COT 6405
ANALYSIS OF ALGORITHMS

Maximum Flow

Computer & Electrical Engineering and Computer Science Department
Florida Atlantic University

Outline

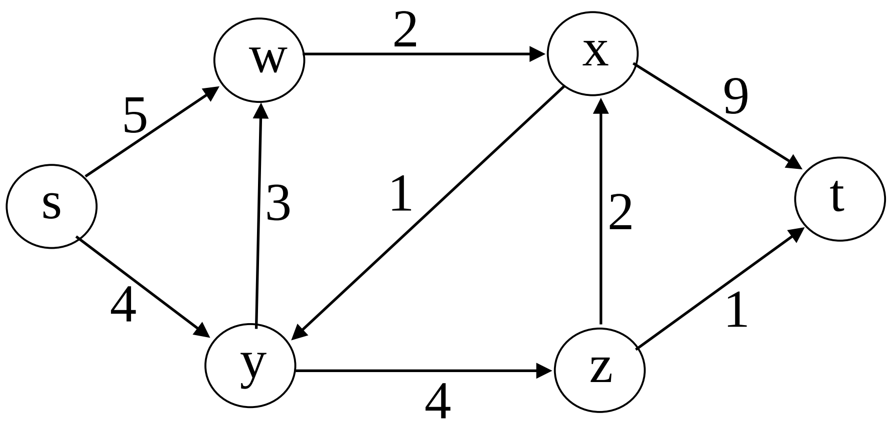
- Flow networks
- Ford-Fulkerson method
- Edmonds-Karp algorithm

Reference: *Introduction to Algorithms*, 3rd edition, by T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, The MIT Press, 2009 (chapter 26)

Flow Networks, motivation

- Use of graphs to model *transportation networks*
  - edges carry some sort of traffic
  - nodes act as “switches”, passing traffic between edges
- Examples
  - highway system: edges are highways, nodes are interchanges
  - computer network: edges are links that carry packets, nodes are switches
  - fluid network: edges are pipes that carry liquid, and nodes are junctures where pipes are plugged together
- Such network models have several ingredients:
  - capacities of the edges: how much they can carry
  - source nodes: nodes that generate traffic
  - sink (or destination) nodes: nodes that “absorb” traffic
  - traffic transmitted across edges

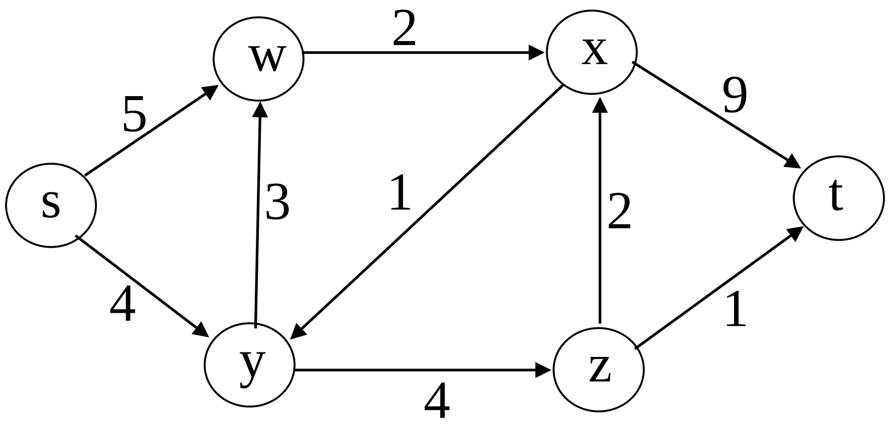
Flow Network, definition



Flow network:

- directed graph $G = (V, E)$
- each edge (u,v) has a capacity $c(u,v) \geq 0$
- if $(u,v) \in E$, then $(v, u) \notin E$
- if $(u, v) \notin E$, then we define $c(u, v) = 0$
- no self-loops
- **source vertex** s, **sink vertex** t
- for each vertex $v \in V$, there is a path $s \leadsto v \leadsto t$

Flow network



The graph of a flow network is connected

$$|E| \geq |V| - 1$$

Flow definition

Given:

- G is a flow network, with capacity function c
- s is the source, t is the sink

A **flow** is a function f : V $\times$ V $\rightarrow$ R, which satisfies the properties:

- **Capacity constraint**: for all u, v $\in$ V

  $$0 \leq f(u,v) \leq c(u,v)$$

- **Flow conservation**: for all u $\in$ V - {s, t}

  $$\sum_{v \in V} f(v,u) = \sum_{v \in V} f(u,v)$$

  "flow in equals flow out"

when (u,v) $\notin$ E, f(u, v) = 0

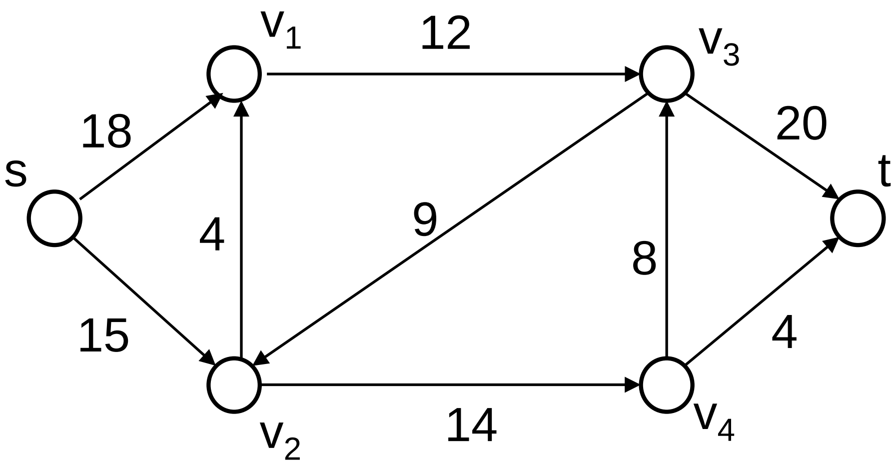
Maximum-flow problem

- f(u,v) – the flow from u to v
- The **value of a flow** is defined as:

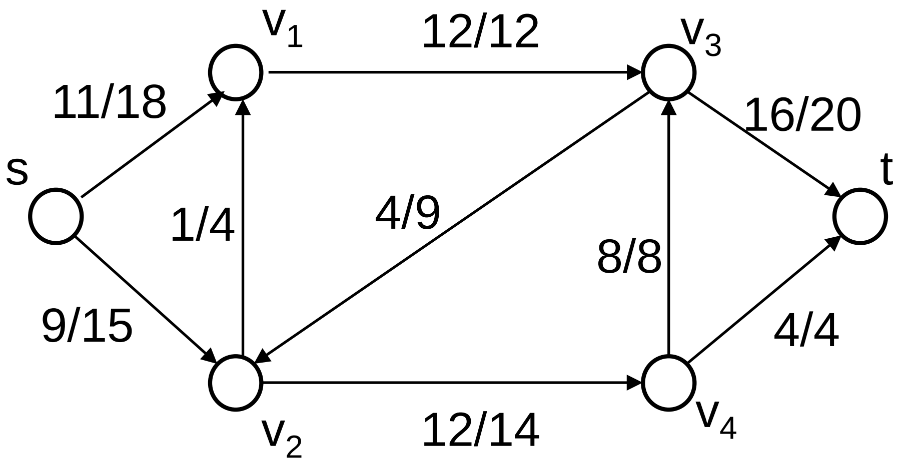
$$|f| = \sum_{v \in V} f(s,v) - \sum_{v \in V} f(v,s)$$

Maximum-flow problem: given a flow network G with source s and sink t, find a flow of maximum value.

