ISSN:2320-0790

An International Journal of Advanced Computer Technology

MATLAB Implementation of Vogel's Approximation and the Modified Distribution Methods

Justice Kwame Appati[1], Gideon Kwadwo Gogovi[1], Gabriel Obed Fosu[2]
[1]Department of Mathematics, Kwame Nkrumah University of Science and Technology, Ghana
[2]Department of ICT and Mathematics, Presbyterian University College, Ghana

Abstract: In the field of operations research, modelling 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 LINDO 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.

Keywords: Modified Distribution; Vogel Approximation; Balance Problem; Basic Feasible Solution; MATLAB, Transportation Model; Cost Matrix

I. INTRODUCTION

Transportation problem in the domain of operations research has been one of the aged linear programming application problems developed by Hitchcock by originality [4]. From the nature of the problem and based on the simplex algorithm, more efficient methods were developed [1][2][6]-[10]. Just like the simplex algorithm, an initial basic feasible solution is also required in solving the transportation problem. In this working environment, methods such as Row Minima, Matrix Minima, Column Minima, North-West Corner Rule, Least Cost or the Vogel Approximation is used.

In theory and practice, all transportation problems can be modeled as a standard linear programming problem implying that, simplex method as a solution technique can be employed. However, transportation problem as a special case of linear programming problem can be solved more efficiently using solution schemes such as Steeping Stone Algorithm or Modified Distribution Method (MODI) in search of an optimal solution to the problem.

Stepping Stone Method that was first derived by Charneset. al [1] was intended to serve as an alternative solution to determining the solution at optimality. However, LINDO (Linear Interactive and Discrete Optimization) package serves as a computational tool with built-in properties which handles transportation problems in its explicit form and solves the problem as a standard linear programming problem.

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. Not withstanding the fact that people need such function to make their computations easier.In this paper, a MATLAB function, that is, vogelModi.m was developed to implement the Vogel Approximation Method, 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.

II. MODEL FORMULATION

A Considering m sources S_i $(i = 1,...,m)$ and n destinations D_j $(j = 1,...,n)$,let a_i be the amount of a homogeneous product that needs to be transported from each source S_i to n destinations D_j, in order to satisfy the demand for b_j units of the product. A penalty c_{ij} is associated with each transport of a unit of product from a source i to destination j.

These penalties could be:

1. transportation cost,
2. delivery time,

3. quantity of goods delivered under-used capacity,
4. Other cost factors

The unknown quantity to be transported from Source S_i to destination D_j is represented by the variable x_{ij}.

This is then formulated mathematically as follows:

$$Minimize: Z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

Subject to the constraints:

$$\sum_{j=1}^{n} x_{ij} = b_i, i = 1,2,\cdots,m$$

$$\sum_{i=1}^{m} x_{ij} = a_i,\ j = 1,2,\cdots,n$$

$$x_{ij} \geq 0, i = 1,2\cdots,m,\ j = 1,2,\cdots,n$$

and

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$$

being the balanced condition which is both a necessary and sufficient condition for the existence of a feasible solution [3].

III. SOLUTION METHOD

All paragraphs must be indented as well as justified, i.e. both left-justified and right-justified.

A. Vogel Approximation Method(VAM)

In this paper, Vogel Approximation was selected over the Northwest Corner Rule and Least Cost Method for the study. This is as a result of its ability to provide an Initial Basic Feasible Solution which is nearly optimal. In the application of VAM, the penalties the organization is likely to be exposed to if it transport a product over best routes instead of least-cost routes are first computed for each row and column.

The steps for Vogel's Approximation method can be summarised in the following list:

1. Determine the penalty cost for each row and column by subtracting the lowest cell cost in the row or column from the next lowest cell cost in the same row or column.
2. Select the row or column with the highest penalty cost whiles breaking ties arbitrarily or choosing the lowest cost cell
3. Allocate as much as possible to the feasible cell with the lowest transportation cost in the row or column with the highest penalty cost
4. Repeat Step 1 to 3 until all demands rims requirements have been met.

