

# Network Flows

## Lecture 16

March 17, 2011

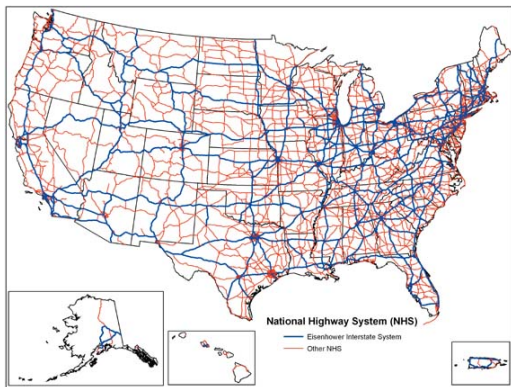
# Everything flows

**Panta rei** – everything flows (literally).  
Heraclitus (535–475 BC)

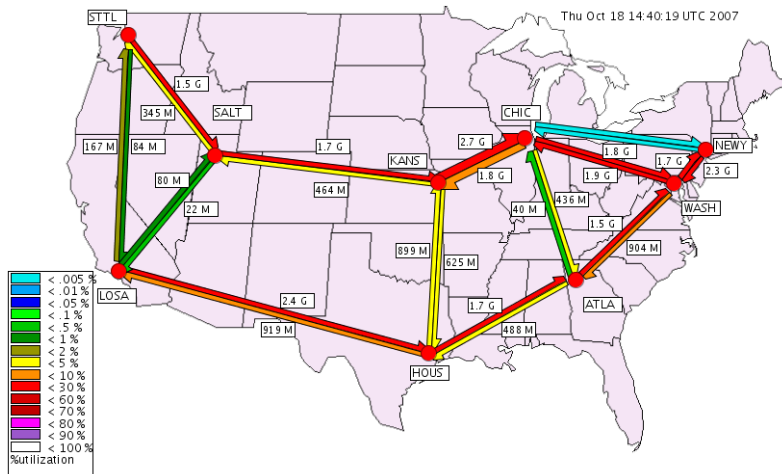
# Part I

## Network Flows: Introduction and Setup

# Transportation/Road Network



# Internet Backbone Network



# Common Features of Flow Networks

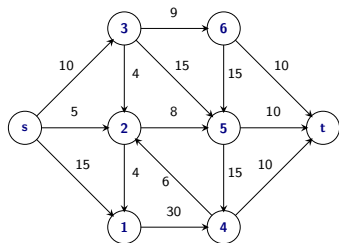
- *Network* represented by a (directed) *graph*  $\mathbf{G} = (\mathbf{V}, \mathbf{E})$
- Each edge  $\mathbf{e}$  has a *capacity*  $\mathbf{c}(\mathbf{e}) \geq \mathbf{0}$  that limits amount of *traffic* on  $\mathbf{e}$
- *Source(s)* of traffic/data
- *Sink(s)* of traffic/data
- Traffic *flows* from sources to sinks
- Traffic is *switched/interchanged* at nodes

**Flow:** abstract term to indicate stuff (traffic/data/etc) that *flows* from sources to sinks.

# Single Source Single Sink Flows

Simple setting:

- single source **s** and single sink **t**
- every other node **v** is an *internal* node
- flow originates at **s** and terminates at **t**



- Each edge **e** has a capacity  $c(e) \geq 0$
- Some times it is convenient to assume that source  $s \in V$  has no incoming edges and sink  $t \in V$  has no outgoing edges

**Assumptions:** All capacities are integer, and every vertex has at least one edge incident to it.

# Definition of Flow

Two ways to define flows:

- edge based
- path based

They are essentially equivalent but have different uses.

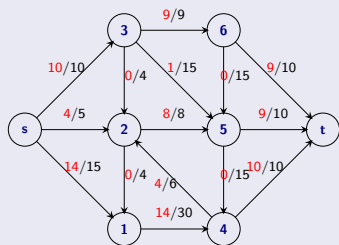
Edge based definition is more compact.



