Dhaka Univ. J. Sci. 61(1): 19-26, 2013 (January)

Application of Bounded Variable Simplex Algorithm in Solving Maximal Flow Model

Sohana Jahan, Marzia Yesmin and Fatima Tuj Jahra

Department of Mathematics ,University of Dhaka,Dhaka-1000,Bangladesh

Email:jahansohana@univdhaka.edu

Received on 20.07.2011. Accepted for published on 10. 7.2012.

Abstract

In this paper, A comparative study is made on the determination of maximum flow in a network by Maximal Flow Algorithm and Bounded Variable Simplex Method. We have introduced a technique to find the route of a flow in any iteration of the bounded variable simplex algorithm.

Keywords: Maximal Flow Model, Bounded Variable simplex method.

I .Introduction

Networking is one of the most important branch of Operation Research. Most of the real world problem can be formulated as network model. There are various type network models such as Minimal spanning tree, Shortest route algorithm , Maximal flow algorithm, Minimum cost capacitated network algorithm. Different type of network problem can be solved by different algorithms. In this paper we have worked on Maximal flow problem. We have formulated the Maximal flow problem as a linear programming problem and solved it using Bounded variable simplex algorithm which is very easy to solve.

In a Maximal Flow Problem, we wish to send as much material as possible from a specified node *s* in a network, called the *source*, to another specified node *t*, called the *sink*.

The maximum flow problem was first formulated in 1954 by T. E. Harris as a simplified model of Soviet railway traffic flow. In 1955, Lester R. Ford and Delbert R. Fulkerson [1],[2] created the first known algorithm, the Ford–Fulkerson algorithm. Over the years, various improved solutions to the maximum flow problem [7],[8] were discovered.

An example of a flow network is given by figure 1. The source node is denoted by 1 and the sink node by 6. Nodes 2, 3, 4,5 are the intermediate nodes. There are nine arcs connecting the various nodes, denoted by (1,2), (1,3), (1,4),(2,3), (2,6), (3,5), (3,6), (4,5), (5,6).

The intermediate nodes must satisfy the strict conservation requirement; that is ,the net flow into these nodes must be zero. However, the source may have a net outflow and the sink a net inflow. As a consequence of the conservation at all intermediate nodes, the outflow f on the source will equal to the inflow of the sink .A set of arc flows satisfying these conditions is said to be a flow in the network of value f. The maximal flow problem is that of determining the maximal flow that can be established in such a network.

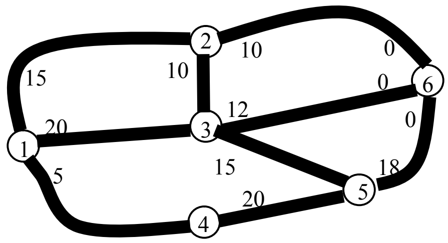

Fig. 1

II. Maximal Flow Algorithm

The Maximum Flow Algorithm [9] is one of the most important methods to obtain maximum flow in a network. The algorithm [4],[6] is based on finding breakthrough paths with net positive flow between the source and sink nodes. Each paths commits part or all the capacities of its arcs to the total flow in the network.

Consider arc (i, j) with (initial) capacities $(\overline{C}_{ij}, \overline{C}_{ji})$. As portion of this capacities are committed to the flow in the arc, the residuals (or the remaining capacities) of the arc are updated. The network with the updated residuals is referred to as the residue network. We use the notation (c_{ij}, c_{ji}) to represent these residuals.

For a node j receives flow from node i, we define a label [a . i], where a is the flow from node i to node j.

The steps of the algorithm are summarized as follows.

Step1. For all arcs (i, j) , set the residual capacity equal to the initial capacity – that $(c_{ij}, c_{ji}) = (\overline{C}_{ij}, \overline{C}_{ji})$. Let $a_1 = \infty$ and label source node 1 with $[\infty, -]$ and go to step 2.

Step2. Determine S_i as the set of unlabeled nodes j that can be reached directly from node i by arcs with positive residuals (that is, $c_a > 0$ for all $j \in S_i$). If $S_i \neq \emptyset$ then go to step 3. Otherwise, go to step 4.

Step 3. Determine $k \in S_i$, such that $c_{ik} = \max_{j \in S} \{c_{ij}\}$.

Set $a_k = c_{ik}$ and label node k with $[a_k$, i]. If k = n, the sink node has been labeled, and a breakthrough path is found, go to step 5. Otherwise, set i = k and go to step 2.

Step 4. (Backtracking) If i = l, no further breakthrough are possible, go to step 6. Otherwise, let r be the node that has been labeled immediately before the current node i and remove i from the set of nodes that are adjacent to r. Set i = r and go to step 2.

Step 5. (Determination of Residue Network):

Let N_p = (l, k_1, k_2,…..n) define the nodes of the p^{th} breakthrough path from source node l to sink node n. The maximum flow along the path is computed as, f_p = min $\{a_1, a_{k1}, a_{k2} ..., a_n\}$.The residual capacity of each arc along the breakthrough path is decreased by f_p in the direction of the flow and increased by f_p in the reverse direction - that is, for nodes i and j on the path, the residual flow is changed from the current (c_{ij}, c_{ji}) to

(a) $(c_{ij} - f_p, c_{ij} + f_p)$ if the flow is from i to j

(b) $(c_{ij} + f_p, c_{ij} - f_p)$ if the flow is from j to i

(c) Reinstate any nodes that were removed in

step 4. Set i = 1 , and return to step 2 to attempt a new breakthrough path.

