Flow Networks

Ching-Chen Huang

Hsi-Yue Hsiao

UTA

CONTENTS

- Flow networks on directed graphs
- Ford-fulkerson Algorithms

  -Residual networks

Flow Networks on directed graphs

- Flow network: A directed graph to model materials that flow through certain paths/conduits
- Each edge represents one conduit, and has a *capacity*, which is an upper bound on the *flow rate* = units/time
- Can think of edges as pipes of different sizes. But flows don't have to be of liquids. A flow can be of trucks per day, which ship hockey pucks between cities

We distinguish two vertices in flow network, source and sink

- The source produces the material at a steady rate
- The sink consumes the material at a steady rate

Objective

- How much of material can be passed through a network of conduits

Or

- Compute max rate that we can send material from a source to a sink

Flow Networks

$G = (V, E)$ directed.

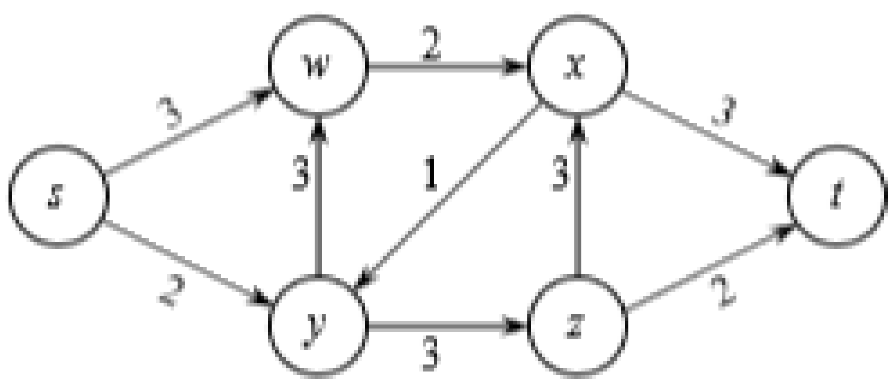
Each edge (u, v) has a ***capacity*** $c(u, v) \geq 0$.

If $(u, v) \notin E$, then $c(u, v) = 0$.

If $(u, v) \in E$, then reverse edge $(v, u) \notin E$. (Can work around this restriction.)

Source vertex s, ***sink*** vertex t, assume $s \leadsto v \leadsto t$ for all $v \in V$, so that each vertex lies on a path from source to sink.

- Example (Edges are labeled with capacities)



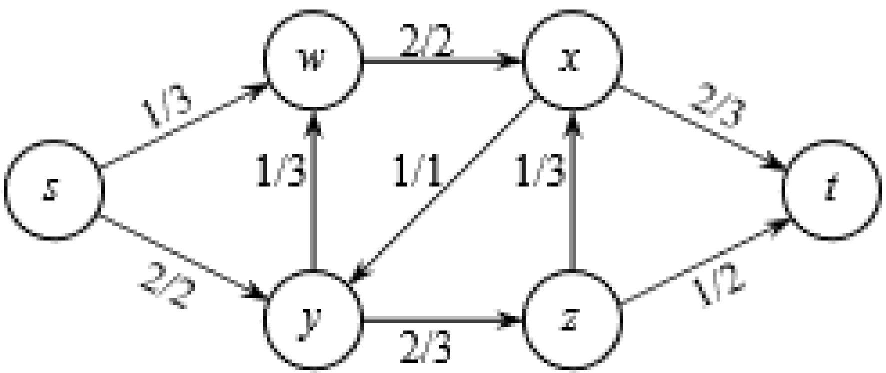
Flow

A function $f : V \times V \to \mathbb{R}$ satisfying

- ***Capacity constraint:*** For all $u, v \in V, 0 \le f(u,v) \le c(u,v)$,
- ***Flow conservation:*** For all $u \in V - \{s,t\}$, $\underbrace{\sum_{v \in V} f(v,u)}_{\text{flow into } u} = \underbrace{\sum_{v \in V} f(u,v)}_{\text{flow out of } u}$.

Equivalently, $\sum_{v \in V} f(u,v) - \sum_{v \in V} f(v,u) = 0$.

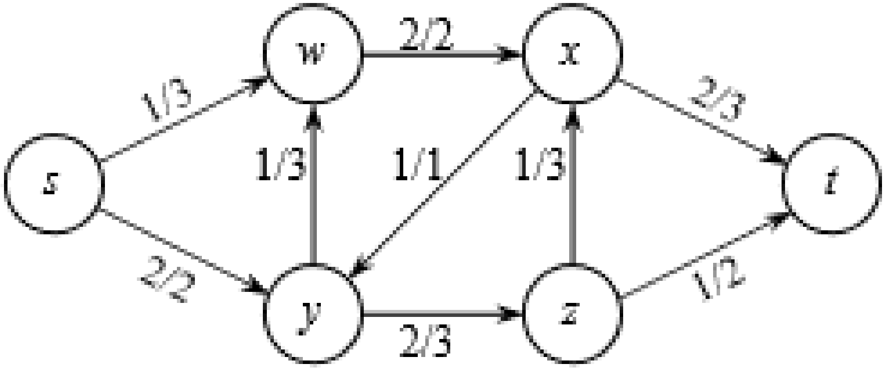
- Edges here are labeled as flow/capacity this time
- Note that all flows are ≤ capacities
- Verify flow conservation by adding up flows at a couple of vertices



