Algorithm Design & Analysis (CSE222)

Lecture-20

Recap

- Bellman Ford
    - Uses dynamic paradigm
    - Distance vector protocols

- Network Flows
    - Flows & Cut

Outline

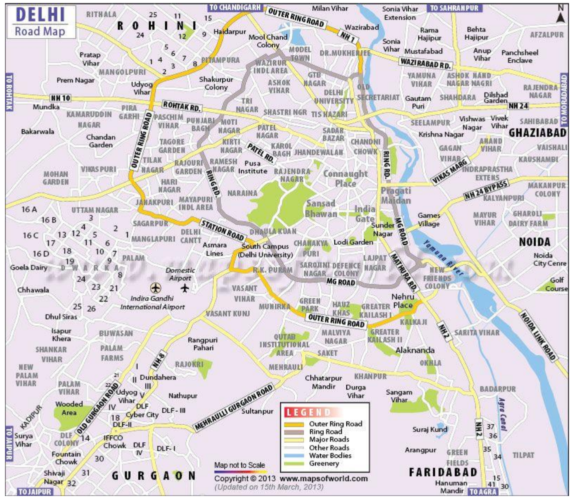
- Network Flows
  - Maximum Flow (Ford-Fulkerson)

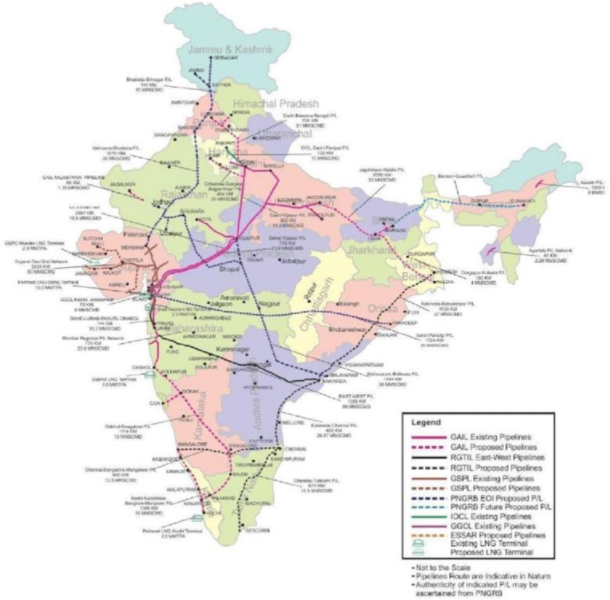
Flow function

Conservation constraint: In flow = out flow at every vertex, but s and t.
Capacity constraint: flow is less or equal to capacity at each edge.

Cut is a partition of vertex set into two disjoint sets.
Capacity of cut: total capacity of all edges whose nodes are in two disjoint sets.

