Technische Universität Braunschweig

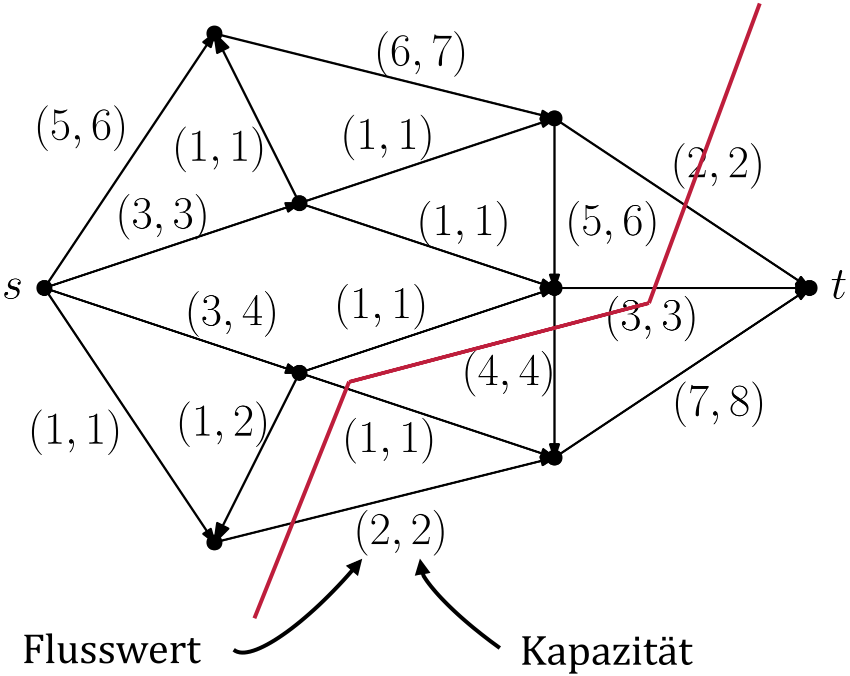
Netzwerkalgorithmen – Vorlesung #8

Arne Schmidt

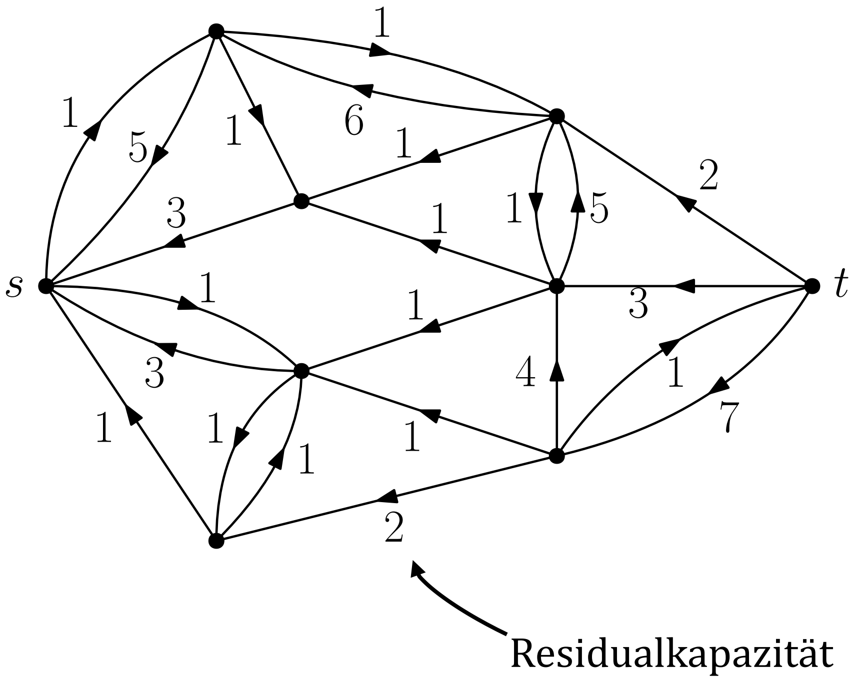
Wiederholung

Wiederholung

Netzwerk



Residualgraph



4.2 Berechnung von maximalen Flüssen

Algorithmus von Ford-Fulkerson

Algorithmus 4.8

Eingabe:

Netzwerk $N = (D, u, s, t)$

Ausgabe:

st-Fluss f mit maximalem Wert

1. Function FordFulkerson(N)
2. $f(e) := 0, \forall e \in A$
3. Bestimme Residualgraph D_f und Residualkapazitäten u_f.
4. Bestimme st-Pfad P in D_f; falls keiner existiert **return** f.
5. Berechne $\gamma := \min_{e \in P} \left(u_f(e)\right)$.
6. Augmentiere f entlang P.
7. Gehe zu Zeile 3.

