Journal of Engineering and Applied Sciences 13 (Special Issue 10): 8270-8276, 2018
ISSN: 1816-949X

An Alternate Simple Approach to Obtain the Maximum Flow in a Network Flow Problem

Protima Dash, Md. Mosfiqur Rahman and F.T. Zohora
Department of GED, Daffodil International University, 1207 Dhaka, Bangladesh

Abstract: The implementation version of Ford Fulkerson algorithm is known as Edmonds-Karp algorithm. The concept of Edmonds-Karp is to use BFS (Breadth First Search) in Ford Fulkerson execution as BFS always choices a path with least. This study likewise is a change form of Edmonds-Karp algorithm. In this study, we need to present an alternate approach for finding the maximum flow concerning less number of iterations and augmentation than Edmonds-Karp algorithm. A numerical illustration is appeared for showing the proposed algorithm to finding the maximal-flow problem by a Breadth First Search (BFS) method.

Key words: Maximum flow, Maximum Flow Problem (MFP), augmenting path, residual network, residual capacity, Breadth First Search (BFS)

INTRODUCTION

The maximum flow is to get a flow from source to sink for which the quantity of the flow amounts for the entire network is as large as possible. The problem is to decide the maximum quantity of flow that can be sent from the source node to the sink node. This type of flow problem is known as maximum flow problem and it is denoted by MFP. The maximal flow problem is one of essential problems for combinatorial optimization in weighted directed graphs. It gives extremely helpful models in a number of practical contexts including communication networks like railway traffic flow (Harris and Ross, 1955), oil, gas, water pipeline systems and power systems and. The maximal flow problem and its varieties have extensive variety of utilizations and have been contemplated broadly.

The powerful arrangement method to acquire the maximum flow in a flow network was presented by Fulkerson and Dantzig (1955). Originally and understood by practicing the simplex method for the linear programming and Lester (Ford and Fulkerson, 1956). Ford and Fulkerson (1962) which is the notable Ford-Fulkerson algorithm explained it by augmenting path algorithm. Keeping in mind the end goal to diminish the computational time, numerous researches about gave diverse algorithms. Freely, Dinic presented Bottleneck algorithm which was the idea of shortest path networks, called layered networks and earned runs in time $O(n^2m)$ where n is the number of nodes, m is the number of arcs and U is an upper bound on the integral arc capacities (Dinic, 1970). The improvement Version of Ford Fulkerson was Edmonds and Karp algorithm (1972) (Edmonds and Karp, 1972) which demonstrated that flows are augmented along shortest paths from source to sink and runs in time $O(nm^2)$. Already there has been a considerable measure of work on maximum flow problems. A distinctive change has been proposed by various researchers (Ahuja and Orlin, 1989; Ahuja *et al*., 1989; Jain and Garg, 2012; Mallick *et al*., 2016). A few researchers additionally proposed different techniques to unravel the MFP basing on the benefits and bad marks of the past strategies. Recently, Ahmed *et al*. (2014) and Khan *et al*. (2013) likewise proposed new approach for finding the maximum flow problem.

We have proposed a simple approach for finding the maximum flow in the networking flow problem from source to sink in this study. A numerical example has been calculated for finding the maximum value of a Maximum flow problem to test the viability and convenience by using proposed algorithm.

MATERIALS AND METHOD

Preliminaries: In this study some basic definitions,

Flow network: Let G = (V, E) be a directed graph with vertex set V and edge set E. A flow network G = (V, E) is a directed graph in which each edge $(u, v)\epsilon E$ has a non negative capacity c (u, v) = 0 and a distinguished source vertex s and sink vertex t (Mallick *et al*., 2016) . If $(u, v)\notin E$, then for convenience, we define c (u, v) = 0. A flow in G is a real-valued function f: $V\times V\rightarrow IR$ that satisfies these constraints:

Corresponding Author: Protima Dash, Department of GED, Daffodil International University, 1207 Dhaka, Bangladesh

Capacity constraints: If an edge (u, v) doesn't exist in the network, then c (u, v) = 0. For all (u, v)∈V×V, then 0 = f (u, v) = c (u, v) (Khan *et al*., 2013).

