COT 6936: Topics in Algorithms

Giri Narasimhan
ECS 254A / EC 2443; Phone: x3748
giri@cs.fiu.edu

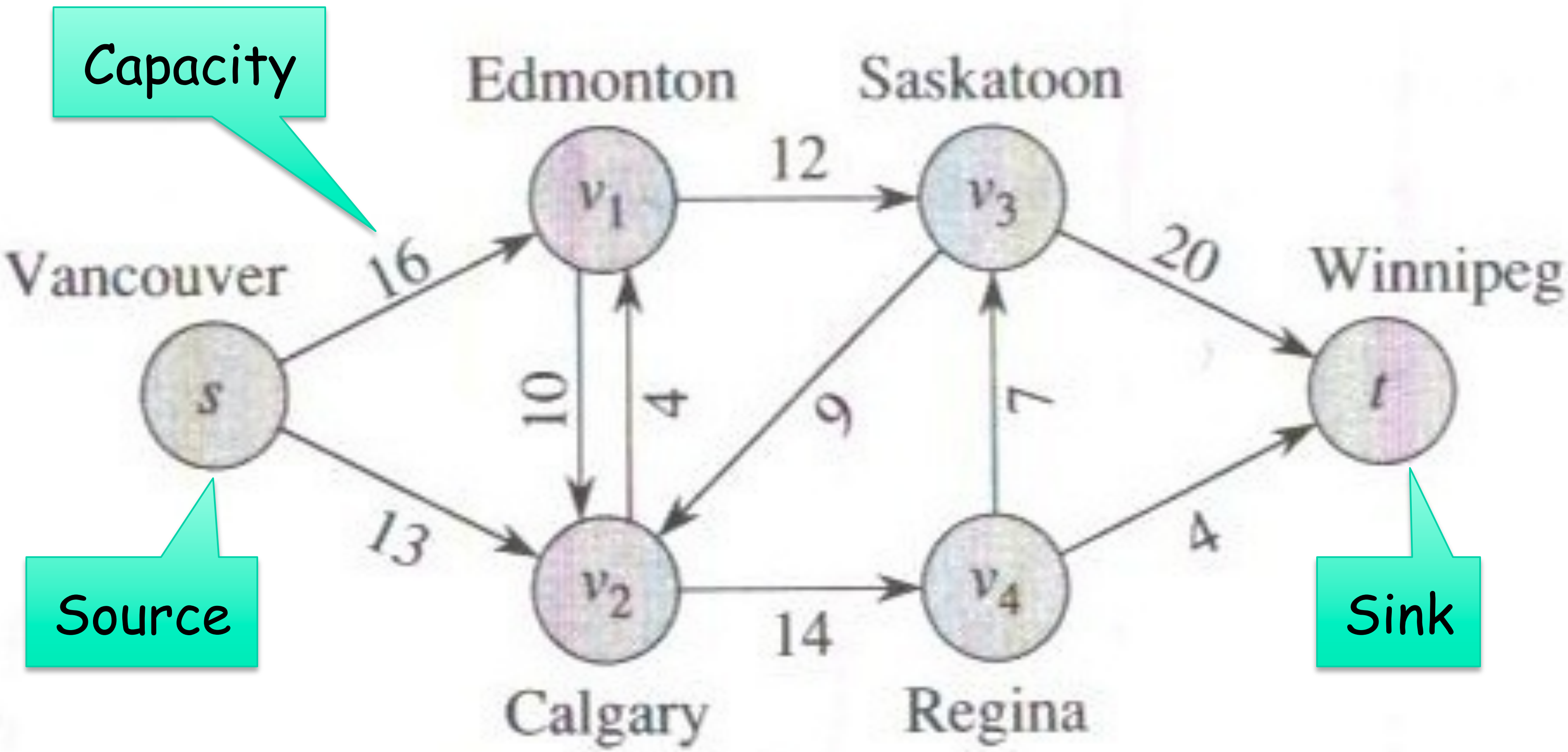
https://moodle.cis.fiu.edu/v2.1/course/view.php?id=**612**