Value of Flow

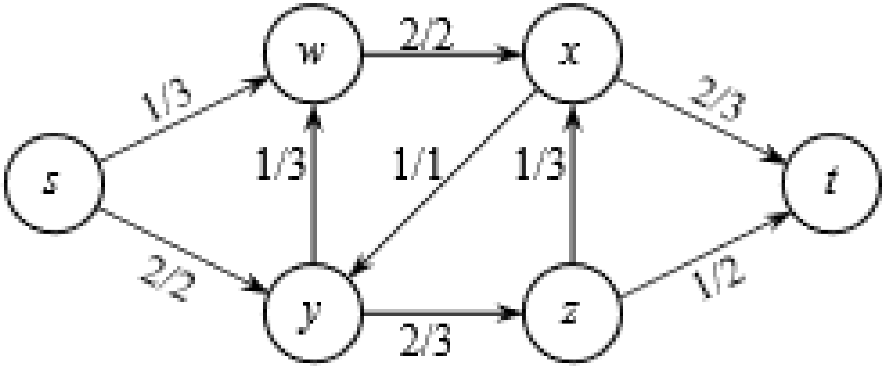
$$\begin{aligned} \textit{Value of flow } f &= |f| \\ &= \sum_{v \in V} f(s,v) - \sum_{v \in V} f(v,s) \\ &= \text{flow out of source} - \text{flow into source} . \end{aligned}$$

- In the example below value of flow is 3



Maximum Flow Problem

Given G, s, t, and c, find a flow whose value is maximum

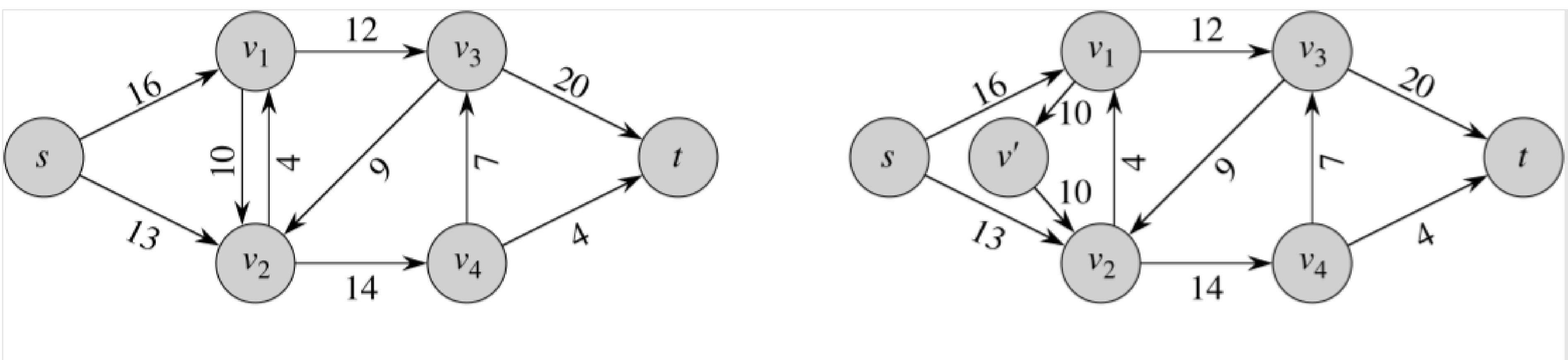


Anti-parallel Edges

Definition of flow network does not allow both (u, v) and (v, u) to be edges. These edges would be ***antiparallel***.

What if we really need antiparallel edges?

- Choose one of them, say (u, v).
- Create a new vertex v'.
- Replace (u, v) by two new edges (u, v') and (v', v), with $c(u, v') = c(v', v) = c(u, v)$.
- Get an equivalent flow network with no antiparallel edges.



(a) (b)

Cuts

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 $t \in T$.

- Similar to cut used in minimum spanning trees, except that here the graph is directed, and we require $s \in S$ and $t \in T$.

For flow f, the ***net flow*** across cut (S, T) is

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

Capacity of cut (S, T) is

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

A ***minimum cut*** of G is a cut whose capacity is minimum over all cuts of G.