Step 6. (Solution)

(a) Given that m breakthrough paths have been determined, the maximal flow in the network is $F = f_1 + f_2 + \ldots + f_m$

(b) Given that the initial and final residuals of arc (i, j) are given by $(\bar{C}_{ij}, \bar{C}_{ji})$ and (c_{ij}, c_{ji}), respectively the optimal flow in arc (i, j) is computed as follows:Let (α , β) = $(\bar{C}_{ij} - c_{ij}, \bar{C}_{ji} - c_{ji})$. If α > 0, the optimal flow from i to j is α. Otherwise, if β > 0, the optimal flow from j to i is β. (It is impossible to have both α and β positive).

Example: Consider the network in Figure 1. The maximal flow problem is to determine the maximum flow from the source node 1 to the sink node 6.

The maximum flow in the network can be determined by Maximal Flow Algorithm as follows:

Iteration 1:

Step1: Set $a_1 = \infty$ and we label node 1 with $[\infty, -]$.

Set $i = 1$

Step 2: $[S_1 = \{2,3,4\} \neq \text{Ø}]$.

Step 3: $k = 3$ because $c_{13} = \max\{c_{12}, c_{13}, c_{14}\}$

= max{15,20,5}=20. Set $a_3 = c_{13} = 20$, and label node 3 with [20,1],Set i=3,and repeat step 2.

Step 2: $S_3 = \{5,6\} \neq \text{Ø}$.

Step 3: k=6 because $c_{36} = 12$. Set $a_6 = c_{36} = 12$

and label node 6 with [12,3], breakthrough is achieved and go to step 5.

Step 5: Breakthrough path is determined from the labels starting at node 6 and ending at node 1 – that is, (6) → [12,3] → (3) → [20,1] → (1). Thus, N_1 = {1,3,6} and $f_1 = \min\{a_1, a_3, a_6\}$ = { ∞, 20,12 } = 12. The residual capacities along path N_1 are

$(c_{13}, c_{31}) = (20 - 12, 0 + 12) = (8,12)$

$(c_{36}, c_{63}) = (12 - 12, 0 + 12) = (0,12)$

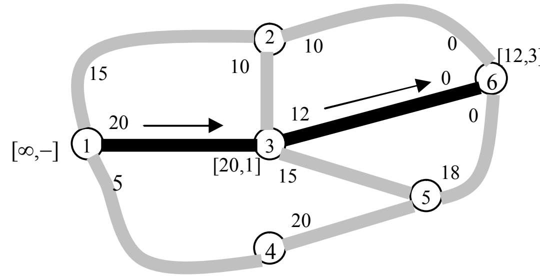

Fig: 1.1

Iteration 2:

Step1: Set $a_1 = \infty$ and we label node 1 with $[\infty, -]$. Set $i = 1$.

Step 2: $S_1 = \{2,3,4\} \neq \Phi$.

Step 3: $k = 2$ because $c_{12} = \max\{c_{12}, c_{13}, c_{14}\}$=

max{15,8,5}=15. Set $a_2 = c_{12} = 15$, and label node 2 with [15,1],Set i=2,and repeat step 2.

Step 2: $S_2 = \{3,6\} \neq \text{Ø}$.

Step 3: k=6 because $c_{26} = 10$. Set $a_6 = c_{26} = 10$

and label node 6 with [10,2], breakthrough is achieved and go to step 5.

Step 5: Breakthrough path is determined from the labels starting at node 6 and ending at node 1 – that is, (6) → [10,2] → (2) → [15,1] → (1). Thus, N_1 = {1,2,6} and $f_2 = \min\{a_1, a_2, a_6\}$ = {∞, 15,10 } = 10. The residual capacities along path N_2 are

$(c_{12}, c_{21}) = (15 - 10, 0 + 10) = (5,10)$

$(c_{26}, c_{62}) = (10 - 10, 0 + 10) = (0,10)$

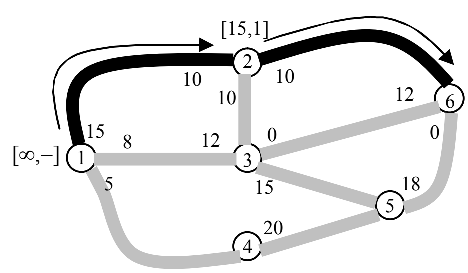

Fig. 1.2

Iteration 3:

Step 1: Let $a_1 = \infty$, and we label node 1 with [∞ , −]. Set i = 1.

Step 2: S_1={2,3,4}≠Ø.

Step 3: k = 3 and a_3 = c_{13} = max {5,8,5} = 8.

Label node 3 with [8,1],Set i=3,and repeat step 2.

Step 2: S_3 = {2,5}≠Ø.[node 1 is already labeled hence it cannot be included in S_3].

Step 3: k = 5 because a_5 = c_{35} =15. Label node 5 with [15,3]. Set i = 5, and repeat step 2.

Step 2: S_5 = {6}≠Ø.

Step 3: k = 6 because a_6 = c_{56} = 18. Label node 6 with [18,6]. Breakthroughs is achieved, go to step 5.