Skew symmetry: If edges (u, v) and (v, u) exist in a directed graph and for all (u, v)∈V×V then f (u, v) = -f (v, u) (Khan *et al*., 2013).

Residual network: A residual network graph indicates how much more flow is allowed in each edge in the network graph.

Residual capacity: The residual capacity of an edge with respect to a flow f, denoted C_f is the difference between the edge's capacity and its flow. That is $C_f(e) = c(e)-f(e)$. Original capacity of the edge minus current flow. Residual capacity is basically the current capacity of the edge.

Augmenting paths: An augmenting pathpin a network G = (V, E) with a flow f is a path from s to t in which every edge has positive capacity in the residual network. We can put more flow from s to t through p. We call the maximum capacity by which we can increase the flow on p the residual capacity of p, given by $C_f(p) = \min \{C_f(u, v): (u, v) \text{ is in } p\}$.

Proposed algorithm:
In this study, the major steps of the proposed algorithm are given as:
Step 1: For each edge (u, v)∈E [G], first initialize the flow f to 0
Step2: f (u, v) = f (v, u) = 0
Step 3: Calculate maximum capacity C in the flow network and then calculate $\Delta = 5\log_5 C$
Step 4: while $\Delta \geq 1$
Step 5: If there exists an augmenting path p from s to t in the residual network G_f with capacity at least Δ then select it and go to step 6: otherwise go to step 9
Step 6: set $C_f(p) = \min \{C_f(u, v): (u, v) \text{ is in } p\}$
Step 7: For each (u, v)∈p, if (u, v)∈E
Set $f(u, v) = f(u, v)+C_f(p)$ else $f(u, v) = -f(v, u)$
Step 8: Calculate the flow value
Step 9: $\Delta = \Delta/5$
Step 10: The flow is maximum

RESULTS AND DISCUSSION

Numerical illustration: A several number of numerical examples have been unraveled for finding the maximum value of a Maximum flow problem by utilizing proposed algorithm which is given underneath.

Example-1: Consider a pipeline system in an Agargaw Government Colony to supply gas in different sectors of an Agargaw Government Colony of Dhaka in the Bangladesh. The pipeline has a communicated capacity in per unit per hour between any two sectors which given as a maximum flow at which gas can flow by the pipe between those two sectors. Presently consider, we need to supply gas from the source sector to the sink sector, assume the source sector is say A and the sink sector is F and gas passes into 7 others sectors before getting from source to sink. Suppose B-H are of these 7 sectors and pipeline between any two sectors has indicated capacity. Demonstrates the input information's which have given to the problem talked above in the following Table 1.

Table 1: Indicate capacities of each pipeline between two sectors

Source part	Destination part	Capacity (Gallons/h)
A	B	27
A	C	18
A	E	38
B	D	20
B	E	11
C	E	22
C	F	15
D	F	10
D	H	18
E	D	37
E	F	15
F	H	34
G	I	48
H	G	20
H	I	30

Table 2: Similarity between sectors and vertices. Now, following similarity between sectors and vertices is used to make the diagram

Sector	Vertex
A	S
B	1
C	2
D	3
E	4
F	5
G	6
H	7
I	t

Calculate the maximum amount of gas which can flow from A-I

Solution of example: Now the problem acknowledged in the Example 1 has shifted in directed graph by portraying sector s as vertices of the diagram and pipelines between any two sectors as edges of the diagram. The capacity of the pipeline in unit every hour is shown as capacity of an edge in units between vertices.

Now, the maximum capacity of the graph is 48. So, the value of the variable C in the above algorithm will be 48. So, the initial graph corresponding to Table 1 and 2 is as follows (Fig. 1).

