International Journal of Innovation and Applied Studies
ISSN 2028-9324 Vol. 31 No. 4 Jan. 2021, pp. 703-711
© 2021 Innovative Space of Scientific Research Journals
http://www.ijias.issr-journals.org/

# Modification of EDMONDS-KARP Algorithm for Solving Maximum Flow Problem

***Dilruba Akter*[1]*, Md. Sharif Uddin*[2]*, and Faria Ahmed Shami*[3]**

[1]Department of Applied Mathematics, Gono Bishwabidyalay, Savar, Dhaka, Bangladesh

[2]Department of Mathematics, Jahangirnagar University, Savar, Dhaka, Bangladesh

[3]Department of Mathematics, Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Gopalganj, Bangladesh

**ABSTRACT:** Edmonds-Karp algorithm is an implementation of the Ford-Fulkerson method for computing the maximum flow in a flow network in much more optimized approach. Edmonds-Karp is identical to Ford-Fulkerson except for one very important trait that is the search order of augmenting paths is well defined. This paper presents some modifications of Edmonds-Karp algorithm for solving maximum flow problem (MFP). Solution of MFP has also been illustrated by using the proposed algorithm to discuss the functionality of proposed method.

**KEYWORDS:** Maximum Flow Problem, Breadth First Search, Augmenting Path, Residual Network, Edmonds-Karp algorithm.

## 1 INTRODUCTION

Maximum flow problems involve finding a feasible maximum flow through a single-source, single-sink flow network. Where each edge is labeled with capacity (the maximum amount of stuff that it can carry). The goal is to figure out how much stuff can be pushed from the vertex s (source) to the vertex t (sink). Maximum flow problem has got a vast application in the field of Mathematics, Computer Science, Engineering sector, Management and Operations Research. At first the effective solution procedure to obtain the maximum flow in a flow network was introduced by Lester R. Ford and Delbert R. Fulkerson [1], [2] in 1955 which is the well-known Ford-Fulkerson algorithm. Later, the improvement of the Ford-Fulkerson method (FFM), " Edmonds-Karp algorithm" [3]; which observed that augmenting along shortest paths leads to a polynomial-time algorithm and performs better than the FFM. Again extensive discussion, further improvement and effectiveness of the solution procedure of MFP are studied by a good number of researchers [1], [2], [4] - [14]. Very recently, Mallick J.B [12] and Ahmed, F. et al [14]. and Khan] also proposed new approaches for finding maximum flow problem. In this paper, a modified Edmonds-Karp algorithm is proposed to compute maximum amount of flow from source to sink for a MFP. Numerical illustration of the proposed algorithm is also done by solving a good number of examples to test the effectiveness and usefulness of the proposed algorithm.

## 2 BASIC IDEA

Some of the basic definitions related to maximum flow problems are given below with an aim to make easy the readers with the article.

### 2.1 CAPACITY CONSTRAINT

The flow $f(u,v)$ through an edge cannot be negative and cannot exceed the capacity of the edge $c(u,v) > 0$; $f(u,v)$ ≤ $c(u,v)$. If an edge $(u,v)$ doesn't exist in the network, then $c(u,v)$ =0.

### 2.2 Flow Conservation

Aside from the source vertex $s$ and sink vertex *t,* each vertex $u$ belongs to $v$ must satisfy the property that the sum of $f(u,v)$ for all edges $(u,v)$ in $E$ (the flow into $u$) must equal the sum of $f(u,w)$ for all edges $(u,w)$ belongs to $E$ (the flow out of $u$).

This property ensures that flow is neither produced nor consumed in the network, except at $s$ and $t$.

### 2.3 Skew Symmetry

For consistency, the quantity $f(v,u)$ represents the net flow from vertex $u$ to $v$. This means that it must be the case that $f(u,v) = - f(v,u)$; this holds even if both edges $(u,v)$ and $(v,u)$ exist in a directed graph.

### 2.4 Residual Network

Intuitively, given a flow network and a flow, the residual network consists of edges that can admit more flow. More formally, let $G = (V, E)$ a flow network with source $s$ and sink$t$. Let $f$ be a flow in $G$, and consider a pair of vertices $u, v \in V$ .The amount of additional flow which can be pushed from $u$ to $v$ before exceeding the capacity $c(u,v)$ is the residual capacity of $(u,v)$, given by $c_f(u,v) = c(u,v) - f(u,v)$.