Types of networks & Types of queries

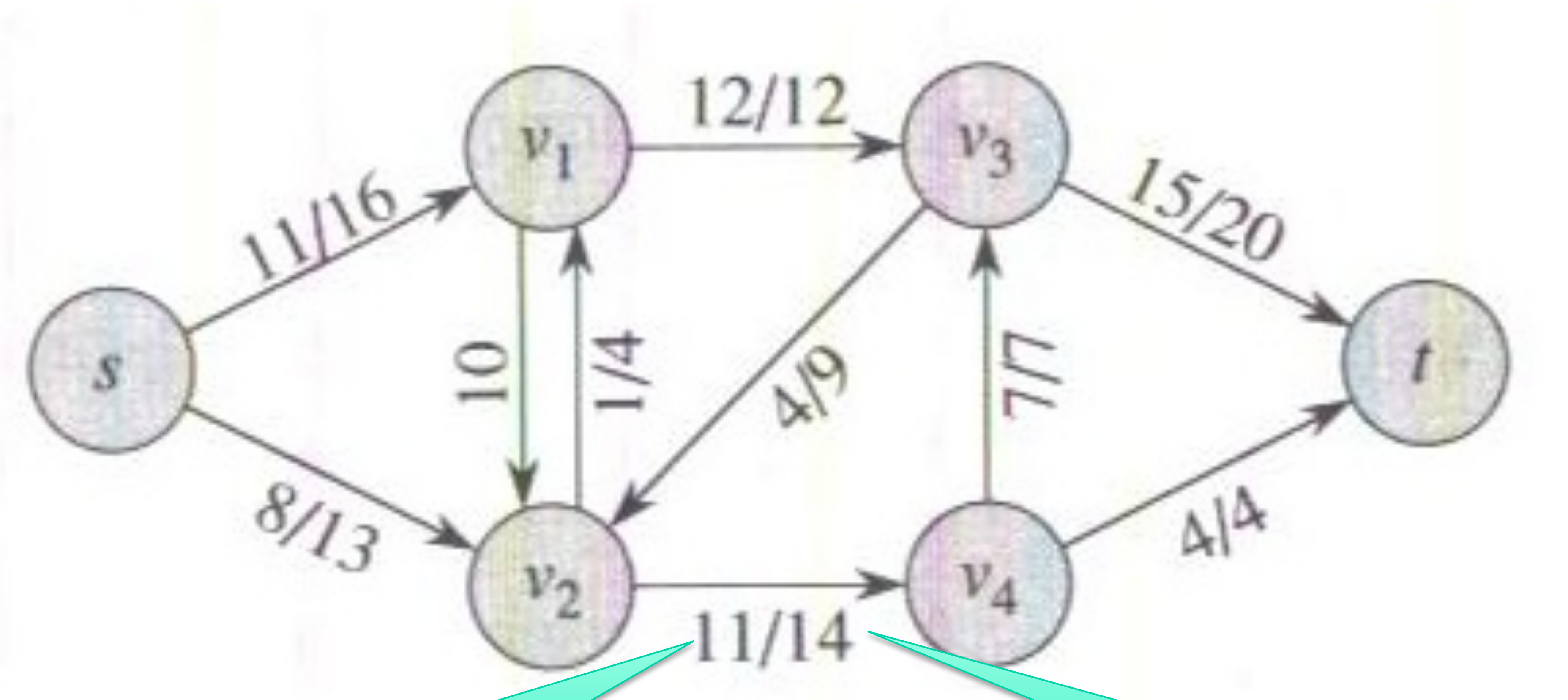
- Road, highway, rail
- Electrical power, water, oil, gas, sewer
- Internet, phone, wireless, sensor
- …

- (1950s) How quickly can Soviet Union get supplies through its rail network to Europe?
- Which links to destroy to reduce flow to under a threshold?

Network Flow: Example



Network Flow: Example of a flow



Flow value along edge

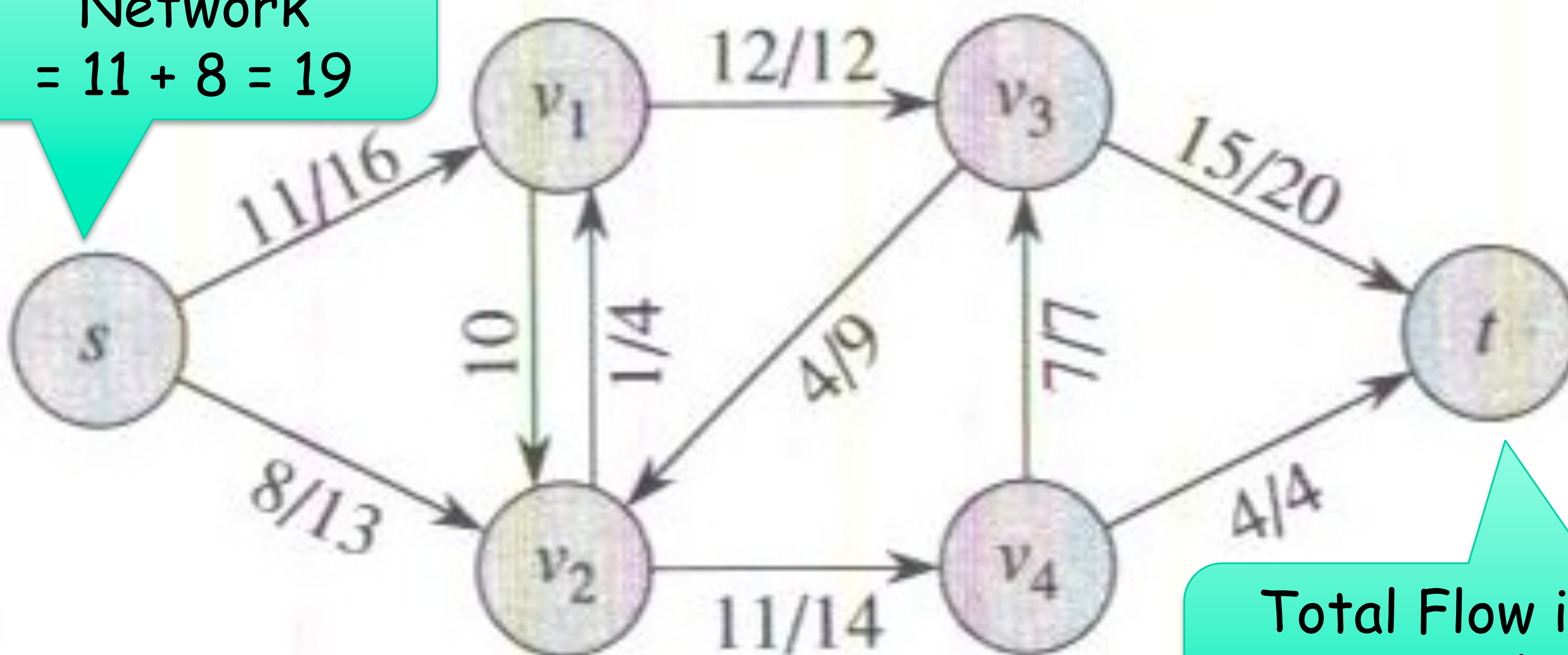
Capacity of edge

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) ≤ 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)

