New Approach to Solve Assignment Problem using MATLAB

Ghadle Kirtiwant P, Muley Yogesh M

Department of Mathematics, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad -431004 (INDIA)

Abstract: **Assignment problem is a special case of transportation problem, in which the objective is to minimized total cost by assigning 'm' jobs to 'n' machines. By using MATLAB coding and some modification in ROA method, optimal solution can be trace for assignment problem within seconds. MATLAB coding result has given for various orders of illustrations.**

Keyword: ROA method, Assignment Problem, Linear Programming, MATLAB coding.

I. INTRODUCTION

Assignment Problem (AP) is completely degenerated form of a Transportation Problem. It appears in some decision-making situations. Such as assign tasks to machines, workers to jobs etc. AP refers to another special class of Linear Programming Problem in which the objective is to assign a number of resources to the equal number of activities at a minimum cost (or maximum profit). Different methods have been presented for Assignment Problem; See [1, 3-7].

AP was developed and published by Harold W. Kuhn (1995), who gave the name "Hungarian Method" because the algorithm was largely based on the earlier works of two Hungarian Mathematicians: Dénes Kőnig and Jenő Egerváry.

Moreover MATLAB is powerful software package. The name MATLAB stands for MATrix LABoratory, it deals with matrix (array). MATLAB can be used for math computations, modeling and simulations, data analysis and processing, visualization and graphics, and algorithm development, and has many built in tools for solving problems. It is also possible to write programs in MATLAB, which are essentially groups of commands that are executed sequentially [8, 9].

Now days MATLAB is widely used mathematics such as MATLAB with Numerical method, Differential Equation, Operation Research, Fuzzy Logic etc., in this article MATLAB coding is used to solve Assignment Problem. This gives optimal solution within fraction of seconds.

II. ASSIGNMENT BASED FORMULATION

Let Assignment Problem of n resources to n activities so as to minimize the overall cost or time in such a way that each resources can associate with one and only one job. The cost matrix (c_{ij}) is given as below.

$$\begin{array}{c} \\ \text{Re}source \end{array} \begin{array}{c} \begin{array}{cccc} A_1 & A_2 & \cdots & A_n \end{array} \\ \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1n} \\ C_{21} & C_{22} & \cdots & C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n1} & C_{n2} & \cdots & C_{nn} \end{pmatrix} \\ \begin{array}{cccc} 1 & 1 & \cdots & 1 \end{array} \end{array} \begin{array}{l} \text{Activity} \\ \text{Available} \\ \begin{array}{c} 1 \\ 1 \\ \vdots \\ 1 \end{array} \\ \\ \end{array}$$

$$\text{Re}quirement$$

The cost matrix is same as that of a Transportation Problem except that availability at each of the resources and the requirement at each of the destinations is unity.

Let x_{ij} denote the assignment of i^{th} resources to j^{th} activity such that,

$$x_{ij} = \begin{cases} 1; & \text{if resource 'i' is assigned to activity 'j'.} \\ 0; & \text{otherwise} \end{cases}$$

Then the mathematical formulation of the assignment problem is,

Subject to the constraints

$$\text{Mininmize } z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \qquad (1)$$

$$\sum_{i=1}^{n} x_{ij} = 1 \text{ and } \sum_{j=1}^{n} x_{ij} = 1 \; ; \; x_{ij} = 0 \text{ or } 1 \qquad (2)$$

For all i=1, 2, …, n and j= 1,2,…,n.

III. ALGORITHM – MODIFIED REVISED ONES ASSIGNMENT METHOD (MROA)

This method is modification of revised ones assignment method [2, 3]

- Divide each row by $\sum_{i=1}^{n} a_i$.
- Divide each column by $\sum_{j=1}^{n} b_j$.
- Check optimality (draw lines to covers all one) if it is equal to *'n'* then go to last step.

- If drawn lines < n, then consider all element less than 1.5 in matrix as 1, and again check optimality.
- If still drawn lines < n, then choose smallest element d_i from uncovered row or column and divide this d_i from uncovered row or column.
- Repeat second last step until optimal solution is attained.

