RESEARCH ARTICLE OPEN ACCESS

Computing Max-Flow by New Method.

Hend El- Morsy
Umm Alqura University.Kingdom Of Saudi Arabia.

ABSTRACT:
In this paper we will compute max flow from source to sink by using shortest path algorithm . We will discuss the uniqness of max flow for the same graph.
Key words:Max flow , Shortest path , Algorithm.

Date Of Submission: 05-09-2019 Date Of Acceptance: 22-09-2019

Difinitions:
Algorithm:
In mathematics and computer science, an algorithm is an effective method expressed as a finite list of well-defined instructions for calculating a function.
In simple words an algorithm is a step-by-step procedure for calculations.

Shortest -path algorithm:

An algorithm that is designed essentially to find a path of minimum length between two specified vertices of connected weighted graph i.e (if there is a path from vertex u to vertex v in network G, any path of minimum length from u to v is a shortest path (SP) from u to v , and its weight is the shortest distance (SD) from u to v [1].

Flow network:
In graph theory, a flow network (also known as a transportation network) is a directed graph where each edge has a capacity and each edge receives a flow. The amount of flow on an edge cannot exceed the capacity of the edge. Often in operations research, a directed graph is called a network, the vertices are called nodes and the edges are called arcs. A flow must satisfy the restriction that the amount of flow into a node equals the amount of flow out of it, unless it is a source, which has only outgoing flow, or sink, which has only incoming flow [2].

Max flow:
In optimization theory, maximum flow problems involve finding a feasible flow through a single-source, single-sink flow network that is maximum [3].

Main Results:
We will illustrate algorithm that computed max flow by using shortest path.

The algorithm:
Input:
Directed , weighted graph with E number of edge ,
f(e) flow of edge .
C(e) capacity of edge .

Algorithm body:
1- Initialized : Max flow = 0
For (each edge e in E do ,
F(e) = 0 ;
2- Repeat search for S-T path P while it exists.
a. Find if there is a path using shortest path ; it exists if f(e) < f(c) for every edge e on P.
b. If no path found , return max flow.
c. Else , find minimum edge value for path P.
3- Repeat step 2.
Until P not reached during shortest path.
4- Output F.
End algorithm.

Example 1:

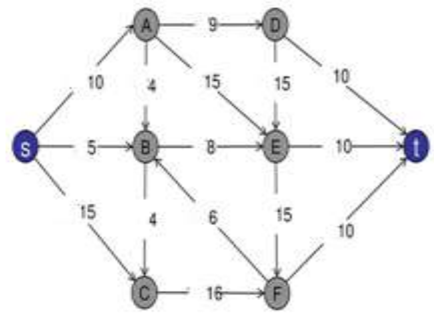

Fig.(1)

If we compute max flow by using shortest path , we will get:
Step 1: From S-T we find shortest path (S B E T) (5+8+10 =23) with minimum flow (5).
Subtract 5 from each edge on path, we get Fig.(1-a)

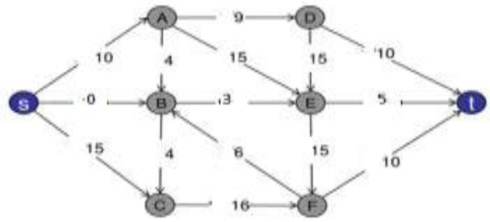

Fig.(1-a)

Step 2: Next shortest path from S – T will be (SABET) with minimum flow(3).
Subtract 3 from each edge on path , Fig.(1-b)

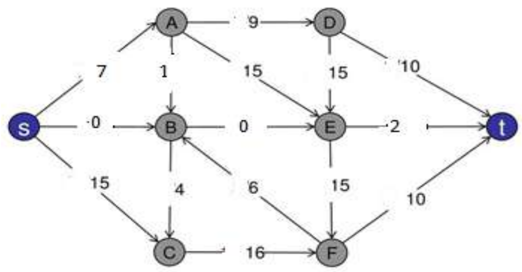

Fig.(1-b)

Step 3: Next shortest path from S – T will be (SAET) with minimum flow(2).
Subtract 2 from each edge on path , Fig.(1-c)

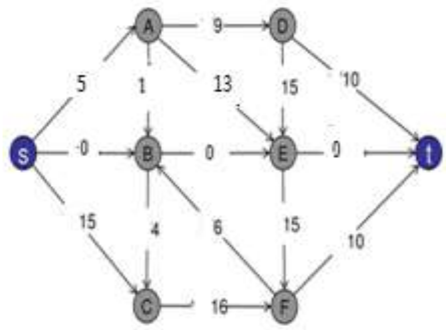

Fig.(1-c)

