THOMSON REUTERS ENDNOTE

[Sengamalaselvi* *et al.*, 6(1): January, 2017]
IC™ Value: 3.00

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

SOLVING TRANSPORTATION PROBLEM BY USING MATLAB

Dr. J. Sengamalaselvi*

* Assistant Professor, Department of Mathematics, SCSVMV University, Kanchipuram Tamilnadu India

DOI: 10.5281/zenodo.259588

ABSTRACT

In the area of Linear Programming Problem (LPP), modeling of Transportation Problem (TP) is fundamental in solving most real life problems as far optimization is concerned. MATLAB is used for treating programming of LPP, a condition referred to as M-File that can result from codes. The Paper discusses to study TP that would calculate the use of MATLAB codes using a mathematical modeling. The model develops the transportation solution for the North West Corner Rule, Least Cost Method, Vogel's Approximation Method, and Modi method for the TP. It is clear that a lot of effort has been involved in by many researchers in inquire about of appropriate solution methods to such problem. Furthermore, analytical approach and MATLAB coding are the methods used by most researchers in the application of these efficient proposed techniques. In this paper, an equivalent MATLAB coding was written that would support in the computation of such problems with easiness especially when the problem at LPP and TP. For each model, we use a combination of analytical method and MATLAB coding to study the easiest way that would be efficient while find the solution of different problems. MATLAB is the powerful computational tool in operation research. The MATLAB coding method is better than analytical method for solving TP. This model gives us good result in Transportation problem.

KEYWORDS: Mat lab commands, MODI Method, Transportation problem, North-west corner method, least cost Method, Vogel's approximation method.

INTRODUCTION

The Transportation problem involves finding the lowest-cost plan for distributing stocks of goods or supplies from multiple origins to multiple destinations that demand the goods. The transportation model can be used to determine how to allocate the supplies available from the various factories to the warehouses that stock or demand those goods, in such a way that total shipping cost is minimized. Usually, analysis of the problem will produce a shipping plan that pertains to a certain period of time (day, week), although once the plan is established, it will generally not change unless one or more of the parameters of the problem (supply, demand, unit shipping cost) changes.

The transportation model starts with the development of a feasible solution, which is then sequentially tested and improved until an optimal solution is obtained. The description of the technique on the following pages focuses on each of the major steps in the process in this order:

TRANSPORTATION PROBLEM

Solve the transportation problem

From	**TO**				**Supply**
	1	**2**	**3**	**4**	**6**
	4	**3**	**2**	**0**	**8**
	0	**2**	**2**	**1**	**10**
Demand	**4**	**6**	**8**	**6**	

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Matlab M-File

```
clc;
clear all;
close all;
x=input('enter the transportation matrix');
x=sd('supply; demand');
[m n]=size(x);
x1=zeros(m,n);
sumc=0;
sumr=0;
for i=1:m-1
sumc=sumc+x(i,n);
end
for j=1:n-1
sumr=sumr+x(m,j);
end
if(sumc == sumr)
for i=1:m
for j=1:n
x11=min(x(i,n),x(m,j));
x1(i,j)=x11;
x(i,n)=x(i,n)-x11;
x(m,j)=x(m,j)-x11;
end
end
else
disp('unbalanced transportation');
end
xre=0;
for i=1:m-1
for j=1:n-1
xre=xre+(x(i,j).*x1(i,j));
end
end
disp(['the transportation cost is ',num2str(xre)]);
```

Matlab Command Window

```
x=input('1 2 3 4; 4 3 2 0; 0 2 2 1');
x=sd('6 8 10; 4 6 8 6');
disp output(['the transportation cost is ',num2str(xre)]);
OUTPUT: The Transportation Cost is  28.
```

NORTH WEST CORNER RULE

Determine basic feasible solution the following transportation problem using North West corner rule

Origin			Sink				Supply
		A	B	C	D	E	
	P	2	11	10	3	7	4
	Q	1	4	7	2	1	8
	R	3	9	4	8	12	9
Demand		3	3	4	5	6	

Matlab M-File

```
clc;
clear all;
```

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

```
close all;
x=input('enter the transportation matrix');
x=sd('supply; demand');
[m n]=size(x);
x1=zeros(m,n);
sumc=0;
sumr=0;
for i=1:m-1
sumc=sumc+x(i,n);
end
for j=1:n-1
sumr=sumr+x(m,j);
end
if(sumc == sumr)
for i=1:m
for j=1:n
x11=min(x(i,n),x(m,j));
x1(i,j)=x11;
x(i,n)=x(i,n)-x11;
x(m,j)=x(m,j)-x11;
end
end
else
disp('unbalanced transportation');
end
xre=0;
for i=1:m-1
for j=1:n-1
xre=xre+(x(i,j).*x1(i,j));
end
end
disp(['the transportation cost is ',num2str(xre)]);
```