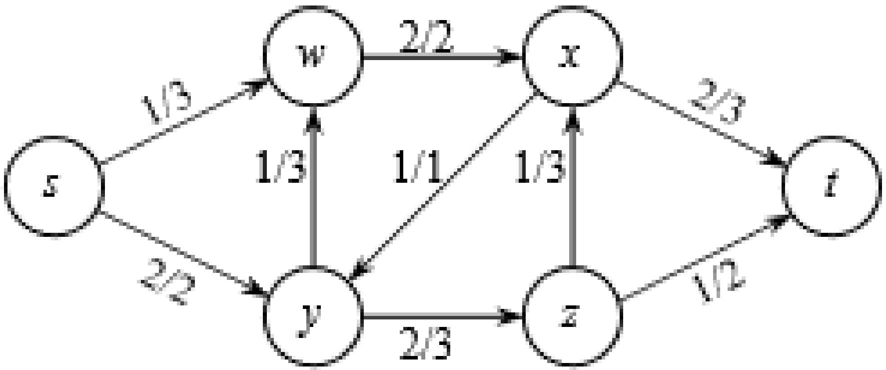
Asymmetry Between Net Flow and Capacity of a Cut

For capacity, count only capacities of edges going from S to T. Ignore edges going in the reverse direction. For net flow, count flow on all edges across the cut: flow on edges going from S to T minus flow on edges going from T to S

Consider the cut, $S = \{s, w, y\}, T = \{x, z, t\}$

$$\begin{aligned} f(S,T) &= \underbrace{f(w,x) + f(y,z)}_{\text{from } S \text{ to } T} - \underbrace{f(x,y)}_{\text{from } T \text{ to } S} \\ &= 2 + 2 - 1 \\ &= 3\,. \end{aligned}$$

$$\begin{aligned} c(S,T) &= \underbrace{c(w,x) + c(y,z)}_{\text{from } S \text{ to } T} \\ &= 2 + 3 \\ &= 5\,. \end{aligned}$$

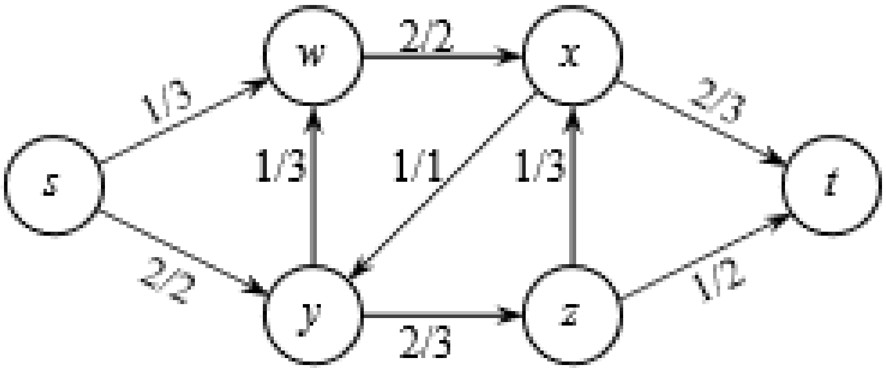


Asymmetry Between Net Flow and Capacity of a Cut

Now consider the cut, $S = \{s, w, x, y\}, T = \{z, t\}$

$$\begin{aligned} f(S,T) &= \underbrace{f(x,t) + f(y,z)}_{\text{from } S \text{ to } T} - \underbrace{f(z,x)}_{\text{from } T \text{ to } S} \\ &= 2 + 2 - 1 \\ &= 3 . \end{aligned}$$

$$\begin{aligned} c(S,T) &= \underbrace{c(x,t) + c(y,z)}_{\text{from } S \text{ to } T} \\ &= 3 + 3 \\ &= 6 . \end{aligned}$$



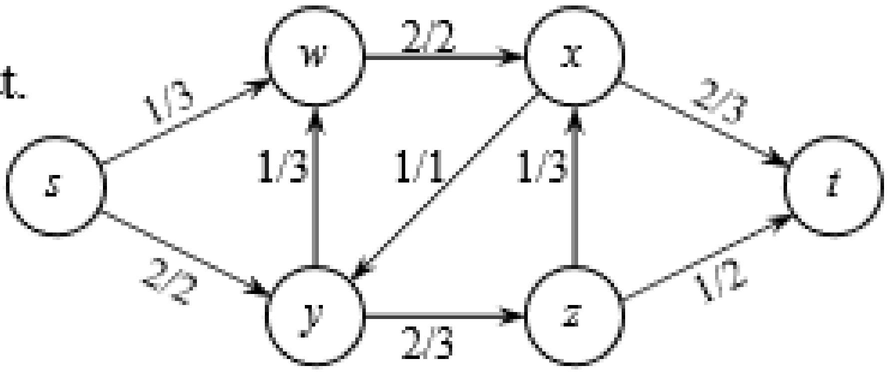
Same flow as previous cut, higher capacity.

Lemma

For any cut (S, T), $f(S, T) = |f|$.
(Net flow across the cut equals value of the flow.)

