Effiziente Algorithmen (SS2015) Kapitel 2 Weitere Flüsse

Walter Unger

Lehrstuhl für Informatik 1

7:05 Uhr, den 6. November 2018

Spezielle Flüsse (Mindestfluss)

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

2 Inhaltsverzeichnis

Inhalt I

Spezielle Flüsse (Mindestfluss)

Mit Mindestfluss

2 Spezielle Flüsse (Alternativen) Mit Alternativen

Flüsse mit Kostenfunktion

Einleitung

Idee

Algorithmus

Verbesserung der Laufzeit

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Das Flussproblem mit Mindestfluss

Definition (Flussproblem mit Mindestfluss)

Das Flussproblem mit Mindestfluss

Definition (Flussproblem mit Mindestfluss)

Eingabe: G = (V, E, s, t, c, c') mit:

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Das Flussproblem mit Mindestfluss

Definition (Flussproblem mit Mindestfluss)

Eingabe:
$$G = (V, E, s, t, c, c')$$
 mit:

• (V, E) ist ein gerichteter Graph (n = |V|, m = |E|)

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Das Flussproblem mit Mindestfluss

Definition (Flussproblem mit Mindestfluss)

Eingabe:
$$G = (V, E, s, t, c, c')$$
 mit:

- (V, E) ist ein gerichteter Graph (n = |V|, m = |E|)
- $s, t \in V \text{ mit } s \neq t$

Das Flussproblem mit Mindestfluss

Definition (Flussproblem mit Mindestfluss)

Eingabe: G = (V, E, s, t, c, c') mit:

- (V, E) ist ein gerichteter Graph (n = |V|, m = |E|)
- $s, t \in V$ mit $s \neq t$
- \circ $c: F \mapsto \mathbb{N}^+$

Das Flussproblem mit Mindestfluss

Definition (Flussproblem mit Mindestfluss)

Eingabe: G = (V, E, s, t, c, c') mit:

- (V, E) ist ein gerichteter Graph (n = |V|, m = |E|)
- $s, t \in V \text{ mit } s \neq t$
- $c: E \mapsto \mathbb{N}^+$
- $c': E \mapsto \mathbb{N}^+$

Das Flussproblem mit Mindestfluss

Definition (Flussproblem mit Mindestfluss)

Eingabe: G = (V, E, s, t, c, c') mit:

•
$$(V, E)$$
 ist ein gerichteter Graph $(n = |V|, m = |E|)$

• $s, t \in V \text{ mit } s \neq t$

•
$$c: E \mapsto \mathbb{N}^+$$

• $c': E \mapsto \mathbb{N}^+$

Ausgabe:
$$f: E \mapsto \mathbb{R}_0^+$$
 mit:

Das Flussproblem mit Mindestfluss

Definition (Flussproblem mit Mindestfluss)

Eingabe: G = (V, E, s, t, c, c') mit:

- (V, E) ist ein gerichteter Graph (n = |V|, m = |E|)
- $s, t \in V \text{ mit } s \neq t$
- $c: E \mapsto \mathbb{N}^+$
- $c': E \mapsto \mathbb{N}^+$

Ausgabe: $f: E \mapsto \mathbb{R}_0^+$ mit:

• $\forall e : c'(e) \leqslant f(e) \leqslant c(e)$

Das Flussproblem mit Mindestfluss

Definition (Flussproblem mit Mindestfluss)

Eingabe:
$$G = (V, E, s, t, c, c')$$
 mit:

- (V, E) ist ein gerichteter Graph (n = |V|, m = |E|)
- $s, t \in V$ mit $s \neq t$
- \bullet $c \cdot F \mapsto \mathbb{N}^+$
- \circ c' · F $\mapsto \mathbb{N}^+$

Ausgabe: $f: E \mapsto \mathbb{R}_0^+$ mit:

- $\forall e : c'(e) \leq f(e) \leq c(e)$
- $\forall v \in V \setminus \{s,t\} : \sum_{(a,v) \in E} f((a,v)) = \sum_{(v,a) \in E} f((v,a))$

Das Flussproblem mit Mindestfluss

Definition (Flussproblem mit Mindestfluss)

Eingabe:
$$G = (V, E, s, t, c, c')$$
 mit:

- (V, E) ist ein gerichteter Graph (n = |V|, m = |E|)
- $s, t \in V$ mit $s \neq t$ \bullet $c \cdot F \mapsto \mathbb{N}^+$
- \circ c' · F $\mapsto \mathbb{N}^+$
- Ausgabe: $f: E \mapsto \mathbb{R}_0^+$ mit:

•
$$\forall e : c'(e) \leqslant f(e) \leqslant c(e)$$

$$\forall c : c (c) \leqslant r(c) \leqslant c(c)$$

•
$$\forall v \in V \setminus \{s, t\} : \sum_{(a, v) \in E} f((a, v)) = \sum_{(v, a) \in E} f((v, a))$$

Ziel: Bestimme, ob es so einen Fluss gibt. Falls ja, dann maximiere $w(f) = \sum_{(s,v) \in E} f((s,v)).$

Spezielle Flüsse (Mindestfluss)

0 0 0 0 0 0 0 0 0

2:2 Mit Mindestfluss

Spezielle Flüsse (Alternativen)

.

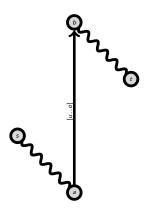
Lösbarkeit

Es muss nicht immer eine Lösung geben. Hier ein einfaches Beispiel:

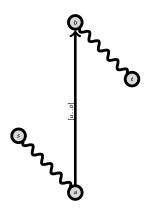


Spezielle Flüsse (Mindestfluss) Spezielle Flüsse (Alternativen) Flüsse mit Kostenfunktion 00000000 2:3 Mit Mindestfluss 1/5 Walter Unger 6.11.2018 7:05 SS2015 RWTH Idee

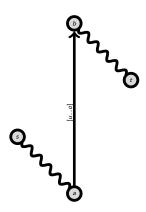
2:3 Mit Mindestfluss 2/5



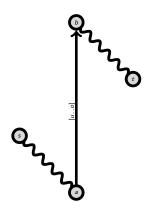
ldoo

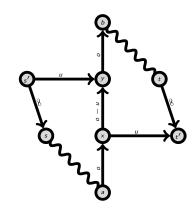


2:3 Mit Mindestriuss 4/5



Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH





00000000 2:4 Mit Mindestfluss 1/11

Spezielle Flüsse (Mindestfluss)

Spezielle Flüsse (Alternativen) Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

00000000 2:4 Mit Mindestfluss 2/11

Spezielle Flüsse (Mindestfluss)

Spezielle Flüsse (Alternativen) Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Verfahren

• Erzeuge aus G = (V, E, s, t, c, c') einen neuen Graphen G' = (V', E', s', t', c'').

- Erzeuge aus G = (V, E, s, t, c, c') einen neuen Graphen G' = (V', E', s', t', c'').
- Füge neue Quelle s' und neue Senke t' hinzu.