Network G is shown with each edge (u, v) labeled as f (u, v)/c (u, v). Now the maximum capacity in the flow network, Max C = 48.

Here:

$$\log_5 C = \log_5 48 = 2.40 \cong 2$$

So:

$$5\log_5 C = 5^2 = 25$$

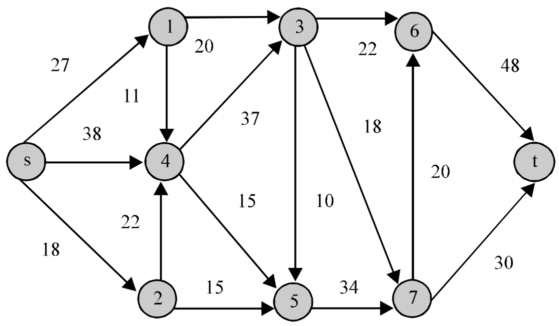

Fig. 1: The inital flow network corresponding to the problem

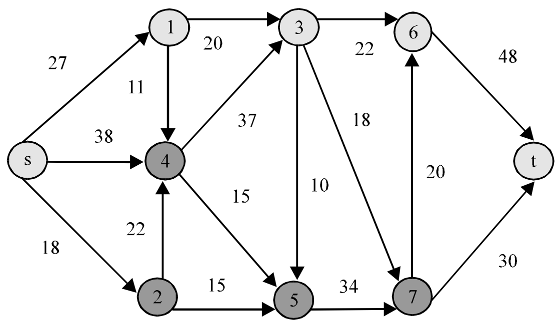

Fig. 2: Residual graph after any augmentation

Iteration 1: Δ = 25, so, now the augmenting path will be searched by the BFS (Breadth First search) procedure with capacity at least 25 in the residual graph. But there is no augmenting path with capacity at least 25. In this way, no flow will be added to the initial flow of the graph which is 0.

Iteration 2:

$$\Delta = \frac{\Delta}{5} = \frac{25}{5} = 5$$

So, now the augmenting path will be searched by the BFS (Breadth First search) till path with capacity at least 5 is establish in the graph which is given in Fig. 2 corresponding to the initial graph. The augmenting path will be searched till path with capacity at least 5 is found in the graph. Now in the consecutive Fig. 2 shows the residual graph and Fig. 3 shows the corresponding flow in the graph.

In the graph the augmenting path is to be arrangement, if there is more than 1 path satisfying the capacity criteria and then the path is unchanging by BFS (Breadth First Search) procedure on the basis of following format of the vertices and corresponding edges which has been given as effort.

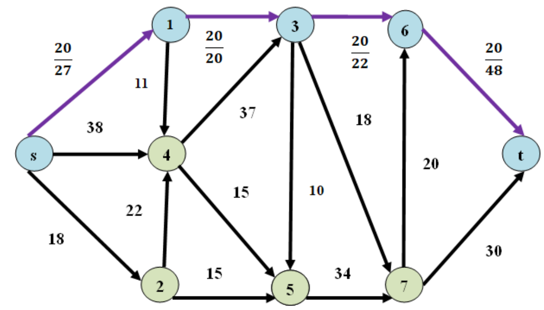

Fig. 3: Flow graph after 1st augmentation

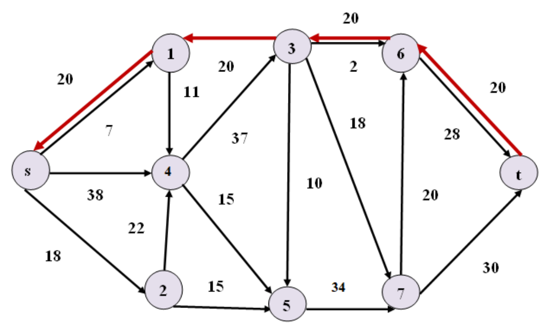

Fig. 4: Residual graph after 1st augmentation

1st augmentation: An augmenting path found in first iteration is s-1-3–6-t with capacity at least 5:

$$C_f(p) = \min\{27, 20, 22, 48\} = 20$$

The maximum flow value, f = 20. Now there is another augmenting path with capacity at least 5. Residual graph after first augmentation is shown in Fig. 4, flow graph shown in Fig. 3.