- Proof is available in the book (too long)
- Intuitively, no matter where you cut the pipes in a network, you'll see the same flow volume coming out of the openings
- Consider the examples of the last two slides

Corollary
The value of any flow $\leq$ capacity of any cut.



Proof of the 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 Let (S, T) be any cut of G and f be any flow. By Lemma above and the capacity constraint,

$$\begin{aligned} |f| = f(S,T) &= \sum_{u \in S} \sum_{v \in T} f(u,v) - \sum_{u \in S} \sum_{v \in T} f(v,u) \\ &\leq \sum_{u \in S} \sum_{v \in T} f(u,v) \\ &\leq \sum_{u \in S} \sum_{v \in T} c(u,v) \\ &= c(S,T) \end{aligned}$$

Thus

$$|f| = f(S,T) \leq c(S,T)$$

Residual Capacity

Given a flow f in network $G = (V, E)$.

Consider a pair of vertices $u, v \in V$.

How much additional flow can we push directly from u to v?
That's the ***residual capacity***,

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

- Example:

  $c(u,v) = 16, f(u,v) = 5 \Rightarrow c_f(u,v) = 11$

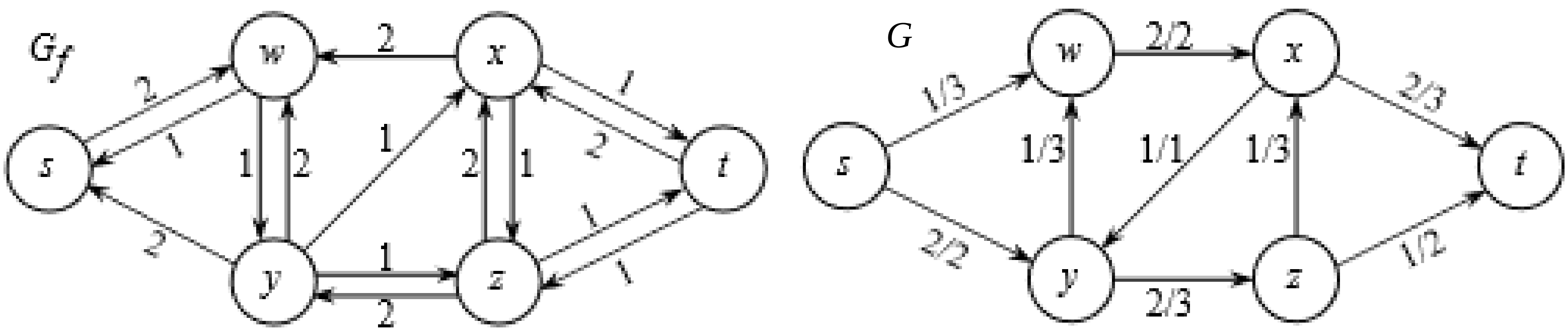
Residual Network

The ***residual network*** 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 can admit a positive flow.

For our example:



Every edge $(u, v) \in E_f$ corresponds to an edge $(u, v) \in E$ or $(v, u) \in E$ (or both).

Therefore, $|E_f| \le 2\,|E|$.

We wish to allow an algorithm to return up to f units to flow from v to u thus making $c_f(v, u) = f(v, u)$

Augmentation and Cancelation

Residual network is similar to a flow network, except that it may contain antiparallel edges ((u, v) and (v, u)). Can define a flow in a residual network that satisfies the definition of a flow, but with respect to capacities c_f in G_f.

Given flows f in G and f' in G_f, define $(f \uparrow f')$, the ***augmentation*** of f by f', as a function $V \times V \to \mathbb{R}$:

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

for all $u, v \in V$.

Intuition: Increase the flow on (u, v) by $f'(u, v)$ but decrease it by $f'(v, u)$ because pushing flow on the reverse edge in the residual network decreases the flow in the original network. Also known as ***cancellation***.

Cancellation is crucial for maximum-flow algorithms

Lemma

Given a flow network G, a flow f in G, and the residual network G_f, let f' be a flow in G_f. Then $f \uparrow f'$ is a flow in G with value $|f \uparrow f'| = |f| + |f'|$.

Proof in the book, please review

Augmenting Path

A simple path $s \leadsto t$ in G_f.