### 2.5 Augmenting Path

Augmenting paths are simply any path from the source to the sink that can currently take more flow. Over the course of the algorithm, flow is monotonically increased. So, there are times when a path from the source to the sink can take on more flow, and that is an augmenting path. This can be found by a breadth-first search, as we let edges have unit length.

## 3 Proposed Algorithm

Here the proposed algorithm is a modified version of Ford-Fulkerson algorithm depending on three important ideas: residual networks, augmenting paths, and cuts. The steps of this algorithm are summarized below. The basic steps of this algorithm are explained below.

**Modified Edmonds-Karp (G, S, T) Algorithm:**

```
Step1 :     First initialize the flow f to '0' for each edge $(u,v) \in E[G]$
Step 2 :    do [f(u,v) ←0 [i.e f (u,v) =0]
Step 3 :    f(v,u) ←0[i.e f (v,u) =0]
Step 4:     $C \leftarrow \max_{(u,v)\in E} c(u,v)$
Step 5:     I ←floor(log$_2$ C)+1
Step 6:     while $I \geq 1$
Step 7:     do while there exists an augmenting path p from s to t in the residual network $G_f$ of capacity at least I
Step 8 :    do $c_f(p) \leftarrow \min \{c_f(u,v) : (u,v) \text{ is in } p\}$
Step 9 :    for each edge $(u,v)$ in p
Step 10 :   do f[u,v] ← f[u,v]+$c_f(p)$
Step 11 :   f[u,v]← -f[v,u]
Step 12:    I$_1$= $\frac{I}{I}$
Step 13:    return f.
```

## 4 MATHEMATICAL ILLUSTRATION

Two numerical examples have been solved for finding the maximum value of a MFP by using proposed algorithm which is given below.

### 4.1 EXAMPLE-1

Suppose one parking system is there in Jahangirnagar University Campus to supply proper parking of vehicles in the Campus. The road between any two places has a stated capacity in number of vehicles per day, given as a maximum flow at which trips can be made through the road between those two stations. Now, suppose we want to route the trips from the entrance of the campus, Suppose A to the station, say I and vehicles trips through 7 other station before reaching from source to sink. Suppose these 7 areas are B, C, D, E, F, G, H and road between any two stations has defined capacity. So, the problem is to find how to route various trips to maximum the number of trips that can be made per day from A to I without violating the limits.

***Table 1. Defined capacities of each road between two stations***

| Source Area | Destination Area | Capacity (gallons/hour) | Source Area | Destination Area | Capacity (gallons/hour) |
|---|---|---|---|---|---|
| A | B | 34 | D | G | 22 |
| A | C | 20 | D | H | 18 |
| A | E | 35 | E | D | 35 |
| B | D | 20 | E | F | 15 |
| B | E | 8 | F | H | 35 |
| C | E | 8 | G | I | 48 |
| C | F | 15 | H | G | 20 |
| D | F | 10 | H | I | 30 |

Calculate the maximum amount of water which can flow from A to I.

**SOLUTION:**

Now the problem discussed in the Example 1 is converted into directed graph by representing stations as vertices of the graph and road between any two areas as edges of the graph. The capacity of the road in number per hour is represented as capacity of an edge in units between vertices.

Now, following correspondence between areas and vertices is used to create the graph.

***Table 2. Correspondence between stations and vertices***

| Stations | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| Vertex | s | 1 | 2 | 3 | 4 | 5 | 6 | 7 | t |

The initial graph corresponding to Table 1 and Table 2 is as follows.

