COT 6936: Topics in Algorithms

Giri Narasimhan

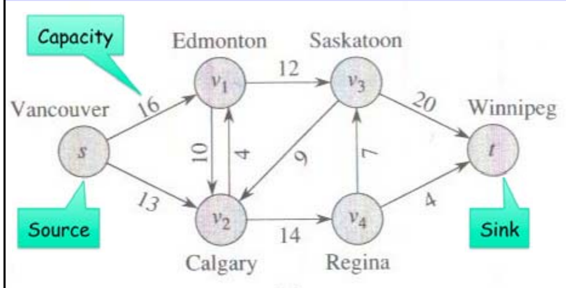
ECS 254A / EC 2443; Phone: x3748

giri@cs.fiu.edu

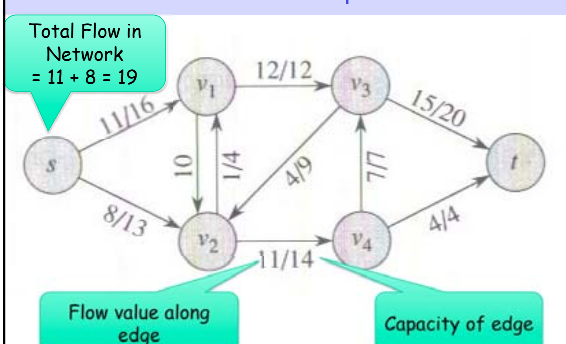
http://www.cs.fiu.edu/~giri/teach/COT6936_S10.html

https://online.cis.fiu.edu/portal/course/view.php?id=427

Network Flow: Example



Network Flow: Example of a flow



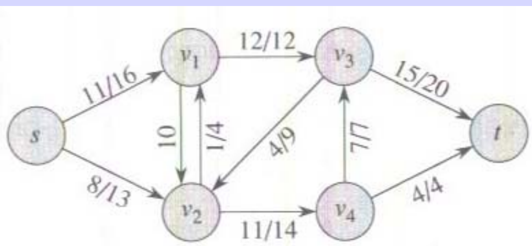
Network Flow

- Directed graph G(V,E) with capacity function on edges given by non-negative function c: E(G) ➔ $\mathcal{R}^+$.
  - Capacity of each edge, e, is given by c(e)
  - Source vertex s
  - Sink vertex t
- Flow function f is a non-negative function of the edges
  - f: E(G) ➔ $\mathcal{R}^+$
  - Capacity constraints: $f(e) \leq c(e)$
  - Flow conservation constraints: For all vertices except source and sink, sum of flow values along edges entering a vertex equals sum of flow values along edges leaving that vertex
- **Flow value**: sum of flow values from source vertex (or sum of flow into sink vertex)

Flow Conservation

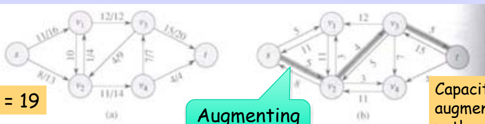
- For any legal flow function:
  - Flow out of source = Flow into sink (Why?)

Network Flow: How to increase flow



Find path with residual capacity and increase flow along path.
- Path s to v_1 to v_3 to t has no residual capacity
  - edge v_1 to v_3 is saturated
- Path s to v_2 to v_3 to t has residual capacity

Residual Flows and Augmenting Paths



Flow = 23

Residual Flow Network: Definition

- Directed Graph G(V,E) with capacity function c and flow function f
- <u>Residual flow network</u> G_f(V,E')
  - For every edge e = (u,v) in E with f(e) < c(e), there are two edges in E': (u,v) and (v,u) with capacities c(e) = f(e) and f(e), respectively
  - For every edge e = (u,v) in E with f(e) = c(e), there is one edge in E': (v,u) with capacity f(e)
  - For every edge e = (u,v) in E with f(e) = 0, there is one edge in E': (u,v) with capacity f(e)

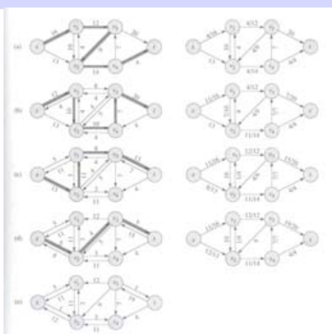
Ford Fulkerson Algorithm

- Initialize flow f to 0.
- While (there exists augmenting path p from s to t) do
  - Augment flow along augmenting path p
- Return flow f as maximum flow from s to t

Ford Fulkerson Algorithm

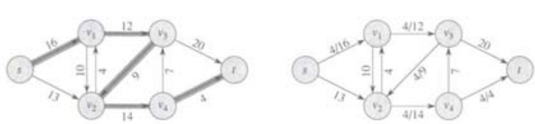
- Initialize flow f to 0.
- While (there exists directed path p from s to t in residual flow network G_f) do
  - Augment flow along augmenting path p
- Return flow f as maximum flow from s to t

Ford-Fulkerson Method: Example

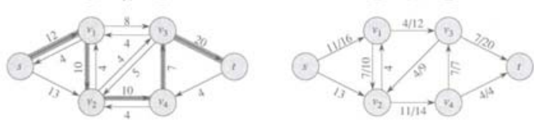


Ford-Fulkerson Method: Example

A



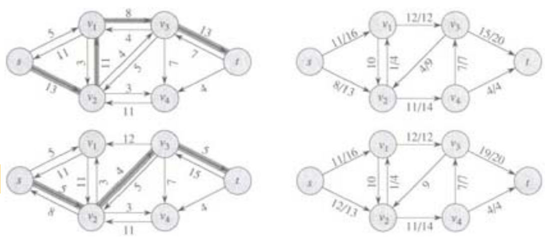
B



Ford-Fulkerson Method: Example

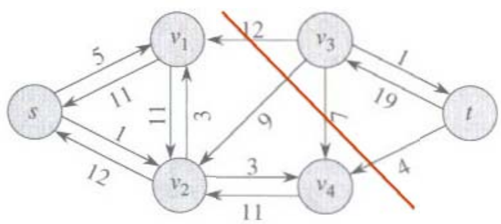
C

D



Ford-Fulkerson Method: Example

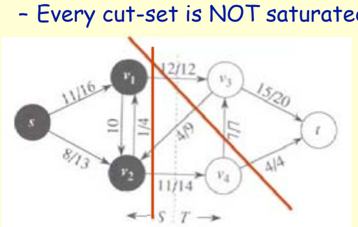
C



- Max-Flow has been reached. Why?
- Cut with zero capacity has been found. Which Cut?
  - $(\{s,v_1,v_2,v_4\},\{v_3,t\})$

Correctness of Ford-Fulkerson Method

- Augmentation is possible if
  - Every cut-set is NOT saturated



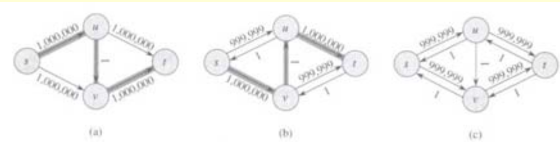
Cut (S,T):
- Capacity = 26
- Flow across cut = 19

Cut (S',T'):
- Capacity = 23
- Flow across cut = 19

- Theorem: Min-Cut = Max-Flow

Time Complexity

- It can be arbitrarily large.



- Solution: When finding augmenting path, find the shortest path
- In that case, time complexity = O(mn)

2/4/10 COT 6936 16