Step 5: N_3 ={1,3,5,6} and f_3 = min{∞ , 8,15,18} = 8.

The residual along the path of N_3 are

(c_{13},c_{31}) = (8 − 8, 12 + 8) = (0, 20); (c_{35},c_{53}) = (15−8, 0 +8) = (7, 8); (c_{56},c_{65}) = (18 − 8, 0 + 8) = (10, 8)

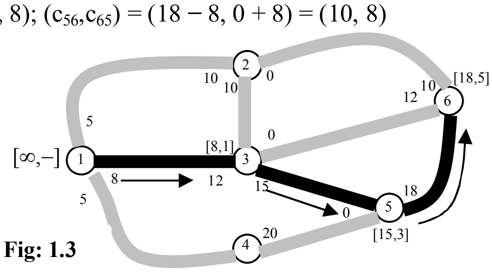

Fig: 1.3

Iteration 4:

Step 1: Let $a_1 = \infty$, and we label node 1 with [∞ , −]. Set i = 1.

Step 2: S_1={2,4}≠Ø.

Step 3: k = 2 and a_2 = c_{12} = max {5,5} = 5. Set i = 2 and label node 2 with [5,1] and repeat step 2.

Step 2: S_2 = {3}≠Ø.[node 1 is already labeled hence it cannot be included in S_2].

Step 3: k = 3 because a_3 = c_{23} =10. Label node 3 with [10,2]. Set i = 3, and repeat step 2.

Step 2: S_3 = {5}≠Ø.

Step 3: k = 5 because a_5 = c_{35} = 7. Label node 5 with [7,3]. Set i=5 and go to step 2.

Step 2: S_5 ={6}≠Ø.

Step 3: k =6 because a_6 = c_{56} =10. Label node 6 with [10,5]. Breakthrough is achieved and go to step 5.

Step 5: N_3 ={1,2,3,5,6} and f_4 = min{∞ , 5,10,7,10} = 5.

The residual along the Path of N_4 are

(c_{12},c_{21}) = (5 − 5, 10 + 5) = (0, 15)

(c_{23},c_{32}) = (10 − 5, 20 + 5) = (5, 25); (c_{35},c_{53}) = (7−5, 8+5) = (2, 13); (c_{56},c_{65}) =(10 −5, 8+5)=(5, 13)

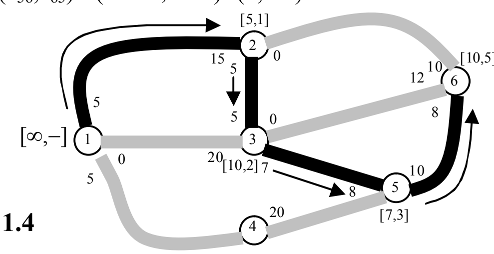

Fig: 1.4

Iteration 5:

Step 1: Let a $a_1 = \infty$, and we label node 1 with [∞ , −]. Set i = 1

Step 2: S_1={4}≠Ø.

Step 3: k = 4and a_4= c_{14} =5. Set i = 4 and label node4 with [5,1] and repeat step 2.

Step 2: S_4 = {5}≠Ø

Step 3: k = 5 because a_5 = c_{45} =15. Label node 5 with [15,4]. Set i = 5, and repeat step 2.

Step 2: S_5 = {6}≠Ø.

Step 3: k = 6 because a_6 = c_{56} = 5. Label node 6 with [5,5]. Breakthrough is achieved, and go to step 5.

Step 5: N_3 ={1,4,5,6} and f_5 = min{∞ , 5,15,5} = 5. The residual along the Path of N_5 are (c_{14},c_{41}) =(5−5, 0+5)= (0, 5); (c_{45},c_{54}) =(15− 5, 0 +5)=(10,5);
(c_{56},c_{65}) = (5 −5, 13 + 5) = (0, 18)

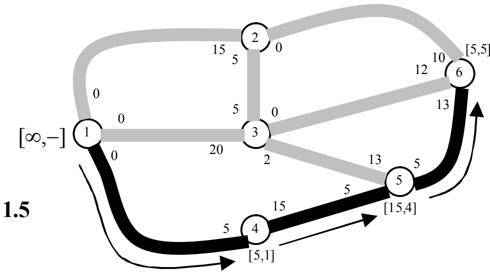

Fig: 1.5

Iteration 6: All the arcs out of node 1 have zero residuals. Hence no further breakthrough is possible. We turn to step 6 to determine the solution.

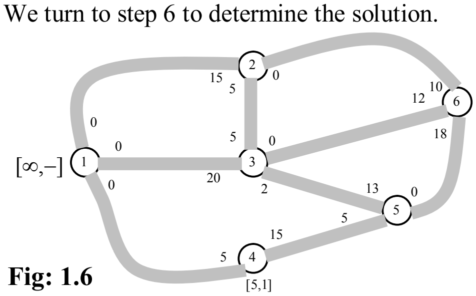

Fig: 1.6

Step 6: Maximal flow in the network is F = f_1 + f_2 + ……+f_5= 12+10+8+5+5= 40 units. The flow in the different arc is computed by subtracting the last residuals ($c_{ij} - c_{ji}$)in iteration 6 from the initial capacities ($\overline{C}_{ij}$, $\overline{C}_{ji}$). as the following table shows.

