x, y-cuts

Definition

Given $x, y \in V(G)$, a set $S \subseteq V(G) \setminus \{x, y\}$ is an x, y-**cut** if $G - S$ has no x, y-path.

x, y-cuts

Definition

Given $x, y \in V(G)$, a set $S \subseteq V(G) \setminus \{x, y\}$ is an **x, y-cut** if $G - S$ has no x, y-path.

$\kappa(x, y) =$ minimum size of an x, y-cut.

x, y-cuts

Definition

Given $x, y \in V(G)$, a set $S \subseteq V(G) \setminus \{x, y\}$ is an **x, y-cut** if $G - S$ has no x, y-path.

$\kappa(x, y) =$ minimum size of an x, y-cut.

Note: If S is an x, y-cut, then $G - S$ has more than one component. Thus

$$\kappa(x, y) \geq \kappa(G).$$

x, y-cuts

Definition

Given $x, y \in V(G)$, a set $S \subseteq V(G) \setminus \{x, y\}$ is an **x, y-cut** if $G - S$ has no x, y-path.

$\kappa(x, y) =$ minimum size of an x, y-cut.

Note: If S is an x, y-cut, then $G - S$ has more than one component. Thus

$$\kappa(x, y) \geq \kappa(G).$$

In fact,

$$\kappa(G) = \min_{x,y \in V(G)} \kappa(x, y).$$

Menger's theorem

Definition

A set of x, y-paths is called **pairwise internally disjoint** if no two paths share an internal vertex.

Menger's theorem

Definition

A set of x, y-paths is called **pairwise internally disjoint** if no two paths share an internal vertex.

$\lambda(x, y) =$ maximum size of a set of pairwise internally disjoint x, y-paths.

Menger's theorem

Definition

A set of x, y-paths is called **pairwise internally disjoint** if no two paths share an internal vertex.

$\lambda(x, y) =$ maximum size of a set of pairwise internally disjoint x, y-paths.

Note that $\kappa(x, y) \geq \lambda(x, y)$

Menger's theorem

Definition

A set of x, y-paths is called **pairwise internally disjoint** if no two paths share an internal vertex.

$\lambda(x, y) =$ maximum size of a set of pairwise internally disjoint x, y-paths.

Note that $\kappa(x, y) \geq \lambda(x, y)$, since no vertex can cut two internally disjoint x, y-paths.

Menger's theorem

Definition

A set of x, y-paths is called **pairwise internally disjoint** if no two paths share an internal vertex.

$\lambda(x, y) =$ maximum size of a set of pairwise internally disjoint x, y-paths.

Note that $\kappa(x, y) \geq \lambda(x, y)$, since no vertex can cut two internally disjoint x, y-paths.

Theorem (Menger '27)

If $x, y \in V(G)$ and $xy \notin E(G)$, then

$$\kappa(x, y) = \lambda(x, y).$$

Menger's theorem

Definition

A set of x, y-paths is called **pairwise internally disjoint** if no two paths share an internal vertex.

$$\lambda(x, y) = \text{maximum size of a set of pairwise internally disjoint } x, y\text{-paths.}$$

Note that $\kappa(x, y) \geq \lambda(x, y)$, since no vertex can cut two internally disjoint x, y-paths.

Theorem (Menger '27)

If $x, y \in V(G)$ and $xy \notin E(G)$, then

$$\kappa(x, y) = \lambda(x, y).$$

Corollary

A graph G is k-connected if and only if, for every $u, v \in V(G)$, there are at least k internally disjoint u, v-paths.

Edge version of Menger's theorem

Definition

Given $x, y \in V(G)$, let

$\kappa'(x, y) =$ minimum size of an **x, y-disconnecting set** of edges.

Edge version of Menger's theorem

Definition

Given $x, y \in V(G)$, let

$\kappa'(x, y) =$ minimum size of an **x, y-disconnecting set** of edges.

$\lambda'(x, y) =$ maximum size of a set of **pairwise edge-disjoint** x, y-paths.