Example



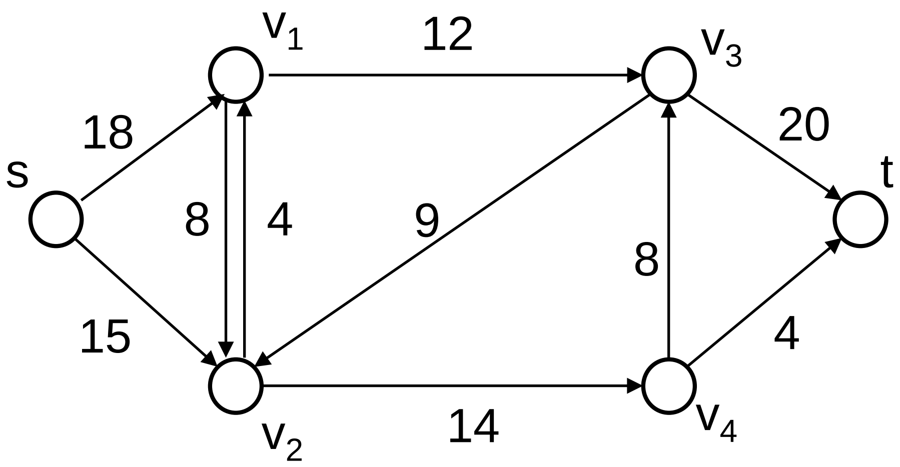
Flow network

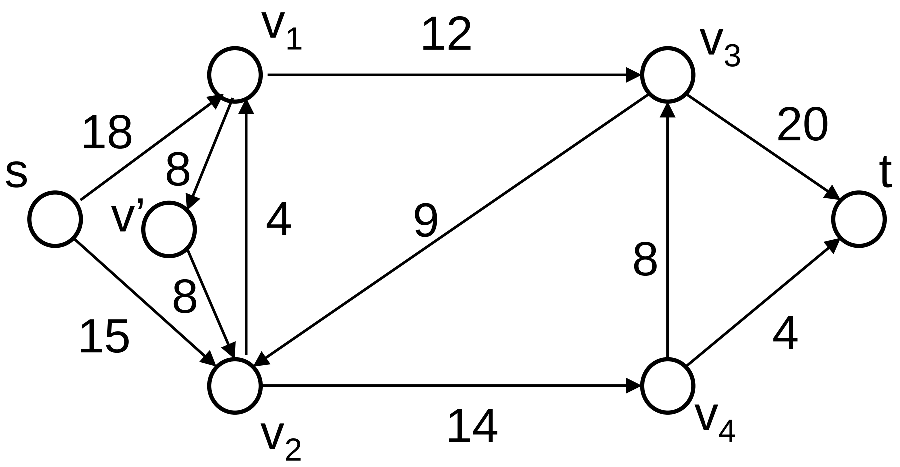


A flow with value $|f| = 20$

- Lucky Puck Company ships hockey pucks from the factory, source city s to the warehouse, city t
- capacity $c(u, v)$ – limit on the number of crates per day that can go from city u to city v
- objective: maximize the flow

