Advanced Algorithms

Nicole Megow (Universität Bremen)

SoSe 2025

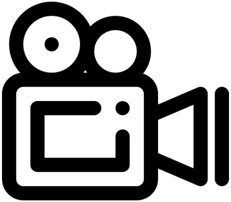
Network Flows

Lecture 5

Recording of this Lecture

This lecture will be recorded

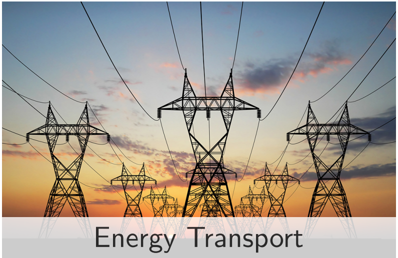
- Recording only of the lecturers by themselves.
- If there are questions from the audience, please make a clear signal if the microphone shall be muted.
- Our goal is to record the lecture, but it is no guarantee that each lecture will be recorded.



Modeling Network Flows . . .


Road Traffic


Data Traffic

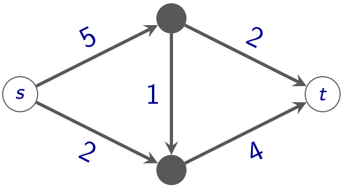

Energy Transport

... and much more.

Network

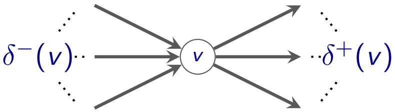
Network

- $G = (V, A, c)$: weighted Digraph
- $s \in V$: source
- $t \in V$: sink
- $c : A \to \mathbb{R}_+$: capacities



Recap: For vertices $v \in V$:

- $\delta^-(v) := \{(u, v) \in A\}$ set of incoming arcs in v
- $\delta^+(v) := \{(v, u) \in A\}$ set of outgoing arcs from v

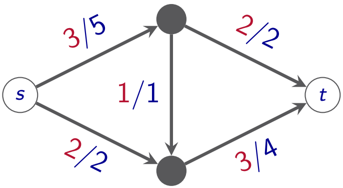


Maximum Flows and Minimum Cuts

s-t-Flow

Networksent

- $G = (V, A, c)$: weighted Digraph
- $s \in V$: source
- $t \in V$: sink
- $c : A \to \mathbb{R}_+$: capacities



A **feasible** s-t**-flow** is a function $f : A \to \mathbb{R}_+$, $a \in A$, which has the following properties:

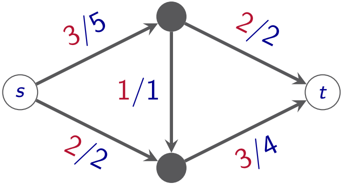
- Capacity constraint: $0 \leq f(a) \leq c(a), \qquad \forall a \in A$
- Flow conservation:

$$\sum_{a \in \delta^-(v)} f(a) = \sum_{a \in \delta^+(v)} f(a), \qquad \forall v \in V \setminus \{s, t\}$$

Max s-t-flow Problem

The excess of a flow f in $v \in V$ is

$$ex_f(v) := \sum_{a \in \delta^+(v)} f(a) - \sum_{a \in \delta^-(v)} f(a).$$



The value of a flow is $val(f) = ex_f(s)$.

Because of flow conservation we have $ex_f(s) = -ex_f(t)$, and in every other vertex $v \neq s, t$ we have $ex_f(v) = 0$.

Max s-t-flow Problem

Given a network $N = (V, A, c, s, t)$. Find a feasible s-t-flow f of maximum value $val(f)$.

Max s-t-flow Problem

Remark: If $f(a) \in \mathbb{N}$, then f is an integer flow. In general, $f(a)$ does not need to be integer (fractional).

$\longrightarrow$ Does not matter for transportation of gas, water, electricity, but is important for routing trucks.

Theorem

The Max s-t-flow Problem with integer capacities $c(a) \in \mathbb{N}$, $a \in A$, always has an integer optimal solution.

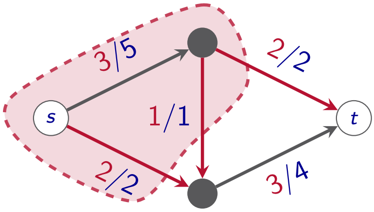
Proof. (later)
Follows from the analysis of the algorithms.

s-t-Cut

For $U \subseteq V$ let $\delta^+(U) := \{(u,v) \in A \mid u \in U, v \in V \setminus U\}$.