Network Flow: Example of a flow

Total Flow in Network
= 11 + 8 = 19

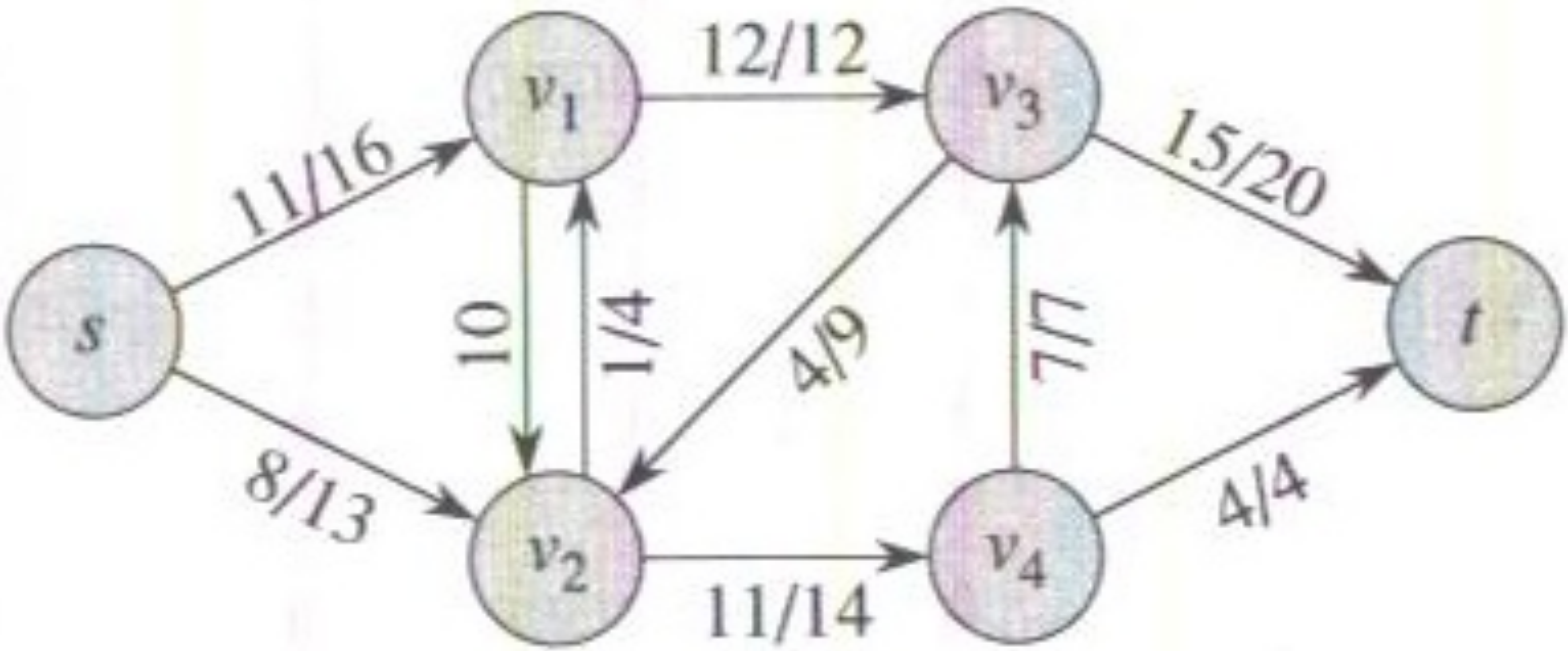


Total Flow in Network
= 15 + 4 = 19

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.

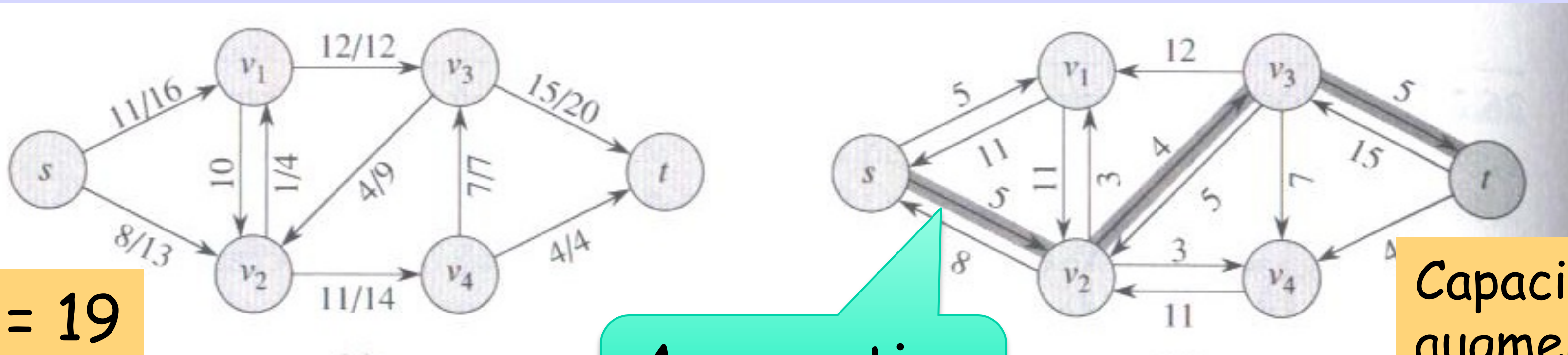
- Path s to v_1 to v_3 to t has no residual capacity
- Path s to v_2 to v_4 to t has no residual capacity
- Path s to v_2 to v_1 to v_3 to t has no residual capacity

Max Network Flow 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

Incorrect Algorithm

Residual Flows and Augmenting Paths



(a) (b)

Flow = 19

Augmenting Path

Capacity of augmenting path = 4

Flow = 23

Find path with residual capacity and increase flow along path.