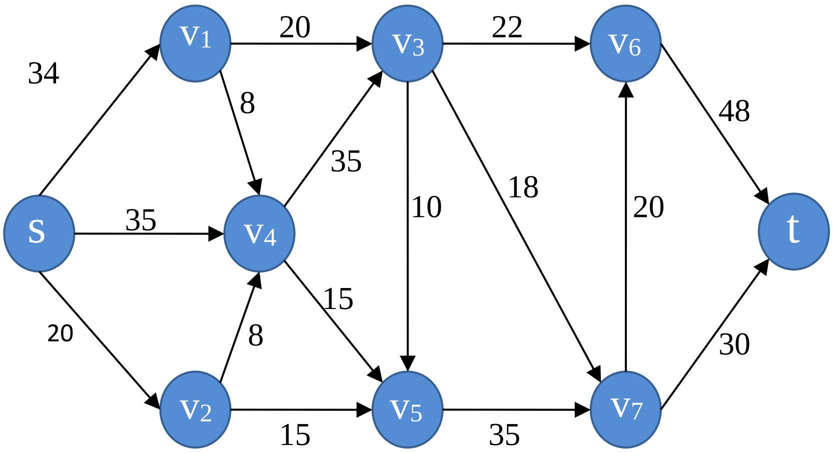


*Fig. 1. The initial flow network corresponding to the problem*

Now we use the modified Edmonds-Karp Method to compute a maximum flow in *G*.

According to the Figure 1, the maximum capacity is 48. So, the value of the variable *C* in the above algorithm will be 48.

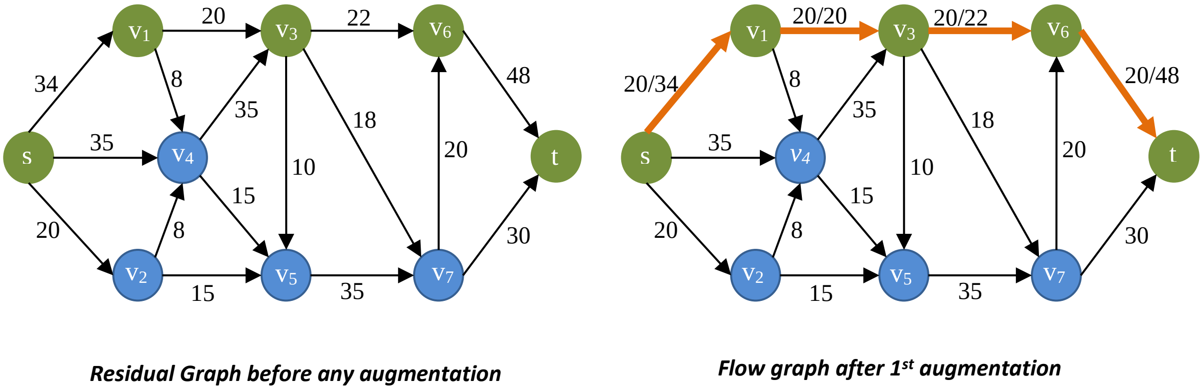
So,

$$\text{floor}(\log_2 48) + 1 = 6$$

which will be value of the variable I in the 1st iteration according to the algorithm.

### 1ST ITERATION: I =6

- So, the augmenting path with capacity at least 6 will be searched by the Breadth First Search procedure in the residual graph which is given below in corresponding to the initial graph.
- The augmenting path will be searched till path with capacity at least 6 is found in the graph.



*Residual Graph before any augmentation*

*Flow graph after 1st augmentation*

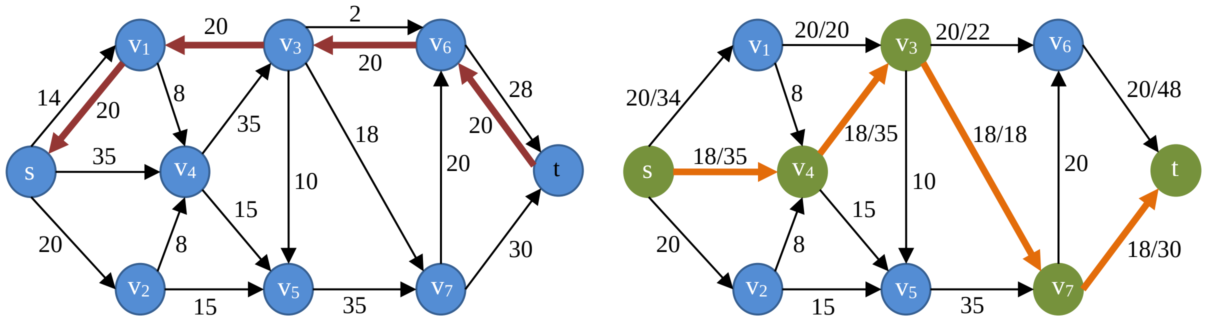
*Fig. 2. Residual Graph before any augmentation and flow graph after 1st augmentation*

Now, in the consecutive figures, the left side figure shows the residual graph and the right side figure shows the corresponding flow in the graph.

