Chapter 7: Maximum Flow Problems

(cp. Cook, Cunningham, Pulleyblank & Schrijver, Chapter 3)

Maximum s-t-Flow Problem

Given: Digraph $D = (V, A)$, arc capacities $u \in \mathbb{R}^A_{\geq 0}$, nodes $s, t \in V$.

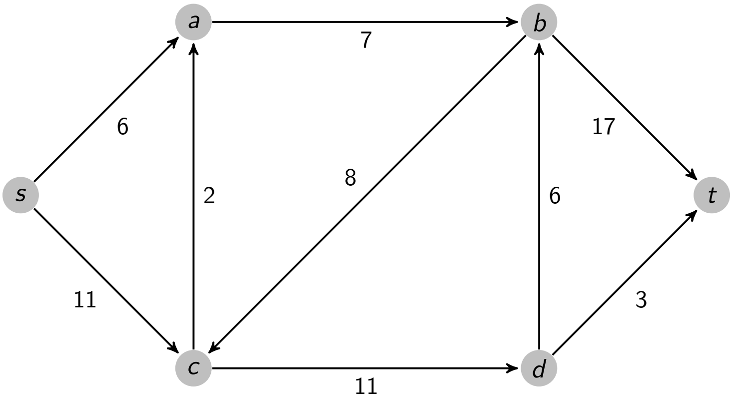
Definition 7.1.

A flow in D is a vector $x \in \mathbb{R}^A_{\geq 0}$. Moreover, a flow x in D

- i obeys arc capacities and is called feasible, if $x_a \leq u_a$ for each $a \in A$;
- ii has excess $\mathrm{ex}_x(v) := x(\delta^-(v)) - x(\delta^+(v))$ at node $v \in V$;
- iii satisfies flow conservation at node $v \in V$ if $\mathrm{ex}_x(v) = 0$;
- iv is a circulation if it satisfies flow conservation at each node $v \in V$;
- v is an s-t-flow of value $\mathrm{ex}_x(t)$ if it satisfies flow conservation at each node $v \in V \setminus \{s, t\}$ and if $\mathrm{ex}_x(t) \geq 0$.

The maximum s-t-flow problem asks for a feasible s-t-flow in D of maximum value.

Example



s-t-Flows and s-t-Cuts

For a subset of nodes $U \subseteq V$, the excess of U is defined as

$$\mathrm{ex}_x(U) := x(\delta^-(U)) - x(\delta^+(U)) \ .$$

Lemma 7.2.

For a flow x and a subset of nodes U it holds that $\mathrm{ex}_x(U) = \sum_{v \in U} \mathrm{ex}_x(v)$. In particular, the value of an s-t-flow x is equal to

$$\mathrm{ex}_x(t) = -\mathrm{ex}_x(s) = \mathrm{ex}_x(U) \quad \text{for each } U \subseteq V \setminus \{s\} \text{ with } t \in U.$$

For $U \subseteq V \setminus \{s\}$ with $t \in U$, the subset of arcs $\delta^-(U)$ is called an s-t-cut.

Lemma 7.3.

Let $U \subseteq V \setminus \{s\}$ with $t \in U$. The value of a feasible s-t-flow x is at most the capacity $u(\delta^-(U))$ of the s-t-cut $\delta^-(U)$. Equality holds if and only if $x_a = u_a$ for each $a \in \delta^-(U)$ and $x_a = 0$ for each $a \in \delta^+(U)$.

Residual Graph and Residual Arcs

For $a = (v, w) \in A$, let $a^{-1} := (w, v)$ be the corresponding backward arc and $A^{-1} := \{a^{-1} \mid a \in A\}$.