Edge version of Menger's theorem

Definition

Given $x, y \in V(G)$, let

$\kappa'(x, y) =$ minimum size of an **x, y-disconnecting set** of edges.

$\lambda'(x, y) =$ maximum size of a set of **pairwise edge-disjoint** x, y-paths.

As in the case of vertices, we have

$$\kappa'(G) = \min_{x,y \in V(G)} \kappa'(x, y).$$

Edge version of Menger's theorem

Definition

Given $x, y \in V(G)$, let

$\kappa'(x,y) =$ minimum size of an **x, y-disconnecting set** of edges.

$\lambda'(x,y) =$ maximum size of a set of **pairwise edge-disjoint** x, y-paths.

As in the case of vertices, we have

$$\kappa'(G) = \min_{x,y \in V(G)} \kappa'(x,y).$$

Also, $\kappa'(x,y) \geq \lambda'(x,y)$, since no edge can cut two edge-disjoint x, y-paths.

Edge version of Menger's theorem

Definition

Given $x, y \in V(G)$, let

$\kappa'(x, y) =$ minimum size of an **x, y-disconnecting set** of edges.

$\lambda'(x, y) =$ maximum size of a set of **pairwise edge-disjoint** x, y-paths.

As in the case of vertices, we have

$$\kappa'(G) = \min_{x,y \in V(G)} \kappa'(x, y).$$

Also, $\kappa'(x, y) \geq \lambda'(x, y)$, since no edge can cut two edge-disjoint x, y-paths.

Theorem

If $x, y \in V(G)$, then

$$\kappa'(x, y) = \lambda'(x, y).$$

The line graph

The proof of the edge version of Menger's theorem uses the notion of line graphs.

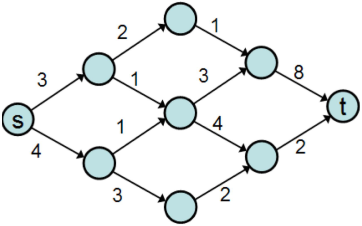
The line graph

The proof of the edge version of Menger's theorem uses the notion of line graphs.

Definition

The **line graph** of G, denoted by $L(G)$, is the graph whose vertices are the edges of G, and with $ef \in E(L(G))$ if e and f are edges of G that share an endpoint.

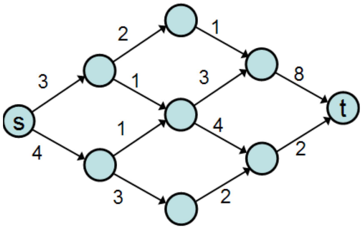
4.3 Network flow problems



Examples:

- Edges represent pipes where water flows in one direction, and the labels indicate their capacity (amount of water per second). What is the maximum flow from s to t?

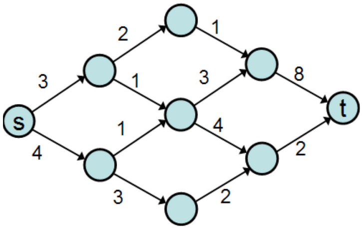
4.3 Network flow problems



Examples:

- Edges represent pipes where water flows in one direction, and the labels indicate their capacity (amount of water per second). What is the maximum flow from s to t?
- Edges represent one-way streets, and the labels indicate the maximum number of cars per hour. What is the maximum number of cars per hour that can travel from s to t?

4.3 Network flow problems



Examples:

- Edges represent pipes where water flows in one direction, and the labels indicate their capacity (amount of water per second). What is the maximum flow from s to t?
- Edges represent one-way streets, and the labels indicate the maximum number of cars per hour. What is the maximum number of cars per hour that can travel from s to t?
- The diagram represents a computer (or electrical) network, and the labels indicate the data (or electricity) transmission capacities. How much data (or current) can be transmitted from s to t?

Networks

Definition

A **network** is a weighted digraph D with two distinguished vertices: the **source** s and the **sink** t.
Each edge has a nonnegative label $c(e)$, called its **capacity**.