2:4 Mit Mindestfluss 5/11

- Erzeuge aus G = (V, E, s, t, c, c') einen neuen Graphen G' = (V', E', s', t', c'').
- Füge neue Quelle s' und neue Senke t' hinzu.
- Ersetze jede Kante (v, w) durch einen Weg der Länge 3:

- Erzeuge aus G = (V, E, s, t, c, c') einen neuen Graphen G' = (V', E', s', t', c'').
- ullet Füge neue Quelle s' und neue Senke t' hinzu.
- Ersetze jede Kante (v, w) durch einen Weg der Länge 3:
 - für jede Kante (v, w) erzeuge zwei neue Knoten x, y und setze:

2:4 Mit Mindestfluss 7/11

rnativen) Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

- Erzeuge aus G = (V, E, s, t, c, c') einen neuen Graphen G' = (V', E', s', t', c'').
- Füge neue Quelle s' und neue Senke t' hinzu.
- Ersetze jede Kante (v, w) durch einen Weg der Länge 3:
 - für jede Kante (v, w) erzeuge zwei neue Knoten x, y und setze:
 - c''(v,x) = c(v,w) und c''(y,w) = c(v,w).

2:4 Mit Mindestfluss 8/11

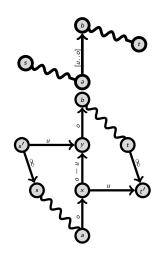
- Erzeuge aus G = (V, E, s, t, c, c') einen neuen Graphen G' = (V', E', s', t', c'').
- Füge neue Quelle s' und neue Senke t' hinzu.
- Ersetze jede Kante (v, w) durch einen Weg der Länge 3:
 - für jede Kante (v, w) erzeuge zwei neue
 Knoten x, y und setze:
 c"(v, v) = c(v, w) und c"(v, w) = c(v, w)
 - c''(v,x) = c(v,w) und c''(y,w) = c(v,w).
 - c''(x,y) = c(v,w) c'(v,w).

Verfahr<u>en</u>

- Erzeuge aus G = (V, E, s, t, c, c') einen neuen Graphen G' = (V', E', s', t', c'').
- Füge neue Quelle s' und neue Senke t' hinzu.
- Ersetze jede Kante (v, w) durch einen Weg der Länge 3:
 - für jede Kante (v, w) erzeuge zwei neue Knoten x, v und setze:
 - c''(v,x) = c(v,w) und c''(v,w) = c(v,w).
 - c''(x, y) = c(v, w) c'(v, w).
 - c''(s', y) = c'(v, w) und c''(x, t') = c'(y, w).

- Erzeuge aus G = (V, E, s, t, c, c') einen neuen Graphen G' = (V', E', s', t', c'').
- ullet Füge neue Quelle s' und neue Senke t' hinzu.
- Ersetze jede Kante (v, w) durch einen Weg der Länge 3:
 - für jede Kante (v, w) erzeuge zwei neue Knoten x, y und setze:
 - c''(v,x) = c(v,w) und c''(y,w) = c(v,w).
 - c''(x,y) = c(v,w) c'(v,w).
 - c''(s', y) = c'(v, w) und c''(x, t') = c'(v, w).
- Setze $c''(t,t') = c''(s',s) = \sum_{e \in E} c(e)$.

- Erzeuge aus G = (V, E, s, t, c, c') einen neuen Graphen G' = (V', E', s', t', c'').
- Füge neue Quelle s' und neue Senke t' hinzu.
- Ersetze jede Kante (v, w) durch einen Weg der Länge 3:
 - für jede Kante (v, w) erzeuge zwei neue Knoten x, y und setze:
 - c''(v,x) = c(v,w) und c''(y,w) = c(v,w).
 - c''(x, y) = c(y, w) c'(y, w).
 - c''(s', y) = c'(v, w) und c''(x, t') = c'(v, w).
- Setze $c''(t,t') = c''(s',s) = \sum_{e \in E} c(e)$.



Spezielle Flüsse (Mindestfluss) Spezielle Flüsse (Alternativen) Flüsse mit Kostenfunktion 000000000 2:5 Mit Mindestfluss 1/8 Walter Unger 6.11.2018 7:05 SS2015 RWTH Aussage

Spezielle Flüsse (Alternativen)

Walter Unger 6.11.2018 7:05 SS2015 **RWTH**

Aussage

Lemma

Es gibt in G einen korrekten Fluss, der die Mindestflussbedingung erfüllt genau dann, wenn es in G' einen maximalen Fluss gibt, der alle Kanten der Form (s',y) und (x,t') saturiert.

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Aussage

Lemma

Es gibt in G einen korrekten Fluss, der die Mindestflussbedingung erfüllt genau dann, wenn es in G' einen maximalen Fluss gibt, der alle Kanten der Form (s', y) und (x, t') saturiert.

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Aussage

Lemma

Es gibt in G einen korrekten Fluss, der die Mindestflussbedingung erfüllt genau dann, wenn es in G' einen maximalen Fluss gibt, der alle Kanten der Form (s', y) und (x, t') saturiert.

Beachte: x und y sind neu eingefügte Knoten, also nicht s oder t.

Beweis:

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Aussage

Lemma

Es gibt in G einen korrekten Fluss, der die Mindestflussbedingung erfüllt genau dann, wenn es in G' einen maximalen Fluss gibt, der alle Kanten der Form (s', y) und (x, t') saturiert.

Beachte: x und y sind neu eingefügte Knoten, also nicht s oder t.

Beweis:

Zeige: ⇒

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Aussage

Lemma

Es gibt in G einen korrekten Fluss, der die Mindestflussbedingung erfüllt genau dann, wenn es in G' einen maximalen Fluss gibt, der alle Kanten der Form (s', y) und (x, t') saturiert.

Beachte: x und y sind neu eingefügte Knoten, also nicht s oder t.

Beweis:

- Zeige: ⇒
- Zeige: ←

Aussage

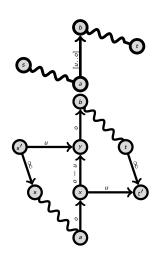
Lemma

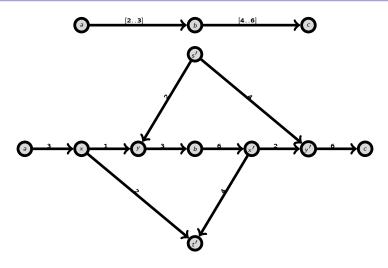
Es gibt in G einen korrekten Fluss, der die Mindestflussbedingung erfüllt genau dann, wenn es in G' einen maximalen Fluss gibt, der alle Kanten der Form (s', y) und (x, t') saturiert.

Beachte: x und y sind neu eingefügte Knoten, also nicht s oder t.