- Path s to v_2 to v_3 to t had residual capacity

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 $f(e)$

Max Network Flow Algorithm

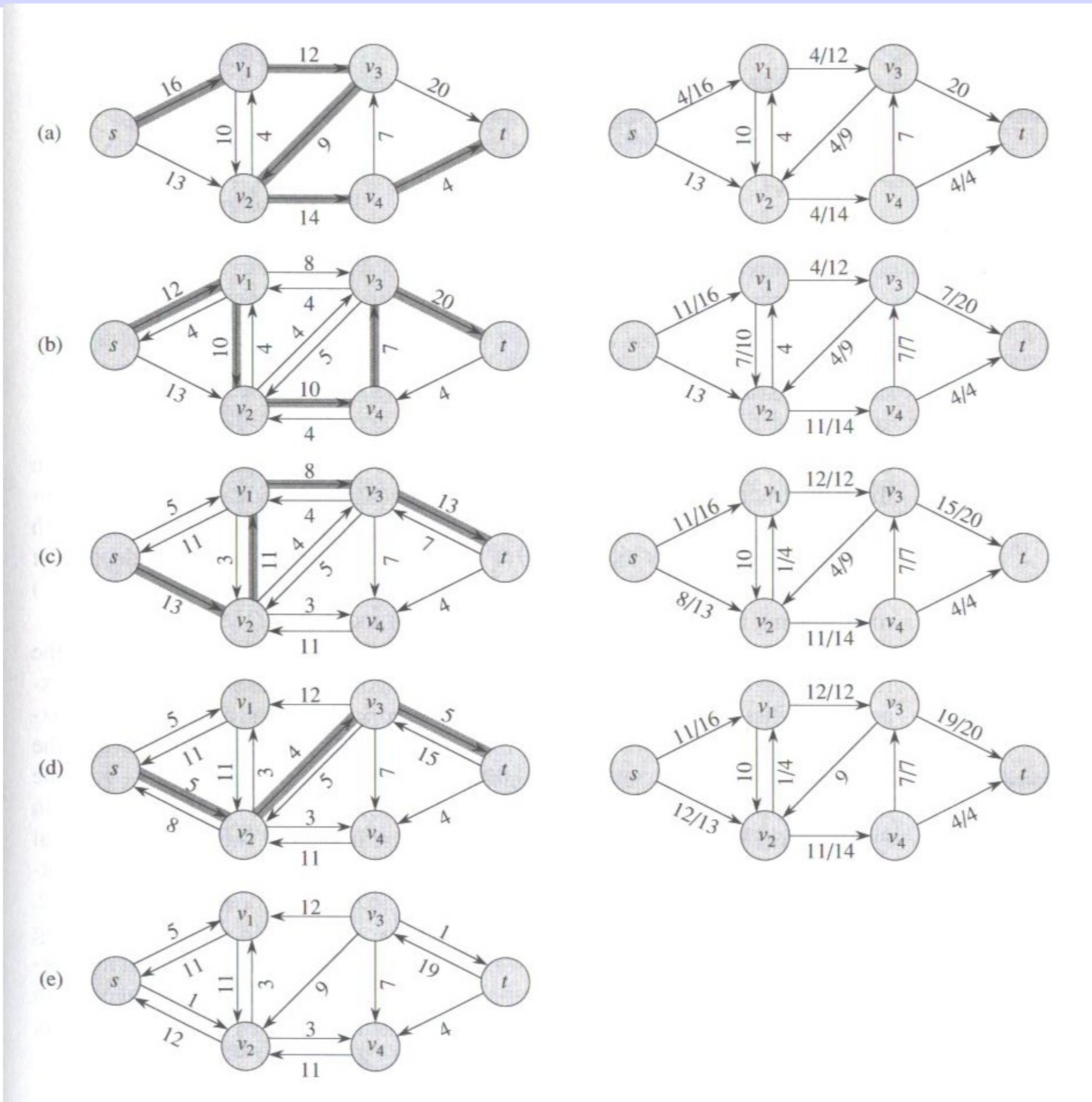
- Initialize flow f to 0.
- While (there exists augmenting path p from s to t) do
  - Augment flow along augmenting path p
- Return flow f as maximum flow from s to t