Networks

Definition

A **network** is a weighted digraph D with two distinguished vertices: the **source** s and the **sink** t.
Each edge has a nonnegative label $c(e)$, called its **capacity**.

Definition

A **feasible flow** is a function $f : E(D) \to \mathbb{R}_{\geq 0}$ that assigns to each edge a non-negative real number, such that

- $0 \leq f(e) \leq c(e)$ for all $e \in E(D)$; (capacity constraints)

Networks

Definition

A **network** is a weighted digraph D with two distinguished vertices: the **source** s and the **sink** t.
Each edge has a nonnegative label $c(e)$, called its **capacity**.

Definition

A **feasible flow** is a function $f : E(D) \to \mathbb{R}_{\geq 0}$ that assigns to each edge a non-negative real number, such that

- $0 \leq f(e) \leq c(e)$ for all $e \in E(D)$; (capacity constraints)
- $f^+(v) = f^-(v)$ for all $v \in V(G) \setminus \{s, t\}$,
  where $f^+(v)$ is the flow on edges leaving v,
  and $f^-(v)$ is the flow on edges entering v.
  (conservation constraints)

Maximum flow

Definition

The **value** of a flow f, denoted by $\mathrm{val}(f)$, is the net flow into the sink, that is,

$$f^-(t) - f^+(t).$$

Maximum flow

Definition

The **value** of a flow f, denoted by $\text{val}(f)$, is the net flow into the sink, that is,

$$f^{-}(t) - f^{+}(t).$$

Equivalently, by the conservation constraints, this equals the net flow out of the source, that is,

$$f^{+}(s) - f^{-}(s).$$

Maximum flow

Definition

The **value** of a flow f, denoted by $\mathrm{val}(f)$, is the net flow into the sink, that is,

$$f^-(t) - f^+(t).$$

Equivalently, by the conservation constraints, this equals the net flow out of the source, that is,

$$f^+(s) - f^-(s).$$

A **maximum flow** is a feasible flow of maximum value.

f-augmenting paths

Definition

If f is a feasible flow, an **f-augmenting path** is a path P from s to t such that, for each $e \in E(P)$,

- if P follows e in the forward direction, then $f(e) < c(e)$,

f-augmenting paths

Definition

If f is a feasible flow, an **f-augmenting path** is a path P from s to t such that, for each $e \in E(P)$,

- if P follows e in the forward direction, then $f(e) < c(e)$,
- if P follows e in the backward direction, then $f(e) > 0$.

f-augmenting paths

Definition

If f is a feasible flow, an **f-augmenting path** is a path P from s to t such that, for each $e \in E(P)$,

- if P follows e in the forward direction, then $f(e) < c(e)$, (let $\epsilon(e) = c(e) - f(e)$)
- if P follows e in the backward direction, then $f(e) > 0$. (let $\epsilon(e) = f(e)$)

The **tolerance** of P is

$$\min_{e \in E(P)} \epsilon(e).$$

f-augmenting paths

Definition

If f is a feasible flow, an **f-augmenting path** is a path P from s to t such that, for each $e \in E(P)$,

- if P follows e in the forward direction, then $f(e) < c(e)$, (let $\epsilon(e) = c(e) - f(e)$)
- if P follows e in the backward direction, then $f(e) > 0$. (let $\epsilon(e) = f(e)$)

The **tolerance** of P is

$$\min_{e \in E(P)} \epsilon(e).$$

An f-augmenting path leads to a feasible flow with larger value, by changing f on the edges of P.

f-augmenting paths

Definition

If f is a feasible flow, an **f-augmenting path** is a path P from s to t such that, for each $e \in E(P)$,

- if P follows e in the forward direction, then $f(e) < c(e)$, (let $\epsilon(e) = c(e) - f(e)$)
- if P follows e in the backward direction, then $f(e) > 0$. (let $\epsilon(e) = f(e)$)

The **tolerance** of P is

$$\min_{e \in E(P)} \epsilon(e).$$

An f-augmenting path leads to a feasible flow with larger value, by changing f on the edges of P.

How can we know when our flow is maximum?