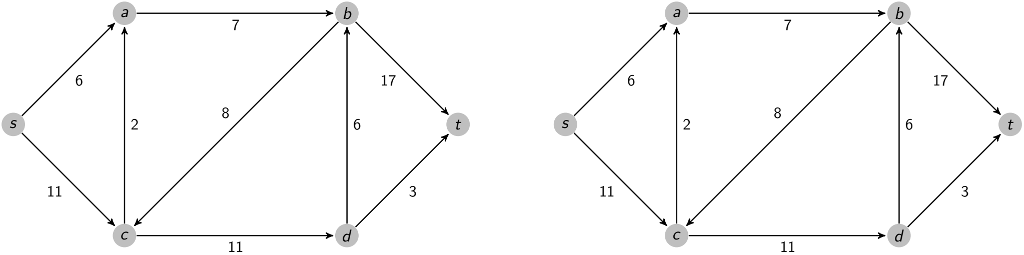
- For a feasible flow x, the set of residual arcs is given by

$$A_x := \{a \in A \mid x_a < u_a\} \cup \{a^{-1} \in A^{-1} \mid x_a > 0\} \ .$$

- For $a \in A$, define the residual capacity $u_x(a)$ as

$$u_x(a) := u(a) - x(a) \quad \text{if } a \in A_x, \quad \text{and} \quad u_x(a^{-1}) := x(a) \quad \text{if } a^{-1} \in A_x.$$

- The digraph $D_x := (V, A_x)$ is called the residual graph of x.



x-augmenting paths

Observation:

- If x is a feasible flow in (D, u) and y a feasible flow in (D_x, u_x), then

$$z(a) := x(a) + y(a) - y(a^{-1}) \quad \text{for } a \in A$$

yields a feasible flow z in D (we write $z := x + y$ for short).

Lemma 7.4.

If x is a feasible s-t-flow such that D_x does not contain an s-t-dipath, then x is a maximum s-t-flow.

Max-Flow Min-Cut Theorem and Ford-Fulkerson Algorithm

Theorem 7.5 (Max-Flow Min-Cut Theorem).

The maximum s-t-flow value equals the minimum capacity of an s-t-cut.

Corollary.

A feasible s-t-flow x is maximum if and only if D_x does not contain an s-t-dipath. □

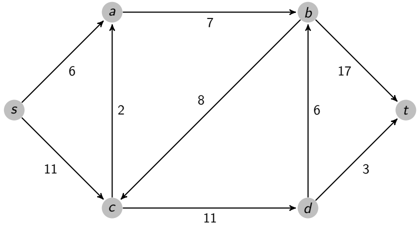
Ford-Fulkerson Algorithm

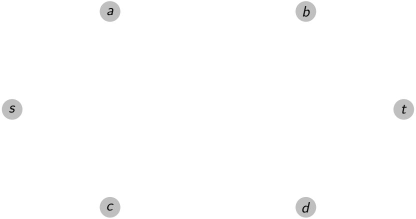
i. set $x := 0$;
ii. while there is an s-t-dipath P in D_x
iii. set $x := x + \delta \cdot \chi^P$ with $\delta := \min\{u_x(a) \mid a \in P\}$;

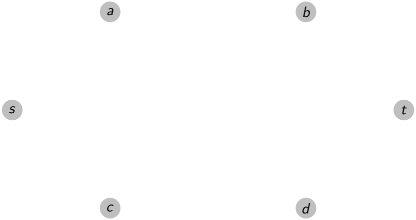
Here, $\chi^P : A \to \{0, 1, -1\}$ is the characteristic vector of dipath P defined by

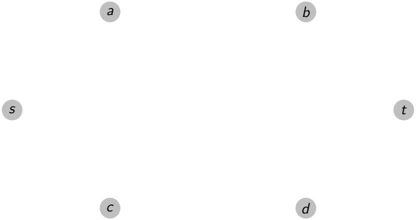
$$\chi^P(a) = \begin{cases} 1 & \text{if } a \in P, \\ -1 & \text{if } a^{-1} \in P, \\ 0 & \text{otherwise,} \end{cases} \qquad \text{for all } a \in A.$$

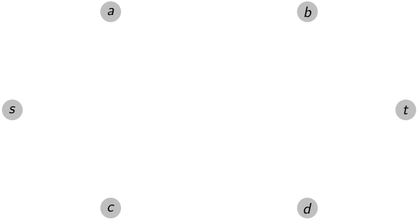
Ford-Fulkerson Example

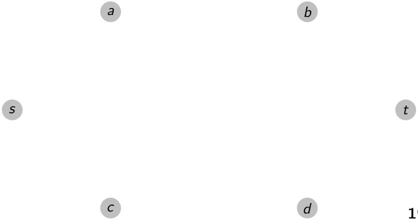












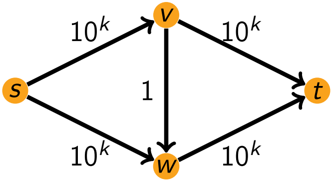
Termination of the Ford-Fulkerson Algorithm

Theorem 7.6.

a. If all capacities are rational, then the algorithm terminates with a maximum s-t-flow.

b. If all capacities are integral, it finds an integral maximum s-t-flow.