Incorrect Algorithm

Ford Fulkerson Algorithm

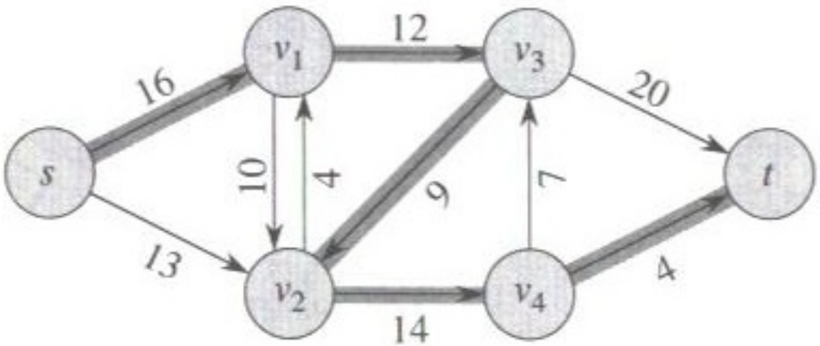
- Initialize flow f to 0.
- While (there exists directed path p from s to t **in residual flow network G_f**) do
  - Augment flow along augmenting path p
- Return flow f as maximum flow from s to t

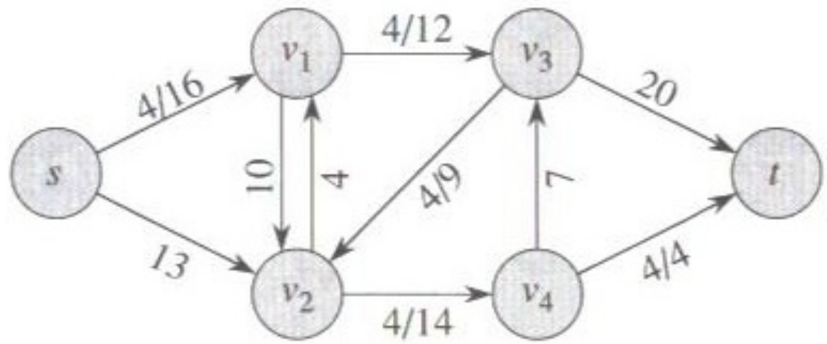
Ford-Fulkerson Method: Example



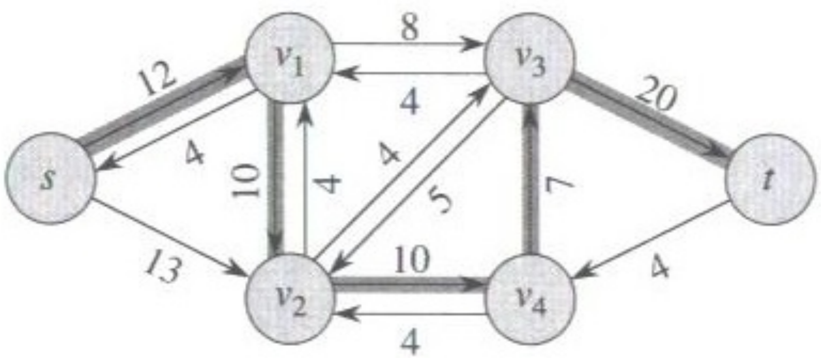