Diese Woche – Zum Überlegen!

Ist der Algorithmus korrekt?

Wie lange braucht der Algorithmus im Worst-Case?

Was passiert bei…

- Ganzzahligen Kapazitäten?
- Rationalen Kapazitäten?
- Irrationalen Kapazitäten?

Korrektheit

Lemma 4.9
Terminiert der Algorithmus von Ford-Fulkerson, dann ist f maximal.

Beweis:
Algorithmus terminiert nur, wenn kein st-Pfad in D_f existiert.
Nach Lemma 4.6 ist f maximal.

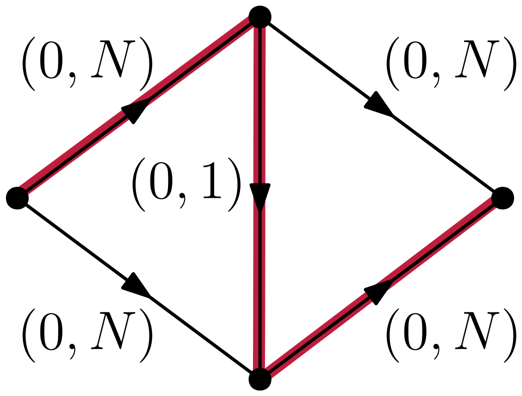
Terminiertheit

Lemma 4.10
Der Algorithmus von Ford-Fulkerson terminiert immer für rationale Kapazitäten.

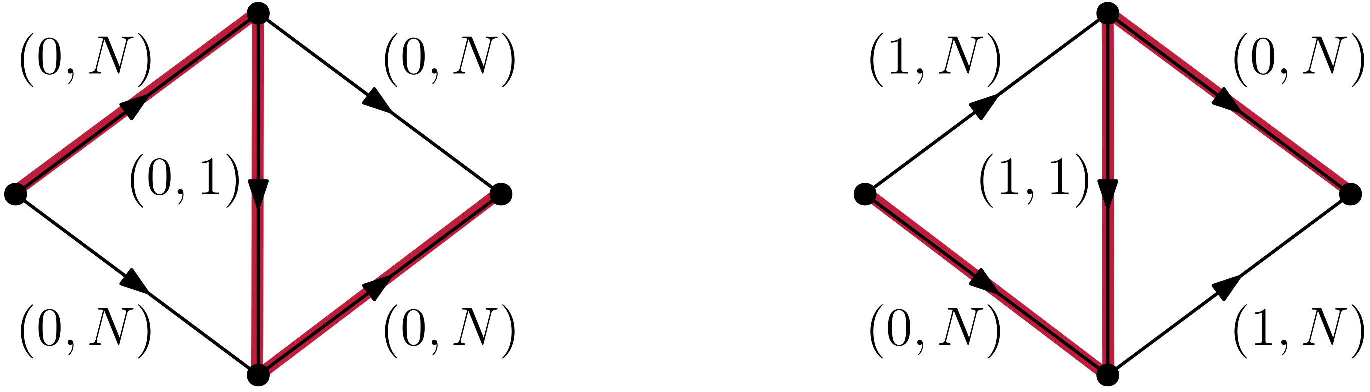
Beweis:
Falls Kapazitäten nicht ganzzahlig:
Betrachte Kapazitäten $u_e = \frac{a_e}{b_e}$ und bestimme $u'_e \coloneqq u_e \cdot \mathrm{kgV}_{e' \in A}(b_{e'})$

Unter ganzzahligen Gewichten:
Jeder st-Pfad augmentiert f um mindestens 1.
$\Rightarrow$ maximal $F = \max_{st-\text{Fluss } f} f$ viele Iterationen.