Arc	$(\bar{C}_{ij}, \bar{C}_{ji})$ - $(c_{ij} - c_{ji})$	Flow Amount	Direction
(1,2)	(15,0) – (0,15) =(15,-15)	15	1→2
(1,3)	(20,0) – (0,20) = (20,-20)	20	1→3
(1,4)	(5,0) – (0,5) = (5,-5)	5	1→4
(2,3)	(10,0) – (5,5) = (5,- 5)	5	2→3
(2,6)	(10,0) – (0,10) = (10,-10)	10	2→6
(3,6)	(12,0) – (0,12) = (12,-12)	12	3→6
(3,5)	(15,0) –(2,13) = (13,-13)	13	3→5
(4,5)	(20,0) – (15,5) = (5,-5)	5	4→5
(5,6)	(18,0)-(0,18)=(18,-18)	18	5→6

III. Bounded – Variable Simplex Algorithm:

In LP models, variables may have explicit positive upper and lower bounds. For example, in production facilities, lower and upper bound can represent the minimum maximum demands for certain products. Bounded variable also arise prominently in the course of solving integer programming problems by the branch and bound algorithm.

The bounded algorithm is efficient computationally because it accounts for the bounds implicitly. We consider the lower bounds first because it is simpler. Given $X \geq L$, we can use the substitution

$X = L + X'$, $X' \geq 0$.

Throughout and solve the problem in terms of X' (whose lower bound now equals zero). The original X is determined by back substitution, which is legitimate because it guarantees that $X = L + X'$ will remain nonnegative for all $X' \geq 0$.

Next, consider the upper bounding constraints, $X \leq U$. The idea of direct substitution (i.e, $X = U - X''$, $X'' \geq 0$) is not correct because back substitution, $X= U - X'$, does not ensure that X will remain nonnegative. A different procedure is thus needed.

Define the upper bounded LP model as

Maximize $z= \{CX \mid (A,I)X = b ,0 \leq X \leq U\}$

The bounded algorithm uses only the constraints $(A,I)X = b$, $X \geq 0$, while accounting for $X \leq U$ implicitly by modifying the simplex feasibility condition.

Let $X_B = B^{-1}b$ be a current basic feasible solution of $(A,I)X = b$, $X \geq 0$ and suppose that, according to the optimality condition, P_j is the entering vector. Then given that all the nonbasic variables are zero, the constraints equation of the ith basic variable can be written as $(X_B)_i = (B^{-1}b)_i – (B^{-1}P_j)_i x_j$.

When the entering variable x_j increases above zero level, $(X_B)_i$ will increase or decrease depending on whether $(B^{-1}P_j)_i$ is negative or positive, respectively. Thus in determining the value of the entering variable x_j, three conditions must be satisfied:

1. The basic variable $(X_B)_i$ remains nonnegative – that is $(X_B)_i \geq 0$.

2. The basic variable $(X_B)_i$ does not exceed its upper bound - that is $(X_B)_i \leq (U_B)_i$, where U_B comprises the ordered elements of U corresponding to X_B .

3. The entering variable x_j cannot assume a value larger than its upper bound – that is $x_j \leq u_j$, where u_j is the jth element of U .The first condition $(X_B)_i \geq 0$ requires that

$(B^{-1}b)_i – (B^{-1}P_j)_i x_j \geq 0$. It is satisfied if $x_j \leq E\,\theta_1 = \min\{(B^{-1}b)_i / (B^{-1}P_j)_i \mid (B^{-1}P_j)_i > 0\}$ .This condition is the same as the feasibility condition of the regular singular method. Next, the condition $(X_B)_i \leq (U_B)_i$ specifies that $(B^{-1}b)_i – (B^{-1}P_j)_i x_j \leq (U_B)_i$. It is satisfied if

$x_j \leq \theta_2 = \min\{[(B^{-1}b)_i - (U_B)_i] / (B^{-1}P_j)_i \mid (B^{-1}P_j)_i < 0\}$.

Combining the three restrictions, x_j enters the solution at the level that satisfies all three conditions – that is,

$x_j = \min \{\theta_1, \theta_2, u_j \}$. The change of basis for the next iteration depends on whether x_j enters the solution at level θ_1, θ_2 or u_j. Assuming that $(X_B)_r$ is the leaving variable, then we have the following rules :

1. $x_j = \theta_1$: $(X_B)_r$ leaves the basic solution (becomes non basic) at level zero. The new iteration is generated using the normal simplex method with x_j and $(X_B)_r$ as the entering and the leaving variables, respectively.

2. $x_j = \theta_2$: $(X_B)_r$ becomes nonbasic at its upper bound. The new iteration is generated in the case of $x_j = \theta_1$, with one modification that accounts for the fact that $(X_B)_r$ will be nonbasic at upper bound. Because the values of θ_1, θ_2 require all nonbasic variables to be at zero level. Convert the new nonbasic $(X_B)_r$ at upper bound to a nonbasic variable at zero level. This is achieved by using the substitution $(X_B)_r = (U_B)_r - (X'_B)_r$, where $(X'_B)_r \geq 0$. It is immaterial whether the substitution is made before or after the new basis is computed.