Beweis:

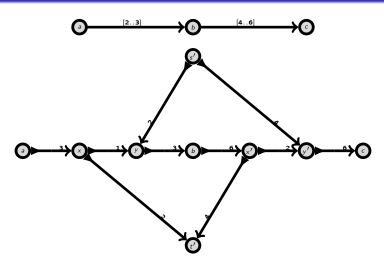
- Zeige: ⇒
- Zeige: ⇐

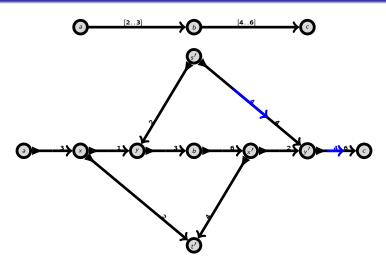


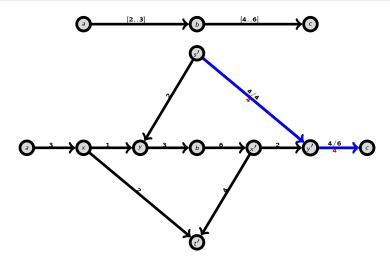


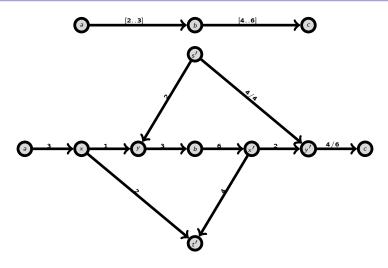
Alternativen) Flüsse mit Kostenfunktion

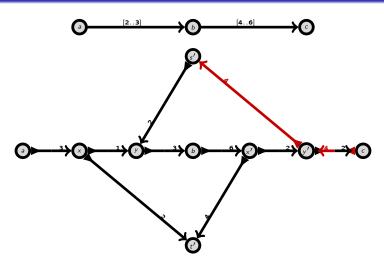
Walter Unger 6.11.2018 7:05 SS2015 RWTH

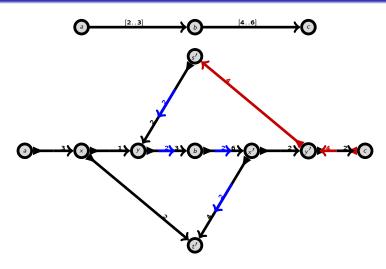








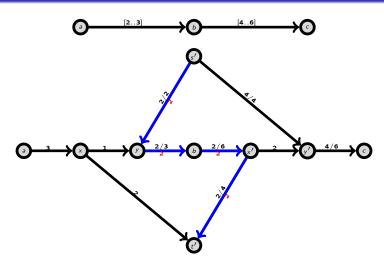


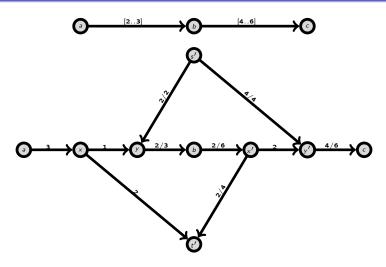


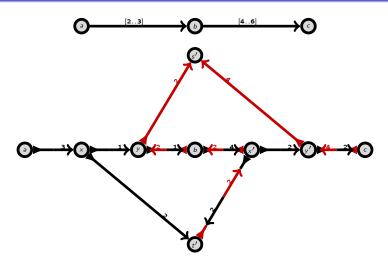
Spezielle Flüsse (Alternativen)

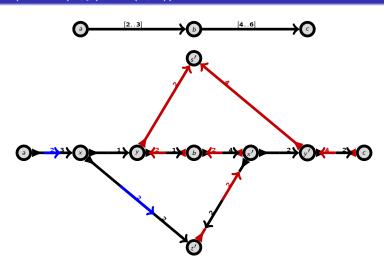
ternativen) Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH



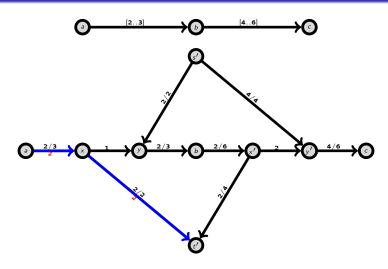


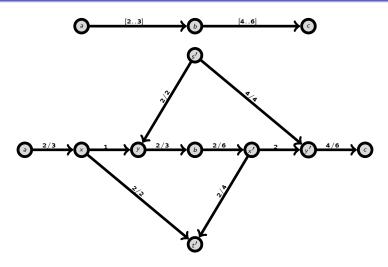


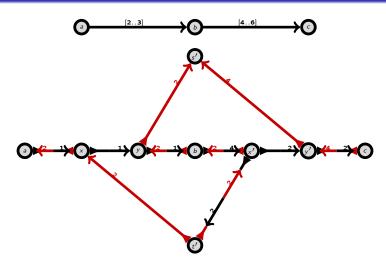


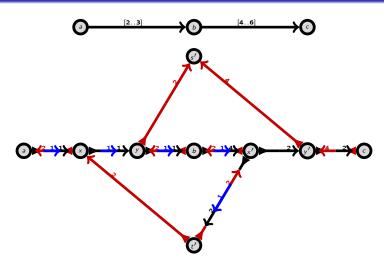
2:6 Mit Mindestfluss 12/17

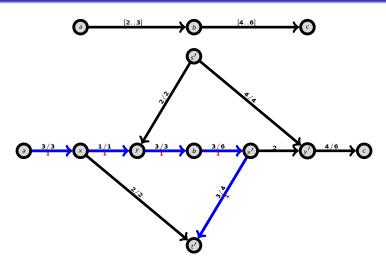
Walter Unger 6.11.20187:05 SS2015 RWTH

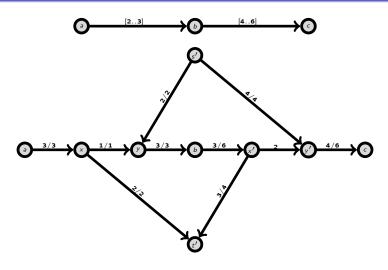












000000000 2:7 Mit Mindestfluss 1/6 $Zeige: \Longrightarrow$

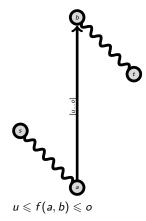
Spezielle Flüsse (Mindestfluss)

Spezielle Flüsse (Alternativen) Flüsse mit Kostenfunktion

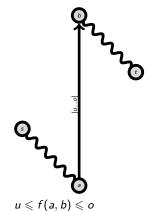
Walter Unger 6.11.2018 7:05 SS2015 RWTH

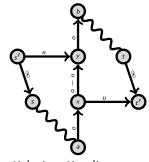
7 -: --- \

 $\mathsf{Zeige:} \Longrightarrow$

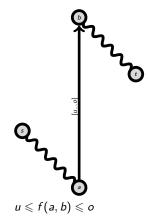


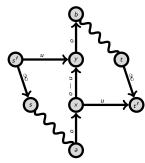
 $Zeige: \Longrightarrow$





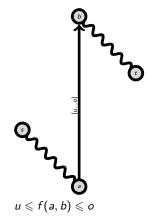
$$f(s',y)=f(x,t')=u$$

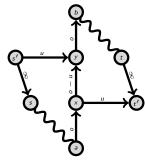




$$f(s', y) = f(x, t') = u$$

 $f(a, x) = f(y, b) = f(a, b)$



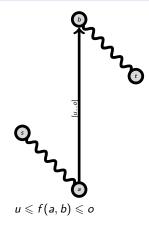


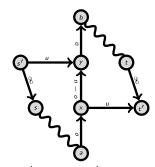
$$f(s', y) = f(x, t') = u$$

 $f(a, x) = f(y, b) = f(a, b)$

7 Mit Mindestfluss 6/6







$$f(s', y) = f(x, t') = u$$

 $f(a, x) = f(y, b) = f(a, b)$
 $f(x, y) = f(a, b) - u$



Spezielle Flüsse (Mindestfluss)

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

00000000 2:8 Mit Mindestfluss 2/6 Zeige: ←

Spezielle Flüsse (Mindestfluss)

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

2:8 Mit Mindestfluss 3/6

Spezielle Flüsse (Alternativen)

Zeige: ←

• Zeige: Wenn es in G' einen maximalen Fluss gibt, der alle Kanten der Form (s', y) und (x, t') saturiert, dann gibt es in G einen korrekten Fluss, der die Mindestflussbedingung erfüllt.