Ford-Fulkerson Method: Example

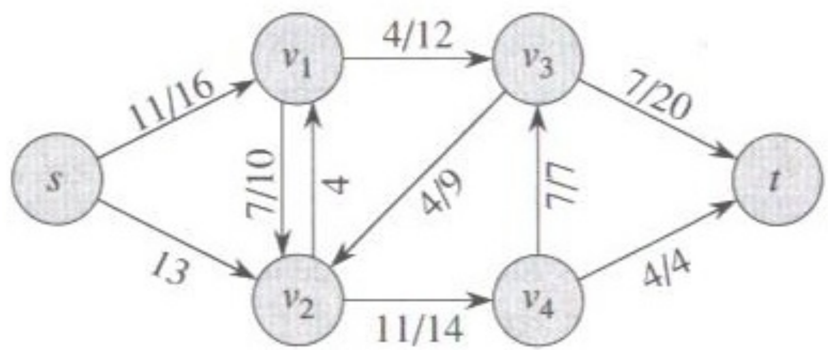
A





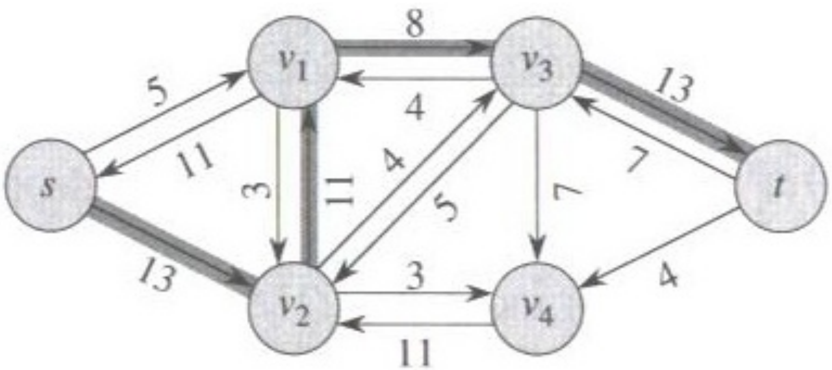
B

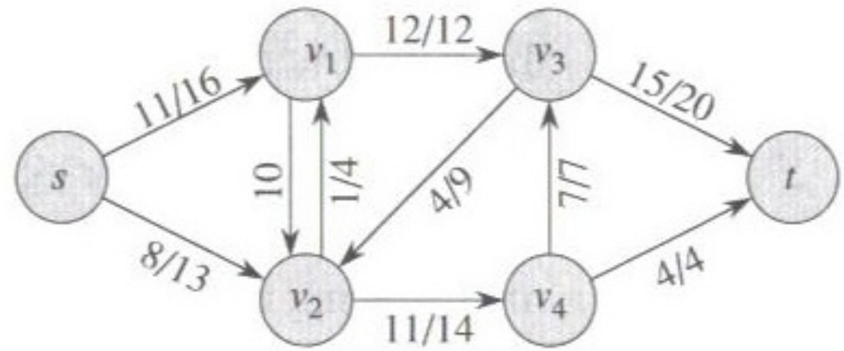




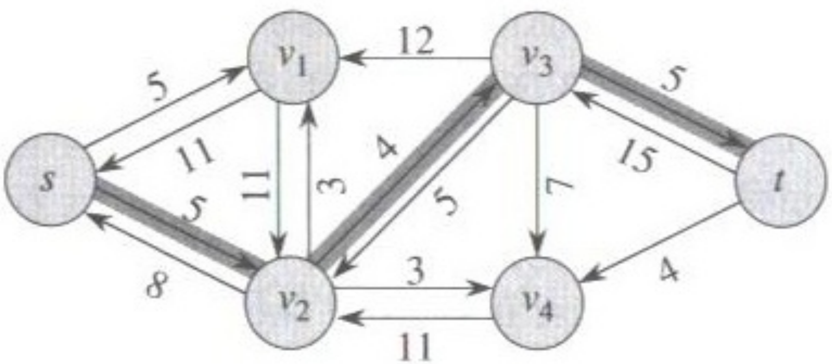
Ford-Fulkerson Method: Example

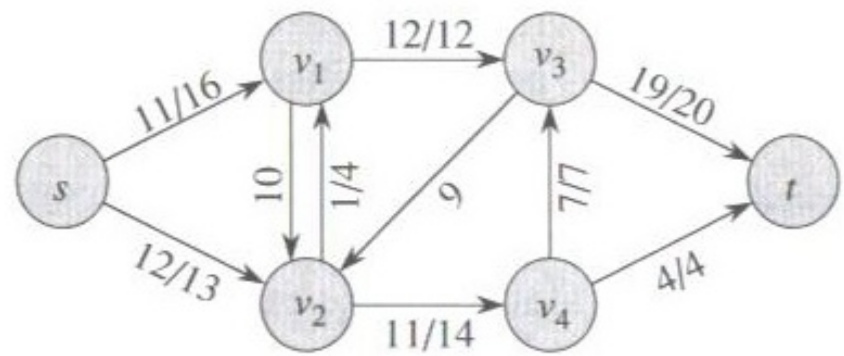
C





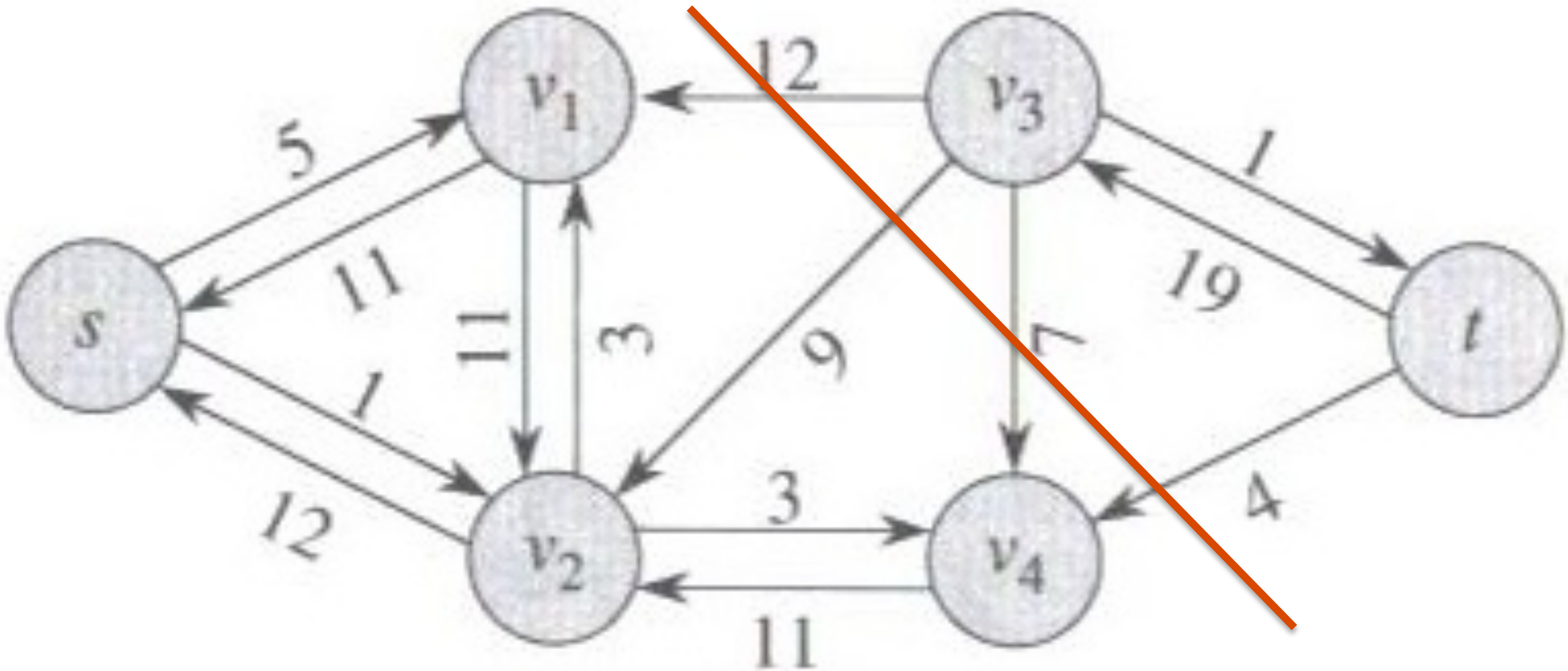
D





Ford-Fulkerson Method: Example