Matlab Command Window

```
x=input('2 11 10 3 7; 1 4 7 2 1; 3 9 4 8 12');
x=sd('4 8 9; 3 3 4 5 6');
disp output(['the transportation cost is ',num2str(xre)]);
Output: The Transportation Cost is 153
```

VOGEL'S APPROXIMATION METHOD

Find the optimal solution of the following problem using vogel's approximation method

Origin		Destination			Supply
		X	Y	Z	
	P	1	2	0	30
	Q	2	3	4	35
	R	1	5	6	35
Demand		30	40	30	

Matlab M-File

```
clc;
clear all;
close all;
function [ibfs,objCost] = vogel's approximation(data)
```

```
x=input('enter the transportation matrix');
x=sd('supply; demand');
cost = data(1:end-1,1:end-1);
demand = data(end,1:end-1);
supply = data(1:end-1,end)';
ibfs = zeros(size(cost));
ctemp = cost;
while length(find(demand==0)) < length(demand)
length(find(supply==0)) < length(supply);
prow = sort(ctemp,1);
 prow = prow(2,:) - prow(1,:);
pcol = sort(ctemp,2);
pcol = pcol(:,2) - pcol(:,1);
[rmax,rind] = max(prow);
[cmax,cind] = max(pcol);
ifrmax>cmax
[~,mind] = min(ctemp(:,rind));
[amt,demand,supply,ctemp] =hkdemandsupply(demand,supply,rind,mind,ctemp);
ibfs(mind,rind) = amt;
elseifcmax>= rmax[~,mind] = min(ctemp(cind,:));
[amt,demand,supply,ctemp] = chkdemandsupply(demand,supply,mind,cind,ctemp);
ibfs(cind,mind) = amt;
end
end
objCost = sum(sum(ibfs.*cost));
disp(['the transportation cost is ', sum(sum(ibfs.*cost))]);
```

Matlab Command Window

```
x=input('1 2 0; 2 3 4; 1 5 6');
x=input('30 35 35; 30 40 30');
disp output(['the transportation cost is ', sum(sum(ibfs.*cost))]);
OUTPUT: The Transportation Cost is 160.
```

MODI METHOD

Find the optimal solution of the following problem using MODI method

origin		**Destination**			**Supply**
		X	**Y**	**Z**	
	P	**1**	**2**	**0**	**30**
	Q	**2**	**3**	**4**	**35**
	R	**1**	**5**	**6**	**35**
Demand		**30**	**40**	**30**	

Matlab M-File

```
clc;
clear all;
close all;
function [ibfs,objCost] = modi method(data)
x=input('enter the transportation matrix');
x=sd('supply; demand');
val = -1;
whileval< 0[prow,pcol]=find(ibfs> 0);
occupiedCells=[prow,pcol];
```

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

```
[prow,pcol]=find(ibfs==0);
unoccupiedCells = [prow,pcol];
r = 0;
k = [column occupied];
fori = 1:length(occupiedCells(1,:)) ri = occupiedCells(1,i);
kj = occupiedCells(2,i);
[r,k] = occupiedSystemSolve(r,k,ri,kj,cost);
end improvementIndex = zeros(length(unoccupiedCells(1,:)),3);
fori = 1:length(unoccupiedCells(1,:)) ri = unoccupiedCells(1,i);
kj = unoccupiedCells(2,i);
e = cost(ri,kj) - r(ri) - k(kj);
improvementIndex(i,:) = [ri,kj,e];
end [val,ind] = min(improvementIndex(:,end));
ifval< 0 %check whether improvement is required ri = improvementIndex(ind,1);
kj = improvementIndex(ind,2);
disp(['Create a circuit around cell (' num2str(ri) ',' num2str(kj) ')' ]);
 circuitImproved = [ri,kj,0];
n = input('Enter number of element that forms the circuit: ');
fori = 1:n nCells = input(['Enter the index of cell ' num2str(i) ' that forms the circuit: ']);
if mod(i,2) == 0 circuitImproved(i+1,:) = [nCells, ibfs(nCells(1),nCells(2))];
else circuitImproved(i+1,:) = [nCells, -ibfs(nCells(1),nCells(2))];
endend ibfs = reallocateDemand(ibfs,circuitImproved);
disp(ibfs) objCost = sum(sum(ibfs.*cost));
endend function [r,k] = occupiedSystemSolve(r,k,ri,kj,cost)
if length(r)>=rik(kj) = cost(ri,kj)-r(ri);
elser(ri) = cost(ri,kj)-k(kj);
end function [y,demand,supply,ctemp] = chkdemandsupply(demand,supply,ded,sud,ctem) tempd
 = demand;
temps = supply;
iftempd(ded) > temps(sud) temps(sud) = 0;
tempd(ded) = demand(ded) - supply(sud);
y = supply(sud);ctem(sud,:) = inf;
else
iftempd(ded) < temps(sud) tempd(ded) = 0;
temps(sud) = supply(sud) - demand(ded);
y = demand(ded);
ctem(:,ded) = inf;
elseiftempd(ded) == temps(sud) tempd(ded) = 0;
temps(sud) = 0;
y = demand(ded);
ctem(:,ded) = inf;
ctem(sud,:) = inf;end demand =tempd;
supply = temps;
ctemp = ctem;
disp(['the transportation cost is ', sum(sum(ibfs.*cost))]);
```