2:8 Mit Mindestfluss 4/6

Zeige: ←

- Zeige: Wenn es in G' einen maximalen Fluss gibt, der alle Kanten der Form (s', y) und (x, t') saturiert, dann gibt es in G einen korrekten Fluss, der die Mindestflussbedingung erfüllt.
- Dann gilt: f(a, x) = f(y, b) für jede ursprüngliche Kante (a, b).

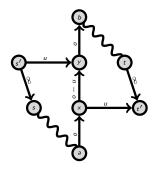
Zeige: ←

- Zeige: Wenn es in G' einen maximalen Fluss gibt, der alle Kanten der Form (s', y) und (x, t') saturiert, dann gibt es in G einen korrekten Fluss, der die Mindestflussbedingung erfüllt.
- Dann gilt: f(a, x) = f(y, b) für jede ursprüngliche Kante (a, b).
- Dann definiert f(a, b) = f(a, x) einen korrekten Fluss auf G.

Zeige: ←

• Zeige: Wenn es in G' einen maximalen Fluss gibt, der alle Kanten der Form (s', y) und (x, t') saturiert, dann gibt es in G einen korrekten Fluss, der die Mindestflussbedingung erfüllt.

- Dann gilt: f(a,x) = f(y,b) für jede ursprüngliche Kante (a,b).
- Dann definiert f(a, b) = f(a, x) einen korrekten Fluss auf G.



Walter Unger 6.11.2018 7:05 SS2015 RWTH

Algorithmus

• Sei f (resp. f') der Fluss auf G (resp. G').

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

- Sei f (resp. f') der Fluss auf G (resp. G').
- Sei weiter f'' der Fluss auf G ohne die untere Schranke c'. Dann gilt, wenn es eine Lösung für G' gibt:

Spezielle Flüsse (Alternativen)

rnativen) Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

- Sei f (resp. f') der Fluss auf G (resp. G').
- Sei weiter f'' der Fluss auf G ohne die untere Schranke c'. Dann gilt, wenn es eine Lösung für G' gibt:
 - f = f'', denn untere Schranken verringern den Fluss auf G nicht mehr.

- Sei f (resp. f') der Fluss auf G (resp. G').
- Sei weiter f'' der Fluss auf G ohne die untere Schranke c'. Dann gilt, wenn es eine Lösung für G' gibt:
 - f = f'', denn untere Schranken verringern den Fluss auf G nicht mehr.
 - $f' = f + \sum_{e \in E(G)} c'(e)$

- Sei f (resp. f') der Fluss auf G (resp. G').
- Sei weiter f'' der Fluss auf G ohne die untere Schranke c'. Dann gilt, wenn es eine Lösung für G' gibt:
 - f = f'', denn untere Schranken verringern den Fluss auf G nicht mehr.
 - $f' = f + \sum_{e \in E(G)} c'(e)$
- Damit haben wir folgendes Verfahren:

- Sei f (resp. f') der Fluss auf G (resp. G').
- Sei weiter f'' der Fluss auf G ohne die untere Schranke c'. Dann gilt, wenn es eine Lösung für G' gibt:
 - f = f'', denn untere Schranken verringern den Fluss auf G nicht mehr.
 - $f' = f + \sum_{e \in E(G)} c'(e)$
- Damit haben wir folgendes Verfahren:
 - 9 Bestimme aus G: G', G'', f, f', f''.

Algorithmus

- Sei f (resp. f') der Fluss auf G (resp. G').
- Sei weiter f'' der Fluss auf G ohne die untere Schranke c'. Dann gilt, wenn es eine Lösung für G' gibt:
 - f = f'', denn untere Schranken verringern den Fluss auf G nicht mehr.
 - $f' = f + \sum_{e \in E(G)} c'(e)$
- Damit haben wir folgendes Verfahren:
 - **1** Bestimme aus G: G', G'', f, f', f''.
 - **3** Bevorzuge auf G' die Kanten der Form (s', y) und (x, t').

Algorithmus

- Sei f (resp. f') der Fluss auf G (resp. G').
- Sei weiter f'' der Fluss auf G ohne die untere Schranke c'. Dann gilt, wenn es eine Lösung für G' gibt:
 - f = f'', denn untere Schranken verringern den Fluss auf G nicht mehr.
 - $f' = f + \sum_{e \in E(G)} c'(e)$
- Damit haben wir folgendes Verfahren:
 - **1** Bestimme aus G: G', G'', f, f', f''.
 - **2** Bevorzuge auf \underline{G}' die Kanten der Form (s', y) und (x, t').
 - **3** Falls $f' < f + \sum_{e \in E(G)} c'(e)$ gilt, so gibt es keine Lösung.

Algorithmus

Mit Mindestfluss 9/9

00000000

- Sei f (resp. f') der Fluss auf G (resp. G').
- Sei weiter f'' der Fluss auf G ohne die untere Schranke c'. Dann gilt, wenn es eine Lösung für G' gibt:
 - f = f'', denn untere Schranken verringern den Fluss auf G nicht mehr.
 - $f' = f + \sum_{e \in F(G)} c'(e)$
- Damit haben wir folgendes Verfahren:
 - **1** Bestimme aus G: G', G'', f, f', f''.
 - 2 Bevorzuge auf G' die Kanten der Form (s', y) und (x, t').
 - **3** Falls $f' < f + \sum_{e \in E(G)} c'(e)$ gilt, so gibt es keine Lösung.
 - 4 Ansonsten bestimme f aus f', d.h. f(a, b) = f(a, x).