Modeling problems with antiparallel edges

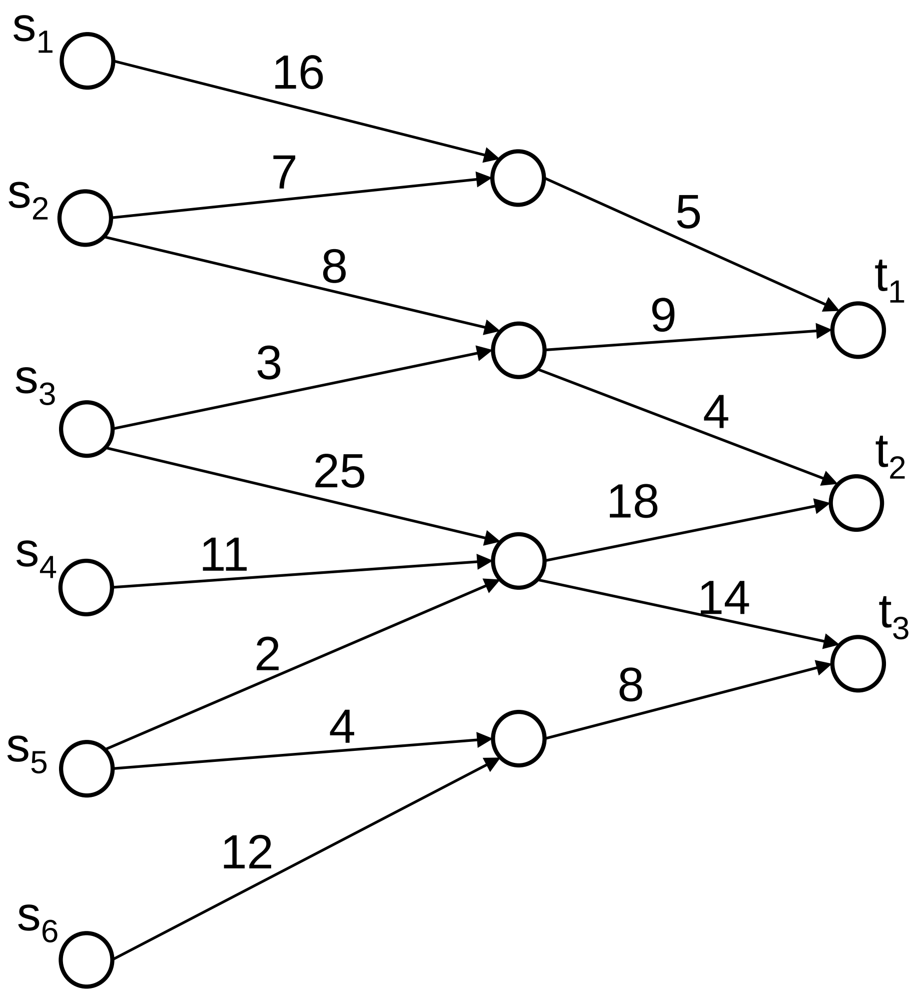




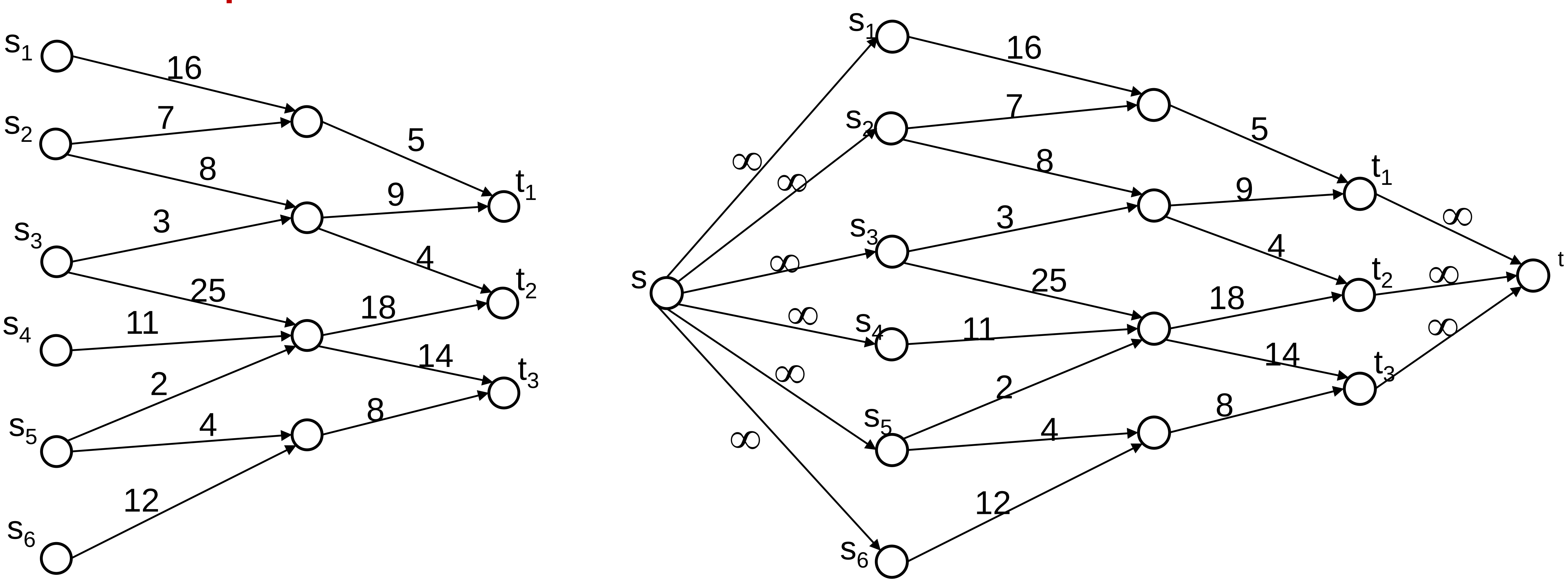
- Previous example: Lucky Puck leases additional space for 8 crates between cities v_1 and v_2
  - if $(v_1, v_2) \in E \Rightarrow (v_2, v_1) \notin E$
  - (v_1, v_2) and (v_2, v_1) are called ***antiparallel***
- Transform to an equivalent flow network w/o antiparallel edges
  - add a new vertex v'
  - replace (v_1, v_2) by two edges (v_1, v') and (v', v_2)

