Journal of Mathematics Research

http://ccsenet.org/jmr

Vol. 2, No. 1

February 2010

Applications of Maximal Network Flow Problems in Transportation and Assignment Problems

Vinai K. Singh

Department of Applied Mathematics

Aryabhatt College of Engineering & Technology

Baghpat (UP) 250601, India

E-mail: vinaiksingh@rediffmail.com

Indu Kala Tripathi & Nimisha

Department of Applied Mathematics, IMS Engineering College, Ghaziabad India

Abstract

This paper presents some modifications of Ford-Fulkerson's labeling method for solving the maximal network flow problem with application in solving the transportation and assignment problems. The modifications involve the tree representation of the nodes labeled and the edges used them. It is shown that after each flow adjustment some of the labels can be retained for the next labeling process. Through certain computational aspects it has been suggested that to indicate that with theses the primal-dual approach for solving the transportation and assignment problems is improved to certain extent.

Keywords: Maximal network flow, Labeling process, Transportation & assignment problems

Mathematics Subject Classification 2000: 76M15, 76M35, 60G20.

1. Introduction

New solution techniques have been suggested to solve transportation, mainly in the tree representation of a basis, location of a better adjacent basis, and labeling and relabeling of trees (Glover, 1982; Klingman, 1983). In this paper some of theses ideas are incorporated into the primal-dual approach to the transportation and assignment problems. Each label computed in the labeling method of Ford-Fulkerson (Ford, L.R., 1962) for the maximum flow problems consists of two element. The second element is used for flow augmentation. After each flow adjustment, all labels are discarded and the labeling process starts from the source again. At each labeling process the nodes labeled and the edges used to label them are represented by a forest on the transportation network. It is shown that by using either the predecessor and distance labels (Singh, Vinai K., 2007) or the triple labels (Johnoson, E.I., 1986) on the nodes of the tree, several of the labeled nodes can be retained as "labeled and unscanned" nodes for the next labeling process.

Based on the Ford-Fulkerson's primal- dual approach for solving the transportation problems, it is found the primal-dual method can be improved considerably making it once again competitive with the MODI method for solving the transportation problems. For the assignment problem our computational experience with these modifications has been very encouraging and since the primal-dual approach is similar in the spirit to Hungarian Method, it appears that we have accelerated the Hungarian Method. It should be pointed out that we have not in any sense attempted to "optimize" our codes. Presumably, the computational times can be reduced by appropriate modifications to the codes.

There are a variety of other problems for which the ideas developed in this paper can be useful. For instance; algorithms for solving the Bottleneck Assignment and Bottleneck Transportation problems (Garfinkel, R.S., 1981) involve the solution of maximal flow problems.

2. Definition and notations

Let G =(N, A) be a finite directed network with N representing the node set and A the arc set Let x =w_0, w_1, ..., w_k = y be a sequence of distinct nodes having the property that either (w_{j-1}, w_j) ∈A or(w_j, w_{j-1}) ∈A for j=1,2, ...,k. Singling out, for each i, one of theses two possibilities, the resulting sequence of nodes is called a path from x to y. Arcs (w_{j-1}, w

$_j$) that belong to the path are referred to as forward arcs while arcs (w $_{j+1}$, w $_j$)belonging to the same path referred to as reverse arcs. A cycle is a path from a node to itself. A tree in G = (N,A) is any set of | N | - 1 arcs without cycles. Each tree is pictured vertically in the plane and extending downwards with the highest node as root. In such a rooted tree, for each node x, a distance label d(x) and the triple- label [consisting of predecessor P(x), right- neighbour B(x), and leftmost successor L(x)] can be defined.