Let $U \subseteq V$ with $s \in U$ and $t \notin U$, then $C := \delta^+(U)$ is an s-t-Cut. The cut capacity is

$$\operatorname{cap}(C) := \sum_{a \in C} c(a).$$



Min s-t-cut Problem

Given a network, find an s-t-cut C of minimum capacity $\operatorname{cap}(C)$.

The cut capacity is obviously an upper bound on the value of a maximum flow.

Application for cuts and flows

Maximum Flows

- Delivery capacities of gas, water, electricity, oil, ...
- Production capacity of production lines
- Absorption capacity of waste water pipes
- Delivery capacities of logistics networks
- Network capacities of telecommunication networks

- typical subproblem for similar but more complicated problems

Minimum Cuts

- Analysis of bottlenecks in the above networks
- Robustness/susceptibility to disruption of the above networks

Flow value $\leq$ cut capacity

Theorem

Let $N = (V, A, c, s, t)$ be a network. Let f be an s-t-flow and $C \subseteq A$ an s-t-cut. Then it holds

$$val(f) \leq \text{cap}(C).$$

Proof. Consider cut $C = \delta^+(U)$ for $U \subset V$ with $s \in U$, $t \notin U$.

- From definition: $val(f) = ex_f(s) = \sum_{a \in \delta^+(s)} f(a) - \sum_{a \in \delta^-(s)} f(a)$.
- Due to flow conservation we have: $\sum_{v \in U \setminus \{s\}} ex_f(v) = \sum_{v \in U \setminus \{s\}} \left(\sum_{a \in \delta^+(v)} f(a) - \sum_{a \in \delta^-(v)} f(a) \right) = 0$
- Therefore,

$$val(f) = ex_f(s) = \sum_{v \in U} ex_f(v) = \sum_{v \in U} \left(\sum_{a \in \delta^+(v)} f(a) - \sum_{a \in \delta^-(v)} f(a) \right)$$
$$= \sum_{a \in \delta^+(U)} f(a) - \sum_{a \in \delta^-(U)} f(a) \leq \text{cap}(C) \quad \square$$

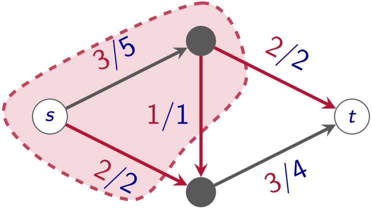
Universität Bremen

Max-Flow = Min-Cut

In fact, the much stronger statement of equality holds...

Theorem (Max-Flow Min-Cut Theorem)

Given a network $N = (V, A, c, s, t)$ with capacities $c(a) \geq 0$, $a \in A$. Then the value of a maximum s-t-flow is equal to the minimum s-t-cut capacity.



max flow = min cut

Proof. later; first recap other important tools.

The residual graph

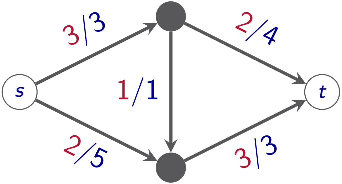
Construction

- Introduce backwards arc: $\overleftarrow{A} := \{\overleftarrow{a} \, : \, a \in A\}$
- Residual capacities for $a \in \overleftrightarrow{A} := A \cup \overleftarrow{A}$ are defined as

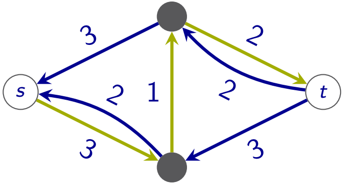
$$\bar{c}_f(a) := \begin{cases} c(a) - f(a) & \text{falls } a \in A \text{ (forward arc)} \\ f(a) & \text{falls } a \in \overleftarrow{A} \text{ (backward arc)} \end{cases}$$

Arcs with $\bar{c}_f(a) = 0$ are deleted.