Networks with multiple sources and sinks

- A maximum-flow problem may have several sources and sinks
- The Lucky Puck Company may have a set of m factories $\{s_1, s_2, \ldots, s_m\}$ and a set of n warehouses $\{t_1, t_2, \ldots, t_n\}$



Converting a multiple-source, multiple-sink problem into an *equivalent* problem with 1 source and 1 sink



- add a **supersource** s and edges (s, s_i) with capacity $c(s,s_i) = \infty$ for all i=1,2,…, m
- add a **supersink** t and edges (t_i, t) with capacity $c(t_i,t) = \infty$ for all i=1,2,…, n

The Ford-Fulkerson method

- has several implementations with different RT
- Idea:
  - start with a flow of 0
  - at each iteration increase the flow by finding an “augmenting path” in the “residual network”
  - repeat until there are no more augmenting paths in the residual network

<u>FORD-FULKERSON-METHOD(G, s, t)</u>
initialize flow *f* to 0
while there exists an augmenting path *p* in the residual network G_f
    augment flow *f* along path *p*
return *f*

Three important concepts

- Residual network
- Augmenting paths
- Cuts

Residual networks

- Suppose we have a flow network $G = (V, E)$ with source s and sink t
- Let f be a flow in G, and let $u,v \in V$
- The *residual capacity* $c_f(u,v)$ is defined as:

$$c_f(u, v) = \begin{cases} c(u,v) - f(u,v) & \text{if } (u, v) \in E \\ f(v,u) & \text{if } (v,u) \in E \\ 0 & \text{otherwise} \end{cases}$$

- Since $(u,v) \in E$ implies $(v,u) \notin E$, exactly one of the above cases applies.

Residual Networks

Given a flow network $G = (V,E)$ and a flow f, the ***residual network of G induced by f*** is $G_f = (V,E_f)$, where

$E_f = \{(u,v) \in V \times V: c_f(u,v) > 0\}$

- each edge of the residual network, called residual edge, can admit a flow > 0
- $|E_f| \leq 2\,|E|$
- residual network is similar to a flow network with capacities given by c_f
  - not a flow network, since it may contain both an edge (u,v) and its reversal (v,u)
- a flow in G_f satisfies the flow properties with respect to capacities c_f
  - a flow in G_f can be used as a roadmap for adding flow in G

Augmenting a flow in the flow network G

- let f be a flow in the flow network G
- let f ' be a flow in the corresponding residual network G_f
- then $f \uparrow f'$ is the **augmentation** of the flow f by f '

$$f \uparrow f' : V \times V \to R$$

$$f \uparrow f' = \begin{cases} f(u,v) + f'(u,v) - f'(v,u) & \text{if } (u, v) \in E \\ 0 & \text{otherwise} \end{cases}$$

- f '(v,u) – pushing flow on the reverse edge in G_f is also known as *cancellation*

Flow augmentation

Lemma

Let G be a flow network

Let f be a flow in G

Let G_f be the residual network of G induced by f

Let f' be a flow in G_f

Then $f \uparrow f'$ is a flow in G with value $|f \uparrow f'| = |f| + |f'|$

Augmenting paths

- *augmenting path* – is a simple path from s to t in the residual network G_f
- Let p be an augmenting path in G_f
- Then the *residual capacity* of p, denoted $c_f(p)$, is defined as:

$$c_f(p) = \min\{c_f(u,v) : (u,v) \text{ is on } p\}$$

- it represents the max amount by which we can increase the flow on each edge in the path p

Augmenting paths

Lemma:

Let $G = (V,E)$ be a flow network, let f be a flow in G, and let p be an augmenting path in G_f. Define a function $f_p : V \times V \to R$ by:

$$f_p(u,v) = \begin{cases} c_f(p) & \text{if } (u, v) \text{ is on } p \\ 0 & \text{otherwise} \end{cases}$$

Then f_p is a flow in G_f with value $|f_p| = c_f(p) > 0$.