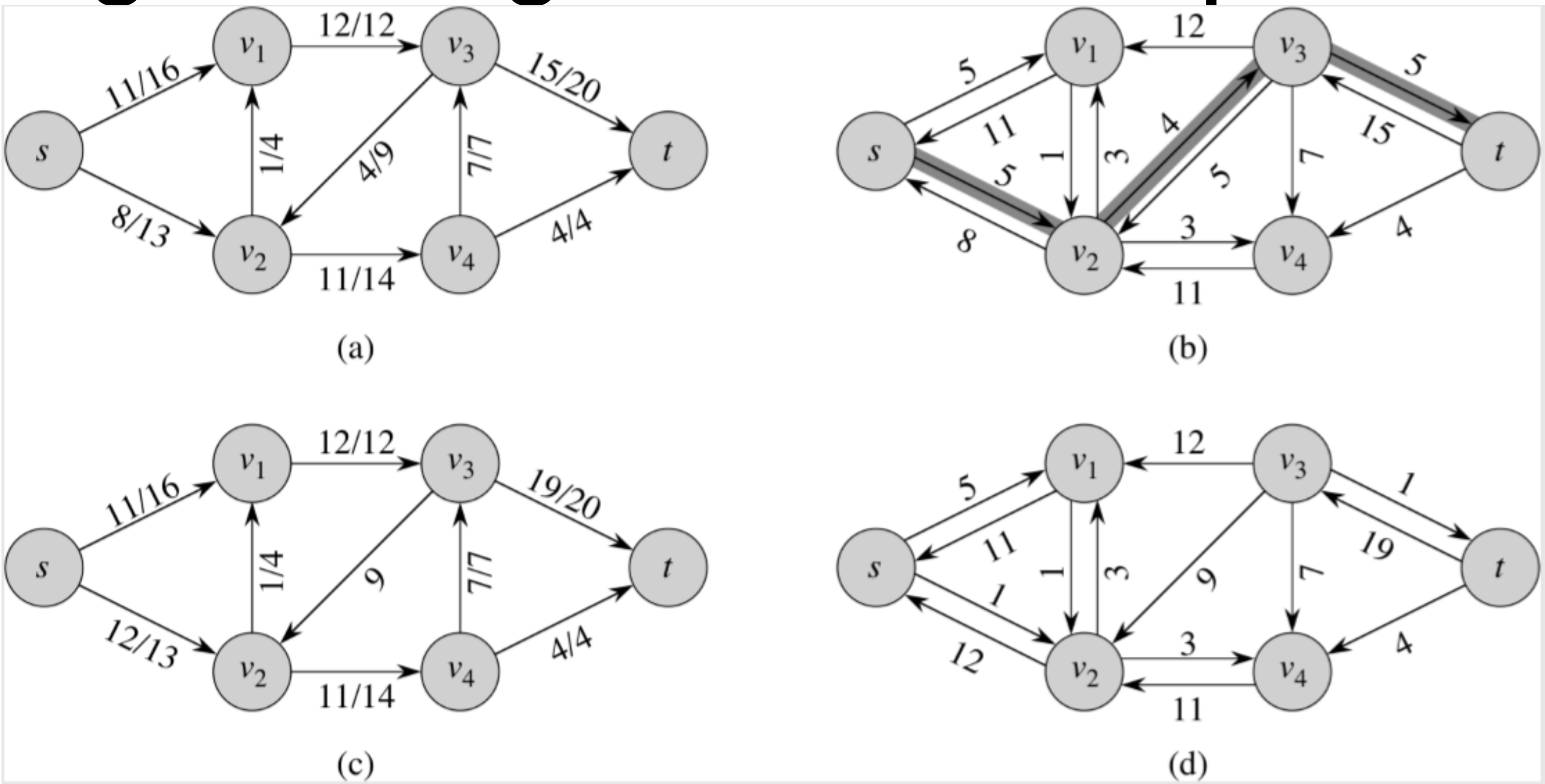
- Admits more flow along each edge.
- Like a sequence of pipes through which we can squirt more flow from s to t.

How much more flow can we push from s to t along augmenting path p?