# Edge Based Definition of Flow

## Definition

A **flow** in a network  $G = (V, E)$ , is a function  $f : E \rightarrow \mathbb{R}^{\geq 0}$  such that



- **Capacity Constraint:** For each edge  $e$ ,  $f(e) \leq c(e)$
- **Conservation Constraint:** For each vertex  $v \neq s, t$

$$\sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e)$$

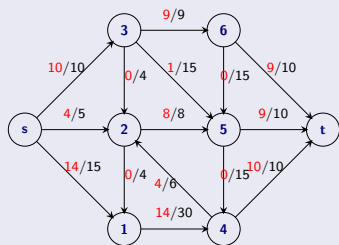
Figure: Flow with value

- **Value of flow:** (total flow out of source) — (total flow in to source)

# Edge Based Definition of Flow

## Definition

A **flow** in a network  $G = (V, E)$ , is a function  $f : E \rightarrow \mathbb{R}^{\geq 0}$  such that



- **Capacity Constraint:** For each edge  $e$ ,  $f(e) \leq c(e)$
- **Conservation Constraint:** For each vertex  $v \neq s, t$

$$\sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e)$$

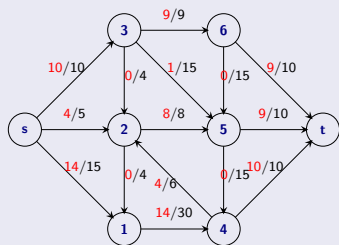
Figure: Flow with value

- **Value of flow:** (total flow out of source) – (total flow in to source)

# Edge Based Definition of Flow

## Definition

A **flow** in a network  $G = (V, E)$ , is a function  $f : E \rightarrow \mathbb{R}^{\geq 0}$  such that



- **Capacity Constraint:** For each edge  $e$ ,  $f(e) \leq c(e)$
- **Conservation Constraint:** For each vertex  $v \neq s, t$

$$\sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e)$$

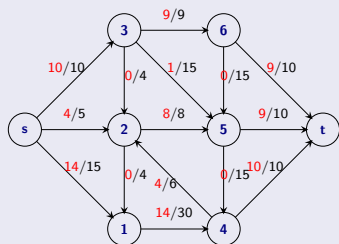
Figure: Flow with value

- **Value of flow:** (total flow out of source) — (total flow in to source)

# Edge Based Definition of Flow

## Definition

A **flow** in a network  $G = (V, E)$ , is a function  $f : E \rightarrow \mathbb{R}^{\geq 0}$  such that



- **Capacity Constraint:** For each edge  $e$ ,  $f(e) \leq c(e)$
- **Conservation Constraint:** For each vertex  $v \neq s, t$

$$\sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e)$$

Figure: Flow with value

- **Value of flow:** (total flow out of source) — (total flow in to source)

Conservation of flow law is also known as **Kirchhoff's law**.

# More Definitions and Notation

## Notation

- The inflow into a vertex  $v$  is  $f^{\text{in}}(v) = \sum_{e \text{ into } v} f(e)$  and the outflow is  $f^{\text{out}}(v) = \sum_{e \text{ out of } v} f(e)$
- For a set of vertices  $A$ ,  $f^{\text{in}}(A) = \sum_{e \text{ into } A} f(e)$ . Outflow  $f^{\text{out}}(A)$  is defined analogously

## Definition

For a network  $G = (V, E)$  with source  $s$ , the **value** of flow  $f$  is defined as  $v(f) = f^{\text{out}}(s) - f^{\text{in}}(s)$

# A Path Based Definition of Flow

Intuition: flow goes from source  $\mathbf{s}$  to sink  $\mathbf{t}$  along a path.

$\mathcal{P}$ : set of all paths from  $\mathbf{s}$  to  $\mathbf{t}$ .  $|\mathcal{P}|$  can be *exponential* in  $\mathbf{n}$ .