3. $x_j= u_j$: The basic vector X_B remains unchanged because $x_j = u_j$ stops short of forcing any of the current basic variables to reach its lower (=0) or upper bound. This means that x_j will remain non basic but at upper bound. Following the argument just presented, the new iteration is generated by using the substitution $x_j= u_j - x'_j$.

A tie among θ_1, θ_2 and u_j may be broken arbitrarily. However, it is preferable, where possible, to implement the rule for $x_j = u_j$ because it entails less computation.

The substitution $x_j = u_j - x'_j$ will change the original c_j, P_j and b to $c'_j = - c_j$, $P'_j = P_j$ and b to $b' = b – u_jP_j$. This means that if the revised simplex method is used, all the computations (e.g., B^{-1},X_B and $z_j – c_j$), should be based on the updated values of C, A and b at each iteration.

IV. LP Formulation of Maximal Flow Model:

If v denotes the amount of material sent from nodes s to node t and x_{ij} denotes the flow from node i to node j over arc $i-j$ the formulation is:

Maximize v,
subject to:

$$\sum_j x_{ij} - \sum_k x_{ki} = \begin{cases} v & \text{if } i = s \text{ (source)}, \\ -v & \text{if } i = t \text{ (sink)}, \\ 0 & \text{otherwise}, \end{cases}$$

$$0 \le x_{ij} \le u_{ij} \qquad (i = 1, 2, \cdots, n;\ j = 1, 2, \cdots, n).$$

As usual, the summations are taken only over the arcs in the network. Also, the upper bound u_{ij} for the flow on arc $i-j$ is taken to be $+\infty$ if arc $i-j$ has unlimited capacity. The interpretation is that v units are supplied at s and consumed at t. Let us introduce a fictitious arc $t-s$ with unlimited capacity; that is, $u_{ts} = +\infty$. Now x_{ts} represents the variable v, since x_{ts} simply returns the v units of flow from node t back to node s, and no formal external supply of material occurs. With the introduction of the arc $t-s$, the problem assumes the following special form of the general network problem:

Maximize x_{ts},
subject to:

$$\sum_j x_{ij} - \sum_k x_{ki} = 0 \qquad (i = 1, 2, \cdots, n).$$

Thus if x_{ij} as the amount of flow in arc (i,j) with capacity C_{ij}. The objective is to determine x_{ij} for all i and j that will maximize the flow between start nodes s and terminate node t subjective to flow restriction (input flow = outflow flow) at all but nodes s and t.

Now we are going to obtain the maximum flow in the network given in Figure 1 using Bounded Variable Simplex method.

The following table summarizes the associated LP with two different , but equivalent, objective functions depending on whether maximize the output from start node 1 (= z_1) or the input to terminal node 6 (= z_2).

	x_{12}	x_{13}	x_{14}	x_{23}	x_{26}	x_{35}	x_{36}	x_{45}	x_{56}	
Max z_1 =	1	1	1							
Max z_2=					1		1		1	
Node 2	1			-1	-1					= 0
Node 3		1		1		-1	-1			= 0
Node 4			1					-1		= 0
Node 5						1		1	-1	= 0
Capacity	15	20	5	10	10	15	12	20	18	

Writing this as an Linear Programming Problem (LPP)

Maximize $z = x_{12}+x_{13}+x_{14}$

s/t $x_{12}-x_{23} - x_{26}= 0$

$x_{13}+x_{23}-x_{35}-x_{36} = 0$

$x_{14}-x_{45} = 0$

$x_{35} +x_{45}-x_{56} = 0$

$0 \le x_{12} \le 15$, $0 \le x_{13} \le 20$, $0 \le x_{14} \le 5$, $0 \le x_{23} \le 10$, $0 \le x_{26} \le 10$, $0 \le x_{35} \le 15$, $0 \le x_{36} \le 12$, $0 \le x_{45} \le 20$, $0 \le x_{56} \le 18$

If we try to solve this LPP by simplex method, we can write the bounded variables as constraints by inserting slack variables. Then we obtain a large set of constraints.

Using Bounded Variable Algorithm this problem can be solved easily. The initial table is:

Table 0

c_j		1	1	1	0	0	0	0	0	0	
Basic		x_{12}	x_{13}	x_{14}	x_{23}	x_{26}	x_{35}	x_{36}	x_{45}	x_{56}	Solution
1	x_{12}	1	0	0	-1	-1	0	0	0	0	0
1	x_{13}	0	1	0	1	0	-1	-1	0	0	0
1	x_{14}	0	0	1	0	0	0	0	-1	0	0
0	x_{56}	0	0	0	0	0	-1	0	-1	1	0
Z		0	0	0	0	-1	-1	-1	-1	0	0

Iteration 1:

We have $B= B^{-1}=I$ and $X_B =(x_{12}, x_{13}, x_{14} ,x_{56})^T =$

$B^{-1}b=(0,0,0,0)^T$. Here x_{45} is the entering variable, we get

$B^{-1}P_{45} =(0,0,-1, -1)^T$ which yields $\theta_1 = \infty$;

$\theta_2 = \min \{- , - , (0 – 5)/(-1) ,(0- 18)/(-1) , - ,0 \}= 5$, corresponding to x_{14}. Next, given the upper bound on the entering variable $x_{45} \le 20$, it follows that