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

Example, next slide

Augmenting Path, Example



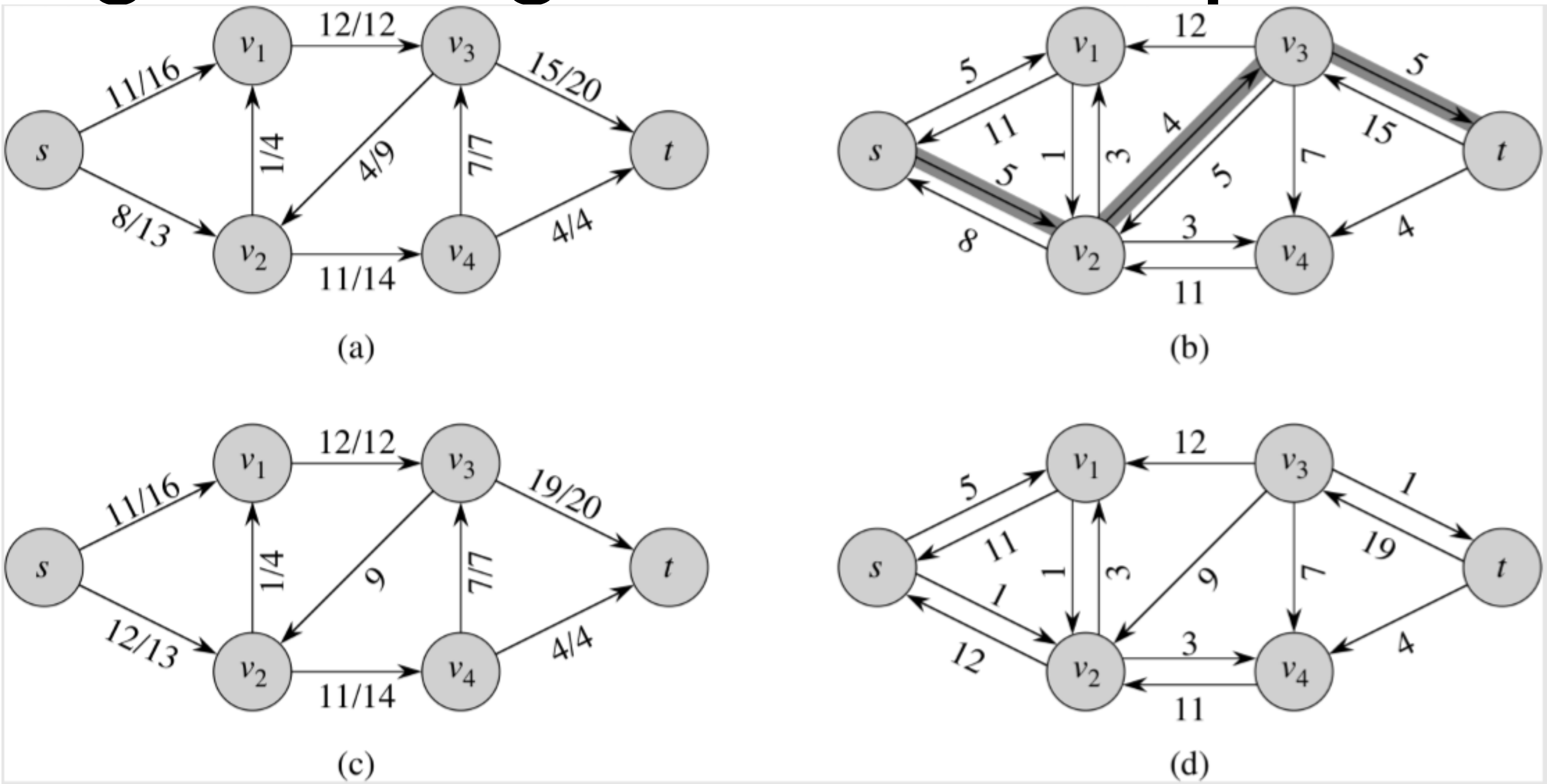
a) A flow network G

b) The residual network G_f, with augmenting path p shaded; residual capacity is $c_f(p) = \min c_f = c_f(v_2, v_3) = 4$

c) The flow in G that results from augmenting along path p by its residual capacity 4

d) The residual network induced by the flow in (c)

Augmenting Path, Example



Observe that G_f (as in (d)) now has no augmenting path

No edges cross the cut $(\{s, v_1, v_2, v_4\}, \{v_3, t\})$ in the forward direction

So no path to get from s to t

This when happens means, that flow in G (as in (c)) is a *maximum flow*

Lemma 26.2

Given flow network G, flow f in G, residual network G_f. Let p be an augmenting path in G_f. Define $f_p : V \times V \to \mathbb{R}$:

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

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

Corollary

Given flow network G, flow f in G, and an augmenting path p in G_f, define f_p as in lemma. Then $f \uparrow f_p$ is a flow in G with value $|f \uparrow f_p| = |f| + |f_p| > |f|$.

Proof

Direct deduction from the above two Lemmas

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.
2. G_f has no augmenting path.
3. $|f| = c(S, T)$ for some cut (S, T).

Proof

(1) $\Rightarrow$ (2): Show the contrapositive: if G_f has an augmenting path, then f is not a maximum flow. If G_f has augmenting path p, then by the above corollary, $f \uparrow f_p$ is a flow in G with value $|f| + |f_p| > |f|$, so that f was not a maximum flow.

Max-flow Min-cut Theorem

(2) $\Rightarrow$ (3): Suppose G_f has no augmenting path. Define

$$
\begin{aligned}
S &= \{v \in V : \text{there exists a path } s \leadsto v \text{ in } G_f\} \ , \\
T &= V - S \ .
\end{aligned}
$$

Must have $t \in T$; otherwise there is an augmenting path.
Therefore, (S, T) is a cut.
Consider $u \in S$ and $v \in T$:

- If $(u, v) \in E$, must have $f(u, v) = c(u, v)$; otherwise, $(u, v) \in E_f \Rightarrow v \in S$.
- If $(v, u) \in E$, must have $f(v, u) = 0$; otherwise, $c_f(u, v) = f(v, u) > 0 \Rightarrow (u, v) \in E_f \Rightarrow v \in S$.
- If $(u, v), (v, u) \notin E$, must have $f(u, v) = f(v, u) = 0$.