Definition (Flussproblem)

Das Flussproblem mit Alternativen

Definition (Flussproblem)

Eingabe: G = (V, E, s, t, c, c') mit:

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Das Flussproblem mit Alternativen

Definition (Flussproblem)

Eingabe:
$$G = (V, E, s, t, c, c')$$
 mit:

• (V, E) ist ein gerichteter Graph (n = |V|, m = |E|)

2:10 Mit Alternativen 4/10

Das Flussproblem mit Alternativen

Definition (Flussproblem)

Eingabe: G = (V, E, s, t, c, c') mit:

- (V, E) ist ein gerichteter Graph (n = |V|, m = |E|)
- $s, t \in V \text{ mit } s \neq t$

Flüsse mit Kostenfunktion

Das Flussproblem mit Alternativen

Definition (Flussproblem)

Eingabe:
$$G = (V, E, s, t, c, c')$$
 mit:

- (V, E) ist ein gerichteter Graph (n = |V|, m = |E|)
- $s, t \in V \text{ mit } s \neq t$
- $c: E \mapsto \mathbb{N}^+$

Definition (Flussproblem)

Eingabe:
$$G = (V, E, s, t, c, c')$$
 mit:

- (V, E) ist ein gerichteter Graph (n = |V|, m = |E|)
- $s, t \in V \text{ mit } s \neq t$
- $c: E \mapsto \mathbb{N}^+$
- $c': E \mapsto \mathbb{N}^+$

Walter Unger 6.11.2018 7:05 SS2015 **RWTH**

Das Flussproblem mit Alternativen

Definition (Flussproblem)

Eingabe:
$$G = (V, E, s, t, c, c')$$
 mit:

- (V, E) ist ein gerichteter Graph (n = |V|, m = |E|)
- $s, t \in V \text{ mit } s \neq t$
- $c: E \mapsto \mathbb{N}^+$
- $\bullet \ c': E \mapsto \mathbb{N}^+$

Ausgabe: $f: E \mapsto \mathbb{R}_0^+$ mit:

Definition (Flussproblem)

Eingabe:
$$G = (V, E, s, t, c, c')$$
 mit:

- (V, E) ist ein gerichteter Graph (n = |V|, m = |E|)
- $s, t \in V$ mit $s \neq t$
- $c: E \mapsto \mathbb{N}^+$
- $c': E \mapsto \mathbb{N}^+$

Ausgabe: $f: E \mapsto \mathbb{R}_0^+$ mit:

•
$$\forall e : c'(e) \leqslant f(e) \leqslant c(e) \text{ oder } f(e) = 0.$$

Definition (Flussproblem)

Eingabe:
$$G = (V, E, s, t, c, c')$$
 mit:

•
$$(V, E)$$
 ist ein gerichteter Graph $(n = |V|, m = |E|)$

•
$$s, t \in V$$
 mit $s \neq t$

•
$$c: E \mapsto \mathbb{N}^+$$

$$\bullet \ c': E \mapsto \mathbb{N}^+$$

Ausgabe: $f: E \mapsto \mathbb{R}_0^+$ mit:

•
$$\forall e : c'(e) \leqslant f(e) \leqslant c(e) \text{ oder } f(e) = 0.$$

•
$$\forall v \in V \setminus \{s,t\} : \sum_{(a,v) \in E} f((a,v)) = \sum_{(v,a) \in E} f((v,a))$$

Definition (Flussproblem)

Eingabe:
$$G = (V, E, s, t, c, c')$$
 mit:

•
$$(V, E)$$
 ist ein gerichteter Graph $(n = |V|, m = |E|)$

•
$$s, t \in V \text{ mit } s \neq t$$

•
$$c: E \mapsto \mathbb{N}^+$$

• $c': F \mapsto \mathbb{N}^+$

Ausgabe:
$$f: E \mapsto \mathbb{R}_0^+$$
 mit:

•
$$\forall e : c'(e) \leqslant f(e) \leqslant c(e) \text{ oder } f(e) = 0.$$

•
$$\forall v \in V \setminus \{s, t\} : \sum_{(a,v) \in E} f((a,v)) = \sum_{(v,a) \in E} f((v,a))$$

Ziel: Bestimme, ob es so einen nicht trivialen Fluss gibt. Falls ja, dann maximiere $w(f) = \sum_{(s,v) \in E} f((s,v)).$

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Reduktion

Theorem

Zu einem gegeben Flussproblem G = (V, E, s, t, c, c') ist es NP-vollständig zu bestimmen, ob es so einen nicht trivialen Fluss gibt.

Beweis: Übung, b.z.w. Reduktion auf Exact-3-SAT.

Exact-3-SAT

uct 5 5/11

Definition

Eine Boolesche Formel \mathcal{F} ist in Exact-3-KNF:

$$\mathcal{F}(x_1, x_2, ..., x_r) = \bigwedge_{i=1}^k c_i$$

2:12 Mit Alternativen 2/6 Exact-3-SAT

Definition

Eine Boolesche Formel \mathcal{F} ist in Exact-3-KNF:

$$\mathcal{F}(x_1, x_2, ..., x_r) = \bigwedge_{i=1}^k c_i$$

00000000

Exact-3-SAT

Definition

Eine Boolesche Formel F ist in Exact-3-KNF:

$$\mathcal{F}(x_1, x_2, ..., x_r) = \bigwedge_{i=1}^k c_i$$

$$(Klauseln)$$
 $c_i = (l_i^1 \lor l_i^2 \lor l_i^3)$

$$\forall \ 1 \leqslant i \leqslant k$$

 $\forall 1 \leq i \leq k$

2:12 Mit Alternativen 4/6

Exact-3-SAT

Definition

Eine Boolesche Formel \mathcal{F} ist in Exact-3-KNF:

$$\mathcal{F}(x_1, x_2, ..., x_r) = \bigwedge_{i=1}^k c_i$$

(Klauseln)
$$c_i = (l_i^1 \lor l_i^2 \lor l_i^3)$$

(Literale)
$$l_i^j = \left\{ \begin{array}{ll} \neg x_l & \text{oder} \\ x_l & \text{für ein } l: 1 \leqslant l \leqslant r \end{array} \right\} \quad \forall \ 1 \leqslant i \leqslant k \text{ und}$$

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 **RWTH**

Exact-3-SAT

Definition

Eine Boolesche Formel \mathcal{F} ist in Exact-3-KNF:

$$\mathcal{F}(x_1, x_2, ..., x_r) = \bigwedge_{i=1}^k c_i$$

$$(Klauseln) \quad c_i = (l_i^1 \lor l_i^2 \lor l_i^3) \qquad \forall 1 \leqslant i \leqslant k$$

$$(Literale) \quad l_i^j = \left\{ \begin{array}{l} \neg x_l & \text{oder} \\ x_l & \text{für ein } l : 1 \leqslant l \leqslant r \end{array} \right\} \quad \forall 1 \leqslant i \leqslant k \text{ und}$$

$$\forall 1 \leqslant i \leqslant k \text{ und}$$

$$\forall 1 \leqslant j \leqslant 3$$

Eine Belegung ist eine Funktion $W: \{x_1, x_2, ..., x_r\} \mapsto \{0, 1\}.$

Exact-3-SAT

Definition

Eine Boolesche Formel F ist in Exact-3-KNF:

$$\mathcal{F}(x_1, x_2, ..., x_r) = \bigwedge_{i=1}^k c_i$$

$$(Klauseln) \quad c_i = (l_i^1 \lor l_i^2 \lor l_i^3) \qquad \forall \ 1 \leqslant i \leqslant k$$

$$(Literale) \quad l_i^j = \left\{ \begin{array}{l} \neg x_l & \text{oder} \\ x_l & \text{für ein } l : 1 \leqslant l \leqslant r \end{array} \right\} \quad \forall \ 1 \leqslant i \leqslant k \text{ und}$$

$$\forall \ 1 \leqslant i \leqslant k \text{ und}$$

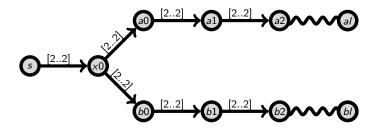
$$\forall \ 1 \leqslant j \leqslant 3$$

Eine Belegung ist eine Funktion $W : \{x_1, x_2, ..., x_r\} \mapsto \{0, 1\}.$

Theorem (Exakt-3-SAT)

Es ist NP-vollständig, festzustellen, ob es für $\mathcal F$ aus Exact-3-KNF eine erfüllende Belegung gibt, bei der in jeder Klausel genau ein Literal "true" ist.