B. Modified Distribution Method (MODI)

Once the IBFS is predetermined by the Vogel's Approximation, the probability of it not being optimal is there, hence the need for another scheme to improve the optimal solution. Again, in this paper, MODI method was considered because it allows the computation of the improvement indices much quicker for each unused square without drawing all the closed path or circuit.

Due to these operations, MODI saves considerable time over the other methods for improving the basic optimal solution. It works by finding the unused route with the largest negative improvement index. Once identified, a closed circuit is then traced out to determine the maximum number of units that can be shipped.In its application, we compute values for each row sayu_1, u_2, $u_3$that is if there are three rows and for each column say v_1, v_2, v_3 in the transport table. By notation, we let

u_i = value assigned to row i

v_j = value assigned to column j

c_{ij} = cost in the square ij(cost of shipping from a source i to destination j)

The steps of the Modified Distribution Method can be summarized as follows:

1. Develop an initial solution using one of the three methods available.
2. Compute u_i and v_j values for each row and column by applying the formula $u_i + v_j = c_{ij}$to each cell that has an allocation.
3. Compute the cost change, k_{ij}, for each empty cell using $c_{ij} - u_i - v_j = k_{ij}$
4. Allocate as much as possible to the empty cell that will result in the greatest net decrease in cost that is the most negative k_{ij}. Allocate according to the stepping-stone path for the selected cell
5. Repeat Steps 2 through 4 until all k_{ij} values are positive or zero.

I) *The Unbalanced Model:* In the context of a balanced transportation problem, methods for determining the IBFS and the optimal solutions are well spelled out in the previous sections. However, in reality, an unbalanced problem is likely to evolve which will invalidate the usage of the initial schemes discussed and hence must be transformed as explained below.

1. If demand is less than supply, create a fictitious destination (or dummy destination)

2.If demand is greater that supply, create a dummy plant having a capacity exactly equal to the additional demand.

II) *Degeneracy:* In this situation there may be an,
1. Excessive number of used cells in a solution
2. Insufficient number of unused cells in a solution.

When degeneracy exists, a very simple "patch-up" will solve the problem by placing ε although an exceedingly small number into some cells. This only serves as a stepping-stone in resolving the problem, but have no effect on the cost of the solution. However, this value can be ignored if it still remains in the final solution..

IV. MATLAB IMPLEMENTATION

A. Code Usage on a Numerical Examples

Assuming on a great occasion of Country's Ceremonial Event where most of its citizens relocate to witness the event. At the end of the program, participants wish to get to their destinations safely using the public transport. Below are the empty cars in the following cities in Ghana and their demands at various event sites respectively.

TABLE I: SUPPLY NODES

Town	Supply of Cars
Accra	35
Kumasi	60
Sunyani	25

TABLE II: DEMAND NODES

Town	Demand for Cars
Jackson Park	30
Rawlings Park	45
Children's Park	25
Market Central	20

Using the city-to-city distance chart, the organizing committee constructs a mileage table for the preceding towns with Table III displaying the result.

TABLE III: CITY-to-CITY COST MATRIX

	To			
From	Jackson Park	Rawlings Park	Children's Park	Market Central
Accra	50	30	60	70
Kumasi	20	80	10	90
Sunyani	100	40	80	30

The objective of the transportation problem here is then to minimize the total miles covered by each car using the Vogel and MODI.

To solve the above explained transportation problem using the MATLAB code (vogelModi.m) as found in appendix, the problem must first be redesigned as follows:

1. Reformulate the table with each row being the Supply whiles the columns being the Demands if is not already so.
2. Any unbalance problem should be made balance since the code only works for a balance problem.

From the above two steps, the Table 4 should now look like this which is actually a balance problem readily solvable by the algorithm implemented in vogelModi.m.