Table 1.1

c_j		1	1	-1	0	0	0	0	0	0	
Basic		x_{12}	x_{13}	x'_{14}	x_{23}	x_{26}	x_{35}	x_{36}	x_{45}	x_{56}	Solution
1	x_{12}	1	0	0	-1	-1	0	0	0	0	0
1	x_{13}	0	1	0	1	0	-1	-1	0	0	0
-1	x'_{14}	0	0	-1	0	0	0	0	-1	0	-5
0	x_{56}	0	0	0	0	0	-1	0	-1	1	0

$x_{45} = \min \{ \infty ,5,20\} = 5 (=\theta_2)$

Because x_{14} becomes nonbasic at its upper bound ,we apply the substitution $x_{14}=5 - x'_{14}$ to obtain

Next, the entering variable x_{45} becomes basic and the leaving variable x'_{14} becomes nonbasic at zero level, which yields:

Table 1.2

	c_j	1	1	-1	0	0	0	0	0	0	
Basic		x_{12}	x_{13}	x'_{14}	x_{23}	x_{26}	x_{35}	x_{36}	x_{45}	x_{56}	Solution
1	x_{12}	1	0	0	-1	-1	0	0	0	0	0
1	x_{13}	0	1	0	1	0	-1	-1	0	0	0
0	x_{45}	0	0	1	0	0	0	0	1	0	5
0	x_{56}	0	0	1	0	0	-1	0	0	1	5
Z		0	0	1	0	-1↑	-1	-1	0	0	0

The flow in this iteration is the minimum of the values of the variable that have changed their values in this iteration. For, x_{14} =5, x_{45} = 5, x_{56} = 5, min{5,5,5} = 5. So flow f_1 = 5 in the route 1 →4→5→6.

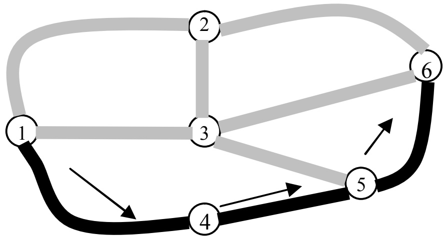

Fig. 2.1

Iteration 2:

Here x_{26} is the entering variable, we get $B^{-1}P_{26}$ =(-1,0, 0,0)^T which yields $\theta_1 = \infty$, θ_2 = min {(0-15)/(-1), -, - , - }= 15 , corresponding to x_{12}. Next, given the upper bound on the entering variable $x_{26} \leq 10$, it follows that x_{26} = min { ∞ ,15, 10} =10 = u_{26}.

Because x_{26} enters at its upper bound, X_B remains unchanged and x_{26} becomes nonbasic at its upper bound. We use the substitution $x_{26} = 10 - x'_{26}$ to obtain Table 2.

Table 2

	c_j	1	1	-1	0	0	0	0	0	0	
Basic		x_{12}	x_{13}	x'_{14}	x_{23}	x'_{26}	x_{35}	x_{36}	x_{45}	x_{56}	Solution
1	x_{12}	1	0	0	-1	1	0	0	0	0	10
1	x_{13}	0	1	0	1	0	-1	-1	0	0	0
0	x_{45}	0	0	1	0	0	0	0	1	0	5
0	x_{56}	0	0	1	0	0	-1	0	0	1	5
Z		0	0	1	0	1	-1↑	-1	0	0	0

x_{12} =10, x_{45} = 5, x_{56} = 5, x_{26} =10.From the last table 1.2 of the iteration 1and from the table 2 we see that the values of x_{45} and x_{56} remained unchanged. the value of x_{12} and x_{26} changed. Thus min{10,10} = 10. So flow f_2 =10 in the rout 1→2→ 6

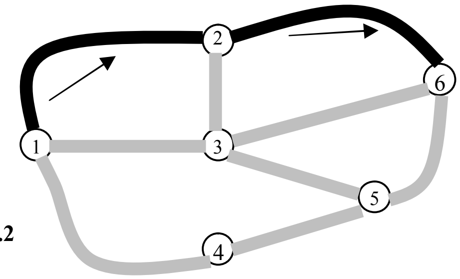

Fig: 2.2

Iteration 3:

Here x_{35} is the entering variable, we get $B^{-1}P_{35}$ =(0, -1,0 ,-1)^T which yields $\theta_1 = \infty$; θ_2 = min {- ,(0 - 20)/(-1) , - ,(5 – 18)/(-1) }= 13 , corresponding to x_{56}. Next, given the upper bound on the entering variable $x_{35} \leq 15$, it follows that x_{35} = min { ∞ ,13,15} =13 (=θ_2)

Because x_{56} becomes nonbasic at its upper bound, we apply the substitution $x_{56}=18 - x'_{56}$ to obtain the following table:

Table 3.1

	c_j	1	1	-1	0	0	0	0	0	0	
Basic		x_{12}	x_{13}	x'_{14}	x_{23}	x'_{26}	x_{35}	x_{36}	x_{45}	x'_{56}	Solution
1	x_{12}	1	0	0	-1	1	0	0	0	0	10
1	x_{13}	0	1	0	1	0	-1	-1	0	0	0
0	x_{45}	0	0	1	0	0	0	0	1	0	5
0	x'_{56}	0	0	1	0	0	-1	0	0	-1	-13

Next, the entering variable x_{35} becomes basic and the leaving variable x'_{56} becomes nonbasic at zero level, which yields:

Table 3.2

	c_j	1	1	-1	0	0	0	0	0	0	
Basic		x_{12}	x_{13}	x'_{14}	x_{23}	x'_{26}	x_{35}	x_{36}	x_{45}	x'_{56}	Solution
1	x_{12}	1	0	0	-1	1	0	0	0	0	10
1	x_{13}	0	1	-1	1	0	0	-1	0	1	13
0	x_{45}	0	0	1	0	0	0	0	1	0	5
0	x_{35}	0	0	-1	0	0	1	0	0	1	13
Z		0	0	0	0	1	0	-1↑	0	1	0

x_{13} =13, x_{35} =13, x_{56} = 18, min{13,13,18} = 13. So flow f_3 = 13 in the route 1→3→5→6.