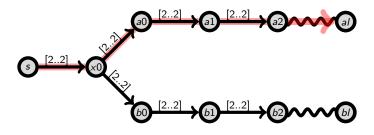
Erste Variable x_0



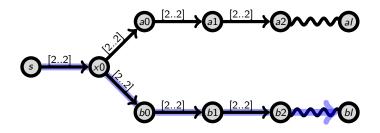
Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Erste Variable x_0

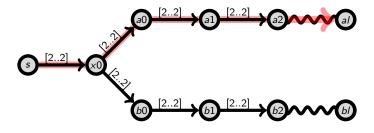


Erste Variable x_0



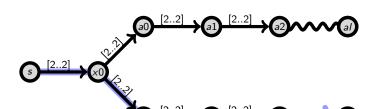
Walter Unger 6.11.2018 7:05 SS2015 RWTH



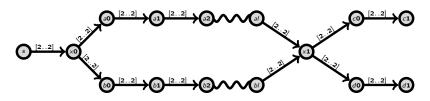


Erste Variable x_0

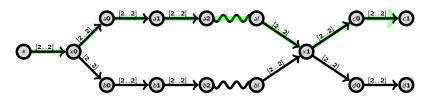
Walter Unger 6.11.2018 7:05 SS2015 RWTH



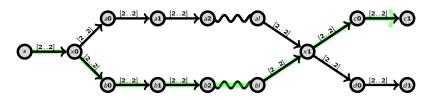
Zweite Variable x_1



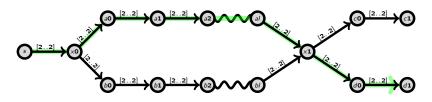
Zweite Variable x_1



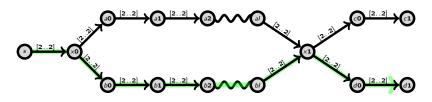
Zweite Variable x₁

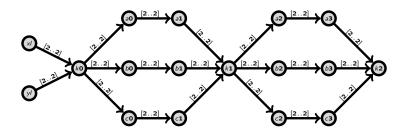


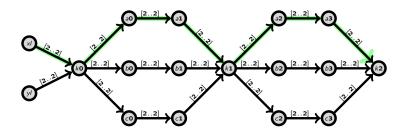
Zweite Variable x₁



Zweite Variable x₁



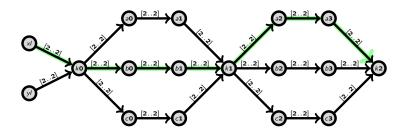




ezielle Flüsse (Alternativen)

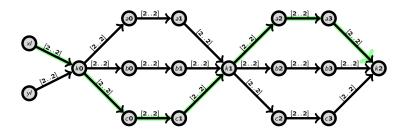
0000●000

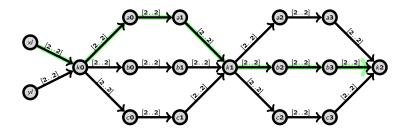
Walter Unger 6.11.2018 7:05 SS2015 RWTH



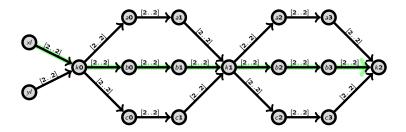
zielle Flüsse (Alternativen) Flüsse mit Kostenfunktion
0000000

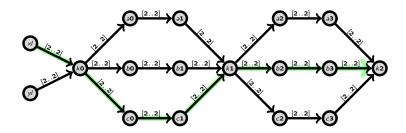
Walter Unger 6.11.20187:05 SS2015 RWTH

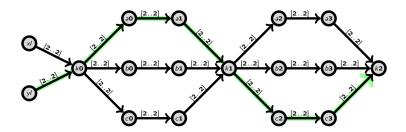




Erste zwei Klauseln k₀ und k₁



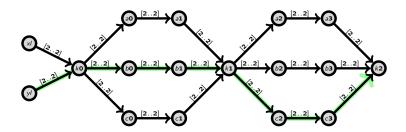




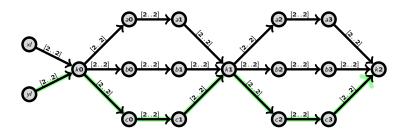
pezielle Flüsse (Alternativen)

0000●000

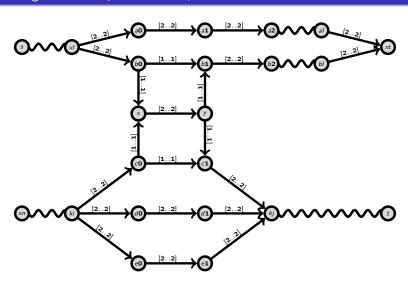
Walter Unger 6.11.2018 7:05 S52015 RWTH



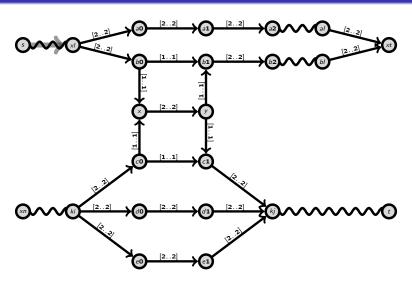
Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH



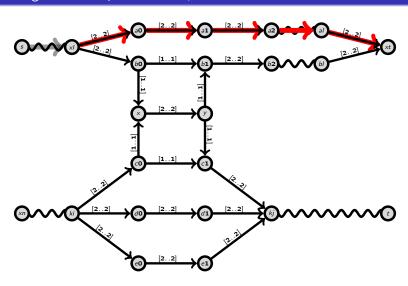
Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH



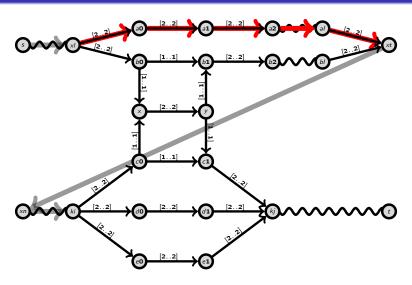
Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH



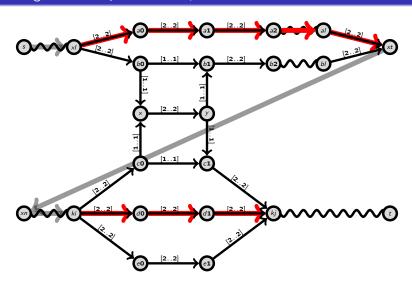
Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH



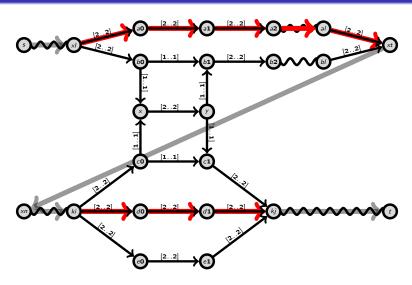
Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH



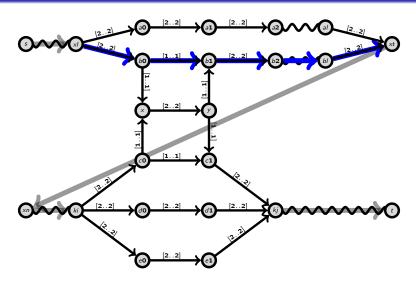
Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH



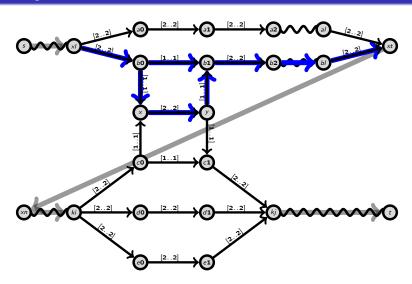
Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH



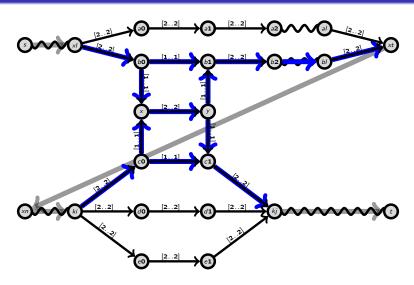
Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH



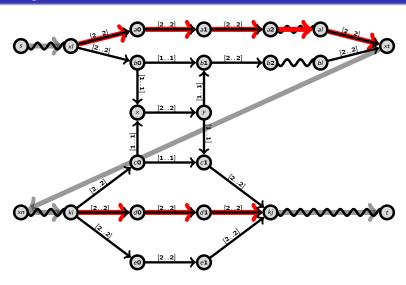
Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH



Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

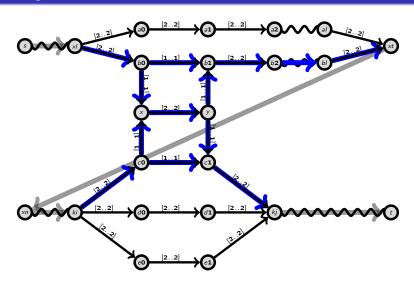


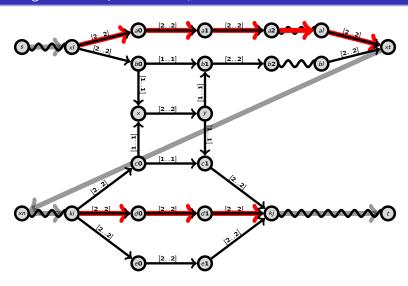
Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

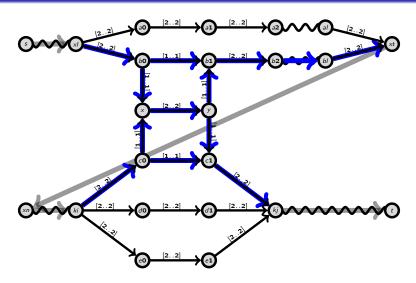


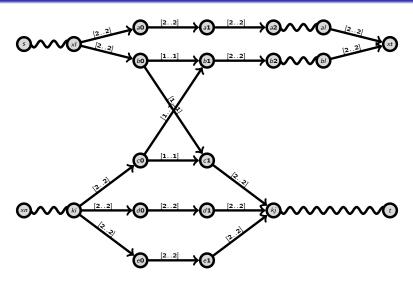
ielle Flüsse (Alternativen) Flüsse mit Kostenfunktion
000000

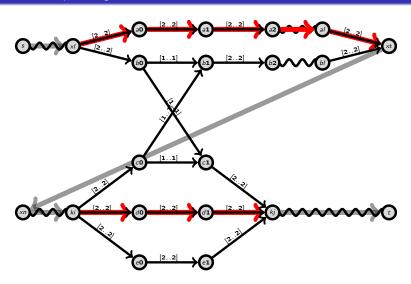
Walter Unger 6:11.20187:05 SS2015 RWTH

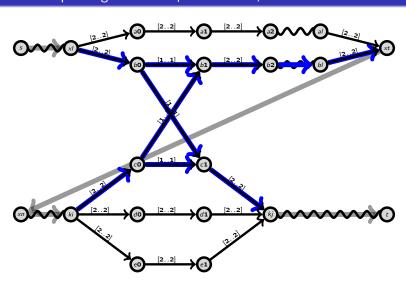


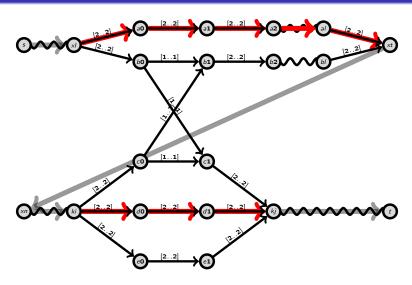


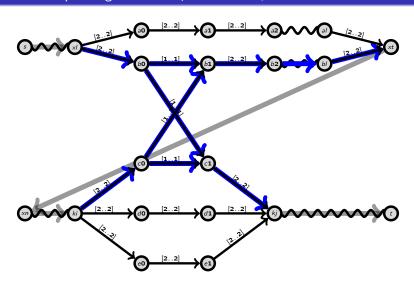












Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Konstruktion

4 Für jede Variable konstruiere einen Baustein, wie oben.

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 **RWTH**

- 4 Für jede Variable konstruiere einen Baustein, wie oben.
 - Der obere Zweig entspricht der Variablen selber.

- 4 Für jede Variable konstruiere einen Baustein, wie oben.
 - Der obere Zweig entspricht der Variablen selber.
 - Der untere Zweig entspricht der negierten Variablen.

- Für jede Variable konstruiere einen Baustein, wie oben.
 - Der obere Zweig entspricht der Variablen selber.
 - Der untere Zweig entspricht der negierten Variablen.
- 2 Für jede Klausel konstruiere einen Baustein, wie oben.

2:18 Mit Alternativen 5/10

Spezielle Flüsse (Alternativen)

- Für jede Variable konstruiere einen Baustein, wie oben.
 - Der obere Zweig entspricht der Variablen selber.
 - Der untere Zweig entspricht der negierten Variablen.
- 2 Für jede Klausel konstruiere einen Baustein, wie oben.
 - Der erste Zweig entspricht dem ersten Literal in der Klausel.

2:18 Mit Alternativen 6/10

Spezielle Flüsse (Alternativen) 00000000

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

- 4 Für jede Variable konstruiere einen Baustein, wie oben.
 - Der obere Zweig entspricht der Variablen selber.
 - Der untere Zweig entspricht der negierten Variablen.
- 2 Für jede Klausel konstruiere einen Baustein, wie oben.
 - Der erste Zweig entspricht dem ersten Literal in der Klausel.
 - Die weiteren Zweige dem zweiten und dem dritten Literal in der Klausel.