## Definition

A flow in a network  $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ , is a function  $\mathbf{f} : \mathcal{P} \rightarrow \mathbb{R}^{\geq 0}$  such that

- **Capacity Constraint:** For each edge  $\mathbf{e}$ , total flow on  $\mathbf{e}$  is  $\leq \mathbf{c}(\mathbf{e})$ .

$$\sum_{\mathbf{p} \in \mathcal{P} : \mathbf{e} \in \mathbf{p}} \mathbf{f}(\mathbf{p}) \leq \mathbf{c}(\mathbf{e})$$

- **Conservation Constraint:** No need! Automatic.

Value of flow:  $\sum_{\mathbf{p} \in \mathcal{P}} \mathbf{f}(\mathbf{p})$

# A Path Based Definition of Flow

Intuition: flow goes from source **s** to sink **t** along a path.

$\mathcal{P}$ : set of all paths from **s** to **t**.  $|\mathcal{P}|$  can be *exponential* in **n**.

## Definition

A flow in a network  $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ , is a function  $\mathbf{f} : \mathcal{P} \rightarrow \mathbb{R}^{\geq 0}$  such that

- **Capacity Constraint:** For each edge **e**, total flow on **e** is  $\leq c(\mathbf{e})$ .

$$\sum_{\mathbf{p} \in \mathcal{P} : \mathbf{e} \in \mathbf{p}} \mathbf{f}(\mathbf{p}) \leq c(\mathbf{e})$$

- **Conservation Constraint:** No need! Automatic.

Value of flow:  $\sum_{\mathbf{p} \in \mathcal{P}} \mathbf{f}(\mathbf{p})$



# A Path Based Definition of Flow

Intuition: flow goes from source **s** to sink **t** along a path.

$\mathcal{P}$ : set of all paths from **s** to **t**.  $|\mathcal{P}|$  can be *exponential* in **n**.

## Definition

A flow in a network  $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ , is a function  $\mathbf{f} : \mathcal{P} \rightarrow \mathbb{R}^{\geq 0}$  such that

- **Capacity Constraint:** For each edge **e**, total flow on **e** is  $\leq c(\mathbf{e})$ .

$$\sum_{\mathbf{p} \in \mathcal{P} : \mathbf{e} \in \mathbf{p}} \mathbf{f}(\mathbf{p}) \leq c(\mathbf{e})$$

- **Conservation Constraint:** No need! Automatic.

Value of flow:  $\sum_{\mathbf{p} \in \mathcal{P}} \mathbf{f}(\mathbf{p})$

# A Path Based Definition of Flow

Intuition: flow goes from source  $\mathbf{s}$  to sink  $\mathbf{t}$  along a path.

$\mathcal{P}$ : set of all paths from  $\mathbf{s}$  to  $\mathbf{t}$ .  $|\mathcal{P}|$  can be *exponential* in  $\mathbf{n}$ .

## Definition

A flow in a network  $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ , is a function  $\mathbf{f} : \mathcal{P} \rightarrow \mathbb{R}^{\geq 0}$  such that

- **Capacity Constraint:** For each edge  $\mathbf{e}$ , total flow on  $\mathbf{e}$  is  $\leq \mathbf{c}(\mathbf{e})$ .

$$\sum_{\mathbf{p} \in \mathcal{P} : \mathbf{e} \in \mathbf{p}} \mathbf{f}(\mathbf{p}) \leq \mathbf{c}(\mathbf{e})$$

- **Conservation Constraint:** No need! Automatic.

Value of flow:  $\sum_{\mathbf{p} \in \mathcal{P}} \mathbf{f}(\mathbf{p})$

# A Path Based Definition of Flow

Intuition: flow goes from source  $\mathbf{s}$  to sink  $\mathbf{t}$  along a path.

$\mathcal{P}$ : set of all paths from  $\mathbf{s}$  to  $\mathbf{t}$ .  $|\mathcal{P}|$  can be *exponential* in  $\mathbf{n}$ .