Residual graph $D_f = (V, A_f)$: $A_f := \{a \in \overleftrightarrow{A} \, : \, \bar{c}_f(a) > 0\}$

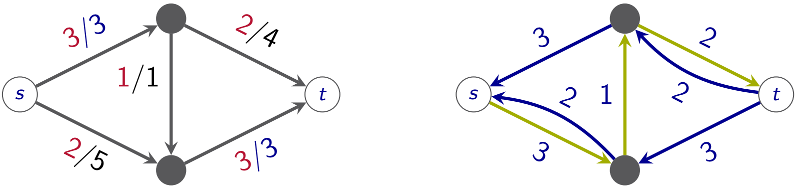


Digraph D with flow f



Residual graph D_f

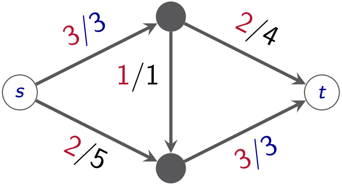
Augmenting Paths



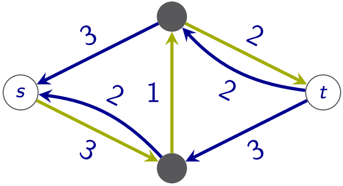
Digraph D with flow f Residual graph D_f

- f-augmenting Path P: s-t-path in D_f

Augmenting Paths



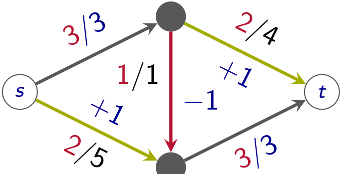
Digraph D with flow f



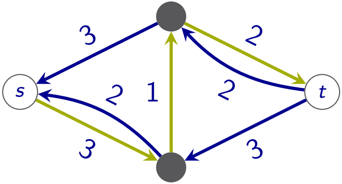
Residual graph D_f

- f-augmenting Path P: s-t-path in D_f
- Bottleneck capacity of P: $\gamma := \min_{a \in P} \bar{c}_f(a)$

Augmenting Paths



Digraph D with flow f

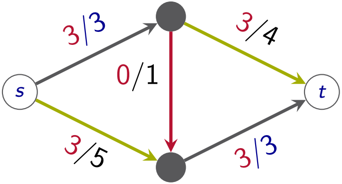


Residual graph D_f

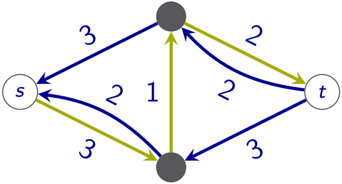
- f-augmenting Path P: s-t-path in D_f
- Bottleneck capacity of P: $\gamma := \min_{a \in P} \bar{c}_f(a)$
- Increase of flow f along P by γ gives a flow f' in D:

$$f'(a) := \begin{cases} f(a) + \gamma & \text{if } a \in P \\ f(a) - \gamma & \text{if } \overleftarrow{a} \in P \\ f(a) & \text{otherwise} \end{cases}$$

Augmenting Paths



Digraph D with flow f

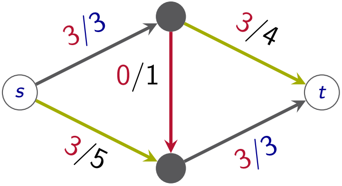


Residual graph D_f

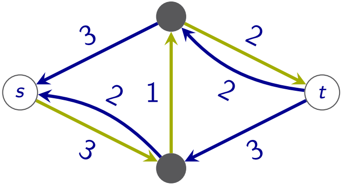
- f-augmenting Path P: s-t-path in D_f
- Bottleneck capacity of P: $\gamma := \min_{a \in P} \bar{c}_f(a)$
- Increase of flow f along P by γ gives a flow f' in D:

$$f'(a) := \begin{cases} f(a) + \gamma & \text{if } a \in P \\ f(a) - \gamma & \text{if } \overleftarrow{a} \in P \\ f(a) & \text{otherwise} \end{cases}$$

Augmenting Paths



Digraph D with flow f



Residual graph D_f

- f-augmenting Path P: s-t-path in D_f
- Bottleneck capacity of P: $\gamma := \min_{a \in P} \bar{c}_f(a)$
- Increase of flow f along P by γ gives a flow f' in D:

$$f'(a) := \begin{cases} f(a) + \gamma & \text{if } a \in P \\ f(a) - \gamma & \text{if } \overleftarrow{a} \in P \\ f(a) & \text{otherwise} \end{cases}$$

Theorem

f' ist a feasible s-t-flow in D with flow value $val(f') = val(f) + \gamma$.