Network Flow



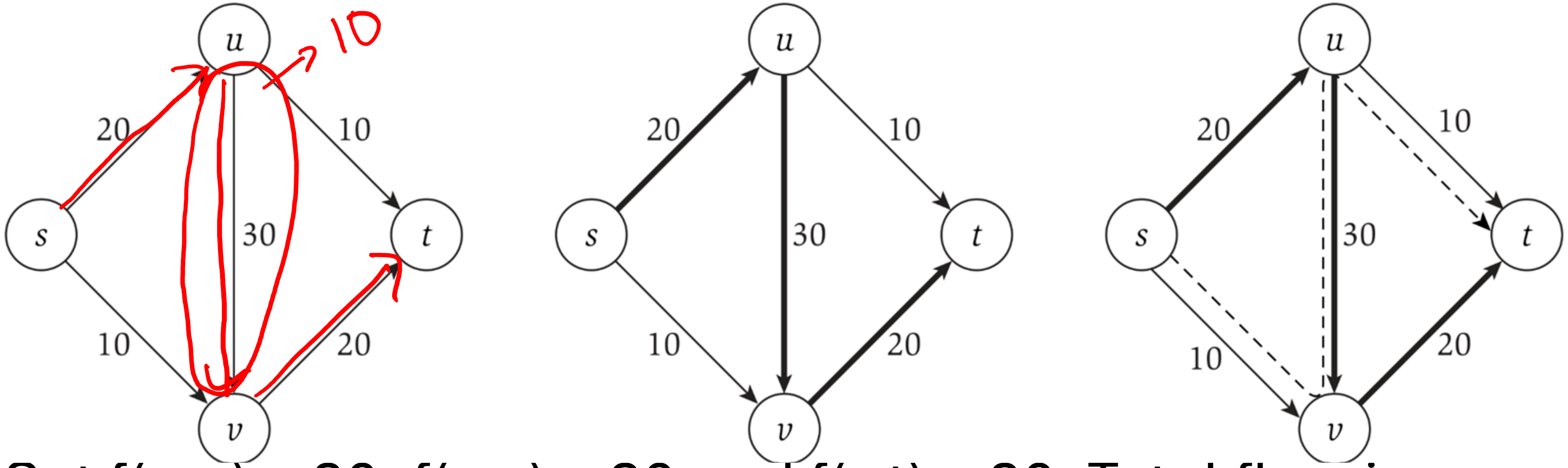


<u>Flow network</u>: single source and single sink.

Here we will only consider integer valued capacities.

Network Flow

<u>Flow network</u>: single source and single sink.

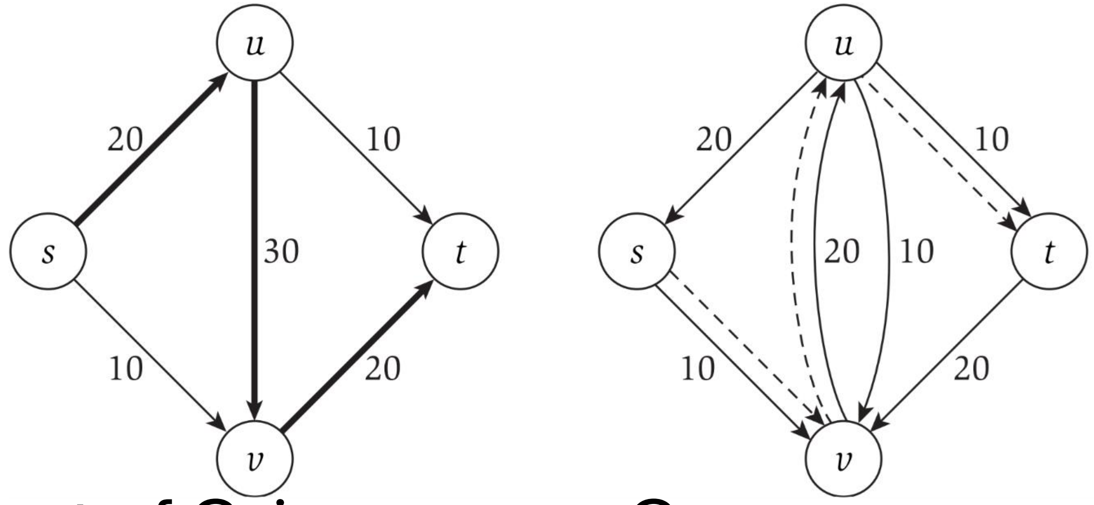


Set f(s,u) = 20, f(u,v) = 20 and f(v,t) = 20. Total flow increased to 20!
Is it the max flow of the graph?

Push forward on edge based on remaining capacity.
Push backward on edge based on flow.

Graph Residual

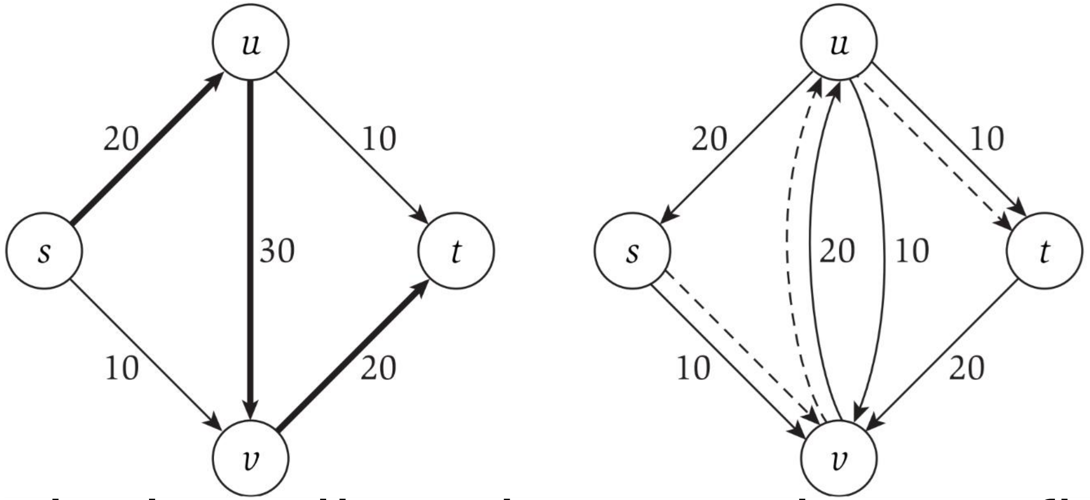
Given flow f on G (flow network), the residual graph G_f is defined as follows.