Then,

$$
\begin{aligned}
f(S, T) &= \sum_{u \in S} \sum_{v \in T} f(u, v) - \sum_{v \in T} \sum_{u \in S} f(v, u) \\
&= \sum_{u \in S} \sum_{v \in T} c(u, v) - \sum_{v \in T} \sum_{u \in S} 0 \\
&= c(S, T) \ .
\end{aligned}
$$

By lemma, $|f| = f(S, T) = c(S, T)$.

(3) $\Rightarrow$ (1): By corollary, $|f| \leq c(S, T)$.
Therefore, $|f| = c(S, T) \Rightarrow f$ is a max flow.

Ford-Fulkerson Algorithm

Keep augmenting flow along an augmenting path until there is no augmenting path.

Represent the flow attribute using the usual dot-notation, but on an edge: $(u, v).f$.

FORD-FULKERSON(G, s, t)

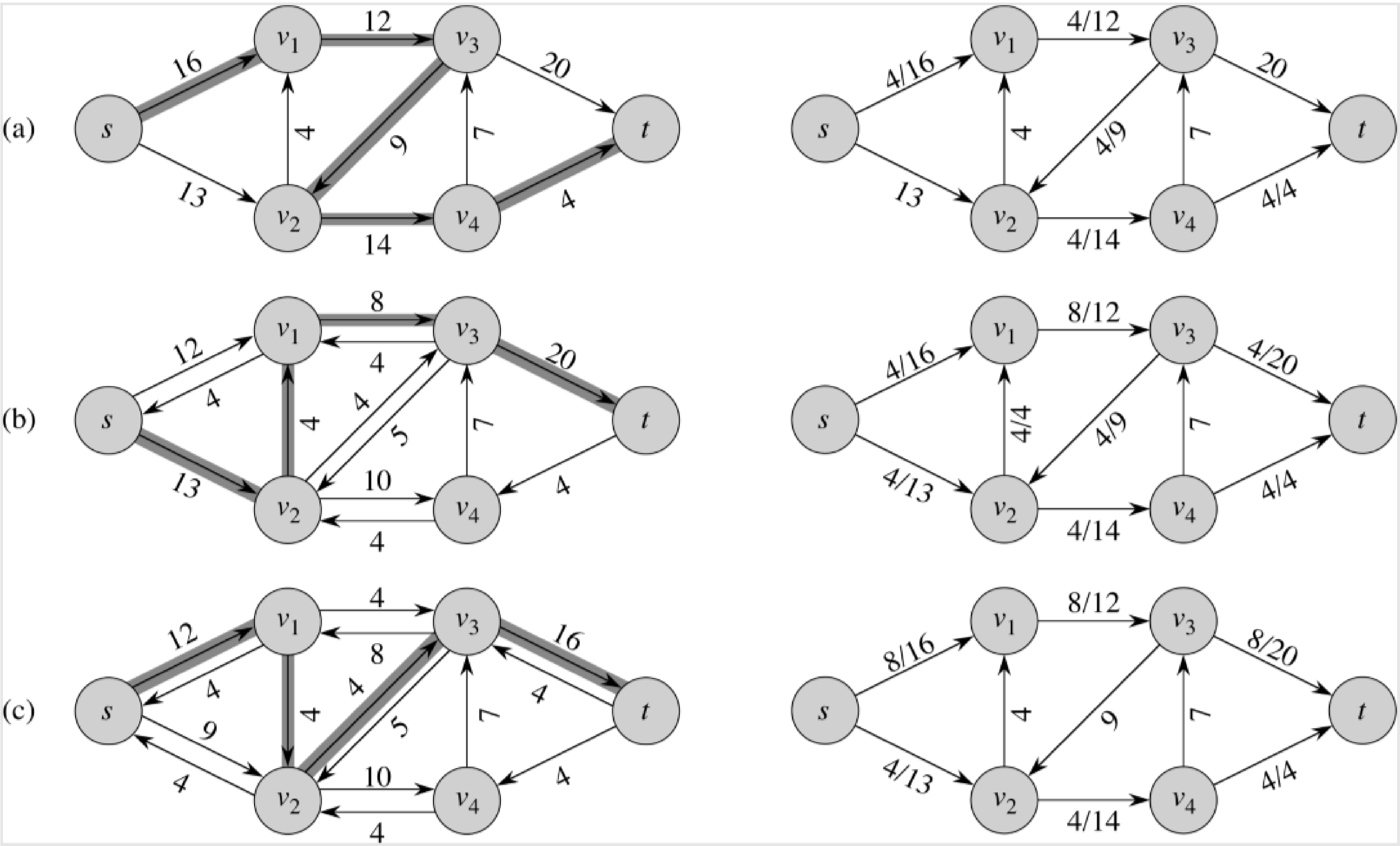
for all $(u, v) \in G.E$

    $(u, v).f = 0$

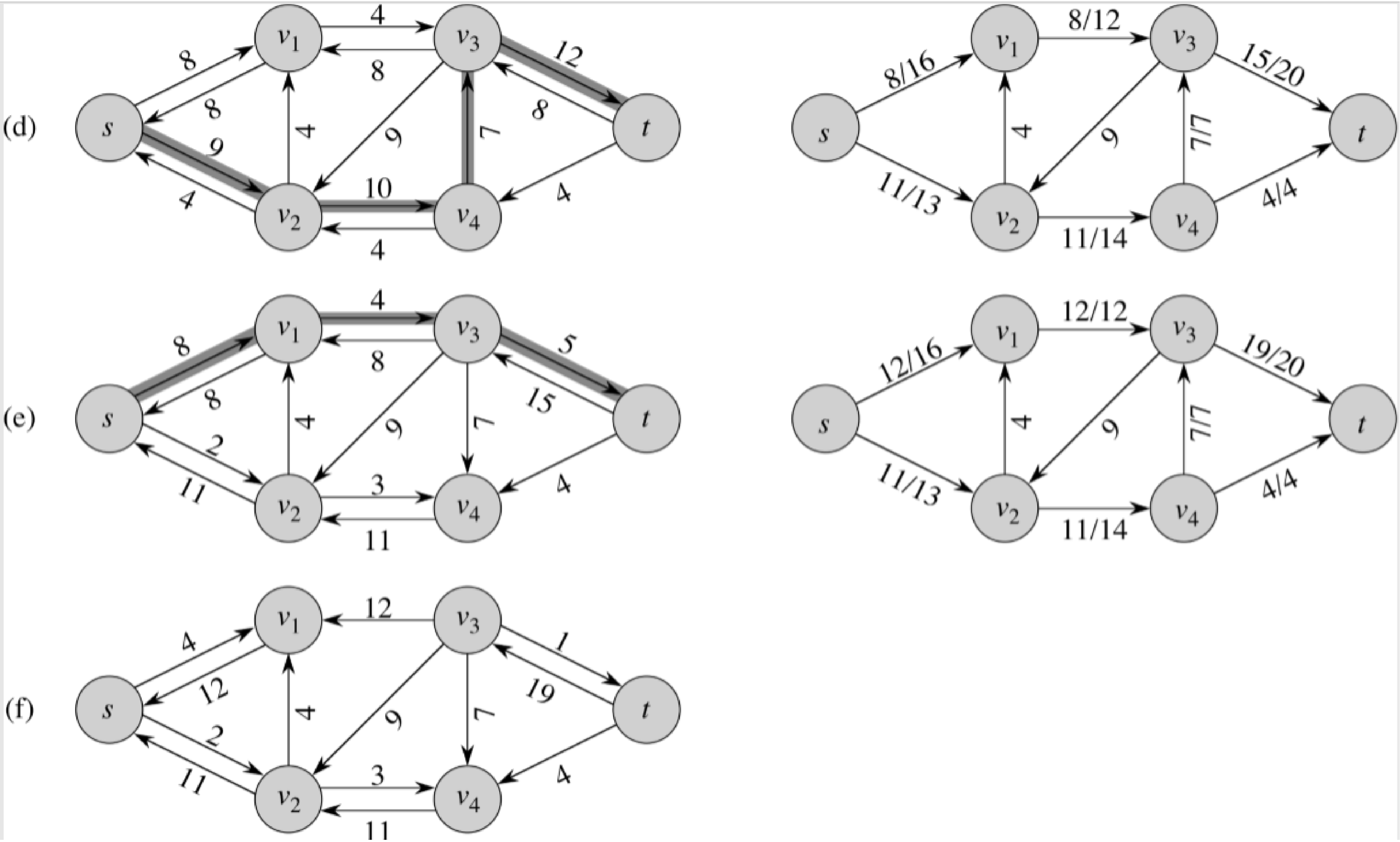
while there is an augmenting path p in G_f

    augment f by $c_f(p)$

Ford-Fulkerson, Example



Ford-Fulkerson, Example



Ford-Fulkerson, Analysis

- If capacities are integer, then each augmenting path raises $|f|$ by ≥ 1. If max flow is f^*, then we need $\leq |f^*|$ iterations
- This means time is $O(E|f^*|)$
- Note that this running time is not polynomial in input size. It depends on $|f^*|$, which is not a function of $|E|$ and $|V|$
- If capacities are irrational, FORD-FULKERSON might never terminate

HW

Exercises

- 26.1-1, 3, and 6
- 26.2-1, 2 & 8

Reference

- Class Notes from Dr. Istvan Jonyer of Oklahoma State University
- Class Notes from CSE 5311 of 2004, made by Hiren Patel, Ujjval Patel