NETWORK FLOW PROBLEMS

Some real life problems like those involving the flow of liquids through pipe, current through wires, and delivery of goods can be modeling by using flow of networks.

Transport network

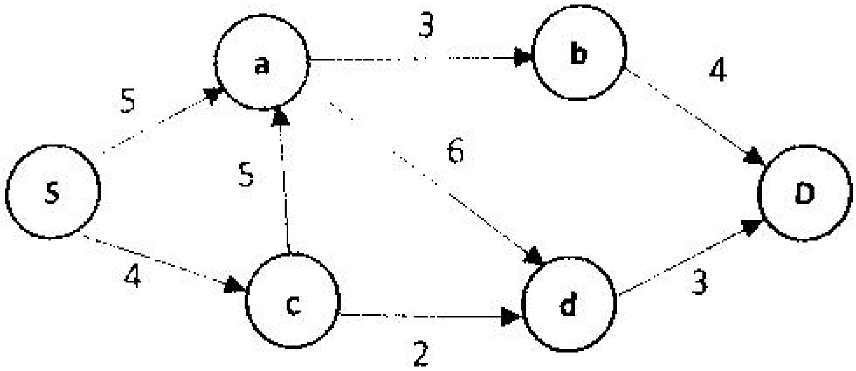
A flow network or transport network is a connected directed graph G = (V, E) that does not contain any loops such that,

- There are exactly two distinguished vertices S and D, called the source and sink (destination) of graph G respectively
- There is a non-negative real valued function K defined on edge E called the capacity function of G.

Then (G, K) is called transport network and function K is called capacity function of G.

The vertices distinct for S and D are called intermediate vertices. If $e \in E$ then K(e) is called capacity of e.

Example: Example of transport network is shown in fig below;

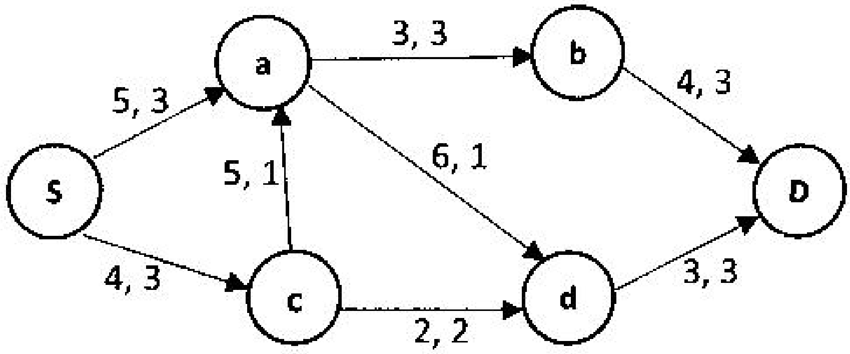


Flow

Let (G, K) be a transport network. Then a flow in G is a non-negative real valued function F defined on edge E such that,

$0 \le F(e) \le K(e)$ for each edge e belongs to E. Where, F(e)=flow to the edge e and K(e)=maximum capacity to the edge e.

Example: Following transport network show that the capacity and flow in each of the edges.



From the above figure the flow-in of vertex 'a' is F(S, a) + F (c, a) = 3+1=4

The flow out of vertex 'a' is F(a, b)+F(a, d)=3+1=4

The flow in S=0

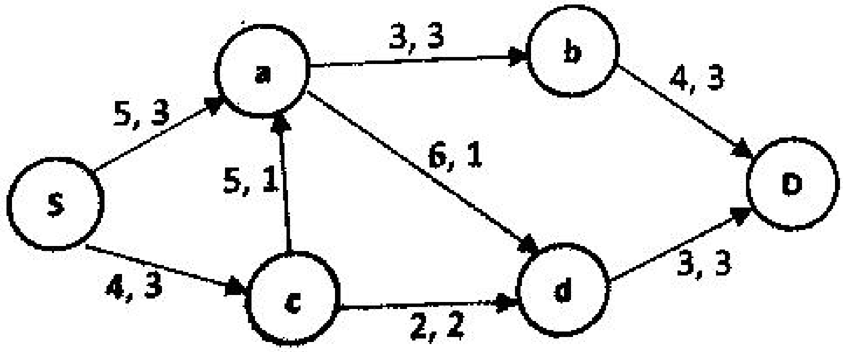
Flow in of vertex D=F(b, D) + F(d, D)=3+3=6 and so on

Note: Flow out of source vertex = flow in of sink vertex

Saturated and Unsaturated edges

The edges for which the flow and capacity are equal are called saturated edges. Otherwise they are called unsaturated edges.