3. Algorithms for Solving the Maximal Flow Problem

Let G =(N,A) be a directed network, where each (x, y) ∈ A has associated with it as non negative number c(x,y) which is the capacity of the arc (x, y). The maximal flow problem from source to sink can be stated as:

$$\text{Max Q} = \sum_{y\in\alpha(x)} f(x,y) - \sum_{y\in\beta(x)} f(y,x) = \begin{cases} Q & if \quad x = s \\ 0 & if \quad x \neq s,t \\ -Q & if \quad x = t \end{cases}$$

0≤f(x,y)≤c(x,y), for all (x,y) ∈A

Where s= source node of he network.

t =sink node of the network.

f (x,y)=flow in arc (x,y)

α(x) ={ y ∈ N : (x,y)∈ A}

β(x)={y ∈ N : (y,x)∈ A}

We define the following characterization of the structure of the labeled nodes and the edges used to label them in a particular application of the labeling process (i.e. routine A):

N_k= {$y_{1,}y_{2,.....,}y_q$} as the set of nodes labeled on the kth application of the labeling

process.

[Note that s ∈ N $_k$. further, t ∈ N $_k$ if there is a breakthrough.]

E $_k$ ={(y $_{j-1}$, y $_j$): y_{j-1}, y $_j$ ∈ N $_k$, P(y $_j$)= y $_{j-1}$ and either (y $_{j-1}$, y $_j$) ∈ A or (y $_j$, y $_{j-1}$) ∈A}

The connected graph (N_k, E_k) is obviously a tree. We define the rooted tree T_k=(N_k, E_k) as the tree formed by the graph $(N_k, E_k$) with s as the root.

Let the flow augmenting path obtained by the kth labeling process be represented by

$$F_k = \{s = y_1,\ (y_1,\ y_2),,\ y_2,\ (y_2,\ y_3),\ y_3.,.....,y_{n-1},\ (y_{n-1},\ y_n),\ y_n = t\}$$

Where ($y_{i,}$ $y_{i+1)}$ ∈ E $_k$ for i= 1, 2,. . . . . , n-1.

Define A_k^+ = {*set of forward arcs in* F_k}

A_k^- = {*set of reverse arcs in* F_k}

After the kth flow adjustment, there exists at least one (y $_{j-1}$, y $_j$) ∈ F_k such that either:

(i) f (y $_{j-1}$, y $_j$)=c(y $_{j-1}$, y $_j$)for (y $_{j-1}$, y $_j$) ∈ A_k^+

Or (ii) f (y $_j$, y $_{j-1}$)=0for (y $_j$, y $_{j-1}$) ∈ A_k^-

Let J_k = {(y $_{j-1}$, y $_j$): (y_{j-1}, y $_j$)a binding edge after he kth flow adjustment}

A backtracking algorithm 2 is given below for flow adjustment at a breakthrough. This algorithm does not use the second element (E) of the label which is computed in Ford-Fulkerson's routine A. The algorithm also identifies the set J_k which is later utilized to identify those nodes whose labels remain valid even after the flow adjustment. One computational experience discussed later in the paper and the results given in (Barr, R., F. Glover, 1982) indicate that there is some computational saving in using a backtracking procedure for flow adjustment instead of the∈ function. Furthermore, we note that the improved labeling algorithms 2 and 3 become computationally less attractive if the ∈ function of Ford-Fulkerson is utilized instead of the backtracking procedure.

Algorithm 1: for flow adjustment along F_k using a backtracking routine.

Step 0: Breakthrough occurs in the kth labeling process.

Step 1 (Backtrack): Let $\epsilon = \{\epsilon_1, \epsilon_2\}$

$$\epsilon_1 = Min\{c(x,y) - f(x,y)$$

$$(x, y) \in A_k^+$$

$$\epsilon_2 = Min\{f(x, y)\}$$

$$(x, y) \in A_k^-$$

Step 2 (Flow Augmentation): Let $J_k = \Phi$

(a) Let f(x, y)=f(x, y)+ ϵ for (x, y) $\in A_k^+$

if f ' (x, y)=c(x, y), then setting $J_k = J_k \cup$(x,y)

(b) Let f' (y, x) =f(y, x) - ϵ for (y,x) $\in A_k^-$

if f'(y,x)=0, then setting $J_k = J_k \cup$ (y, x).