## Definition

A flow in a network  $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ , is a function  $\mathbf{f} : \mathcal{P} \rightarrow \mathbb{R}^{\geq 0}$  such that

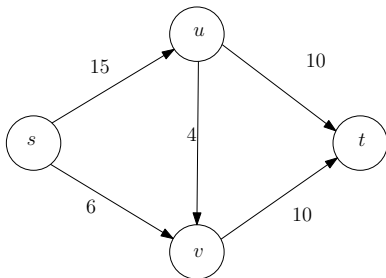
- **Capacity Constraint:** For each edge  $\mathbf{e}$ , total flow on  $\mathbf{e}$  is  $\leq \mathbf{c}(\mathbf{e})$ .

$$\sum_{\mathbf{p} \in \mathcal{P} : \mathbf{e} \in \mathbf{p}} \mathbf{f}(\mathbf{p}) \leq \mathbf{c}(\mathbf{e})$$

- **Conservation Constraint:** No need! Automatic.

Value of flow:  $\sum_{\mathbf{p} \in \mathcal{P}} \mathbf{f}(\mathbf{p})$

# Example



$$\mathcal{P} = \{p_1, p_2, p_3\}$$

$$p_1 : s \rightarrow u \rightarrow t$$

$$p_2 : s \rightarrow u \rightarrow v \rightarrow t$$

$$p_3 : s \rightarrow v \rightarrow t$$

$$f(p_1) = 10, f(p_2) = 4, f(p_3) = 6$$

# Path based flow implies Edge based flow

## Lemma

Given a path based flow  $\mathbf{f} : \mathcal{P} \rightarrow \mathbb{R}^{\geq 0}$  there is an edge based flow  $\mathbf{f}' : \mathbf{E} \rightarrow \mathbb{R}^{\geq 0}$  of the same value.

## Proof.

For each edge  $\mathbf{e}$  define  $\mathbf{f}'(\mathbf{e}) = \sum_{\mathbf{p}: \mathbf{e} \in \mathbf{p}} \mathbf{f}(\mathbf{p})$ .

**Exercise:** verify capacity and conservation constraints for  $\mathbf{f}'$ .

**Exercise:** verify that value of  $\mathbf{f}$  and  $\mathbf{f}'$  are equal □

# Path based flow implies Edge based flow

## Lemma

Given a path based flow  $\mathbf{f} : \mathcal{P} \rightarrow \mathbb{R}^{\geq 0}$  there is an edge based flow  $\mathbf{f}' : \mathbf{E} \rightarrow \mathbb{R}^{\geq 0}$  of the same value.

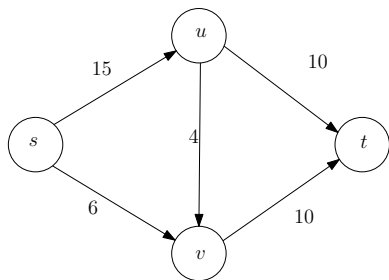
## Proof.

For each edge  $\mathbf{e}$  define  $\mathbf{f}'(\mathbf{e}) = \sum_{\mathbf{p}: \mathbf{e} \in \mathbf{p}} \mathbf{f}(\mathbf{p})$ .

**Exercise:** verify capacity and conservation constraints for  $\mathbf{f}'$ .

**Exercise:** verify that value of  $\mathbf{f}$  and  $\mathbf{f}'$  are equal □

# Example



$$\mathcal{P} = \{p_1, p_2, p_3\}$$

$$p_1 : s \rightarrow u \rightarrow t$$

$$p_2 : s \rightarrow u \rightarrow v \rightarrow t$$

$$p_3 : s \rightarrow v \rightarrow t$$

$$f(p_1) = 10, f(p_2) = 4, f(p_3) = 6$$

$$f'((s, u)) = 14$$

$$f'((u, v)) = 4$$

$$f'((s, v)) = 6$$

$$f'((u, t)) = 10$$

$$f'((v, t)) = 10$$

# Flow Decomposition

## Edge based flow to Path based Flow

### Lemma

Given an edge based flow  $\mathbf{f}' : \mathbf{E} \rightarrow \mathbb{R}^{\geq 0}$ , there is a path based flow  $\mathbf{f} : \mathcal{P} \rightarrow \mathbb{R}^{\geq 0}$  of same value. Moreover,  $\mathbf{f}$  assigns non-negative flow to at most  $\mathbf{m}$  paths where  $|\mathbf{E}| = \mathbf{m}$  and  $|\mathbf{V}| = \mathbf{n}$ . Given  $\mathbf{f}'$ , the path based flow can be computed in  $\mathbf{O}(\mathbf{mn})$  time.