Example:



Here edges (a, b), (c, d) and (d, D) are called saturated edges. And remaining edges are called unsaturated edges.

Slack of edge

The slack of edge 'e' is the difference of capacity and maximum flow of that edge. The slack of each saturated edge is zero.

Example: In the above figure the slack of edge (S, a) = 5-3=2

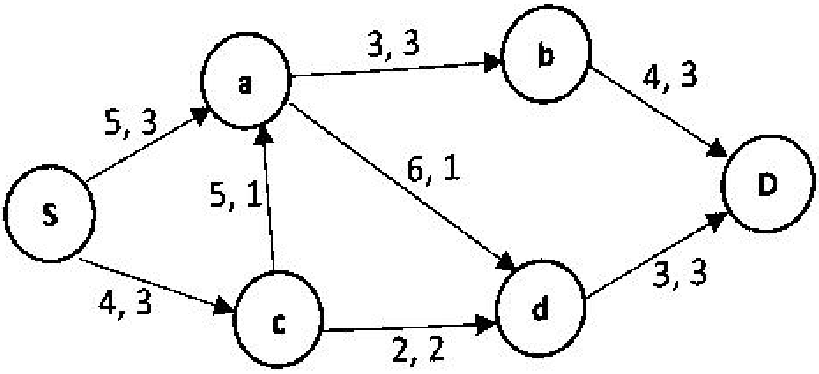
Slack of edge (a, b) = 3-3 = 0

Slack of edge (a, d) = 6-1=5 and so on

Cut

Let (G, K) is a transport network with source S and sink D. let X is subset of V and Y= V-X where X contains at least source S and Y contains at least sink D. Then the edge list (X, Y) are called cut edges of given network flow.

Example:



Let X={S, a, d} and Y= {b, c, D} then

(X, Y)= {(S, c), (a, b), (d, D)} is called cut edge of given transport network.

Minimum cut

A cut edge (X, Y) is said to be minimum cut edge of transport network (G, K) if there is no any cut edge (W, Z) that is less than (X, Y). I.e. K(W, Z) < K(X, Y) is false.

Example: In the above figure possible cut edges and their capacity are,

Case 1: let X={S} and Y= {a, b, c, d, D} then

(X, Y)= {(S, a), (S, c)} =5+4=9

Case 2: let X={S, a} and Y= {b, c, d, D} then

(X, Y)= {(S, c)} =4

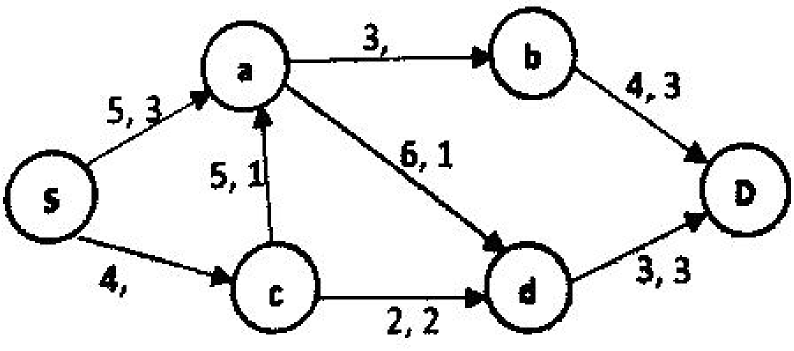
Case 3: let X={S} and Y= {a, b, c, d, D} then

(X, Y)= {(S, a), (S, c)} =5+4=9

F-augmenting path or flow augmenting path

A flow-augmenting path from source *s* to sink *t* is a sequence of edges from *s* to *t* such that, for each edge in this path, the flow F(e) < K(e) on forward edges and F(e) > 0 on backward edges. It means that such a path is not optimally used yet, and it can transfer more units than it is currently transferring. If the flow for at least one edge of the path reaches its capacity, then obviously the flow cannot be augmented.