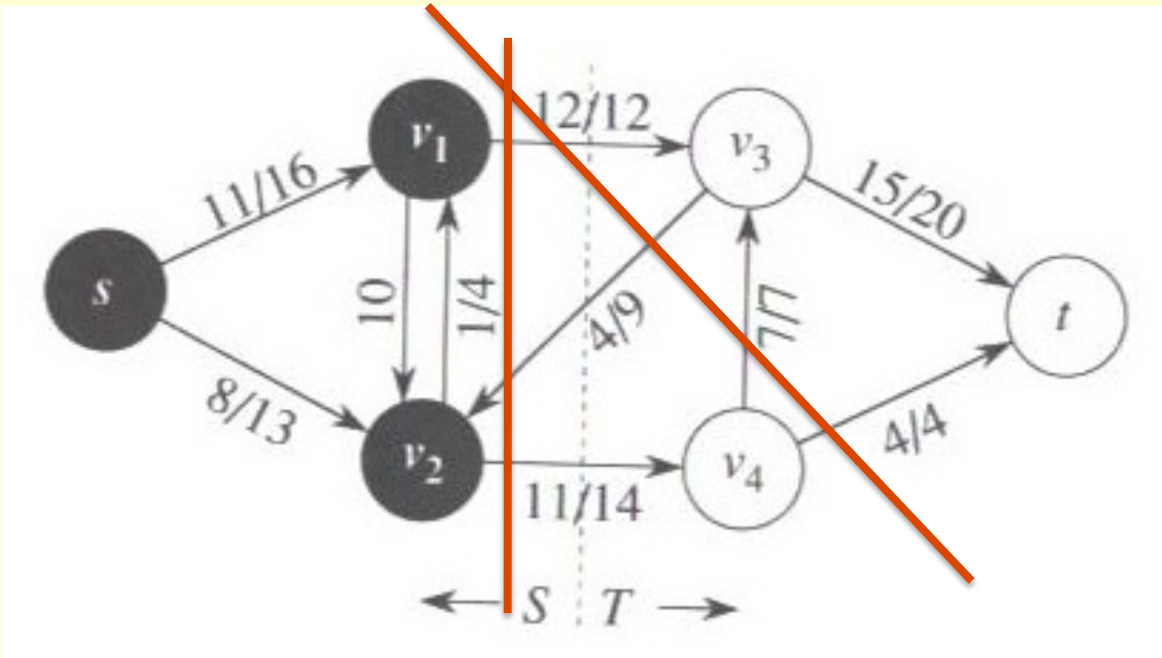
C



- 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



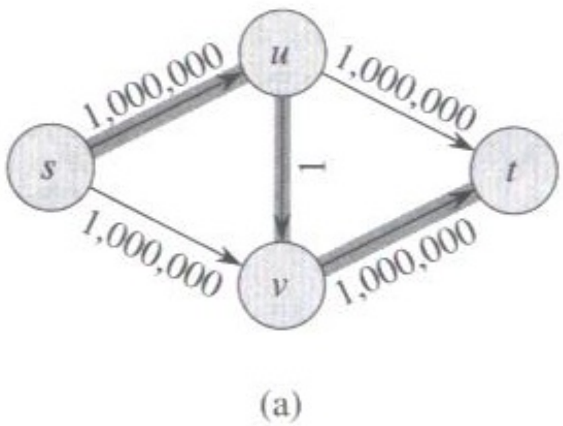
Cut (S,T):
- Capacity = 26
- Flow across cut = 19

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

- Theorem: Min-Cut = Max-Flow

Time Complexity

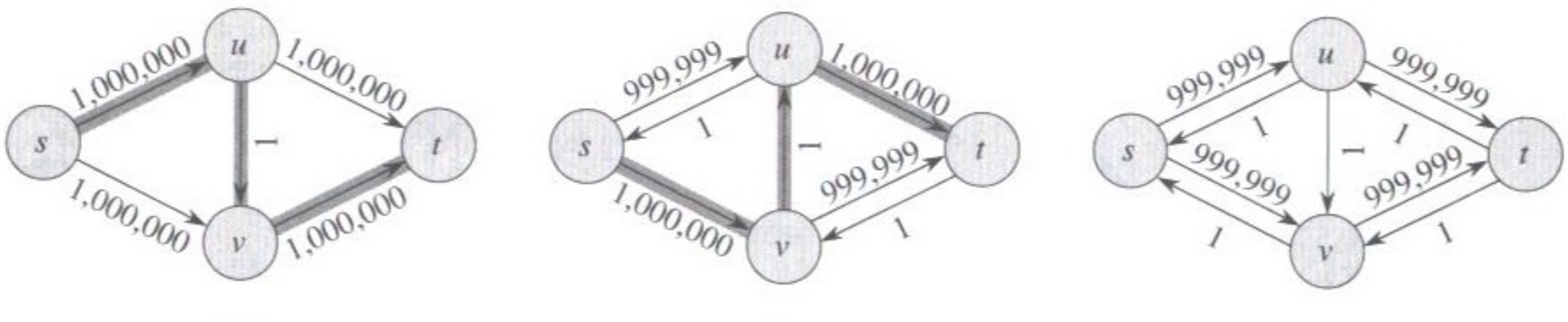
- It can be arbitrarily large.