Augmenting paths

Corollary: Let

- G be a flow network
- f be a flow in G
- p be an augmenting path in G_f
- f_p defined as in the previous lemma
- suppose that we augment f by f_p

Then $f \uparrow f_p$ is a flow in G with value:

$$|f \uparrow f_p| = |f| + |f_p|$$

Cuts of flow networks

- Ford-Fulkerson method repeatedly augments the flow along augmenting paths until it has found a max flow
- How do we know when the flow is maximum (when does the algorithm terminates)?
  - Answer: max-flow min-cut theorem
- Next, we introduce the cut of a flow network:

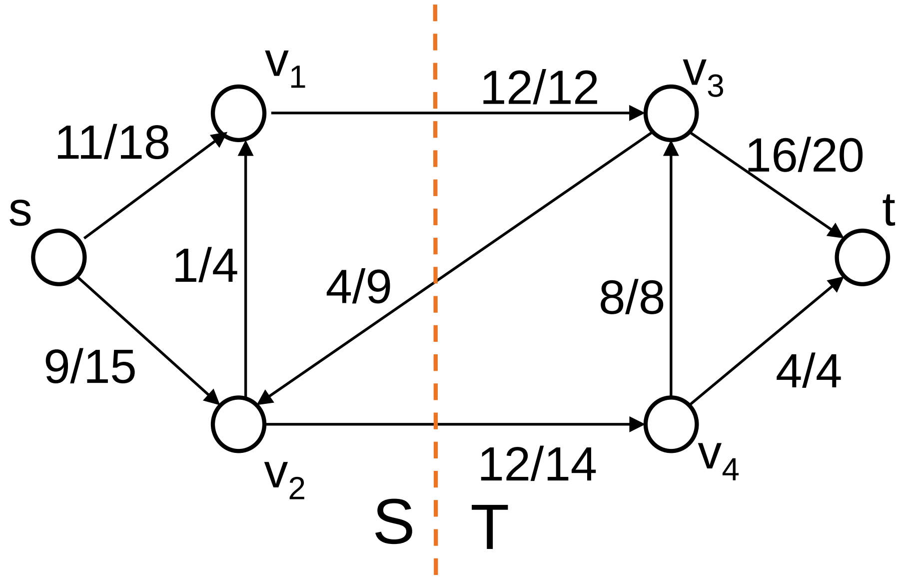
A ***cut*** (S,T) of flow network G = (V,E) is a partition of V into S and T = V – S, such that s $\in$ S and t $\in$ T.

More on cuts …

- If f is a flow, then the ***net flow*** f(S,T) across the cut (S,T) is:

$$f(S,T) = \sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{u \in S} \sum_{v \in T} f(v,u)$$

- The ***capacity*** of the cut (S,T) is: $c(S,T) = \sum_{u \in S} \sum_{v \in T} c(u,v)$



$S = \{s, v_1, v_2\}$
$T = \{t, v_3, v_4\}$
$f(S,T) = ?$
$c(S,T) = ?$

More on cuts …

Lemma: Let f be a flow in a flow network G, and let (S,T) be a cut of G. Then the *net flow* across (S,T) is f(S,T) = | f |.

Corollary: The value of any flow f in a flow network G is bounded from above by the capacity of any cut of G.

Proof:

$$|f| = f(S,T) = \sum_{u\in S}\sum_{v\in T} f(u,v) - \sum_{u\in S}\sum_{v\in T} f(v,u)$$

$$\leq \sum_{u\in S}\sum_{v\in T} f(u,v) \leq \sum_{u\in S}\sum_{v\in T} c(u,v) = c(S,T)$$

Max-flow min-cut theorem

If f is a flow in a flow network $G = (V,E)$ with source s and sink t, then the following conditions are equivalent:

1. f is a maximum flow in G
2. the residual network G_f contains no augmenting paths
3. $|f| = c(S,T)$ for some cut (S,T) of G

Proof:

(1) $\Rightarrow$ (2) if G_f has an augmenting path p, then $f \uparrow f_p$ is a flow in G with value $> |f|$, thus f is not maximum.