Worst-Case Laufzeit

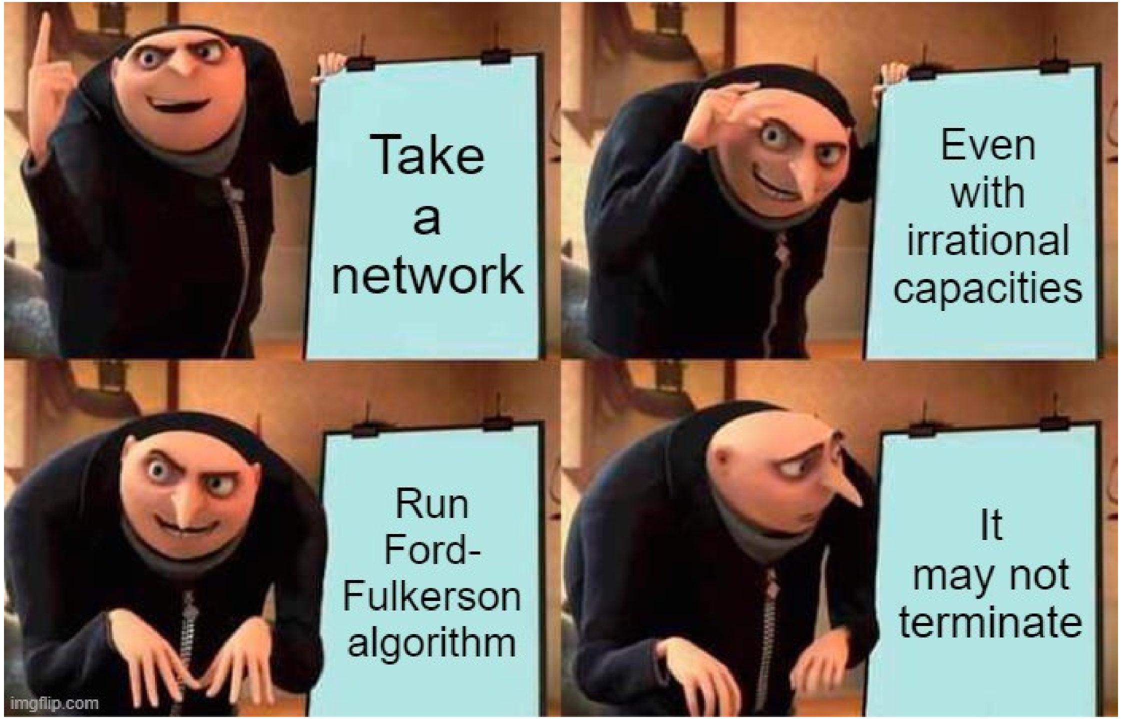


Worst-Case Laufzeit



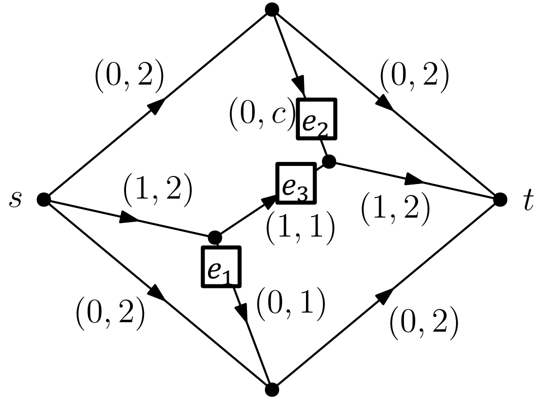
$2N$ Iterationen!

Worst Worst-Case



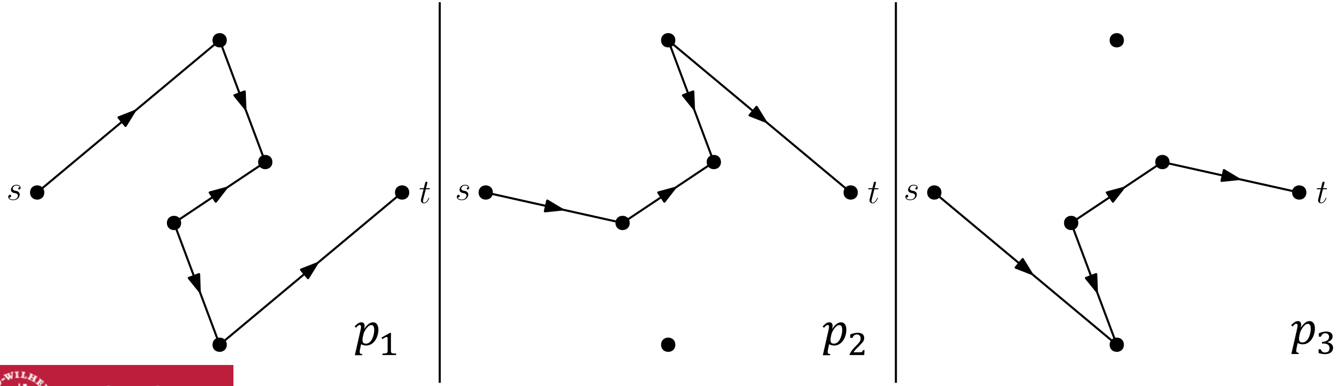
Irrationale Kapazitäten

$$\boxed{c := \frac{\sqrt{5}-1}{2}}$$



Residualkapaizitäten von (e_1, e_2, e_3) nach Anwenden von p_1, p_2, p_1, p_3.