Optimality criteria

Theorem

Let $N = (V, A, c, s, t)$ be a network and f a feasible s-t-flow. Then we have:

f maximum $\Leftrightarrow \nexists$ f-augmenting s-t-path in the residual graph.

Proof.

"$\Rightarrow$": clear, since an f-augmenting s-t-path would increase the flow value.
"$\Leftarrow$": Let $U \subset V$ be the set of vertices that are reachable from s in the residual graph via directed paths. $\delta^+(U) = C$ is an s-t-cut, because $s \in U$ and $t \notin U$.

$$\Rightarrow val(f) = ex_f(s) \leq \text{cap}(C). \qquad (*)$$

- For arcs $a = (x, y) \in \delta^+(U)$ in N it holds $f(a) = c(a)$ (otherwise $y \in U$).
- For arcs $a = (u, v) \in \delta^-(U)$ in N it holds $f(a) = 0$ (otherwise $u \in U$).

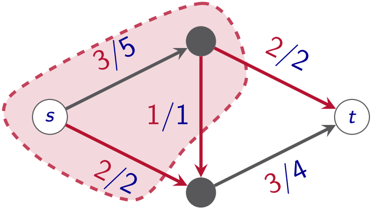
$$val(f) = ex_f(s) = \sum_{a \in \delta^+(U)} f(a) - \sum_{a \in \delta^-(U)} f(a) = \sum_{a \in \delta^+(U)} c(a) = \text{cap}(C)$$

With $(*)$ it follows that f is maximum. □

Max-Flow Min-Cut Theorem

Theorem (Max-Flow Min-Cut Theorem)

Given a network $N = (V, A, c, s, t)$ with capacities $c(a) \geq 0$, $a \in A$. Then the value of a maximum s-t-flow is equal to the value of a minimum s-t-cut capacity.



max flow = min cut

Proof. Let f be a maximum flow. Then due to the previous theorem there is no f-augmenting s-t-path in the residual graph. Construct cut C as in the proof of the previous theorem. Then $val(f) = \text{cap}(C)$.

Algorithms for the Max-flow Problem

Augmenting Path Algorithm

Algorithm (Ford & Fulkerson, 1957)

1. $f := 0$
2. As long as there exists an f-augmenting path P in the residual graph, increase the flow f along P by $\min_{a \in P} \bar{c}_f(a)$.
3. Return f.

…Example at the board.

Augmenting Path Algorithm

Algorithm (Ford & Fulkerson, 1957)

1. $f := 0$
2. As long as there exists an f-augmenting path P in the residual graph, increase the flow f along P by $\min_{a \in P} \bar{c}_f(a)$.
3. Return f.

Theorem

If the arc capacities are integer, then the algorithm computes an integer maximum s-t-flow in running-time $\mathcal{O}(m \cdot M)$, where M is the value of a maximum flow.

Proof Idea.

- Flow is maximum if and only if there is no augmenting path.
- Algorithm increases/decreases flow on an arc by γ, which is integer for $c(a) \in \mathbb{Z}_+$. $\Rightarrow$ flow integer
- In every iteration is $\gamma > 0$ and the flow value increases. $\Rightarrow$ Flow is maximum when algorithm terminates.

Running time of Augmenting Path Algorithm

Running time Ford-Fulkerson $\mathcal{O}(|A| \cdot M)$, with M value of maximum flow

M can be large, roughly $|A| \cdot c_{\max}$ where $c_{\max} := \max_{a \in A} c(a)$. This is only pseudo-polynomial in input size.

...see also the exercise.

Theorem (Edmonds and Karp, 1969)

The variant of the Ford-Fulkerson Algorithm, in which the selected augmenting path is always a shortest s-t path in the residual graph (w.r.t. number of arcs), has a running time of $\mathcal{O}(m^2 \cdot n)$.

$=$ polynomial running time

Proof. At the board.

Recap

- Networks and maximum flows in networks
- Residual graphs and augmenting paths
- Ford-Fulkerson algorithm (pseudopolynomial running time)
- Edmonds-Karp algorithm: special path selection, polynomial
- Max-flow min-cut theorem
  - Question: Given a max flow, how to find a minimum cut?