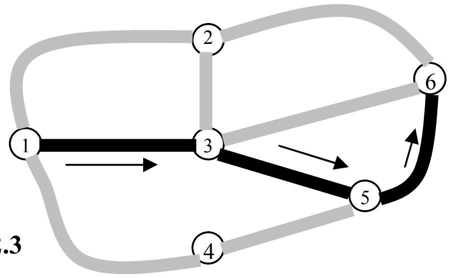

Fig: 2.3

Iteration 4:

Here x_{36} is the entering variable, we get $B^{-1}P_{36}=(0, -1,0 ,0)^T$ which yields

$\theta_1 = \infty$, θ_2 = min {- ,(13 - 20)/(-1) , - ,0 }= 7 ,

corresponding to x_{13}

Next, given the upper bound on the entering variable $x_{36} \le 12$, it follows that

x_{36} = min { ∞ ,7,12} =7 (= θ_2)

Because x_{13} becomes nonbasic at its upper bound, we apply the substitution x_{13}=10 - x'_{13} and obtain Table 4.1.

Table 4.1

Basic	c_j	1	-1	-1	0	0	0	0	0	0	
		x_{12}	x'_{13}	x'_{14}	x_{23}	x'_{26}	x_{35}	x_{36}	x_{45}	x'_{56}	Solution
1	x_{12}	1	0	0	-1	1	0	0	0	0	10
-1	x'_{13}	0	-1	-1	1	0	0	-1	0	1	-7
0	x_{45}	0	0	1	0	0	0	0	1	0	5
0	x_{35}	0	0	-1	0	0	1	0	0	1	13

Next, the entering variable x_{36} becomes basic and the leaving variable x'_{13} becomes nonbasic at zero level, which yields:

Table 4.2

Basic	c_j	1	-1	-1	0	0	0	0	0	0	
		x_{12}	x'_{13}	x'_{14}	x_{23}	x'_{26}	x_{35}	x_{36}	x_{45}	x'_{56}	Solution
1	x_{12}	1	0	0	-1	1	0	0	0	0	10
0	x_{36}	0	1	1	-1	0	0	1	0	-1	7
0	x_{45}	0	0	1	0	0	0	0	1	0	5
0	x_{35}	0	0	-1	0	0	1	0	0	1	13
Z		0	1	1	-1↑	1	0	0	0	0	0

x_{13} =20, x_{36} =7, min{20,7} = 7 So flow f_4 = 7 in the route 1→3→6

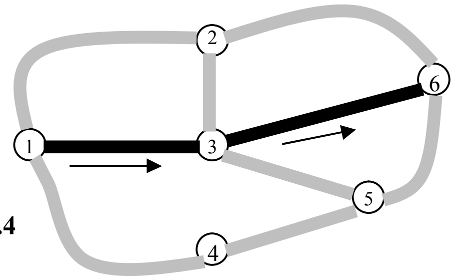

Fig: 2.4

Iteration 5:

Here x_{23} is the entering variable, we get

Table 5.1

Basic	c_j	1	-1	-1	0	0	0	0	0	0	
		x_{12}	x'_{13}	x'_{14}	x_{23}	x'_{26}	x_{35}	x'_{36}	x_{45}	x'_{56}	Solution
1	x_{12}	1	0	0	-1	1	0	0	0	0	10
0	x'_{36}	0	1	1	-1	0	0	-1	0	-1	-5
0	x_{45}	0	0	1	0	0	0	0	1	0	5
0	x_{35}	0	0	-1	0	0	1	0	0	1	13

$B^{-1}P_{23}=(-1, -1,0 ,0)^T$ which yields $\theta_1 = \infty ; \theta_2$ = min {(10 – 20)/(-1) ,(7 - 12)/(-1) ,- ,- }= 5 , corresponding to x_{36}.

Next, given the upper bound on the entering variable $x_{23} \le 10$, it follows that

x_{23} = min { ∞ ,5,10} = 5 (= θ_2)

Because x_{36} becomes nonbasic at its upper bound, we apply the substitution x_{36}=15 - x'_{36} to obtain Table 5.1.