Iteration 0: $(1, c, 0)$
Iteration 1: $(1-c, 2c-1, 0)$
Iteration 2: $(2-3c, 5c-3, 0)$
Iteration n: $(a, b, 0)$
Iteration $n+1$: $(a-b, 2b-a, 0)$

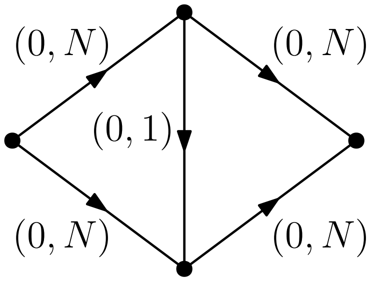


Irrationale Kapazitäten

Problem: Das Wählen beliebiger Pfade.

Lösungsideen:

- Pfad mit bester Verbesserung?
- Pfad mit vielen Kanten?
- Pfad mit wenig Kanten?



+

Finden langer Pfade ist schwer!

Beste Verbesserung

Problem 4.11: Widest Path Problem

Gegeben:

Graph $D = (V, A)$ mit Kostenfunktion $c: A \to \mathbb{R}$ und Knoten $s \in V$.

Gesucht:

Für jeden Knoten $v \in V$ einen sv-Pfad P mit $\min_{e \in P} c(e)$ maximal.

Vgl: Präsenzaufgabe auf Blatt 2 $\Rightarrow$ Lässt sich analog mit modifiziertem Dijkstra lösen.
Laufzeit: $O(m + n \log n)$

Zu zeigen: Es gibt immer einen Pfad, der eine „große“ Verbesserung bringt.

Flusszerlegung

Satz 4.12
Sei $N = (D, u, s, t)$ ein Netzwerk und f ein st-Fluss in N.
Dann gibt es eine Menge
- $\mathcal{P}$ von st-Pfaden
- $\mathcal{C}$ von Kreisen

sodass folgendes gilt:

a) $f(e) = \sum_{\substack{p \in \mathcal{P} \cup \mathcal{C} \\ e \in p}} w(p)$

b) $Wert(f) = \sum_{p \in \mathcal{P}} w(p)$

c) $|\mathcal{P}| + |\mathcal{C}| \leq |E|$

Dabei ist $w(p)$ der Flusswert des Pfades bzw. Kreises.

Zeige nun: Die Konstruktion erfüllt Bedingungen a) bis c) $\Rightarrow$ Hausaufgabe!