Whenever the augmenting path is to be found in the graph, if there are more than 1 path satisfying the capacity criteria, then the path is determined by BFS procedure on the basis of sequential ordering of the vertices and corresponding edges which have as input.

### 1ST AUGMENTATION

- So, the augmenting path found in 1st iteration is ***s- $v_1$- $v_3$ -$v_6$ – t*** with capacity 20.
- So, the initial flow is augmented by 20 units and the flow in the graph is shown in the above right side figure 2 giving maximum flow value ***f*=20**.
- The residual graph after 1st augmentation is shown below in



*Residual Graph after 1st augmentation*

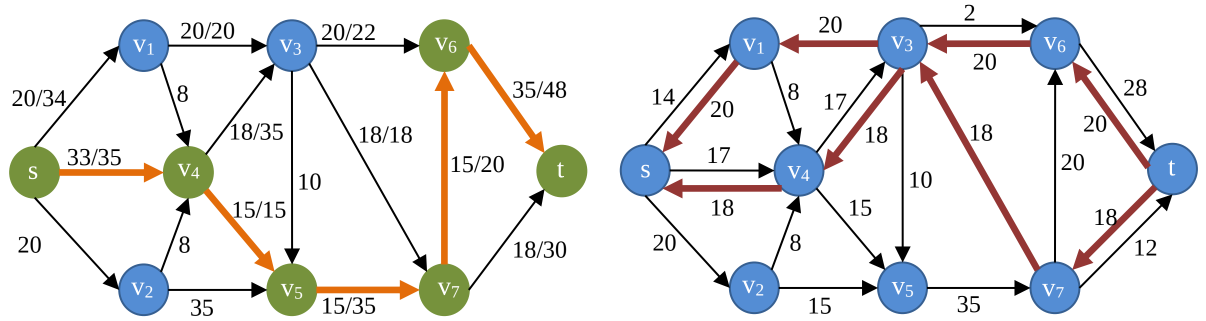
*Flow Graph after 2nd augmentation*

*Fig. 3. Residual Graph after 1st augmentation and flow graph after 2nd augmentation*

### 2ND AUGMENTATION

- Now, again there is a path with capacity at least 6 and the path found in the same 2nd iteration is ***s-$v_4$-$v_3$-$v_7$-t*** with capacity 18.
- So, the maximum flow is augmented by 18 units and the flow in the graph is shown in the above figure 3 giving maximum flow value ***f*= 20 + 18 = 38**.
- Now, there is no path with capacity at least 18.

Same procedure will be followed until variable I becomes < 6



*Residual Graph after 2nd augmentation*

*Flow graph after 3rd augmentation*

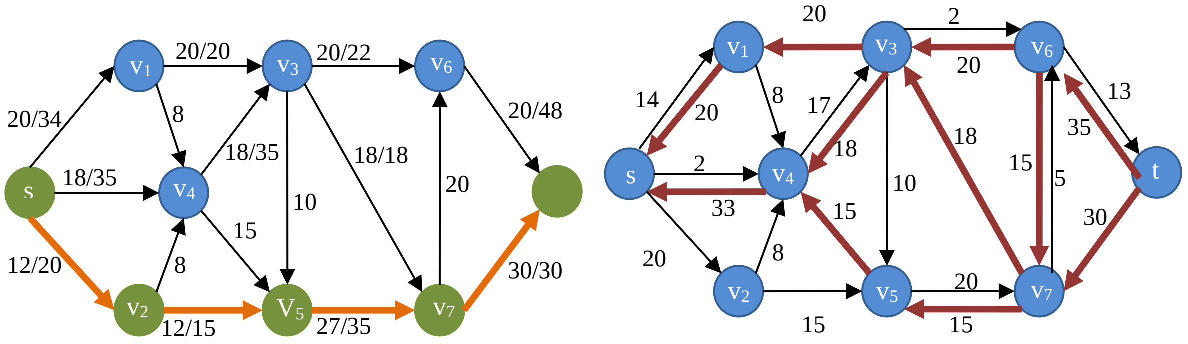
*Fig. 4. Residual Graph after 2nd augmentation and flow graph after 3rd augmentation*

### 3RD AUGMENTATION

- Now, again there is a path with capacity at least 6 and the path found in the same 2nd iteration is ***s – v₄ - v₅ - v₇ – v₆ –t*** with capacity 15.
- So, the maximum flow is augmented by 15 units and the flow in the graph is shown in the above figure 4 giving maximum flow value ***f*= 38 + 15 = 53**.
- Now, there is no path with capacity at least 15.