Example:



In the above figure the possible f-augmenting paths are: {S→a→b→D}, {S→c→a→b→D}

If we reach the sink t, the flows of the edges on the augmenting path that was just found are updated by increasing flows of forward edges and decreasing flows of backward edges, and the process restarts in the quest for another augmenting path. Here is a summary of the algorithm.

```
augmentPath (network with source s and sink t)
        for each edge e in the path from s to t
                if forward(e)
                        f(e) += slack(t);
                else
                        f(e) -= slack(t);
```

Maximum flow

A flow F in a network (G, K) is called a maximum flow if $|F'| \leq |F$, for every flow F′ in (G, K). Simply a flow F in a network (G, K) is called maximum flow if the value of F is the largest possible value of any flow other possible flows in (G, K).

Ford Fulkerson algorithm

This algorithm is used to find the maximum flow of given network (G, K). Let a network flow diagram with source s and sink t.

```
1.  Start
2.  set flow of all edges and vertices to 0;
        Label = (null, ∞);
        Labeled = {s};
3.  while labeled is not empty // while not stuck;
    3.1.    Detach a vertex v from labeled;
    3.2.    for all unlabeled vertices u adjacent to v
        3.2.1.  if forward(edge(vu)) and slack(edge(vu)) > 0
                        Label (u) = (v+, min (slack (v), slack (edge (vu))))
        3.2.2.  else if backward(edge(vu)) and f(edge(uv)) > 0
                        Label (u) = (v-, min (slack (v), f (edge (uv))));
        3.2.3.  if u got labeled
            3.2.3.1.    if u == t
                            augmentPath (network);
                                Labeled = {s};   // look for another path;
            3.2.3.2.    else
                            Include u in labeled;
4.  Stop
```

Simple idea behind Ford-Fulkerson Algorithm

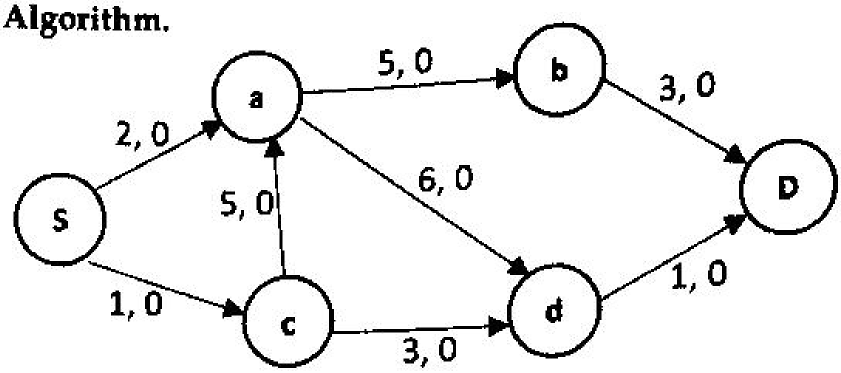
The following is simple idea of Ford-Fulkerson algorithm

1. Start with initial flow as 0
2. While there is 'a' augmenting path from source to sink.
   - Add this path-flow to flow
3. Return flow.

Time Complexity

Time complexity of the above algorithm is O(max_flow * E). We run a loop while there is an augmenting path. In worst case, we may add 1 unit flow in every iteration. Therefore the time complexity becomes O(max_flow * E).

Example 1: Find max flow of the following network flow graph by using Ford Fulkerson Algorithm.