- Node set of G_f is same as G.
- For any e ∈ E, if f(e)<c(e) then include edge e in G_f with a capacity of c(e)-f(e). The edge is called forward edge with left over capacity c(e)-f(e).
- For any e = (u,v), if f(e)>0, then include e' = (v,u) with a capacity f(e). This edge is called backward edge.

Graph Residual

Given flow f on G (flow network), the residual graph G_f is defined as follows.



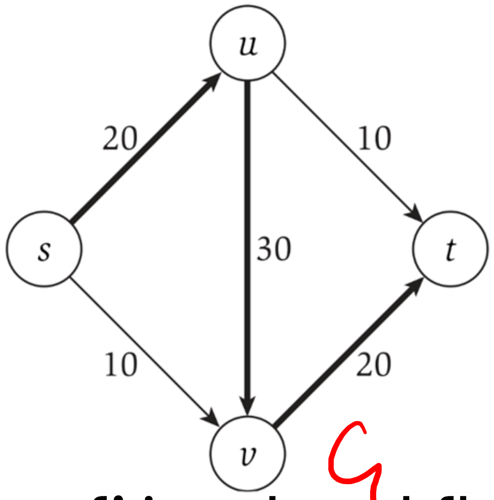
- Forward edge tells us how much more flow can be passed through the edge.
- Backward edge tells us how much flow can be undone through the edge.

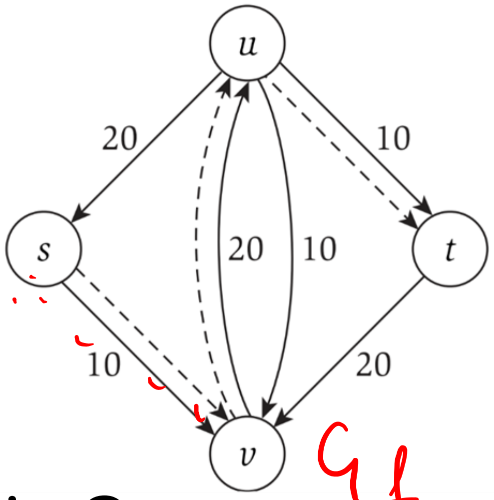
Claim: G_f has most twice as many edges as G.

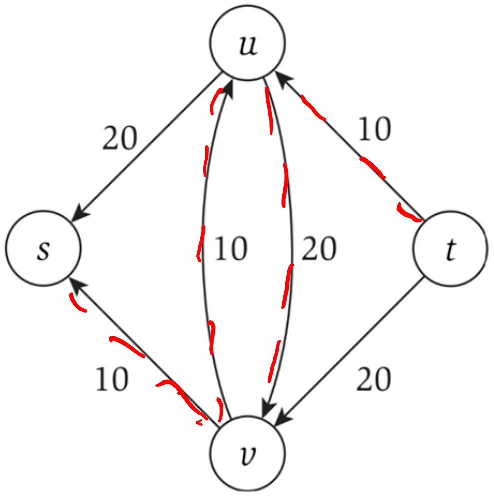
Augmenting Path

- Let P be a simple path in G_f.
- Let b is bottleneck of P in G_f.
- For every edge (u,v) in G_f,
  - If (u,v) is a forward edge then increase f(e) by b in G.
  - Else decrease f(e) by b in G.
- Return f.

(s, v, u, t) → 10







Claim: Returned flow f' is a legal flow in G.

Augmenting Path

<u>Claim</u>: Returned flow f' is a legal flow in G.

<u>Proof</u>: *Capacity Condition*

- The change between f and f' is only due to edge of P.
- If $e = (u,v)$ is forward edge then f' ensures $f'(e) \leq c(e)$

$0 \leq f(e) \leq f'(e) = f(e) + \text{bottleneck}(P, f) = f(e) + c(e) - f(e) = c(e).$

- If $e' = (u,v)$ is backward edge then f' ensures $f'(e) \geq 0$, where $e = (v,u)$

$c(e) \geq f(e) \geq f'(e) = f(e) - \text{bottleneck}(P, f) \geq f(e) - f(e) = 0.$

Augmenting Path

<u>Claim</u>: Returned flow f’ is a legal flow in G.