### 4TH AUGMENTATION

- Now, again there is a path with capacity at least 6 and the path found in the same 2nd iteration is ***s - v₂ - v₅ - v₇ - t*** with capacity 12.
- So, the maximum flow is augmented by 12 units and the flow in the graph is shown in the following figure 5 giving maximum flow value ***f*= 53 + 12 = 65**.
- Now, there is no path with capacity at least 6



***Residual Graph after 3rd augmentation***

***Flow Graph after 4th augmentation***

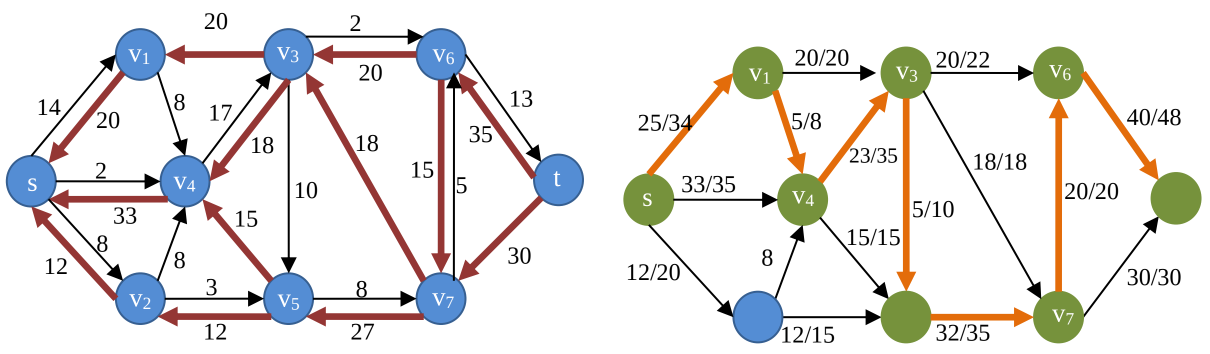
***Fig. 5. Residual Graph after 3rd augmentation and flow graph after 4th augmentation***

### 2ND ITERATION: I = FLOOR $\frac{I}{I} = \frac{6}{6} = 1$

So, now the augmenting path with capacity at least 1 will be searched in the residual graph which is given below in figure 6 corresponding to the initial graph.

### 5TH AUGMENTATION

- So, the augmenting path found in 3rd iteration is **s- *v₁- v₄ -v₃ – v₅ – v₇ - v₆ - t*** with capacity 5.
- So, the initial flow is augmented by 5 units and the flow in the graph is shown below in the right side of the figure 6.
- giving maximum flow value ***f*= 65+5 = 70**.
- The residual graph after 4th augmentation is shown below in Figure 6