Solution: At first listing f-augmenting paths as

{S→a→b→D},{S→c→d→D},{S→a→d→D},{S→a→c→d→D}, {S→c→d→a→b→D}

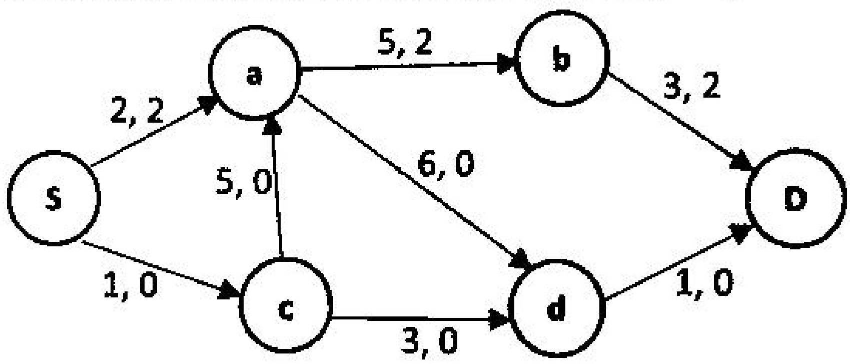
Step 1: In the f-augmenting path {S→a→b→D},

Slack of edge (S, a)=2-0=2 (minimum)

Slack of edge (a, b)=5-0=5

Slack of edge (b, D)=3-0=3

Since the minimum slick is 2 hence add 2 to every edge's flow of the path {S→a→b→D},



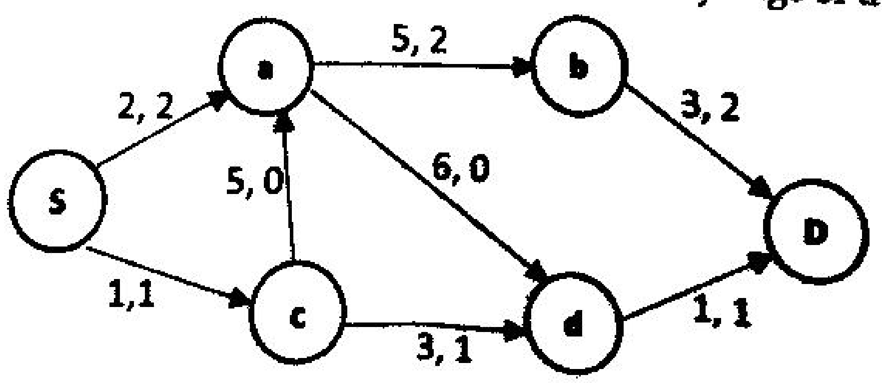
Step 2: In the f-augmenting path {S→c→d→D},

Slack of edge (S, c) =1-0=1 (minimum)

Slack of edge (c, d) =3-0=3

Slack of edge (d, D) =1-0=1

Since the minimum slick is 1 hence add 1 to every edge of the path {S→c→d→D},

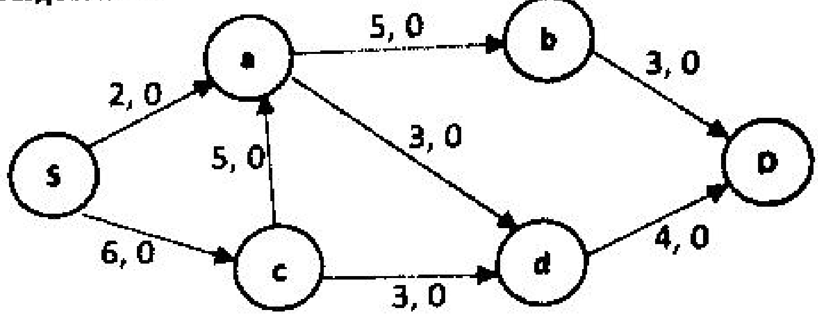


Now there is no any possible path form source to sink without saturated edge

Hence maximum flow of given network graph is 2+1=3

Hence flow out from source=2+1=3=flow in to the sink=2+1=3.

Example 2: Find max flow of the following network flow graph by using Ford Fulkerson Algorithm.