The adjusted flows are represented by f'(x, y). Stop

It will now be shown that with the use of either the predecessor – distance labels or the triple – labels, some of the labeled nodes can be retained for the next labeling process.

Lemma 1: after the k th breakthrough and flow adjustment let M = $\{m_1, m_2, \ldots, m_l\}$ be the set of labeled nodes such that d $(m_1) \leq$d (x_b) for i= 1,2,. . . .,l where d (x_b) = Min [d(x_j)]

$$(x_i,\ y_i) \in J_k$$

The predecessor and distance labels of the node $m_1 \in$ M remain valid and these nodes can be retained as labeled and unscanned for the next labeling process.

Proof: After the k th flow adjustment, all edges $(m_{j-1}, m_j) \in E_k$with

d $(m_j) \leq$d (x_b) have flows f(m_{j-1}, m_j) or f(m_j, m_{j-1})

such that f$(m_{j-1}, m_j) <$ c(m_{j-1}, m_j) if m_j is labeled m_{j-1}^+

or f$(m_{j-1}, m_j) > 0$ if m_j is labeled m_{j-1}^-

This is true since $(m_{j-1}, m_j) \notin J_k$. On the next application of the labeling process, node m_j can be labeled from node m_{j-1} again. Since s$\in$ M, all $m_i \in$M can be labeled in the same sequence again. The labels, therefore, remain valid, and the nodes $m_i \in$M can be retained in the labeled and unscanned state for the next labeling process.

Corollary to Lemma 1: After the kth breakthrough and flow adjustment, if P(t)=x_b, the labels and the status (scanned or unscanned) of each labeled node except nodes x_b and t can be retained for the next labeling process. Nodex_b is set to the labeled and unscanned state while t is set to the unlabeled state.

Lemma 2: After the kth breakthrough and flow adjustment, we need to discard only the triple –label of the nodes in the tree $T_k^* = (N_k^*, E_k^*)$ with root y_b, where y_b is such that

$$d(y_b) = Min[d(x_i)]$$

$$(x_i\ ,\ y_i) \in J_k$$

In addition, if L $(x_b) = y_b$ we have to set L(x_b) =B (y_b) , where node x_b is such that P$(y_b) = x_b$. Otherwise we have to set B(u)=B(y_b), where node u is such that B(u)= y_b.

Proof: The proof of the first part of lemma is similar to proof of lemma 1. The second part of the lemma then follows from the fact that the subtree with $y_b$as root has to be disconnected from the tree with s as the root.

Using the results of Lemmas 1 and 2, we propose the following improved labeling algorithms from solving the maximal flow problem.

Algorithm 2: Improved labeling algorithm for solving the maximal flow problem using the predecessor and distance functions.

Step 0: (i) At the start all nodes are in the unlabeled state

(ii) Node s receives labels (Φ, 0) (s is now labeled and unscanned).

Step 1 (labeling Process): Select any labeled unscanned node x with label ($z^{\pm}$, h). To each unlabeled node y such that f(x,y) < c(x,y), assign the label ($x^{\pm}$, h+1). To each unlabeled node y such that f(y,x)>0, assign the label (x^{-}, h+1) [y is now labeled and unscanned, x is labeled and scanned].Repeat until t is labeled [breakthrough] or until no more labels can be assigned and t is unlabeled. In the breakthrough case go to step 2, in the latter case, stop.

Step 2 (Flow Change):

(i) Apply Algorithm 1.

(ii) Let d(y_b)=Min [d(x_i)]. If P(t)= x_b, go to Step 2(iii); otherwise go to step 2(iv).

$$(x_i, y_i) \in J_k$$

(iii) Set $x_b$to the labeled and unscanned state. Erase the label on t. Go to step 1.

(iv) Set all labeled nodes y_j such that d(y_j) ≤ d(x_b) to the labeled and unscanned state and erase the labels on the other labeled nodes. Go to step 1.