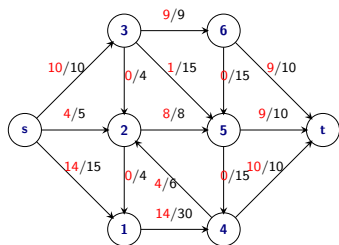
# Flow Decomposition

## Edge based flow to Path based Flow

### Proof Idea.

- remove all edges with  $f'(e) = 0$
- find a path  $p$  from  $s$  to  $t$
- assign  $f(p)$  to be  $\min_{e \in p} f'(e)$
- reduce  $f'(e)$  for all  $e \in p$  by  $f(p)$
- repeat until no path from  $s$  to  $t$
- in each iteration at least one edge has flow reduced to zero; hence at most  $m$  iterations. Can be implemented in  $O(m(m+n))$  time.  $O(mn)$  time requires care. □

# Example



# Edge vs Path based Definitions of Flow

Edge based flows:

- *compact* representation, only  $m$  values to be specified
- need to check flow conservation explicitly at each internal node

Path flows:

- in some applications, paths more natural
- not compact
- no need to check flow conservation constraints

Equivalence shows that we can go back and forth easily.

# The Maximum-Flow Problem

## Problem

**Input** A network  $G$  with capacity  $c$  and source  $s$  and sink  $t$

**Goal** Find flow of *maximum* value

**Question:** Given a flow network, what is an *upper bound* on the maximum flow between source and sink?

# The Maximum-Flow Problem

## Problem

**Input** A network  $G$  with capacity  $c$  and source  $s$  and sink  $t$

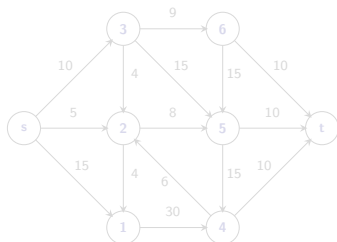
**Goal** Find flow of *maximum* value

**Question:** Given a flow network, what is an *upper bound* on the maximum flow between source and sink?

## Definition

Given a flow network an **s-t cut** is a set of edges  $E' \subset E$  such that removing  $E'$  *disconnects*  $s$  from  $t$ : in other words there is no directed  $s \rightarrow t$  path in  $E - E'$ .

The **capacity** of a cut  $E'$  is  $\sum_{e \in E'} c(e)$ .

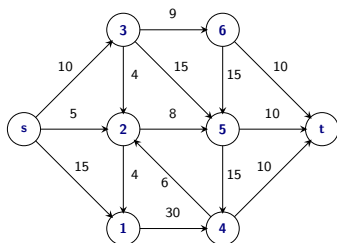


**Caution:** cut may leave  $t \rightarrow s$  paths!

## Definition

Given a flow network an **s-t cut** is a set of edges  $E' \subset E$  such that removing  $E'$  *disconnects*  $s$  from  $t$ : in other words there is no directed  $s \rightarrow t$  path in  $E - E'$ .

The **capacity** of a cut  $E'$  is  $\sum_{e \in E'} c(e)$ .

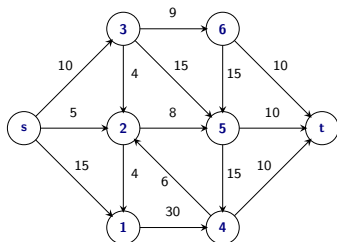


**Caution:** cut may leave  $t \rightarrow s$  paths!

## Definition

Given a flow network an **s-t cut** is a set of edges  $E' \subset E$  such that removing  $E'$  *disconnects*  $s$  from  $t$ : in other words there is no directed  $s \rightarrow t$  path in  $E - E'$ .

The **capacity** of a cut  $E'$  is  $\sum_{e \in E'} c(e)$ .