Max-flow min-cut theorem, cont.

(2) $\Rightarrow$ (3) If G_f has no augmenting path:

Let $S = \{v \in V$: there is a path from s to v in $G_f\}$

$T = V - S$

then (S,T) is a cut

Let $u \in S$ and $v \in T$.

- if $(u,v) \in E$, then $f(u,v) = c(u,v)$ since otherwise $(u,v) \in E_f$, then $v \in S$
- if $(v,u) \in E$, then $f(v,u) = 0$. Otherwise $c_f(u,v) = f(v,u)$, then $(u,v) \in E_f$, then $v \in S$
- If neither (u,v) or $(v,u) \in E$, then $f(u,v) = f(v,u) = 0$

$$f(S,T) = \sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{v \in T} \sum_{u \in S} f(v,u)$$

$$= \sum_{u \in S} \sum_{v \in T} c(u,v) - \sum_{v \in T} \sum_{u \in S} 0 = c(S,T)$$

Therefore, $|f| = c(S, T)$ for some cut (S, T) of G.

Max-flow min-cut theorem, cont.

(3) $\Rightarrow$ (1)

- from the corollary, $|f| \leq c(S,T)$ for every cut (S,T)
- since $|f| = c(S,T)$, it implies that f is a max flow.

Ford-Fulkerson algorithm

<u>FORD-FULKERSON(G, s, t)</u>

for each edge (u, v) $\in$ G.E

    (u, v).f = 0

while there exists a path *p* from *s* to *t* in the residual network G_f

    $c_f(p) = \min\{c_f(u,v): (u,v) \text{ is in } p\}$

    for each edge (u, v) in p

        if $(u, v) \in E$

            $(u, v).f = (u, v).f + c_f(p)$

        else $(v, u).f = (v, u).f - c_f(p)$

Analysis of Ford-Fulkerson

- RT depends on how we find the augmenting paths *p*
  - If we choose poorly, it may not even terminate when capacities are irrational numbers
- In practice, usually capacities are integer numbers
- If they are rational numbers, apply a scaling transformation to make them integral
- For integral capacities, the while loop executes at most |f*| times, where f* is the max flow
  - Flow increases by at least one unit each iteration

Analysis of Ford-Fulkerson

- How to find an augmenting path p?

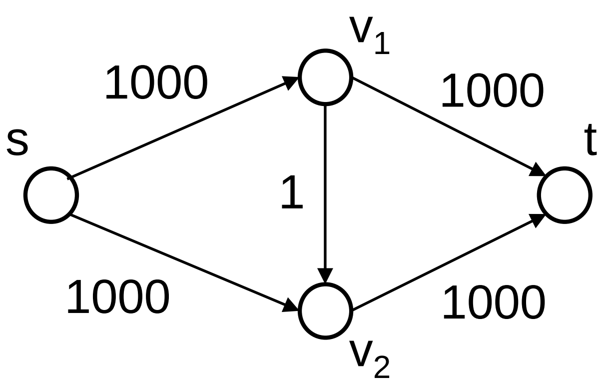
- Let G' = (V, E') – be the graph where we store the residual network G_f
  - $|E'| \leq 2|E|$
- Use Breadth-First-Search (BFS) or Depth-First-Search (DFS) to find an augmenting path between s and t
  - RT for BFS/DFS is O(V + E') = O(E)

Then the total RT of the Ford-Fulkerson is

RT = O(E |f*|)

Analysis of Ford-Fulkerson

- When | f* | is small and capacities are integral, the RT is good
- When | f* | is large, max-flow may converge slow



$|f^*| = 2{,}000$

- If the algorithm alternates in selecting the augmenting paths $<s, v_1, v_2, t>$ and $<s, v_2, v_1, t>$ then it performs 2,000 augmentations each increasing the flow by 1 unit

Integrality Theorem

If the capacity function c takes only integral values, then the max-flow f produced by Ford-Fulkerson is an integer. Moreover, for all vertices u and v, the value of $f(u,v)$ is an integer.

Edmonds-Karp algorithm

- Finds the augmenting path using BFS
- Then the augmenting path is a shortest-path (in terms of number of edges) from s to t in the residual network
- Edmonds-Karp has $\boldsymbol{RT = O(VE^2)}$