Algorithm 3: Improved labeling algorithm for the maximal flow problem using the triple labels.

Step 0: (i) At the start all nodes are in the unlabeled state.

(ii) Node s receives the labels (Φ, 0) (s is now labeled and unscanned).

Step 1 (Labeling Process):

Select any labeled unscanned node x with label ($x^{\pm}$, B(x), L(x)). To each unlabeled node y such that f(x, y) < c(x, y), assign the label (x^{+}, B(y), L(y)) where B(y) = L(x) and L(y)=Φ Set L(x) =y. To each unlabeled node y such that f(x,y)>0, assign the label (B(y), L(y)) where B(y)=L(x) and L(y)=Φ [x is now labeled and unscanned, y is labeled and unscanned]. Repeat until t is labeled [breakthrough] or until no more labels can be assigned and t is unlabeled. In the breakthrough case, go to step 2; in the latter case, stop.

Step 2 (Flow Change):

(i) Apply algorithm 1.

(ii) Let d (y_b)=Min [d(y_i)].Let x = P (y_b). If L(x) = B (y_b)

$$(x_i, y_i) \in J_k$$

and go to step 2 (iv). Otherwise go to Step 2 (iii).

(iii) Let node u be such that B (u) = y_b. Let B(u) and go to Step 2 (iv).

(iv) Disconnect the subtree with y_b as root from the tree with s as root. Erase all labels on the subtree with y_b as root.

(v) Set all remaining labeled nodes to labeled and unscanned state; go to Step 1.

The predecessor- distance and triple-label algorithms use the same criterion as Ford-Fulkerson's procedure in the search for flow augmenting paths. Optimality of solution at termination and finiteness of the algorithms for integers capacity functions are not in question. The main computational advantage of the predecessor- distance and triple –label algorithms is that after each flow adjustment, the labels that remain valid after a flow adjustment, it requires the use of one more label than the predecessor-distance algorithm.

4. Application To Transportation And Assignment Problems

The transportation problem (Ford, 1962) can be stated as:

$$\min Z = \sum_{i \in U} \sum_{j \in V} c_{ij} x_{ij}$$

subject to $\sum_{j \in V} x_{ij} = a_i$ i ∈ U =1, 2, ……., m }, set of rows

$\sum_{i \in U} x_{ij} = b_j$ j ∈ V = {1, 2, ……., n }, set of columns

$x_{ij} \geq 0$, for all i∈ U and j∈V

with $\sum_{i \in U} a_i = \sum_{j \in V} b_j$

The network(N,A) representation of this problem is a bipartite graph with i∈ U , j ∈V forming the two sets of nodes and arcs (i, j) connecting each i ∈ U to each j∈V.

The dual to the transportation problem maybe stated as:

Max Z' = $\sum_{i\in U} a_i u_i + \sum_{j\in V} b_j v_j$

subject to $u_i + v_j \le c_{ij}$, for all $i \in$U,j $\in$ V

The primal- dual algorithm (Florian, M., 1990) solves a sequence of restricted primal problems. Each such problem is a maximum flow problem on a subset of the extended transportation network (N^*, A^*) which is formed from (N,A) by the addition of a source s^*, sink t^*, arcs (s^*, i) with c (s^*, i) =a_i and arcs(j, t^*) with c(j, t^*)= b_j. Thus it is clear that either Algorithm 2 or 3 can be used for solving the maximum flow problems. However, since the transportation network is a special type of multi-source, muti-sink network, and alternative algorithm is computationally interesting. This algorithm (predecessor- root algorithm) is especially suitable and computationally very efficient for the assignment problem.

Definition and Notation: Let x^*_{ij} represent a flow such that $0 \le a^*_i = a_i - \sum_{j\in V} x^*_{ij}$ and $0 \le b^*_j = b_j - \sum_{i\in U} x^*_{ij}$. A row i with $a^*_i > 0$ is defined as a source row i.

