 22, Issue 4, 373-383

[4] A._I. € Olc_er and A.Y. Odabas_i, "A new fuzzy multiple attributive group decision making methodology and its application to propulsion system selection problem", European Journal of Operational Research, Year 2005,Vol.-166, Issue 1, 93-114.

[5] Jian Chen and Song Lin, "An interactive neural network-based approach for solving multiple criteria decision-making problems", Elsevier Science Publishers, Year 2003, Vol- 36, Issue 2, 137-146.

[6] Jian Ma, "An Approach to Multiple Attribute Decision Making Based on Preference Information on Alternatives", Elsevier North-Holland, Year 2002, Vol.-131, Issue 1, 101-106.

[7] Layek Albde-Malek and Lars Johan Resare, "Algorithm based decision support system for the concerted selection of equipment in machining/assembly cells", International Journal of Production Research, Year 2000, Vol-38, Issue 2,323-339.

[8] Haymwantee P. Singh & Wilfred V. Huang, "A decision support system for robot selection", Elsevier Science Publishers B. V., Year 1991, vol.-7, issue 3, 273-283