2nd augmentation: An augmenting path found in first iteration is s-4-3–7-t with capacity at least 5:

$$C_f(p) = \min\{38, 37, 18, 30\} = 18$$

The maximum flow value, f = 20 + 18 = 38. Now there is another augmenting path with capacity at least 5. Residual graph after 2nd augmentation is shown in Fig. 5, flow graph shown in Fig. 6.

3rd augmentation: An augmenting path found in first iteration is s-4-5-7-t with capacity at least 5:

$$C_f(p) = \min\{20, 19, 34, 12\} = 12$$

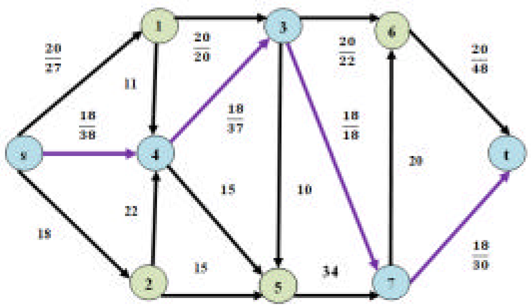

Fig. 5: Flow graph after 2nd augmentation

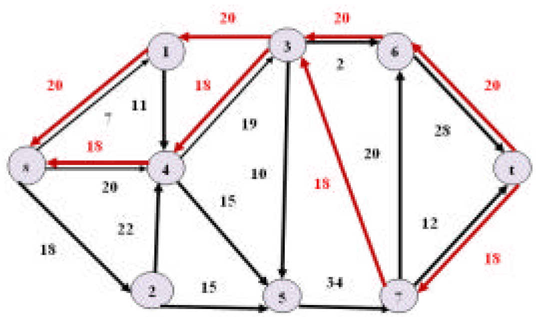

Fig. 6: Residual graph after 2nd augmentation

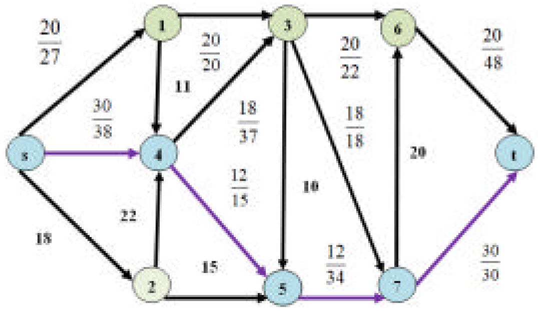

Fig. 7: Flow graph after 3rd augmentation

The maximum flow value, f = 38 + 12 = 50. Now there is another augmenting path with capacity at least 5. Residual graph after 3rd augmentation is shown in Fig. 7, flow graph shown in Fig. 8.

4th augmentation: An augmenting path found in first iteration is s-2-5–7-6-t with capacity at least 5:

$$C_f(p) = \min\{18, 15, 22, 20, 20, 28\} = 15$$

The maximum flow value, f = 50 + 15 = 65. Now Residual graph after 4th augmentation is shown in Fig. 9, flow graph shown in Fig. 10.