It is clear that a source node i, together with its descendant labeled nodes and the edges used to label them, forms a tree T^j_k =(N^j_k, E^j_k). The source rows are not connected to each other by labeling. Thus, the labeled nodes (rows and columns), together with the edges used to label them, form a forest F $_k$ = $\{T^1_k, T^2_k, ..., T^q_k\}$.

Lemma 3: if on the k^{th} application of the labeling process the flow augmenting path k is formed by edges from the treeT^j_k=(N^j_k, E^j_k)then after the flow adjustment is sufficient to discard the labels only on nodes $y_r \in N^j_k$. The predecessor and root labels on the other labeled nodes (i.e$y_r \in \bigcup_{i=1}^{q} N^i_k, i \ne$ j) remain valid, and the nodes can be retained as labeled and unscanned nodes for the next labeling process.

Proof: The flow in any edge $(y_{r-1}, y_r) \in \bigcup_{i=1}^{q} E^i_k (i \ne$j) is not change at the k^{th} flow adjustment. Therefore at the next labeling process all nodes $y_r \in \bigcup_{i=1}^{q} N^i_k (i \ne$j) can be labeled in the same sequence again; they can therefore be retained as labeled and unscanned nodes.

Lemma 4: Suppose that at the k th breakthrough a particular column g, say, with b^*_g >0 and F_k have been detected in the tree T^j_k=(N^j_k,E^j_k)with source row i as root. If Min [a^*_i, e(g)]> b^*_g, then the predecessor and root labels on nodes $y_r \in N^j_k$ remain valid for the next labeling process. Therefore, the labels and the status (scanned or unscanned) of all $y_r \in \bigcup_{j=1}^{q} N^j_k$ can be retained for the next labeling process.

Proof: All edges$(y_{r-1}, y_r) \in E^j_k$ are non- binding. Therefore, each $y_r \in N^j_k$ can be labeled in the same sequence again. Furthermore, from a labeled and scanned node, no further labeling is possible. Hence the labels and status of all $y_r \in \bigcup_{j=1}^{q} N^j_k$ can be retained for the next labeling process.

We propose the following predecessor-root labeling algorithm, based on the results of Lemmas 3 and 4.

Algorithm 4: Predecessor- root labeling algorithm for solving the transportation problem.

Step 0: Begin with any dual feasible solution (u*,v*). Let $x^*_{ij} = 0$, for all (i, j). At the start all i $\in$U and j $\in$ V are in the unlabeled state.

Step 1 (Labeling Process): Each unlabeled source row i with a^*_i >0 is labeled as (j, R(i)) where R(i) = i. Next select a labeled row, say row I, and scan it for all unlabeled columns such that $c_{ij} - u_i - v_j$ =0; label these columns as (i, R (j)) where R(j)= R(i). Repeat until all labeled rows have been scanned. The select labeled column, say column j, and scan it for all unlabeled rows i such that x_{ij} >0 ; label these rows as (j, R(i)) where R(i)=R(j). Repeat until all labeled columns have been scanned. Now revert to row scanning of new labeled rows and then the column and so on. If a column l with $b^*_l > 0$ is labeled (breakthrough) go to step 2. Otherwise continue until no more labels can be assigned (non-breakthrough) and go to step 3.

Step 2 (Flow Adjustment):

(i) Set s = R(l). Backtrack along the detected flow augmenting path F_k from s to l and determine e(l) = Min [x^*_{ij} (i, j)$\in A^*_k$] where A^+_k and A^-_k are as defined in section 3.

(ii) If Min[e(l), a^*_s]> b^*_l, set e(l) and go to step 2(iii).Otherwise set e(l)= Min [e(l), a^*_s] and go to Step 2 (iv)

(iii) Let $x^*_{ij} = x^*_{ij} + \varepsilon(l); for (i, j) \in A^+_k$

$$x^*_{ij} = x^*_{ij} - \varepsilon(l); for (i, j) \in A^-_k$$

$a^*_s = a^*_s - \varepsilon(l)$ and $b^*_l = b^*_l - \varepsilon$(l)

Retain all labels and go to step 1.