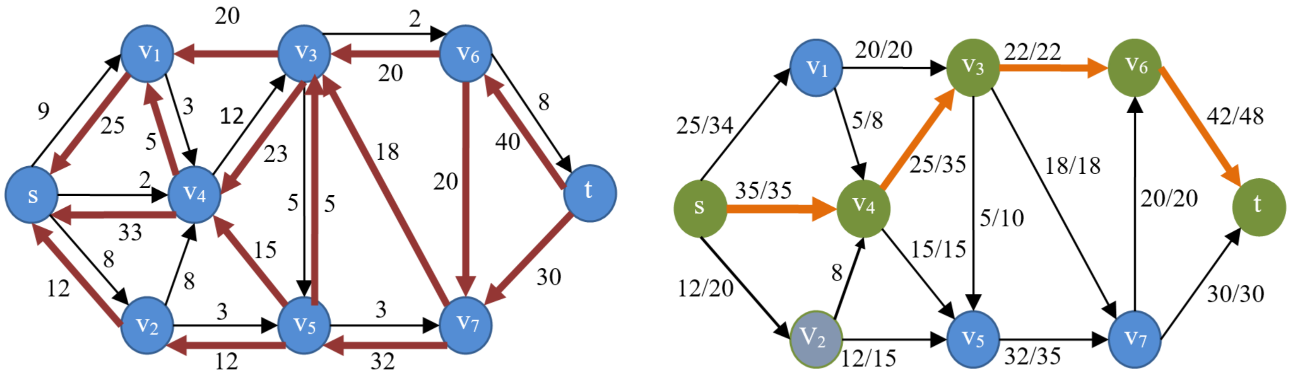
*Residual Graph after 4th augmentation*

*Flow graph after 5th augmentation*

***Fig. 6.*** ***Residual Graph after 4th augmentation and flow graph after 5th augmentation***

6TH AUGMENTATION

- Now, again there is a path with capacity at least 1 and the path found in the same 2nd iteration is **s- *$v_4$* -*$v_3$* – *$v_6$* - *t*** with capacity 2.
- So, the initial flow is augmented by 2 units and the flow in the graph is shown below in the right side figure 7 giving maximum flow value **f=70+2=72**.



***Residual Graph after 5th augmentation***

***Flow graph after 6th augmentation***

***Fig. 7.*** ***Residual Graph after 5th augmentation and flow graph after 6th augmentation***

The residual graph after 6th augmentation is shown below:

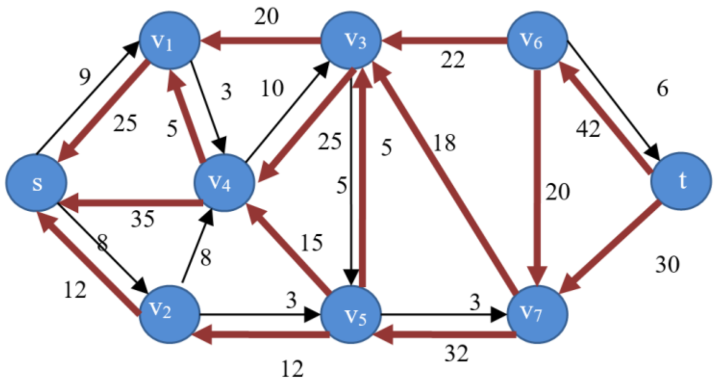


***Fig. 8.*** ***Resulting Residual Network***

Now, there is no augmenting path with respect to above residual graph.

So, the maximum flow value ***f* = 72**.

### 4.2 EXAMPLE-2

Suppose one pipeline system is there in any certain area to supply water in different dwellers of any areas. The pipeline between any two dwellers has a stated capacity in per unit per hour, given as a maximum flow at which water can flow through the pipe between those two dwellers. Now, suppose we want to supply water from the source area, suppose A to the sink area, say F and water passes through 4 other areas before reaching from source to sink. Suppose these 4 areas are B, C, D, E and pipeline between any two areas has defined capacity. So, the problem is to calculate the maximum amount of gas which can flow from A to F.

*Table 3. Defined capacities of each pipeline between two areas*

| Production Plant (Source Area) | Distribution Spot (Destination Area) | Capacity (unit / hour) |
|---|---|---|
| A | B | 10 |
| A | C | 10 |
| B | C | 2 |
| B | D | 4 |
| B | E | 8 |
| C | E | 9 |
| D | F | 10 |
| E | D | 6 |
| E | F | 10 |

Calculate the maximum amount of water which can flow from A to F.

**SOLUTION:**

Now the problem discussed in the Example 2 is converted into directed graph by representing areas as vertices of the graph and pipelines between any two areas as edges of the graph. The capacity of the pipeline in unit per hour is represented as capacity of an edge in units between vertices.

Now, following correspondence between areas and vertices is used to create the graph.

*Table 4. Correspondence between areas and vertices*

| Area | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| Vertex | s | 1 | 2 | 3 | 4 | t |

So, the initial graph corresponding to and is shown in

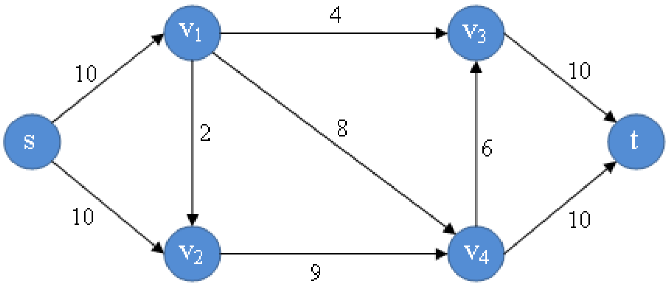


*Fig. 9. The initial flow network corresponding to the problem*

Solving the above example using proposed algorithm, the maximum flow value f = 19.