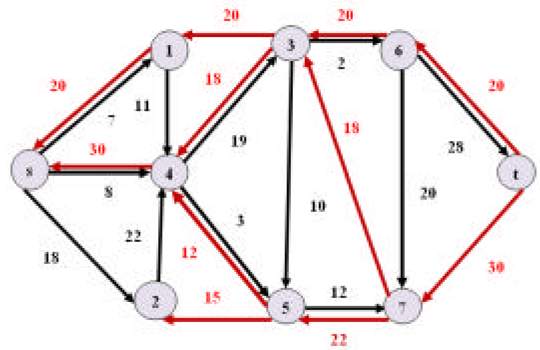

Fig. 8: Residual graph after 3rd augmentation

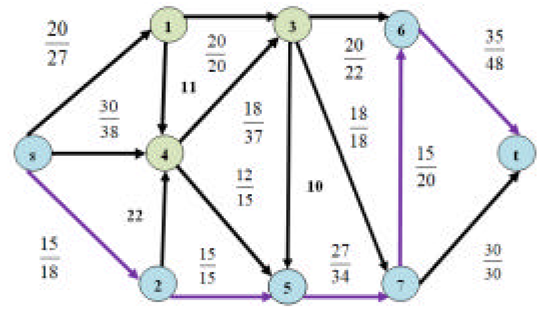

Fig. 9: Flow graph after 4th augmentation

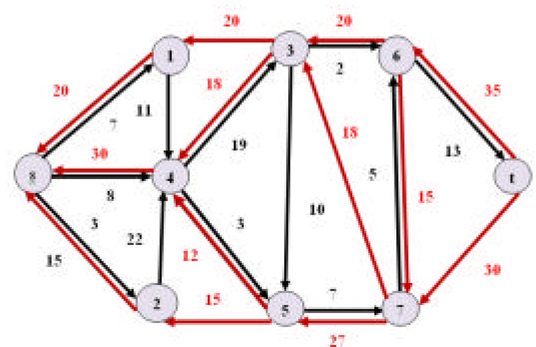

Fig. 10: Resudial graph after 4th augmentation

5th augmentation: An augmenting path found in 2nd iteration is s-4–3-5-7-6-t with capacity at least 5:

$$C_f(p) = \min\{6, 19, 10, 7, 5, 13\} = 5$$

The maximum flow value, f = 65 + 5 = 70. Now Residual graph after 5th augmentation is shown in Fig. 11, flow graph shown in Fig. 12.

Iteration 3:

$$D = \frac{D}{5} = \frac{5}{5} = 1$$

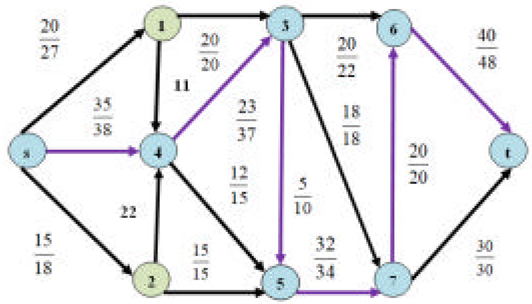

Fig. 11: Flow graph after 5th augmentation

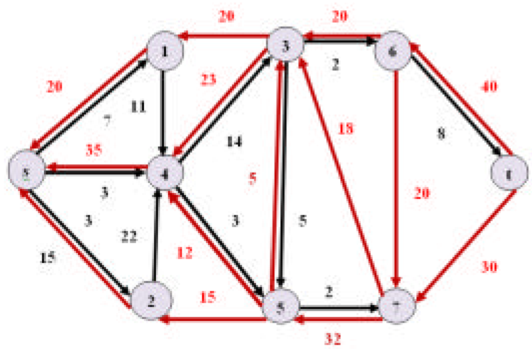

Fig. 12: Residual graph after 5th augmentation

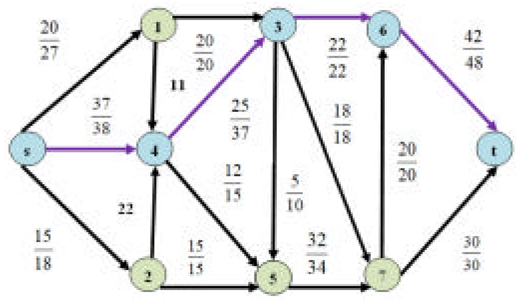

Fig. 13: Flow graph after 6th augmentation

6th augmentation: An augmenting path found in 3rd iteration is s-4-3–6-t with capacity at least 1:

$$C_f(p) = \min\{3, 14, 2, 8\} = 2$$

The maximum flow value, f = 70 + 2 = 72. Now Residual graph after 6th augmentation is shown in Fig. 13, flow graph shown in Fig. 14.