Matlab Command Window

```
x=input('1 2 0; 2 3 4; 1 5 6');
x=input('30 35 35; 30 40 30');
disp output(['the transportation cost is ', sum(sum(ibfs.*cost))]);
```

Output: The Transportation Cost is 160

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

HUNGARIAN METHOD

The assignment cost of assigning any one operator to any one machine is given in the following table

		Operator			
		I	II	III	IV
Machine	A	10	5	13	15
	B	3	9	18	3
	C	10	7	3	2
	D	5	11	9	7

Matlab M-File

```
clc;
clear all;
close all;
function [assignment,cost] = munkres(costMat)
x=input('enter the transportation matrix');
 [assignment,cost] = munkres(magic(5));
disp(assignment);
disp(cost);
n=16;
A=rand(n);
tic[a,b]=munkres(A);
A=rand(10,7);
A(A>0.7)=Inf;
[a,b]=munkres(A);
A = [1 3 Inf; Inf Inf 5; Inf Inf 0.5];
[a,b]=munkres(A);
assignment = zeros(1,size(costMat,1));
cost = 0;
validMat = costMat == costMat & costMat < Inf;
bigM = 10^(ceil(log10(sum(costMat(validMat))))+1);
costMat(~validMat) = bigM;
validCol = any(validMat,1);
validRow = any(validMat,2);
nRows = sum(validRow);
nCols = sum(validCol);
n = max(nRows,nCols);
if ~n
   return
end
maxv=10*max(costMat(validMat));
dMat = zeros(n) + maxv;
dMat(1:nRows,1:nCols) = costMat(validRow,validCol);
minR = min(dMat,[],2);
minC = min(bsxfun(@minus, dMat, minR));
zP = dMat == bsxfun(@plus, minC, minR);
starZ = zeros(n,1);
while any(zP(:))
[r,c]=find(zP,1);
starZ(r)=c;
zP(r,:)=false;
zP(:,c)=false;
end
while
   if all(starZ>0)
```

```
        break
    end
    coverColumn = false(1,n);
    coverColumn(starZ(starZ>0))=true;
    coverRow = false(n,1);
    primeZ = zeros(n,1);
[rIdx, cIdx] =
find(dMat(~coverRow,~coverColumn)==bsxfun(@plus,minR(~coverRow),minC(~coverColumn)));
    while
        cR = find(~coverRow);
        cC = find(~coverColumn);
        rIdx = cR(rIdx);
        cIdx = cC(cIdx);
                while ~isempty(cIdx)
            uZr = rIdx(1);
            uZc = cIdx(1);
            primeZ(uZr) = uZc;
            stz = starZ(uZr);
            if ~stz
                break;
            end
            coverRow(uZr) = true;
            coverColumn(stz) = false;
            z = rIdx==uZr;
            rIdx(z) = [];
            cIdx(z) = [];
            cR = find(~coverRow);
            z = dMat(~coverRow,stz) == minR(~coverRow) + minC(stz);
            rIdx = [rIdx(:);cR(z)];
            cIdx = [cIdx(:);stz(ones(sum(z),1))];
        end
[minval,rIdx,cIdx]=outerplus(dMat(~coverRow,~coverColumn),minR(~coverRow),minC(~coverColumn));
            minC(~coverColumn) = minC(~coverColumn) + minval;
            minR(coverRow) = minR(coverRow) - minval;
        else
            break
        end
    end
    rowZ1 = find(starZ==uZc);
    starZ(uZr)=uZc;
    while rowZ1>0
        starZ(rowZ1)=0;
        uZc = primeZ(rowZ1);
        uZr = rowZ1;
        rowZ1 = find(starZ==uZc);
        starZ(uZr)=uZc;
    end
end
rowIdx = find(validRow);
colIdx = find(validCol);
starZ = starZ(1:nRows);
vIdx = starZ <= nCols;
assignment(rowIdx(vIdx)) = colIdx(starZ(vIdx));
pass = assignment(assignment>0);
pass(~diag(validMat(assignment>0,pass))) = 0;
assignment(assignment>0) = pass;
cost = trace(costMat(assignment>0,assignment(assignment>0)));
```

```
function [minval,rIdx,cIdx]=outerplus(M,x,y)
ny=size(M,2);
minval=inf;
for c=1:ny
   M(:,c)=M(:,c)-(x+y(c));
   minval = min(minval,min(M(:,c)));
end
[rIdx,cIdx]=find(M==minval);
disp([' minimum assignment cost ', outerplus(M,x,y)]);
```