Next, the entering variable x_{36} becomes basic and the leaving variable x'_{13} becomes nonbasic at zero level, which yields

Table 5.2

Basic	c_j	1	-1	-1	0	0	0	0	0	0	
		x_{12}	x'_{13}	x'_{14}	x_{23}	x'_{26}	x_{35}	x'_{36}	x_{45}	x'_{56}	Solution
1	x_{12}	1	-1	-1	0	1	0	1	0	1	15
0	x_{23}	0	-1	-1	1	0	0	1	0	1	5
0	x_{45}	0	0	1	0	0	0	0	1	0	5
0	x_{35}	0	0	-1	0	0	1	0	0	1	13
Z		0	0	0	0	1	0	1	0	0	0

x_{12} =15, x_{23} = 5, x_{36} = 12, min{15,5,12} = 5. So flow f_5 = 5 in the route 1→2 → 3→ 6

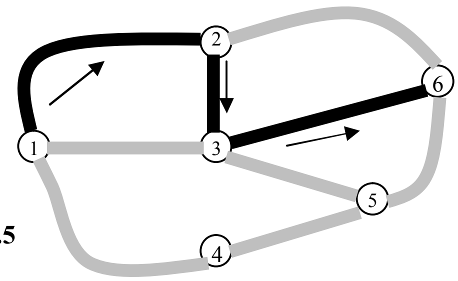

Fig: 2.5

The last table is feasible and optimal.
The optimal values are obtained by back substitution.
$x'_{13} = 0$ gives $x_{13} = 20 - x'_{13} = 20 - 0 = 20$;
$x'_{14} = 0$ gives $x_{14} = 5 - x'_{14} = 5 - 0 = 5$;
$x'_{26} = 0$ gives $x_{26} = 10 - x'_{26} = 10 - 0 = 10$;
$x'_{36} = 0$ gives $x_{36} = 12 - x'_{36} = 12 - 0 = 12$;
$x'_{56} = 0$ gives $x_{56} = 18 - x'_{56} = 18 - 0 = 18$;
The optimal solution is:

$x_{12}=15$, $x_{13}=20$, $x_{14}=5$, $x_{23}=5$, $x_{26}= 10$, $x_{35}=13$, $x_{36}= 12$, $x_{45}= 5$, $x_{56} = 18$.

So, the associated maximum flow is $z = x_{12} + x_{13} + x_{14}= 15 + 20 + 5 = 40$.

Remark1: In the first iteration since all the non basic variables that can enter the basis has relative cost factor -1 so any variable can be chosen arbitrarily. We choose the variable x_{45} because only one arch x_{45} is from node 4 and node 4 is directly linked with the source node 1 and no other node is linked with node 4, so the route 1→ 4 →5 must be used for any flow to pass through the node 4.

Remark 2: The flow in iteration is the value of the variable that has been is chosen to enter the basis. Note that in iteration 1, x_{45} is chosen to enter the basis and thus the flow in that iteration is $f_1 = x_{45}=5$; in the second iteration x_{26} has chosen which remained nonbasic at its upper bound. So the flow in that iteration is $f_2 = u_{26} =10$.Similarly for other iterations the flow can be determined. Now, the route in any iteration can be determined from the value of the variables that have changed in that iteration. For, in the second iteration $x_{12} = 10$, $x_{26}=10$, the values of x_{13}, x_{45}, x_{56} remained unchanged. So the route is the 1→2 → 6.

Again in the comparing the values of the variables in the fifth iteration with their values in previous iteration we see that,

In fourth iteration: $x_{12} = 10$, $x_{36} = 7$, $x_{45} = 5$, $x_{35} = 13$

In Fifth iteration: $x_{36} = 12$ $x_{12} = 15$, $x_{23} = 5$, $x_{45} = 5$,

$x_{35} = 13$.

Thus x_{23} entered the basis and the variables whose values have changed are x_{12}, x_{23} and x_{36}.

Thus the route in fifth iteration is 1→2→3→6 with a flow $f_5 = x_{23} = 5$.

In general to determine a route in any we set the variables with changed value in a sequence such that x_{si} , x_{ij}, x_{jk}, ….x_{pt} where s denotes the source node and t denotes the terminal node.

V. Conclusion

Bounded variable simplex method is very useful technique. It makes a large system of constraint set a small one. If there are upper bounds for the decision variables, Bounded variable simplex method takes less time to solve a problem then the simplex method. Using Bounded Variable simplex method we have solved a maximum flow problem and determine a technique to obtain the maximum flow in each iteration from the simplex table of that iteration. The route thorough which the maximum flow has passed is also determined.

....

1. Ford,L.R.,Jr.; Fulkerson, D.R.,"Maximal Flow through a Network",*Canadian Journal of Mathematics* (1956),pp.399-404.
2. Ford, L.R., Jr.; Fulkerson, D.R., *Flows in Networks*, Princeton University Press (1962).
3. G. B. Dantzig. *Linear programming and extensions*. Princeton University Press, Princeton, NJ, 1963.
4. Hamdy A. Taha ,*Operation Research- An Introduction*, 7th Edition.
5. Kambo N.S., *Introduction to Operation Research*.
6. Ralph P. Grimaldi *Discrete and Combinatorial Mathematics-An Applied Introduction*, 4th Edition.
7. R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. Network Flows: Theory, Algorithms, and Applications. Prentice Hall, Englewood Cliffs, NJ, 1993.
8. 8. Schrijver, Alexander, "On the history of the transportation and maximum flow problems", *Mathematical Programming* 91(2002) 437-445.
9. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein (2001). "26. Maximum Flow". *Introduction to Algorithms, Second Edition*. MIT Press and McGraw-Hill. pp. 643–668. ISBN 0-262-03293-7.