Now, there is no augmenting path with capacity at least 1. So, the algorithm finishes and the flow stops in iteration 3 and augmentation 6.

Therefore, the flow stops in iteration 3 and augmentation 6 and no path with capacity at least 1. Thus,

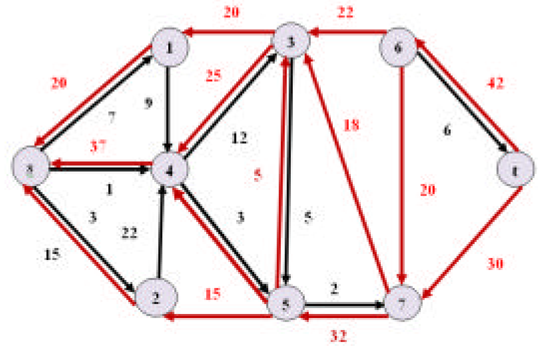

Fig. 14: Resulting residual network

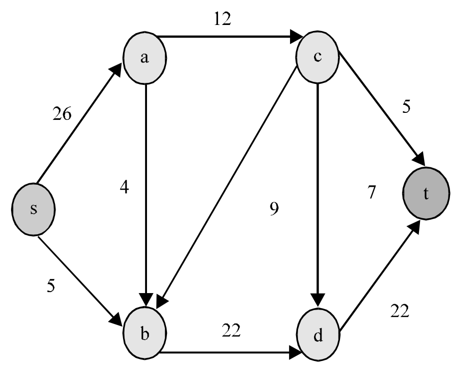

Fig. 15: The initial flow graph corresponding to the problem

the algorithm come to an ends and the resulting flow in network returns the maximum flow. Therefore, maximum flow value f = 72.

Example 2: An Oil company has the following pipeline network where each pipeline is labeled with its maximum flow rate (in thousands of gallons per hour).The specified capacities are shown in every pipeline between any two quarters which can between corresponding two quarters in the following Table 3.

Calculate the maximum amount of Oil which can flow from A-F

Solution of Example 2: Now the problem realized in the Example 1 has moved in directed graph by expressing sectors as vertices of the graph and pipelines between any two sectors as edges of the graph. The capacity of the pipeline in unit per hour is illustrated as capacity of an edge in units between vertices.

Now, following likeness between quarters and vertices is used to make the graph. So, the initial graph corresponding to Table 3 and Table 4 is shown in Fig. 15. Therefore, calculating the above problem, we get the value of maximum flow in the network is 21.

Table 3: signify capacities of each pipeline between two sectors

Source sector	Destination sector	Capacity (Gallons/h)
A	B	26
A	C	5
B	C	4
B	D	12
D	C	9
C	E	22
D	F	5
D	E	7
E	F	22

Table 4: Relationship between sectors and vertices

Sector	Vertex
A	S
B	a
C	b
D	c
E	d
F	t

Solution using edmondskarp algorithm: Presently there has been tackle a same network flow problem by utilizing Edmonds-Karp algorithm. The methodology in every iteration is quickly outlined under:

1st iteration: Chosen the augmenting path is s-2-5-7-t along capacity 15. The maximum flow value, f =15.

2nd iteration: Chosen the augmenting path is s-2-4-5-7-t along capacity 3. The maximum flow value, f = 15+3 = 18.

3rd iteration: Chosen the augmenting path is s-4-5-7-t along capacity 12. The maximum flow value, f = 18+12 = 30.