Matlab Command Window

```
x=input('10 5 13 15; 3 9 18 3; 10 7 3 2; 5 11 9 7');
disp output([' minimum assignment cost ', outerplus(M,x,y)]);
Output: Minimum Assignment Cost is 16.
```

CONCLUSION

In the field of operations research, modeling of transportation problem is fundamental in solving most real life problems as far optimization is concerned. It is clear that a lot more effort has been put in by many researchers in seek of appropriate solution methods to such problem. Vogel's Approximation Method (VAM), among the class of algorithms provided to solve the Initial Basic Feasible Solution (IBFS) proved to be best. Likewise is the Modified Distribution Method in testing the optimality of the IBFS. However, for some time now, manual calculations and MATLAB are the tools used by most researchers in the application of these efficient proposed techniques. In this work, an equivalent MATLAB program was written that would aid in the computation of such problems with ease especially when the problem at hand has a larger cost matrix. To this effect and to the best of our knowledge, no MATLAB function has been written to handle this problem, although is now obvious that more scientist in the scientific world are into the usage of MATLAB environment. Notwithstanding the fact that people is need such function to make their computations easier. In this paper, a MATLAB function, that is developed to implement the VAM, MODI which helps get the IBFS and Modified Distribution Method, which also test for the optimality of the IBFS based on the assumption that the problem is balanced. MATLAB is used to find easy way for the solution of the Transportation methods.

REFERENCES

[1] Alexander Schrijver, Theory of Linear and Integer Programming. John Wiley & sons
[2] Andersen, D. R., Dennis J. Sweeny, and Thomas A. Williams. 2001. Quantitative
[3] Andersen, D.R., D.J. Sweeney, & T.A. Williams, An Introduction to Management
[4] Anderson, Michael Q. 1982. Quantitative Management Decision Making, Belmont,
[5] Bernd Gärtner, Jiří Matoušek (2006). Understanding and Using Linear Programming, Berlin: Springer
[6] Bierman, H., C.P. Bonini, and W.H. Hausman. Quantitative Analysis for Business Decisions (Richard D. Irwin, Inc., Homewood, IL, 1977). 642 pp. California: Brooks/ Cole Publishing Co.,
[7] Charnes, A., Cooper, W. W., and Henderson, A. (1953),"An Introduction to Linear Programming", Wiley, New York.
[8] Dantzig G. B. and Thapa. M. N. (1997). Linear programming 1: Introduction. Springer-Verlag.
[9] Dantzig G. B. and Thapa. M. N. (2003).Linear Programming 2: Theory and Extensions. Springer-Verlag.
[10] Dantzig, G.B (1963),"Linear Programming and Extensions", Princeton University Press,Princeton, N J,
[11] Dunn, Robert A. and K. D. Ramsing. 1981. Management Science: A Practical Approach
[12] Dykstra, D.P. Mathematical Programming for Natural Resource Management (McGraw-Hill, Inc., New York, 1984). 318 pp.
[13] Edward S. A., and M. Venkatachalapathy, (2011),"Modified Vogel's Approximation Method for Fuzzy Transportation Problems" Applied Mathematical Sciences, Vol. 5 (28), 1367-1372.
[14] Hillier, F.S., and G.J. Lieberman. Introduction to Operations Research, sixth edition (McGraw-Hill, Inc., New York, 1995). 998 pp.
[15] Hitchcock, F.L (1941),"The distribution of a product from several sources to numerouslocalities", J. Math. Phys. 20, 224-230.