Step 4: Next shortest path from S – T will be (SADT) with minimum flow(5).
Subtract 5 from each edge on path , Fig.(1-d)

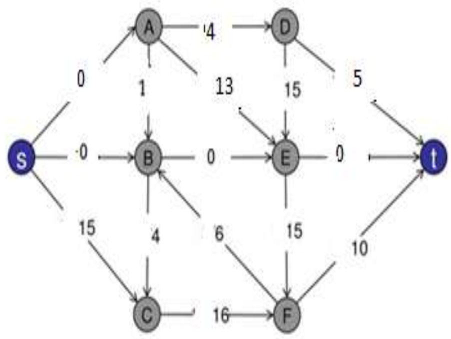

Fig.(1-d)

Step 5: The last shortest path from S – T will be (SCFT) with minimum flow(10).
Subtract 10 from each edge on path , Fig.(1-e)

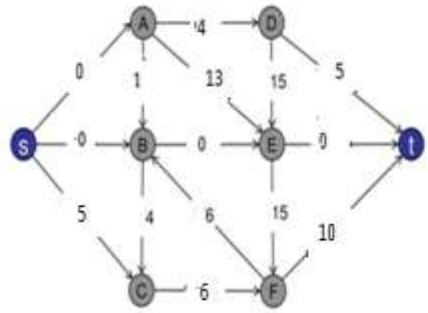

Fig.(1-e)

Max flow = 5 + 3 + 2 + 5 + 10 = 25 .

Note:
If we compute max flow regardless of our method (Shortest path) we may get another results.

Show Fig.(1) in the previous example.
If we compute max flow starting with any path from S to T with the same steps illustrated in example 1 :

1- Start with path(SADT) with minimum flow (10).
2- Then path(SCFT) with minimum flow (10).
3- The path (SBET) with minimum flow (5).
4- Finally path (SCFBET) with minimum flow(3).

Then max flow = 10 + 10 + 5 + 3 = 28

We find that:
Max flow computed by shortest path method ≤ Max flow computed by any other method.

Example 2:
We will compute max flow by using shortest path algorithm:

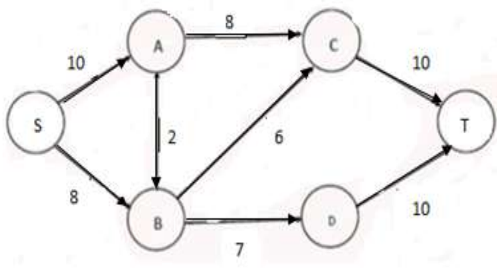

Fig.(2)

Step 1: Shortest path from S – T will be (SBCT) with minimum flow(6).
Subtract 6 from each edge on path , Fig.(2-a)

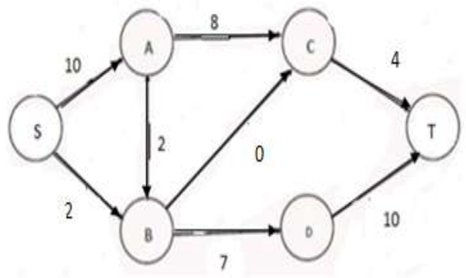

Fig.(2-a)

Step 2: Next Shortest path from S – T will be (SBDT) with minimum flow(2).
Subtract 2 from each edge on path , Fig.(2-b)

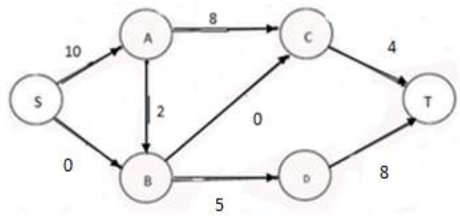

Fig.(2-b)

Step 3: Next Shortest path from S – T will be (SACT) with minimum flow(4).
Subtract 4 from each edge on path , Fig.(2-c)

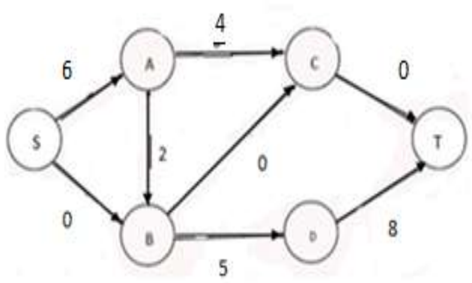

Fig.(2-c)

Step 4: The last Shortest path from S – T will be (SABDT) with minimum flow(2).