Solution: At first listing f-augmenting paths as

{S→a→b→D},{S→c→d→D},{S→a→d→D},{S→a→c→d→D},{S→c→a→b→D}, ,{S→c→a→d→D}, {S→c→d→a→b→D}

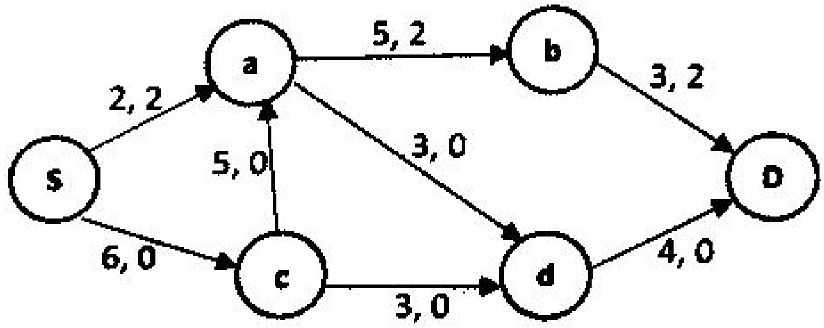
Step 1: In the f-augmenting path {S→a→b→D},

Slack of edge (S, a)=2-0=2 (minimum)

Slack of edge (a, b) =5-0=5

Slack of edge (b, D) =3-0=3

Since the minimum slick is 2 hence add 2 to every edge's flow of the path {S→a→b→D},



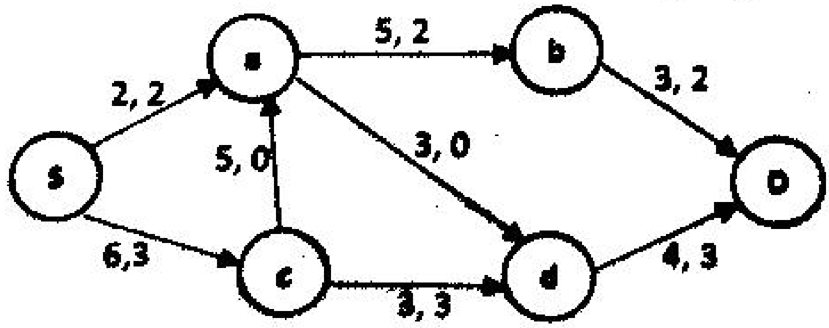
Step 2: In the f-augmenting path {S→c→d→D},

Slack of edge (S, c) =6-0=6

Slack of edge (c, d) =3-0=3 (minimum)

Slack of edge (d, D) =4-0=4

Since the minimum slick is 3 hence add 3 to every edge of the path {S→c→d→D}.



Step 3: In the f-augmenting path {S→a→d→D},

No change

Step 4: In the f-augmenting path {S→a→c→d→D},

No change

Step 5: In the f-augmenting path {S→c→a→b→D},

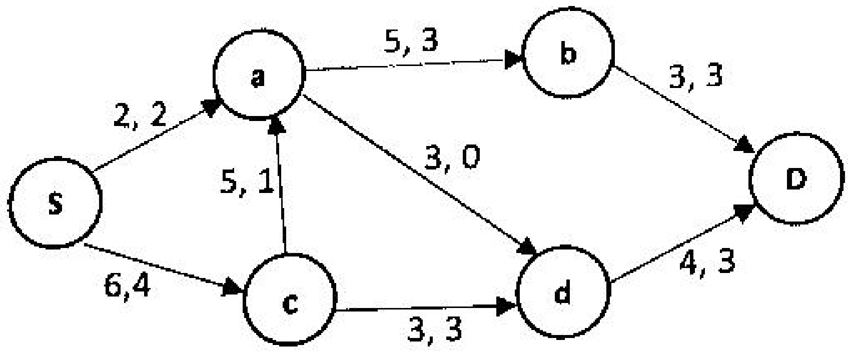
Slack of edge (S, c) =6-3=3

Slack of edge (c, a) =5-0=5

Slack of edge (a, b) =5-2=3

Slack of edge (b, D) =3-2=1 (minimum)

Since the minimum slick is 1 hence add to every edge of the path {S→c→a→b→D},



Step 6: In the f-augmenting path {S→c→a→d→D},

Slack of edge (S, c) =6-4=2

Slack of edge (c, a) =5-1=4

Slack of edge (a, d) =3-0=3

Slack of edge (d, D) =4-3=1 (minimum)

Since the minimum slick is 1 hence add to every edge of the path {S→c→a→d→D},