When an arbitrary x-augmenting path is chosen in every iteration, the Ford-Fulkerson Algorithm can behave badly:



Running Time of the Ford-Fulkerson Algorithm

Theorem 7.7.

If all capacities are integral and the maximum flow value is $K < \infty$, then the Ford-Fulkerson Algorithm terminates after at most K iterations. Its running time is $O(m \cdot K)$ in this case.

Proof: In each iteration the flow value is increased by at least 1. □

A variant of the Ford-Fulkerson Algo. is the Edmonds-Karp Algorithm:

- In each iteration, choose shortest s-t-dipath in D_x (edge lengths=1)

Theorem 7.8.

The Edmonds-Karp Algorithm terminates after at most $n \cdot m$ iterations; its running time is $O(n \cdot m^2)$.

Remark: The Edmonds-Karp Algorithm can be implemented with running time $O(n^2 \cdot m)$.

Arc-Based LP Formulation

Straightforward LP formulation of the maximum s-t-flow problem:

$$
\begin{aligned}
\max \quad & \sum_{a \in \delta^+(s)} x_a - \sum_{a \in \delta^-(s)} x_a \\
\text{s.t.} \quad & \sum_{a \in \delta^-(v)} x_a - \sum_{a \in \delta^+(v)} x_a = 0 && \text{for all } v \in V \setminus \{s, t\} \\
& x_a \le u(a) && \text{for all } a \in A \\
& x_a \ge 0 && \text{for all } a \in A
\end{aligned}
$$

Dual LP:

$$
\begin{aligned}
\min \quad & \sum_{a \in A} u(a) \cdot z_a \\
\text{s.t.} \quad & y_w - y_v + z_{(v,w)} \ge 0 && \text{for all } (v, w) \in A \\
& y_s = 1, \quad y_t = 0 \\
& z_a \ge 0 && \text{for all } a \in A
\end{aligned}
$$

Dual Solutions and s-t-Cuts

$$
\begin{aligned}
\min \quad & \sum_{a \in A} u(a) \cdot z_a \\
\text{s.t.} \quad & y_w - y_v + z_{(v,w)} \geq 0 && \text{for all } (v,w) \in A \\
& y_s = 1, \quad y_t = 0 \\
& z_a \geq 0 && \text{for all } a \in A
\end{aligned}
$$

Observation: An s-t-cut $\delta^+(U)$ (with $U \subseteq V \setminus \{t\}$, $s \in U$) yields feasible dual solution (y, z) of value $u(\delta^+(U))$:

- let y be the characteristic vector χ^U of U (i. e., $y_v = 1$ for $v \in U$, $y_v = 0$ for $v \in V \setminus U$)
- let z be the characteristic vector $\chi^{\delta^+(U)}$ of $\delta^+(U)$ (i. e., $z_a = 1$ for $a \in \delta^+(U)$, $z_a = 0$ for $a \in A \setminus \delta^+(U)$)

Theorem 7.9.

There exists an s-t-cut $\delta^+(U)$ (with $U \subseteq V \setminus \{t\}$, $s \in U$) such that the corresponding dual solution (y, z) is an optimal dual solution.

Application: Kőnig's Theorem

Definition 7.10.

Consider an undirected graph $G = (V, E)$.

i. A matching in G is a subset of edges $M \subseteq E$ with $e \cap e' = \emptyset$ for all $e, e' \in M$ with $e \neq e'$.
ii. A vertex cover is a subset of nodes $C \subseteq V$ with $e \cap C \neq \emptyset$ for all $e \in E$.

Theorem 7.11.

In bipartite graphs, the maximum cardinality of a matching equals the minimum cardinality of a vertex cover.

Observation: In a bipartite graph $G = (P \dot{\cup} Q, E)$, a maximum cardinality matching can be found by a maximum flow computation.