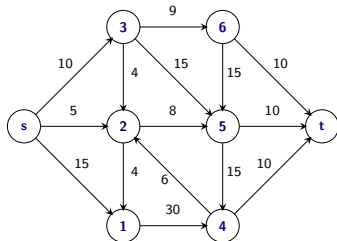
**Caution:** cut may leave  $t \rightarrow s$  paths!



# Minimal Cut

## Definition

Given a flow network an  $s$ - $t$ ,  $E'$  is a **minimal cut** if for all  $e \in E'$ ,  $E' - \{e\}$  is not a cut.

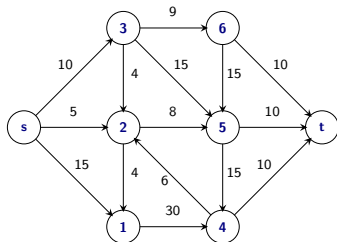


**Observation:** given a cut  $E'$ , can check efficiently whether  $E'$  is a minimal cut or not. How?

# Minimal Cut

## Definition

Given a flow network an  $s$ - $t$ ,  $E'$  is a **minimal cut** if for all  $e \in E'$ ,  $E' - \{e\}$  is not a cut.



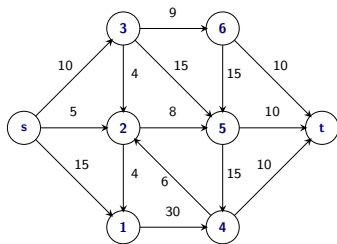
**Observation:** given a cut  $E'$ , can check efficiently whether  $E'$  is a minimal cut or not. How?

# Cuts as Vertex Partitions

Let  $A \subset V$  such that

- $s \in A, t \notin A$
- $B = V - A$  and hence  $t \in B$

Define  $\text{cut}(A, B) = \{(u, v) \in E \mid u \in A, v \in B\}$  : edges leaving  $A$



# Cuts as Vertex Partitions

Let  $A \subset V$  such that

- $s \in A, t \notin A$
- $B = V - A$  and hence  $t \in B$

Define  $\text{cut}(A, B) = \{(u, v) \in E \mid u \in A, v \in B\}$  : edges leaving  $A$

## Claim

$(A, B)$  is an  $s$ - $t$  cut.

## Proof.

Let  $P$  be any  $s \rightarrow t$  path in  $G$ . Since  $t$  is not in  $A$ ,  $P$  has to leave  $A$  via some edge  $(u, v)$  in  $(A, B)$ . □

# Cuts as Vertex Partitions

## Lemma

Suppose  $E'$  is an  $s$ - $t$  cut. Then there is a cut  $(A, B)$  such that  $(A, B) \subseteq E'$ .

## Proof.

$E'$  is an  $s$ - $t$  cut implies no path from  $s$  to  $t$  in  $(V, E - E')$ .

- Let  $A$  be set of all nodes reachable by  $s$  in  $(V, E - E')$ .
- Since  $E'$  is a cut,  $t \notin A$ .
- $(A, B) \subseteq E'$ . Why? If some edge  $(u, v) \in (A, B)$  is not in  $E'$  then  $v$  will be reachable by  $s$  and should be in  $A$ , hence a contradiction. □

## Corollary

Every minimal  $s$ - $t$  cut  $E'$  is a cut of the form  $(A, B)$ .

# Cuts as Vertex Partitions

## Lemma

Suppose  $E'$  is an  $s$ - $t$  cut. Then there is a cut  $(A, B)$  such that  $(A, B) \subseteq E'$ .

## Proof.

$E'$  is an  $s$ - $t$  cut implies no path from  $s$  to  $t$  in  $(V, E - E')$ .

- Let  $A$  be set of all nodes reachable by  $s$  in  $(V, E - E')$ .
- Since  $E'$  is a cut,  $t \notin A$ .
- $(A, B) \subseteq E'$ . Why? If some edge  $(u, v) \in (A, B)$  is not in  $E'$  then  $v$  will be reachable by  $s$  and should be in  $A$ , hence a contradiction. □

## Corollary

Every minimal  $s$ - $t$  cut  $E'$  is a cut of the form  $(A, B)$ .