Or

- Divide each row by $\sum_{i=1}^{n} a_i$.
- Divide each column by $\sum_{j=1}^{n} b_j$.
- Check optimality (draw lines to covers all one) if it is equal to *'n'* then go to last step.
- If drawn lines < n, then consider all element less than 1.5 in matrix as 1, and again check optimality.
- To select 1 from each row and column apply following rules
    1. Select only one 1 from each rows, i.e. if more than one 1 occurred in row then ignore temporarily.
    2. Select only one 1 from each columns, i.e. if more than one 1 occurred in column then ignore temporarily.
    3. After selection of single 1 from each row and column, then delete corresponding row and column.
    4. If still value remains in matrix then select minimum value from each row and delete corresponding row and column.
    5. Repeat step 3 and 4 until optimal solution is obtained.

III. MATLAB CODING FOR ASSIGNMENT PROBLEM BY MROA

It is assumed that reader has basic knowledge of MATLAB programming. In this problem, 100 random number of sample illustration solved for matrix 3x3, 4x4, 5x5. This gives optimal result within fraction of seconds, for this we used system which contains intel core i3 processor, 4 GB RAM, Window 7 operating system.

```
%% Simply copy paste this program in M-script file
%% MATLAB PROGRAMMING FOR ASSIGNMENT PROBLEM
%% BY: Dr. Kirtiwant P Ghadle, Yogesh M Muley
clc
tic;
x= randi([2,10],3,3);
xnv=x;
 x1=x;
minr= min(x,[],2);%nnminr(x);
 [r c]=size(x);
for i=1:r
     for j=1:c
          x(i,j)=x(i,j)/minr(i);
     end
 end
 x;
  minc=min(x,[],1);%nnminr(x')';
for i=1:r
     for j=1:c
          x(j,i)=x(j,i)/minc(i);
     end
end
 x;
  y=x;
 for i=1:r
     for j=1:c
          if x(i,j)<=1.1
               x(i,j)=1;
          end
     end
 end
 x;
count=0;
 for i=1:r
 for i=1:r
     for j=1:c
        if x(j,i)==1
         count = count+ 1;
        end
     end
     if count <=1
     for I=i
          for j=1:c
               if x(j,I)==1
                x(j,I)= x(j,I)*100;
               end
          end
          end
     end
      count=0;
 end
 x;
 cmp= nnmaxr(x);
 iv= find(cmp==100)';
 k= length(iv);
 for i=1:k
     for j=1:c
          if x(iv(i),j)~=100
               x(iv(i),j)=0;
          end
       end
 end
 x;
cmp1= nnmaxr(x');
 iv1= find(cmp1==100)';
 k1= length(iv1);
 for i=1:k1
     for j=1:c
          if x(j,iv1(i))~=100
```

```
            x(j,iv1(i))=0;
        end
      end
end
end
x;
for i=1:r
    for j=1:c
        if x(i,j)==100
            x(i,j)=1;
        end
    end
end
x;
for i=1:r
    for j=1:c
if x(i,j)==1
        x(i,j)=x(i,j)*x1(i,j);
end
    end
end
x;
y1=x;
count=0;
for i=1:r
    for j=1:c
        if x(i,j)~=0
            count=count+1;
        end
    end
end
  if count~=r
   y
   x=y;
    [r c]=size(y);
    y=x;
for i=1:r
    for j=1:c
        if x(i,j)<1.5
            x(i,j)=100;
        end
    end
end
x
 for i=1:r
 minx=min(min(x))
for i=1:r
    for j=1:c
        if x(i,j)~=100
            x(i,j)=x(i,j)/minx;
        end
    end
end
 end
for i=1:r
    for j=1:c
        if x(i,j)==100
            x(i,j)=1;
        end
    end
end
  count=0;
for i=1:r
for i=1:r
    for j=1:c
       if x(j,i)==1
        count = count+ 1;
       end
    end
    if count <=1
    for I=i
        for j=1:c
            if x(j,I)==1
             x(j,I)= x(j,I)*100;
            end
        end
        end
    end
     count=0;
end
x;
cmp= nnmaxr(x);
iv= find(cmp==100)';
k= length(iv);
for i=1:k
    for j=1:c
        if x(iv(i),j)~=100
            x(iv(i),j)=0;
        end
      end
end
x;
cmp1= nnmaxr(x');
iv1= find(cmp1==100)';
k1= length(iv1);
for i=1:k1
    for j=1:c
        if x(j,iv1(i))~=100
            x(j,iv1(i))=0;
        end
      end
end
end
 for i=1:r
    for j=1:c
        if x(i,j)==100
            x(i,j)=1;
        end
        if x(i,j)==0
            x(i,j)=100;
        end
    end
end
x
```

```
y= x
y=y.*xnv
      for i=1:r
          minr= min(min(y))
                 [yc yr]= find(y'==minr)
                    k1= length(yr);
     for m=1:1 % instead of k1 we will
take 1st whatever value come
      y(yr(m),yc(m)) = y(yr(m),yc(m))+9;
     for j=1:c
                 if
y(yr(m),j)~=y(yr(m),yc(m))
                           y(yr(m),j)=100;
                end
              end
                    for j=1:c
                         if
y(j,yc(m))~=y(yr(m),yc(m))

y(j,yc(m))=100;
                           end
                      continue
                   end
                end
          end
 y
 x=y;
 for i=1:r
     for j=1:c
         if x(i,j)==100
             x(i,j)=0;
         end
         if x(i,j)~=0
             x(i,j)=1;

         end
     end
 end
 x;
  z=xnv.*x
Minvalue=(sum(sum(z)))
 end
  if count ==r
     x
     Minvalue_z=sum(sum(x))
  end
xnv
toc;
```