(iv)Adjust flows as in Step 2 (iii) above. $\sum_{i\in U}\sum_{j\in V} x^*_{ij} = \sum_{j\in V} b_j$, stop.

Otherwise erase the labels of all labeled rows i and labeled columns j such that R(i) =R(j)=s.Go to Step 1.

Step 3 (Dual variable change):

1. Let I and J be the index set of labeled rows and columns and

$$I^* = U - I, J^* = V - J,$$

Set $u_i = \begin{cases} u_i + \delta & i \in I \\ u_i & i \in I^* \end{cases}$

$$v_j = \begin{cases} v_j - \delta & j \in J \\ v_j & j \in J^* \end{cases}$$

Where $\delta = \min_{i,j} (c_{ij} - u_i - v_j)$

Go to Step 1 with the new dual feasible solution.

After each flow change, the labels discarded by the predecessor-root method include some labels that can be retained. If the triple – label algorithm discussed in the previous section is adopted for the primal-dual approach to the transportation problem, all valid labels after a flow change would be retained. However, the triple –label algorithm require the use of one more element in the label then the predecessor-root algorithm.

The predecessor – root algorithm specializes very well to accelerate the primal- dual method for solving the Assignment Problem (Ford, 1962).

For this problem since the a^*_i's and b^*_j's are equal to one, at each breakthrough, the net availability at the root of the treeT^j_k= (N^j_k,E^j_k) which contains F_k, becomes zero after flow adjustment. Therefore, all the labels on nodes $y_r \in N^j_k$ have to be discarded. In the next section we present some computational results using the predecessor- root algorithm to solve the assignment Problem.

5. Computational Results

In this section we discuss some computational experience with the application of the predecessor-root algorithm to the transportation and assignment problems and also the triple label algorithm applied to the transportation problem.

In table 1, the mean solution times as a function of problem size is presented for comparison, the rim and cost parameters were kept identical of those tested in (Singh, 2007). Our mean solution times indicate that the modified primal-dual method is computationally as attractive as the primal method. Approximate mean solution times with the 1971 code 100 ×100 and 150×150 transportation problems reported in (Singh, 2007) are 1.9 and 4.81 seconds respectively compared to our mean solution times of 6.56 and 14.55 seconds respectively. For 100×100 transportation problems the range of solution times with our code was4.3 seconds to 8.5 seconds. In table 1, we also analyzed the breakdown of mean solution times for the transportation problems. In spite of our modifications about 70% of the mean solution time was still taken up in the labeling process. This explains why our attempts in reducing the number of labels to be discarded after each flow change had been effective in reducing the mean solution time. Since such a large proportion of the mean solution time is spent in the labeling process, any significant reduction in scanning during the labeling process would reduce the mean solution time substantially. Lemma 4 was found to be very effective in the sense. For example, in the 100×100 transportation problem it was found that out of 250 average number of breakthroughs, about 100(or about 40%) of them did not require the discarding of any labels.

We tested the effect, on the mean solution time, of the variation in range of the parameters of the transportation problem. We found that mean solution time was relatively independent of the range of the rim parameters, but dependent on the range of the cost parameters. This can be explained by the fact that for problems with a large range in the cost parameter, an initial dual feasible solution would probable not have a large number of admissible cells. Also, at each dual variable change, only a few new cells would probably become admissible. Therefore, a large number of iterations is required before optimality is attained. This fact is brought out in table 2. This suggests that for transportation problems with a large range in cost parameter, the maximum dual change rule as suggested in (Ford, L.R., 1962) may be more efficient than Ford-Fulkerson's routine C.