- Solution: When finding augmenting path, find the shortest path
- In that case, # of augmentations = O(mn)

Time Complexity Analysis

- It can be arbitrarily large.



(a) (b) (c)

- Solution: When finding augmenting path, find the shortest path
- In that case, # of augmentations = O(mn)

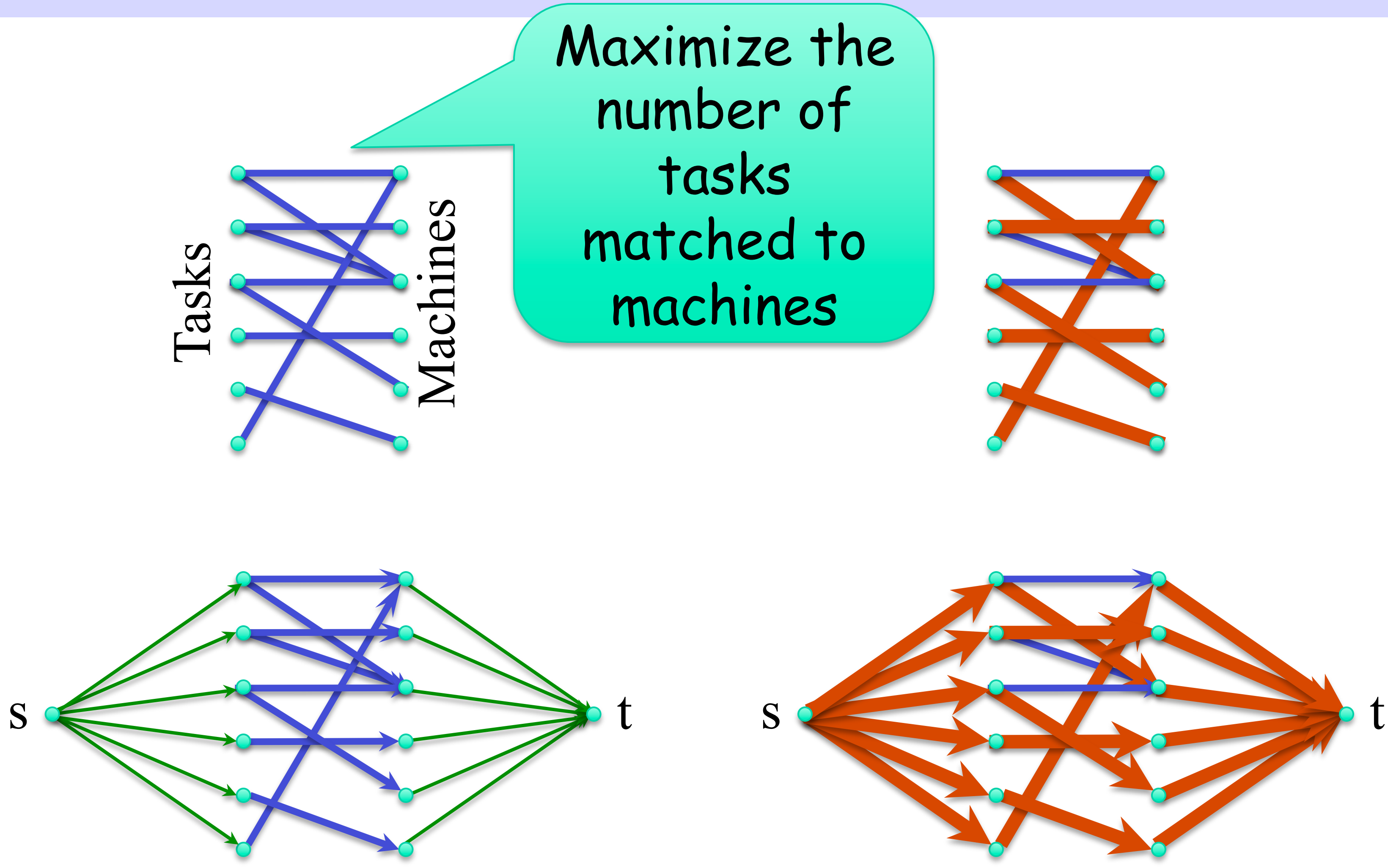
More efficient Network Flow algorithms

- Push-relabel algorithms [Goldberg, '87]
  - Local algorithm, works on one vertex at a time
  - Avoids maintaining flow conservation rule
    - Excess flow in each node
    - Height function
  - $O(mn^2)$ time complexity
  - Can be improved to $O(n^3)$

Generalizations

- Multiple sources and sinks.
  - Can be reduced to single source and sink

Bipartite Matching



Network Flow

- Input: Directed graph G(V,E) with capacity function on edges given by non-negative function $c: E(G) \rightarrow \mathcal{R}^+$.
  - Capacity of each edge, e, is given by c(e)
  - Source vertex s
  - Sink vertex t
- Question: Find a flow function f with the maximum flow value

Min-Cost Network Flow

- Input: 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)
  - Flow cost of each edge, e, is given by a(e)
    - Implies that cost of flow in e is a(e)•f(e)
    - Total cost of flow = Σ a(e)•f(e)
  - Source vertex s
  - Sink vertex t
  - Flow required = F
- Question: Find **min-cost** flow function f with flow value = F

Minimum Path Cover in DAGs

- Path Cover: set of vertex disjoint paths that cover all vertices
- Minimum Path cover in directed acyclic graphs can be reduced to Network Flow