By using this coding the following result obtain:

Comparison with optimal solutions for random instances				
Sr. No	**Matrix Size (mxn)**	**Number of Sample solved**	**Efficiency (%)**	**Time (sec)**
1	3x3	100	88	0.01 sec
2	4x4	100	82	0.02 sec
3	5x5	100	73	0.03 sec

The performance of this MATLAB coding is satisfactory, reliable, efficient and faster.

CONCLUSION

New approach is used to solve Assignment Problem using MATLAB, which is faster and reliable. Assignment Problem with MATLAB has great practical and theoretical importance. The programming in MATLAB will save lot of time from complex and iterative calculations. It is also applicable for Travelling Salesman Problem.

REFERENCES

[1] Dimitri. P. Bertsekas., A New Algorithm for the Assignment Problem, *Mathematical Programming*, 21 (1981), 152-171.

[2] Ghadle Kirtiwant P, Muley Yogesh M, Revised Ones Assignment Method for Solving Assignment Problem, *Journal of Statistics and Mathematics*, ISSN: 0976-8807 & E-ISSN: 0976-8815, Volume 4, Issue 1 (2013), 147-150.

[3] Hadi Basirzadeh, Ones Assignment Method for solving Assignment Problems, *Applied Mathematical Sciences*, Vol. 6, 2012, No. 47 (2012), 2345-2355.

[4] M.B. Wright, Speeding up the Hungarian Algorithm, *Computers Opns Res*., Vol.17, No.1(1990), 95-96.

[5] M.L. Balinski, R.E. Gomory., A Primal Method for the Assignment and Transportation Problems, *Management Science*, Vol.10 (1964), No.3.

[6] Ming.S. Hung, Walter. O. Rom., Solving Assignment Problem by Relaxation, *Operation Research*, Vol.28, No.4 (1980), 969-982.

[7] Roy Jonker, Ton Volgenant., Improving the Hungarian Assignment Algorithm, *Operation Research letters*, Volume 5 (1986), Number 4.

[8] Amous Gilat,MATLAB: An Introduction with Applications, John Wiley & Sons, Inc.(2004).

[9] Stormy Attaway, MATLAB: A Practical Introduction to Programming and Problem Solving, Elsevier, Inc (2009).