# Cuts as Vertex Partitions

## Lemma

Suppose  $E'$  is an  $s$ - $t$  cut. Then there is a cut  $(A, B)$  such that  $(A, B) \subseteq E'$ .

## Proof.

$E'$  is an  $s$ - $t$  cut implies no path from  $s$  to  $t$  in  $(V, E - E')$ .

- Let  $A$  be set of all nodes reachable by  $s$  in  $(V, E - E')$ .
- Since  $E'$  is a cut,  $t \notin A$ .
- $(A, B) \subseteq E'$ . Why? If some edge  $(u, v) \in (A, B)$  is not in  $E'$  then  $v$  will be reachable by  $s$  and should be in  $A$ , hence a contradiction. □

## Corollary

Every minimal  $s$ - $t$  cut  $E'$  is a cut of the form  $(A, B)$ .

# Cuts as Vertex Partitions

## Lemma

Suppose  $E'$  is an  $s$ - $t$  cut. Then there is a cut  $(A, B)$  such that  $(A, B) \subseteq E'$ .

## Proof.

$E'$  is an  $s$ - $t$  cut implies no path from  $s$  to  $t$  in  $(V, E - E')$ .

- Let  $A$  be set of all nodes reachable by  $s$  in  $(V, E - E')$ .
- Since  $E'$  is a cut,  $t \notin A$ .
- $(A, B) \subseteq E'$ . Why? If some edge  $(u, v) \in (A, B)$  is not in  $E'$  then  $v$  will be reachable by  $s$  and should be in  $A$ , hence a contradiction. □

## Corollary

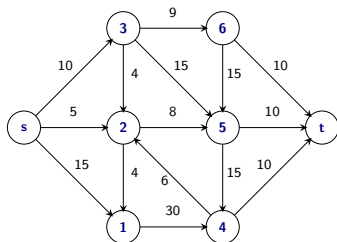
Every minimal  $s$ - $t$  cut  $E'$  is a cut of the form  $(A, B)$ .



# Minimum Cut

## Definition

Given a flow network an **s-t minimum** cut is a cut  $E'$  of smallest capacity amongst all **s-t** cuts.

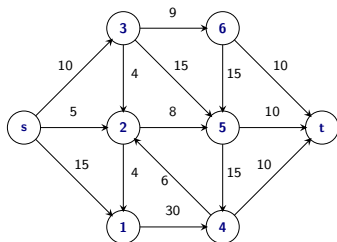


**Observation:** exponential number of **s-t** cuts and no “easy” algorithm to find a minimum cut.

# Minimum Cut

## Definition

Given a flow network an **s-t minimum cut** is a cut  $E'$  of smallest capacity amongst all **s-t** cuts.



**Observation:** exponential number of **s-t** cuts and no “easy” algorithm to find a minimum cut.

# The Minimum-Cut Problem

## Problem

**Input** A flow network **G**

**Goal** Find the capacity of a *minimum* **s-t** cut

# Flows and Cuts

## Lemma

For any **s-t** cut  $E'$ , **maximum s-t flow**  $\leq$  capacity of  $E'$ .

## Proof.

Formal proof easier with path based definition of flow.

Suppose  $f : \mathcal{P} \rightarrow \mathbb{R}^{\geq 0}$  is a max-flow.

Every path  $p \in \mathcal{P}$  contains an edge  $e \in E'$ . Why?

Assign each path  $p \in \mathcal{P}$  to exactly one edge  $e \in E'$ .

Let  $\mathcal{P}_e$  be paths assigned to  $e \in E'$ . Then

$$v(f) = \sum_{p \in \mathcal{P}} f(p) = \sum_{e \in E'} \sum_{p \in \mathcal{P}_e} f(p) \leq \sum_{e \in E'} c(e)$$



# Flows and Cuts

## Lemma

For any **s-t** cut  $E'$ , **maximum s-t flow**  $\leq$  capacity of  $E'$ .

## Proof.

Formal proof easier with path based definition of flow.

Suppose  $f : \mathcal{P} \rightarrow \mathbb{R}^{\geq 0}$  is a max-flow.