TABLE IV: A BALANCED COST-DEMAND-SUPPLY TABLE

	DEMAND				
SUPPLY	50	30	60	70	35
	20	80	10	90	60
	100	40	80	30	25
	30	45	25	20	**120**

At the MATLAB command prompt ">>" type any variable say, test, and assigned the matrix to it. Call the vogelModi function with the argument being the variable to compute the problem. That is:

```
>>test = [50, 30, 60, 70, 35; 20, 80, 10, 90, 60; ...100, 40, 80, 30, 25; 30, 45, 25, 20, 120];
>> [ibfs,objCost] = vogelModi(test)
```

Whiles the program is on the run, it might request for the closed circuit to be typed manually to improve the IBFS found till an optimal solution is reached.

V. CONCLUSION

In this study, a MATLAB code was developed for the solving of the transportation problem based on the Vogel's Approximation and the Modified Distribution Method. It was observed that this program will aid most Operation Researchers modeling large transportation problem and wish to use Vogel's Approximation method and the Modified distribution method as its solution techniques for arriving at an optimal solution with less effort

VI. REFERENCES

[1] Charnes, A., Cooper, W. W., and Henderson, A. (1953),"An Introduction to Linear Programming", Wiley, New York.

[2] Dantzig, G.B (1963),"Linear Programming and Extensions", Princeton University Press,Princeton, N J,

[3] Edward S. A., and M. Venkatachalapathy, (2011),"Modified Vogel's Approximation Method for Fuzzy Transportation Problems" Applied Mathematical Sciences, Vol. 5 (28), 1367-1372.

[4] Hitchcock, F.L (1941),"The distribution of a product from several sources to numerouslocalities", J. Math. Phys. 20, 224-230.

[5] MATLAB Classes and Objects (Programming and Data Types). http://www.mathworks.com/access/helpdesk r13/help/techdoc/matlabprog/ch14 oop.html.

[6] Dantzig G. B. and Thapa. M. N. (1997). Linear programming 1: Introduction. Springer-Verlag.

[7] Dantzig G. B. and Thapa. M. N. (2003).Linear Programming 2: Theory and Extensions. Springer-Verlag.

[8] Karl-Heinz B. (1987).The Simplex Algorithm: A Probabilistic Analysis, Algorithms and Combinatorics, Volume 1, Springer-Verlag.

[9] Nering E. D. andTucker, A. W.(1993). Linear Programs and Related Problems, Academic Press.