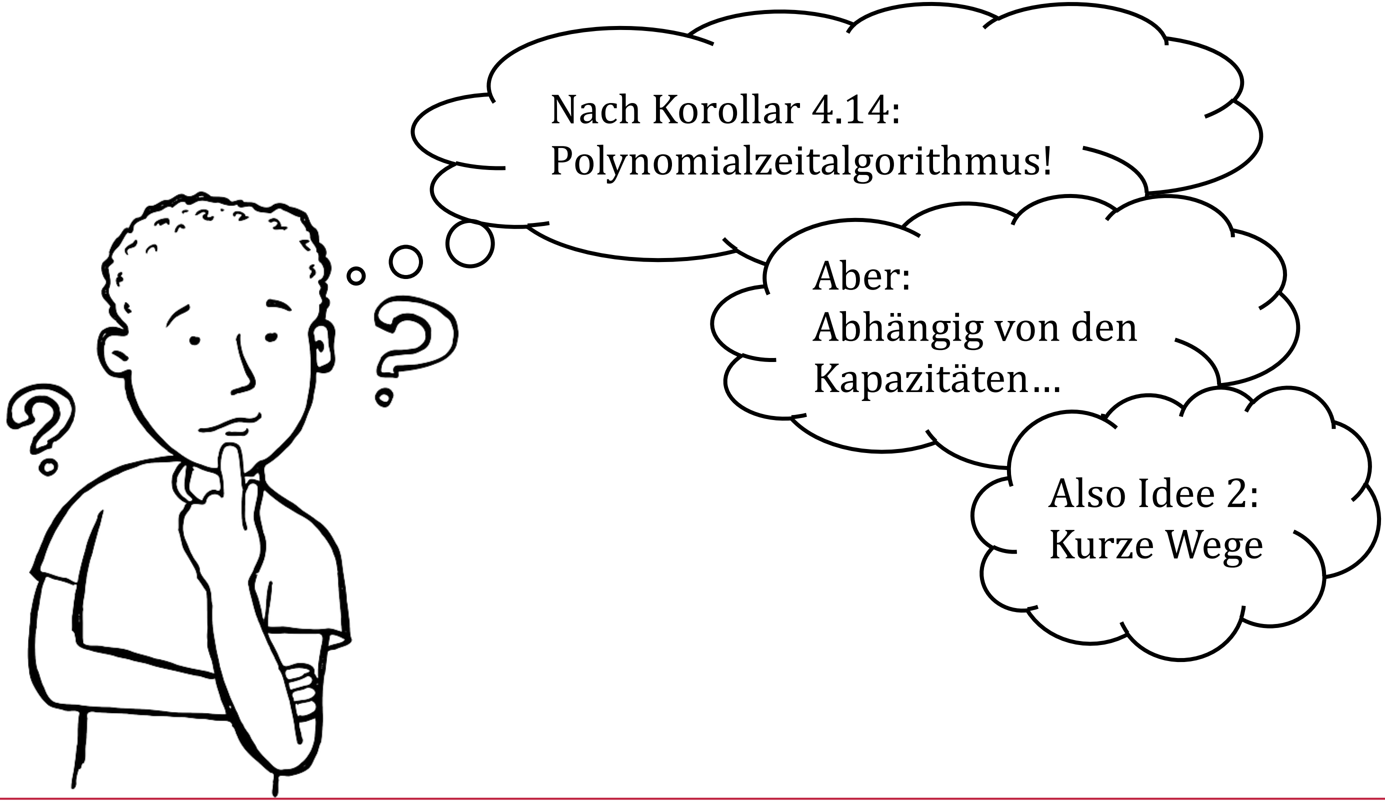
Pfade mit großer Verbesserung

Lemma 4.13
Sei $N = (D, u, s, t)$ ein Netzwerk mit maximalem Flusswert opt.
Dann gibt es einen augmentierenden Pfad mit Kapazität $\frac{opt}{|E|}$.

Beweis:
Sei f ein maximaler Fluss.
Nach Satz 4.12 existieren Pfade $P_1, \dots, P_k$ mit
$Wert(f) = \sum_{i=1}^{k} w(P_i)$ und $k \leq |E|$.
$\Rightarrow$ Es ex. Pfad P_i mit $w(P_i) \geq \frac{Wert(f)}{|E|}$

Korollar 4.14
Für ein Netzwerk $N = (D, u, s, t)$ mit $u: A \to \mathbb{Q}^+$ maximalem Flusswert f^* benötigt der Algorithmus von Ford-Fulkerson maximal $m \log f^*$ Iterationen, wenn immer ein Pfad mit maximaler Verbesserung gewählt wird.

Kurze Wege



Nach Korollar 4.14:
Polynomialzeitalgorithmus!
Aber:
Abhängig von den
Kapazitäten…
Also Idee 2:
Kurze Wege

Algorithmus von Edmonds-Karp

Algorithmus 4.15

Eingabe:

Netzwerk $N = (D, u, s, t)$

Ausgabe:

st-Fluss f mit maximalem Wert

1. Function $\textsc{EdmondsKarp}(N)$
2. $f(e) := 0, \forall e \in A$
3. Bestimme Residualgraph D_f und Residualkapazitäten u_f.
4. Bestimme **kürzesten** st-Pfad P in D_f; falls keiner existiert **return** f.
5. Berechne $\gamma := \min_{e \in P} \left(u_f(e) \right)$.
6. Augmentiere f entlang P.
7. Gehe zu Zeile 3.

Analyse Edmonds-Karp

Lemma 4.16
Sei $N = (D, u, s, t)$ ein Netzwerk und $f_1, \dots, f_k$ eine Folge von Flüssen, wobei f_{i+1} aus f_i durch Augmentieren entlang eines kürzesten f_i-augmentierenden Pfades P_i entsteht.
Dann gilt:

a) Für alle $1 \leq i < k$ gilt $|E(P_i)| \leq |E(P_{i+1})|$
b) Falls P_i und P_j mit $i < j$ ein Paar entgegengesetzter Kanten enthält, dann gilt $|E(P_i)| \leq \left|E\left(P_j\right)\right| - 2$

Nächste Woche

Lemma 4.17
Für ein Netzwerk $N = (D, u, s, t)$ mit $u: A \to \mathbb{R}^+$ terminiert der Algorithmus von Edmonds-Karp nach maximal $\frac{mn}{2}$ Iterationen.

Satz 4.18
Der Algorithmus von Edmonds-Karp löst Problem 4.2 (MaxFlow) in Zeit $O(nm^2)$.