## 5 Result And Discussion

shows a comparison of no. of iteration and no. of augmentation required to obtain the maximum flow by using various existing methods and this mentioned proposed Modified Edmonds-Karp algorithm by means of the above two sample examples and it is seen that our proposed algorithm requires less number of iterations and augmentation paths to reach the maximum flow.

*Table 5. Comparison of the result obtained by different methods*

| Example | Method | Number of Iteration | Number of Augmentation |
|---|---|---|---|
| 1 | Edmonds-Karp | 7 | 7 |
| | Modified Edmonds-Karp (proposed) | 2 | 6 |
| 2 | Edmonds-Karp | 5 | 5 |
| | Modified Edmonds-Karp (proposed) | 1 | 3 |

## 6 Conclusion

In this paper, Edmonds-Karp algorithm is being modified to compute maximum amount of flow from source to sink in a flow network. Two real life problem of maximum flow problems is solved by using mentioned proposed algorithm, Ford-Fulkerson algorithm, Edmonds-Karp algorithm, Faruque Ahmed et al.'s algorithm, Mallick et al.'s algorithm and to test the efficiency of the proposed algorithm. During this process, it is observed that, this proposed algorithm performs better than other mentioned algorithm and this method can be used to solve any type of maximum flow problems.

## References

[1] Ford Jr., L.R. and Fulkerson D.R.: Flows in Networks, Princeton University Press, Princeton, NJ, 1962.

[2] Ford, Jr. L.R. and Fulkerson D.R., "Maximal Flow through a Network.", Canadian Journal of Mathematics, " vol.8, pp. 399-404, 1956.

[3] Edmonds, J. and Karp, R.M., "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems.", Journal of the Association for Computing Machinery, vol.19, pp. 248-264, 1972.

[4] Goldberg, A.V., Tardos, E. and Tarjan, R.E., "Network Flow Algorithms. In: Korte, B., Lovasz, L., Promel, H.J. and Schriver, A., Eds., Paths, Flows, and VLSI-Layout, vol. 9, Springer-Verlag, Berlin Heidelberg, 101-164, 1990.

[5] Goldberg, A.V. and Tarjan, R.E., "A New Approach to the Maximum-Flow Problem",. Journal of the Association for Computing Machinery, vol.35, pp. 921-940,1988.

[6] Goldberg, E. and Rao, S., "Beyond the Flow Decomposition Barrier.", Journal of the ACM, vol45, pp. 783-797,1998.

[7] Cherkasky, B.V., "An Algorithm for Constructing Maximal Flows in Networks with Complexity of Operations.", Mathematical Methods of Solution of Economical Problems, vol.7, pp. 117-125. (In Russian) (1977).

[8] Jain, C. and Garg, D., "Improved Edmond Karps Algorithm for Network Flow Problem.", International Journal of Computer Applications, vol.37, pp. 48-53, 2012.

[9] Dinic, E.A., "Algorithm for Solution of a Problem of Maximum Flow in a Network with Power Estimation", Soviet Math Doklady, vol.11, pp. 1277-1280.

[10] Goldfarb, D. and Jin, Z.Y., "A Faster Combinatorial Algorithm for the Generalized Circulation Problem.", Mathematics of Operations Research, vol.21, pp. 529-539, 1996.

[11] Hamdy, A.T., Operations Research: An Introduction, 8th Edition, Pearson Prentice Hall, Upper Saddle River, New Jersey. 2007.

[12] Mallick, J.B., "An Efficient Network Flow Approach: On Development of New Idea & A Modification of EDMONDS-KARP Algorithm." M. Phil. Thesis, Department of Mathematics, Jahangirnagar University. 2014.

[13] Galil, Z. and Naamad, A., "An Algorithm for the Maximal Flow Problem.", Journal of Computer and System Sciences, vol.21, pp.203-217, 1980.

[14] Ahmed, F., Khan, Md.A., Khan, A.R., Ahmed, S.S. and Uddin, "An Efficient Algorithm for Finding Maximum Flow in a Network-Flow.", Journal of Physical Sciences, vol.19, pp. 41-50, 2014.