Table 3(a) gives the mean solution times, using the predecessor- root algorithm, for solving the assignment problem. The range for was maintained as integers between 0 and 50 in the order to compare with (Florian, M., 1990). For 50×50 and 100 ×100 problems, our mean solution times are 0.47 and 1.74 seconds respectively compared to 0.51 and 2.0 seconds, respectively as respectively as reported in (Florian, M., 1990). For 100 × 100 assignment problems the range of solution times with our code was 1.43 to 2.0 seconds. The computational saving was especially significant for large problems. For

the 100×100 and 175×175 assignment problems [c_{ij} integers in the range 0 -99] our mean solution times are 2.01 and 5.85 seconds respectively as shown in Table 3(b).

Table 4 gives the effect, on the mean solution time, of the variation in range of the cost parameter for the assignment problem. Again we find that the number of nonbreakthrough and the mean solution times are dependent on the number of significant digits in the cost parameter. This again suggests that even for the assignment problem with a large number of significant digits in the cost parameter, the maximum dual change rule may be more efficient computationally then Ford-Fulkerson's routine C.

< Table 1 >

All solution times are exclusive of input/output and based on 10 randomly generated problems, each solved by using QSB_+ Computer Software. The a_i and b_j are integers drawn from a uniform distribution between 1 and 99 and the $c_{ij}$are integers drawn from a uniform distribution between 0 and 99.

< Table 2 >

Note: NBT =Nonbreakthroughs

DVC = Time for Dual Variable Change

MST =Mean Solution Time

All solution times are exclusive of input/output and based on 10 randomly generated problems, each solved by using QSB_+ Computer Software. The parameters are integers drawn from a distribution between the ranges specified.

< Table 3(a) >

All Solution times are exclusive of input/output and based on 10 randomly generated problems, each solved on by using QSB_+ Computer Software. The $c_{ij}$are integers drawn from a uniform distribution between 0 and 50.

< Table 3(b) >

All solution times are exclusive of input/output and based on 10 randomly generated problem, each solved by using QSB_+ Computer Software. The c_{ij}re integers drawn from a uniform distribution between 0 and 99.

< Table 4 >

Note: DVC = Dual Variable Change

All solution times are exclusive of input/output and based on 10 randomly generated problems each solved on by using QSB_+ Computer software. The c_{ij} are integers drawn from a uniform distribution between the ranges specified.

References

Barr, R., F. Glover, and D. Klingman. (1982). An improved Version of the Out-of Kilter Method and a Comparative Study of Computer Codes, Working paper Series 102, Center for Cybernetic Studies, The University of Texas, Austin, Texas, November.

Bennington, G.E. (1983). An Efficient Minimal Cost Flow algorithm. *Management Science*, No. 9, May 1983.

Florian, M. and M. Klein. (1990). An Experimental Evaluation of some Methods For Solving the assignment problem. *Canadian operational Research society*, No. 2.

Ford, L.R. and D.R. Flukerson. (1962). *Flows in Networks*. Princeton University Press, Princeton, New Jersey.

Garfinkel, R.S. (1981). An improved Algorithm for the Bottleneck assignment Problem. *Operation Research*, No. 6.

Glover, F., D. Karney and D. Klingman. (1980). Locating Stepping Stone paths in Distribution Problems via the Predecessor Index Method. *Transportation Science*, No. 4.

Glover, F., D. Karney and D. Klingman. (1982). The augmented Predecessor Index Method for locating Stepping Stone paths and Assigning Dual Prices in Distribution Problems. *Transportation Science*, No. 2.

Grafinkel, R.S. and M.R.Rao. (1981). The Bottleneck Transportation Problems. *Naval Research Logistics Quarterly*, No. 4.

Henier Muller-Merback. (1986). An Improved Starting Algorithm for the Ford-Fulkerson Approach to the Transportation Problem. *Management Science*, 13.

Johnoson, E.I. (1986). Networks and Basic Solutions. *Operation Research*.

Klien, M. (1987). A Primal Method for Minimal Cost Flows with Applications to the Assignment and Transportation Problems. *Management Science*.