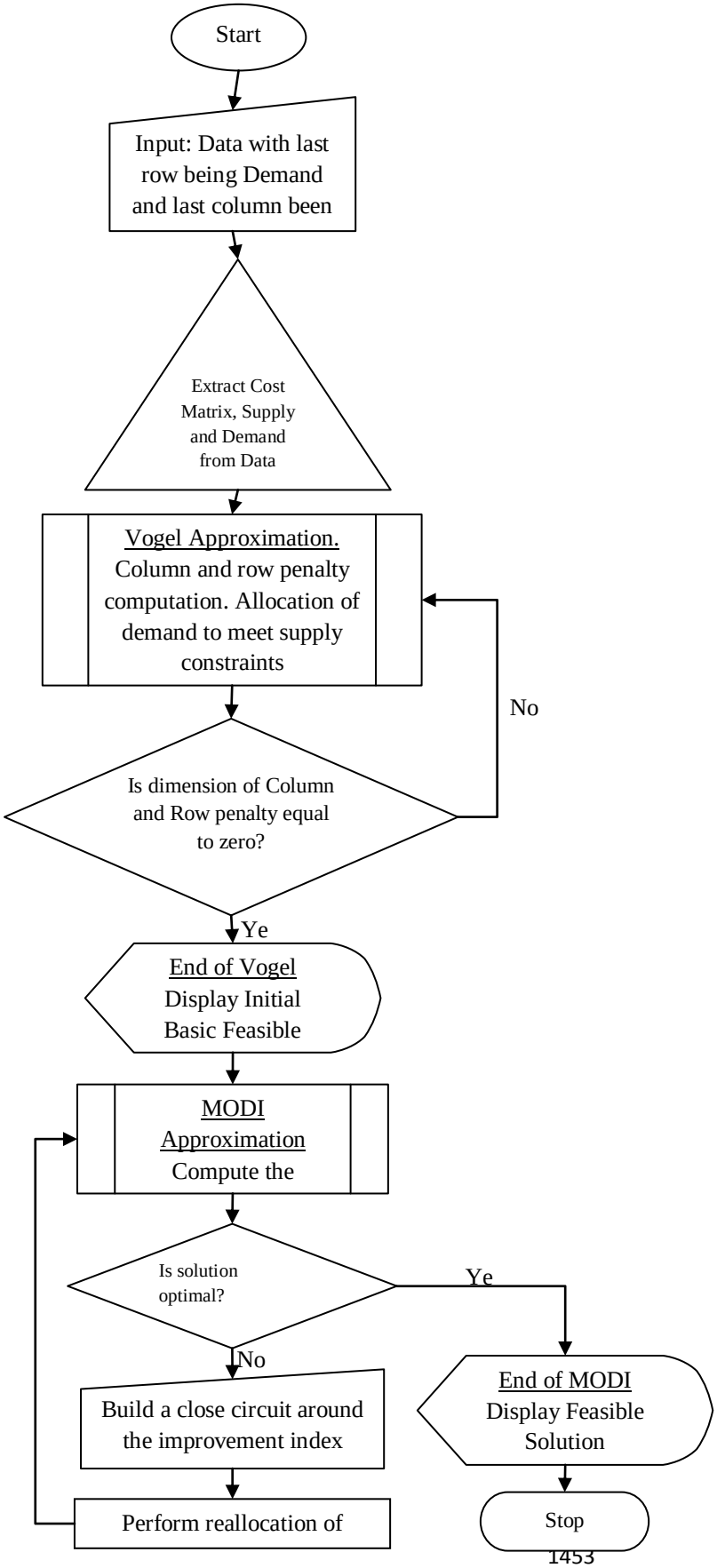
[10] Padberg M. (1999).Linear Optimization and Extensions, Second Edition, Springer-Verlag.

APPENDIX I

```
function [ibfs,objCost] = vogelModi(data)

% ===DATA PREPARATION===

cost = data(1:end-1,1:end-1);demand = data(end,1:end-1);supply = data(1:end-1,end)';ibfs = zeros(size(cost));

% ===VOGEL APPROXIMATION===

ctemp = cost; %temporal cost matrix

while length(find(demand==0)) < length(demand) || length(find(supply==0)) < length(supply)

prow = sort(ctemp,1); prow = prow(2,:) - prow(1,:); %row penalty

pcol = sort(ctemp,2);pcol = pcol(:,2) - pcol(:,1); %column penalty

[rmax,rind] = max(prow);[cmax,cind] = max(pcol);ifrmax>cmax

[~,mind] = min(ctemp(:,rind));[amt,demand,supply,ctemp] = chkdemandsupply(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;endendobjCost = sum(sum(ibfs.*cost));

%===MODIFIED DISTRIBUTION ===

val = -1; whileval< 0[prow,pcol]=find(ibfs> 0);occupiedCells=[prow,pcol]';
[prow,pcol]=find(ibfs==0);unoccupiedCells = [prow,pcol]'; r = 0;k = [];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)

% ===OTHER REQUIRED FUNCTIONS===

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;elseiftempd(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;
```

APPENDIX II

Flowchart of Vogel and Modi


Start
Input: Data with last row being Demand and last column been
Extract Cost Matrix, Supply and Demand from Data
Vogel Approximation. Column and row penalty computation. Allocation of demand to meet supply constraints
No
Is dimension of Column and Row penalty equal to zero?
Ye
End of Vogel Display Initial Basic Feasible
MODI Approximation Compute the
Is solution optimal?
Ye
No
Build a close circuit around the improvement index
End of MODI Display Feasible Solution
Perform reallocation of
Stop