4th iteration: Chosen the augmenting path is s-1-3-6-t along capacity 20. The maximum flow value, f = 30 + 20 = 50.

5th iteration: Chosen the augmenting path is s-4-3-6-t along capacity 2. The maximum flow value, f = 50+2 = 52.

6th iteration: Chosen the augmenting path is s-4-3-5-7-6-t along capacity 4. The maximum flow value, f = 52 +4= 56.

7th iteration: Chosen the augmenting path is s-4-3-7-6-t along capacity 16. The maximum flow value, f = 56 +16 = 72.

Comparison: Now there has been constructed the following table to compare among different algorithm and proposed algorithm.

Outcome: Table 5 shows a comparison of no. of iteration and No. of augmentation stated to obtain the maximum flow and it is perceived that this proposed algorithm states less number of iterations and augmentations to achieve the maximum flow.

Table 5: Assessment of the residual obtained by different methods

	Number of iterations		Number of augmentation	
Name of algorithm	Example 1	Example 2	Example 1	Example 2
Ford-Fulkerson	8	4	8	4
Edmonds-Karp	7	4	7	4
Chintan and Deepak Garg	5	3	6	4
Md.Al-Amin Khan	5	4	6	4
Faruque Ahmed	6	4	6	4
Modified Edmonds-Karp	4	3	6	4
A simple Approach (Proposed)	3	2	6	4

CONCLUSION

Change in any numerical algorithm is a constant procedure to discover more exact outcome. By methods for this, we needed to propose an algorithm that will be brought about maximum flow in a network flow problem involving less number of iterations and less number of augmentations. For this, we considered various types of algorithm like Ford-Fulkerson, Edmonds-Karp, Md. Al-Amin Khan, Faruque Ahmed, Chintan and Deepak Garg. algorithm. It is observed that, this proposed algorithm is receiving less number of iterations and less number of augmentations to calculate the maximum flow in a flow network.

REFERENCES

Ahmed, F., M. Khan, A.R. Khan, S.S. Ahmed and M. Uddin, 2014. An efficient algorithm for finding maximum flow in a network-flow. J. Phys. Sci., 19: 41-50.

Ahuja, R.K. and J.B. Orlin, 1989. A fast and simple algorithm for the maximum flow problem. Oper. Res., 37: 748-759.

Ahuja, R.K., J.B. Orlin and R.E. Tarjan, 1989. Improved time bounds for the maximum flow problem. SIAM. J. Comput., 18: 939-954.

Dinic, E.A., 1970. Algorithm for solution of a problem of maximum flow in networks with power estimation. Sov. Math. Dokl., 11: 1277-1280.

Edmonds, J. and R.M. Karp, 1972. Theoretical improvements in algorithmic efficiency for network flow problems. J. ACM., 19: 248-264.

Ford, L.R. and D.R. Fulkerson, 1956. Maximal flow through a network. Can. J. Math., 8: 399-404.

Ford, L.R. and D.R. Fulkerson, 1962. Flows in Networks. Princeton University Press, Princeton, New Jersey, USA., Pages: 194.

Fulkerson, D.R. and G.B. Dantzig, 1955. Computation of maximum flow in network. Nav. Res. Logist. Q., 2: 277-283.

Harris, T.E. and F.S. Ross, 1955. Fundamentals of a method for evaluating rail net capacities (No. RM-1573). Rand Corporation, Santa Monica, California.

Jain, C. and D. Garg, 2012. Improved edmond karps algorithm for network flow problem. Intl. J. Comput. Appl., 37: 48-53.

Khan, M., A. Rashid, A.R. Khan and M.S. Uddin, 2013. An innovative approach for solving maximal-flow problems. J. Phys. Sci., 17: 143-154.

Mallick, K.K., A.R. Khan, M.M. Ahmed, M.S. Arefin and M.S. Uddin, 2016. Modified EDMONDS-KARP algorithm to solve maximum flow problems. Open J. Appl. Sci., 6: 131-140.