Klingman, D., A. Napier and J. Stutz. (1983). A Program for Generating large Scale Capacitated assignment, Transporta-

tion and Minimum Cost Flow Network Problems, working Paper Series 109, center for Cybernetic Studies, the university of Texas, Austin, Texas, February 1983.

Singh, Vinai K. and Nimisha. (2007). An Algorithm for Labeling Network Flow Problems. *Indian Journal of Mathematics and Mathematical Sciences*, Vol. 3, No. 2, Dec. 2007.

Srinivasan, V. and G. l. Thompson. (1983). Benefit- Cost analysis of Coding Technique for the Primal Transportation algorithm. *Journal of the Association for Computing machinery*, No. 2.

Table 1. Breakthrough of Mean Solution Time for solving the Transportation Problem as a Function of Problem Size

Problem Size	Average Number of nonbreak-Throughs	Average Number of break-throughs	Time for initialization (Sec.)	Time for labeling (Sec.)	Time for flow augmentation (Sec)	Time for dual vaiable Change (Sec.)	Mean solution time (Sec.)
30× 30	19	79	0.02	0.50	0.08	0.10	0.70
50×50	15	125	0.13	1.23	0.25	0.37	1.98
80×80	10	182	0.33	2.99	0.49	0.58	4.39
100×100	10	250	0.52	4.52	0.74	0.78	6.56
130×130	8	331	0.86	8.49	1.13	1.00	11.48
150×150	6	376	1.13	10.93	1.25	1.24	14.55
175×175	7	452	1.53	17.33	2.04	1.55	22.45

Table 2. Effect on the Mean Solution Time, of the Variation in range of the Parameters of the Transportation Problem (Size 100 x100)

Range for c_{ij}	0 -9	0 – 99	0 -999	0 -9999
Range fora_i and b_j	Average DVC % MST Number MST (Sec.) Of NBT	Average DVC % MST Number MST (Sec.) Of NBT	Average DVC % MST Number MST (Sec.) Of NBT	Average DVC % MST Number MST (Sec.) Of NBT
1-9	0 1.0% 0.96	10 16% 4.99	62 44.5% 13.97	138 58.0% 28.16
1-99	1 1.5% 1.36	10 12% 6.56	69 35.5% 18.27	132 48.5% 28.51
1-999	1 1.6% 1.29	10 11.6% 6.85	68 35.0% 18.07	132 46.5% 29.87

Table 3(a). Breakdown of Mean Solution Time for Solving The Assignment Problem as a Function of Problem Size

Average problem size	Time for number of nonbreak-throughs	Initialization (Sec.)	Time for labeling (Sec.)	Time for flow augmentation (Sec.)	Mean dual variable Change (Sec.)	Solution time (Sec.)
30×30	5	0.02	0.10	0.08	0.04	0.24
50×50	4	0.05	0.16	0.18	0.08	0.47
80×80	3	0.10	0.68	0.20	0.15	1.13
100×100	3	0.16	1.08	0.26	0.24	1.74

Table 3 (b). Breakthrough of Mean Solution Time for Solving the assignment Problem as a Function of Problem Size

Average Problem Size	Time for number of nonbreakthroughs	Initialization (Sec.)	Time for labeling (Sec.)	Time for flow augmentation (Sec.)	Mean dual variable Change (Sec.)	Solution time (Sec.)
100×100	5	0.17	1.20	0.20	0.44	2.01
130×130	4	0.29	2.15	0.25	0.61	3.30
150×150	3	0.36	2.74	0.35	0.64	4.09
175x175	3	0.49	3.83	0.47	0.79	5.58

Table 4. Effect on Mean Solution Time of the Variation in Range of the Cost Parameters of the Assignment Problem (Size 100x100)

Range for c_{ij}	Average number of nonbreakthrough	Time for DVC Mean Solution Time	Mean Solution Time (Sec.)
0- 9	0	1.0%	0.69
0-99	5	21.9%	2.01
0-999	40	57.0%	7.13
0-9999	86	68.5%	13.34