<u>Proof</u>: *Capacity Condition*

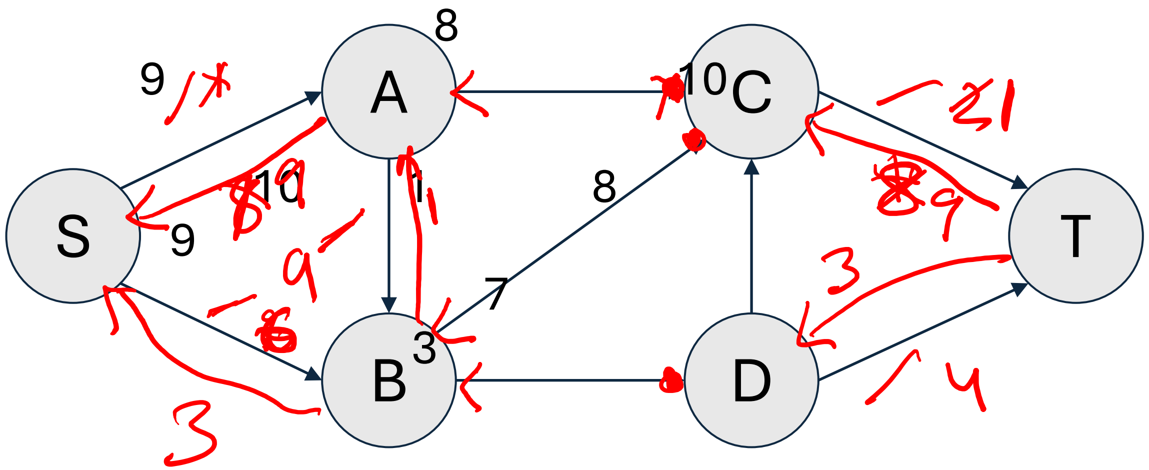
- The change between f and f’ is only due to edge of P.
- If e = (u,v) is forward edge then f’ ensures $f'(e) \leq c(e)$
- If e’ = (u,v) is backward edge then f’ ensures $f'(e) \geq 0$, where e = (v,u)

Conservation condition

- f followed conservation condition in G.
- The flow on the augmenting path ensures conservation condition in G.
- So, f’ follows conservation condition in G.

Maximum FLow (Ford Fulkerson)

```
Max-Flow
  Initially f(e) = 0 for all e in G
  While there is an s-t path in the residual graph G_f
    Let P be a simple s-t path in G_f
    f' = augment(f, P)
    Update f to be f'
    Update the residual graph G_f to be G_f'
  Endwhile
  Return f
```



S, A, C, T – 8
S A B C T – 1
S B D T – 3
12

Ford Fulkerson

Claim: At every intermediate step the flow values f(e) and the residual capacities in G_f are integers.

Does Ford Fulkerson algorithm always terminates?

While applying augmentations we have the following guarantee.

Claim: If f be a flow in G & P be a simple path s-t in G_f, $|f'| = |f| +$ bottleneck(P,f). Since, bottleneck(P,f) > 0, hence $|f'| > |f|$.

Ford Fulkerson

Claim: If f be a flow in G & P be a simple path s-t in G_f, |f'| = |f| + bottleneck(P,f). Since, bottleneck(P,f) > 0, hence |f'| > |f|.

Proof:

- The first edge of P must be an edge out of s in G_f. It doesn't visit s again as P is a simple path.
- So, the first edge must be a forward edge, to which we increase the flow value by bottleneck(P,f).
- Since we do not change flow value of any edge going in s, |f'| > |f|.

Ford Fulkerson

Claim: The Ford-Fulkerson algorithm terminates in at most C iterations, where C is the total capacity from s.

Proof: We know the following statements to be true at each iteration

- $|f'|$ is an integer.
- $|f'| > |f|$
- For every edge $f'(e) \leq c(e)$.

So, the number of iterations of the while loop is bounded by C.

Ford Fulkerson

Running time:

We know that at most C times the algorithm computes augmenting path.

Let T be the running time of augmenting path.

The running time of Ford-Fulkerson is O(TC).

What is the most efficient value of T?

Can you find maximum flow in a flow graph that rational valued capacities?

Can you find maximum flow in a flow graph that rational valued capacities?

Reference

Slides

Algorithms Design by Kleinberg & Tardos - Chp 7.1