Subtract 2 from each edge on path , Fig.(2-d)

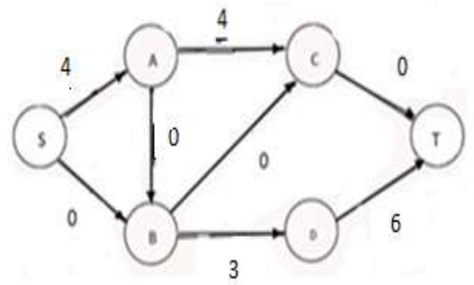

Fig.(2-d)

Then max flow = 6 + 2 + 4 + 2 = 14.
But if we compute max flow regardless of our method (Shortest path) we may get another results.

1- Path (SACT) with minimum flow(8).
2- Path (SBDT) with minimum flow(7).
3- Pah (SABCT) with minimum flow(2).

Then max flow = 8+ 7 + 2 = 17.
Which greater than the first.

Example 3 :

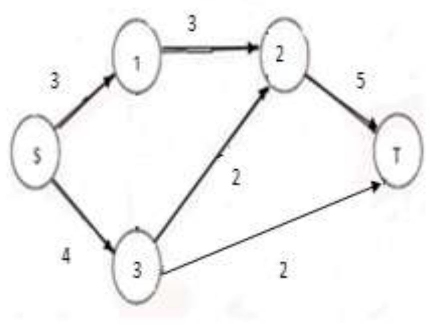

Fig.(3)

We will compute max flow by shortest path algorithm with the same steps of the previous examples:

1- Path (S3T) with minimum flow 2 As in Fig.(3-a).
2- Path (S32T) with minimum flow 2 As in Fig.(3-b).
3- Then path (S12T) with minimum flow 3 As in Fig.(3-c).

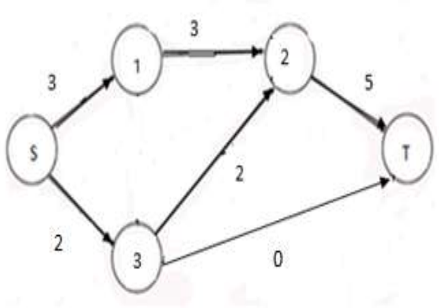

Fig.(3-a)

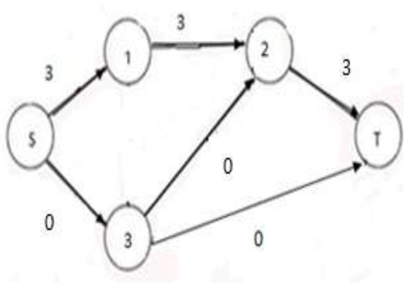

Fig.(3-b)

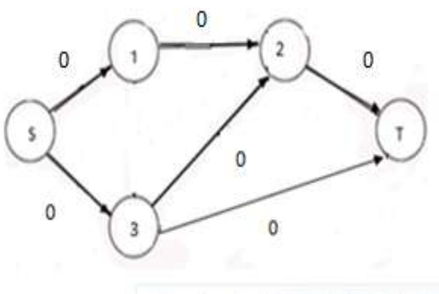

Fig.(3-c)

Then max flow = 3 + 2 + 2 = 7.
*If we compute max flow regardless of method (Shortest path) we get:
1- path (S12T) with minimum flow 3.
2- Path (S3T) with minimum flow 2.
3-Path (S32T) with minimum flow 2.
Then max flow = 3+ 2 + 2 = 7.
Which equal to max flow getting by the first method.

Theorem:
Max flow computed by shortest path algorithm $\leq$ Max flow computed by the other algorithms.

Proof:
The proof of this theorem comes directly from the above discussion.

REFERENCES

[1]. Balakrishnan V.K , "Schaum's Outline of theory and problems of
[2]. graph theory" . University of Maine . Orono , Maine , 1995.
[3]. A.V. Goldberg, É. Tardos and R.E. Tarjan, Network flow algorithms, Tech. Report STAN-CS-89-1252, Stanford University CS Dept., 1989
[4]. Schrijver, A. (2002). "On the history of the transportation and maximum flow problems". Mathematical Programming. 91 (3): 437–445.

Hend El- Morsy" Computing Max-Flow by New Method." International Journal of Engineering Research and Applications (IJERA), vol. 9, no. 9, 2019, pp. 16-19