Every path  $p \in \mathcal{P}$  contains an edge  $e \in E'$ . Why?

Assign each path  $p \in \mathcal{P}$  to exactly one edge  $e \in E'$ .

Let  $\mathcal{P}_e$  be paths assigned to  $e \in E'$ . Then

$$v(f) = \sum_{p \in \mathcal{P}} f(p) = \sum_{e \in E'} \sum_{p \in \mathcal{P}_e} f(p) \leq \sum_{e \in E'} c(e)$$



# Flows and Cuts

## Lemma

For any s-t cut  $E'$ , **maximum s-t flow**  $\leq$  capacity of  $E'$ .

## Proof.

Formal proof easier with path based definition of flow.

Suppose  $f : \mathcal{P} \rightarrow \mathbb{R}^{\geq 0}$  is a max-flow.

Every path  $p \in \mathcal{P}$  contains an edge  $e \in E'$ . Why?

Assign each path  $p \in \mathcal{P}$  to exactly one edge  $e \in E'$ .

Let  $\mathcal{P}_e$  be paths assigned to  $e \in E'$ . Then

$$v(f) = \sum_{p \in \mathcal{P}} f(p) = \sum_{e \in E'} \sum_{p \in \mathcal{P}_e} f(p) \leq \sum_{e \in E'} c(e)$$



# Flows and Cuts

## Lemma

For any s-t cut  $E'$ , **maximum s-t flow**  $\leq$  capacity of  $E'$ .

## Proof.

Formal proof easier with path based definition of flow.

Suppose  $f : \mathcal{P} \rightarrow \mathbb{R}^{\geq 0}$  is a max-flow.

Every path  $p \in \mathcal{P}$  contains an edge  $e \in E'$ . Why?

Assign each path  $p \in \mathcal{P}$  to exactly one edge  $e \in E'$ .

Let  $\mathcal{P}_e$  be paths assigned to  $e \in E'$ . Then

$$v(f) = \sum_{p \in \mathcal{P}} f(p) = \sum_{e \in E'} \sum_{p \in \mathcal{P}_e} f(p) \leq \sum_{e \in E'} c(e)$$



# Flows and Cuts

## Lemma

For any **s-t** cut  $E'$ , **maximum s-t flow**  $\leq$  capacity of  $E'$ .

## Corollary

Maximum **s-t** flow  $\leq$  minimum **s-t** cut.



# Max-Flow Min-Cut Theorem

## Theorem

*In any flow network the maximum **s-t** flow is equal to the minimum **s-t** cut.*

Can compute minimum-cut from maximum flow and vice-versa!  
Proof coming shortly.

Many applications:

- optimization
- graph theory
- combinatorics

# Max-Flow Min-Cut Theorem

## Theorem

*In any flow network the maximum **s-t** flow is equal to the minimum **s-t** cut.*

Can compute minimum-cut from maximum flow and vice-versa!

Proof coming shortly.

Many applications:

- optimization
- graph theory
- combinatorics

# Max-Flow Min-Cut Theorem

## Theorem

*In any flow network the maximum **s-t** flow is equal to the minimum **s-t** cut.*

Can compute minimum-cut from maximum flow and vice-versa!

Proof coming shortly.

Many applications:

- optimization
- graph theory
- combinatorics

# The Maximum-Flow Problem

## Problem

**Input** A network  $G$  with capacity  $c$  and source  $s$  and sink  $t$

**Goal** Find flow of *maximum* value from  $s$  to  $t$

**Exercise:** Given  $G, s, t$  as above, show that one can remove all edges into  $s$  and all edges out of  $t$  without affecting the flow value between  $s$  and  $t$ .

# The Maximum-Flow Problem

## Problem

**Input** A network  $G$  with capacity  $c$  and source  $s$  and sink  $t$

**Goal** Find flow of *maximum* value from  $s$  to  $t$

**Exercise:** Given  $G, s, t$  as above, show that one can remove all edges into  $s$  and all edges out of  $t$  without affecting the flow value between  $s$  and  $t$ .

# Notes

# Notes

# Notes



# Notes