- Für jede Variable konstruiere einen Baustein, wie oben.
 - Der obere Zweig entspricht der Variablen selber.
 - Der untere Zweig entspricht der negierten Variablen.
- Für jede Klausel konstruiere einen Baustein, wie oben.
 - Der erste Zweig entspricht dem ersten Literal in der Klausel.
 - Die weiteren Zweige dem zweiten und dem dritten Literal in der Klausel.
- 4 Hänge alle Bausteine für die Variablen und Klauseln hintereinander.

- Für jede Variable konstruiere einen Baustein, wie oben.
 - Der obere Zweig entspricht der Variablen selber.
 - Der untere Zweig entspricht der negierten Variablen.
- Für jede Klausel konstruiere einen Baustein, wie oben.
 - Der erste Zweig entspricht dem ersten Literal in der Klausel.
- Die weiteren Zweige dem zweiten und dem dritten Literal in der Klausel.
- 4 Hänge alle Bausteine für die Variablen und Klauseln hintereinander.
- Für jedes Auftreten eines Literals in einer Klausel mache die obige Anpassung.

- Für jede Variable konstruiere einen Baustein, wie oben.
 - Der obere Zweig entspricht der Variablen selber.
 - Der untere Zweig entspricht der negierten Variablen.
- Für jede Klausel konstruiere einen Baustein, wie oben.
 - Der erste Zweig entspricht dem ersten Literal in der Klausel.
- Die weiteren Zweige dem zweiten und dem dritten Literal in der Klausel.
- 4 Hänge alle Bausteine für die Variablen und Klauseln hintereinander.
- Für jedes Auftreten eines Literals in einer Klausel mache die obige Anpassung.
- § Falls es eine Belegung der Variablen gibt, die die Formel erfüllt, dann:

- Für jede Variable konstruiere einen Baustein, wie oben.
 - Der obere Zweig entspricht der Variablen selber.
 - Der untere Zweig entspricht der negierten Variablen.
- Für jede Klausel konstruiere einen Baustein, wie oben.
 - Der erste Zweig entspricht dem ersten Literal in der Klausel.
- Die weiteren Zweige dem zweiten und dem dritten Literal in der Klausel.
- 4 Hänge alle Bausteine für die Variablen und Klauseln hintereinander.
- Für jedes Auftreten eines Literals in einer Klausel mache die obige Anpassung.
- § Falls es eine Belegung der Variablen gibt, die die Formel erfüllt, dann:
 - geht ein Fluss von 2 durch jeweils den Zweig, der der Belegung der Variablen entspricht.

Walter Unger 6.11.2018 7:05 SS2015 **RWTH**

Motivation

• Benutzung der Kanten (Transportwege) im allgemeinen nicht umsonst.

isse (Alternativen)

Flüsse mit Kostenfunktion

OOOOOOOOOOOOOOOOOOO

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Motivation

- Benutzung der Kanten (Transportwege) im allgemeinen nicht umsonst.
- Daher sollten wir die Kosten minimieren.

Motivation

• Benutzung der Kanten (Transportwege) im allgemeinen nicht umsonst.

- Daher sollten wir die Kosten minimieren.
- Kosten sind minimal, wenn der Fluss Null ist.

Motivation

- Benutzung der Kanten (Transportwege) im allgemeinen nicht umsonst.
- Daher sollten wir die Kosten minimieren.
- Kosten sind minimal, wenn der Fluss Null ist.
- Daher suchen wir:

Kostenminimalen Fluss mit Wert W.

Motivation

- Benutzung der Kanten (Transportwege) im allgemeinen nicht umsonst.
- Daher sollten wir die Kosten minimieren.
- Kosten sind minimal, wenn der Fluss Null ist.
- Daher suchen wir:

Kostenminimalen Fluss mit Wert W.

 Wichtig: G sollte keine Kreise mit negativen Kosten (Gewichtssumme) enthalten. Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Das Flussproblem mit Kosten

Definition (Min-Cost-Flow-Problem)

Das Flussproblem mit Kosten

Definition (Min-Cost-Flow-Problem)

Einleitung 3/10

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion

Das Flussproblem mit Kosten

Definition (Min-Cost-Flow-Problem)

Eingabe:
$$G = (V, E, s, t, c, I), W$$
 mit:

• (V, E) ist ein gerichteter Graph (n = |V|, m = |E|)

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Das Flussproblem mit Kosten

Definition (Min-Cost-Flow-Problem)

- (V, E) ist ein gerichteter Graph (n = |V|, m = |E|)
- $s, t \in V$ mit $s \neq t$

Einleitung 5/10

Das Flussproblem mit Kosten

Definition (Min-Cost-Flow-Problem)

- (V, E) ist ein gerichteter Graph (n = |V|, m = |E|)
- $s, t \in V \text{ mit } s \neq t$
- $c: E \mapsto \mathbb{N}^+$

Definition (Min-Cost-Flow-Problem)

- (V, E) ist ein gerichteter Graph (n = |V|, m = |E|)
- $s, t \in V$ mit $s \neq t$
- $c: E \mapsto \mathbb{N}^+$
- $I: F \mapsto \mathbb{Z}$ und $W \in \mathbb{N}$

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion

Das Flussproblem mit Kosten

Einleitung 7/10

Definition (Min-Cost-Flow-Problem)

Eingabe: G = (V, E, s, t, c, I), W mit:

•
$$(V, E)$$
 ist ein gerichteter Graph $(n = |V|, m = |E|)$

•
$$s, t \in V \text{ mit } s \neq t$$

•
$$c: E \mapsto \mathbb{N}^+$$

$$\bullet \ \mathit{I} : E \mapsto \mathbb{Z} \ \mathsf{und} \ \mathit{W} \in \mathbb{N}$$

Ausgabe: $f: E \mapsto \mathbb{R}_0^+$ mit:

Definition (Min-Cost-Flow-Problem)

•
$$(V, E)$$
 ist ein gerichteter Graph $(n = |V|, m = |E|)$

•
$$s, t \in V \text{ mit } s \neq t$$

•
$$c: E \mapsto \mathbb{N}^+$$

• $I: E \mapsto \mathbb{Z}$ und $W \in \mathbb{N}$

Ausgabe:
$$f: E \mapsto \mathbb{R}_0^+$$
 mit:

•
$$\forall e: f(e) \leqslant c(e)$$
.

Definition (Min-Cost-Flow-Problem)

Eingabe:
$$G = (V, E, s, t, c, I), W$$
 mit:

•
$$(V, E)$$
 ist ein gerichteter Graph $(n = |V|, m = |E|)$

•
$$s, t \in V \text{ mit } s \neq t$$

•
$$c: E \mapsto \mathbb{N}^+$$

• $I: E \mapsto \mathbb{Z}$ und $W \in \mathbb{N}$

Ausgabe:
$$f: E \mapsto \mathbb{R}_0^+$$
 mit:

•
$$\forall e : f(e) \leqslant c(e)$$
.

•
$$\forall v \in V \setminus \{s,t\} : \sum_{(a,v) \in E} f((a,v)) = \sum_{(v,a) \in E} f((v,a))$$

Definition (Min-Cost-Flow-Problem)

Eingabe:
$$G = (V, E, s, t, c, I), W$$
 mit:

•
$$(V, E)$$
 ist ein gerichteter Graph $(n = |V|, m = |E|)$

•
$$s, t \in V \text{ mit } s \neq t$$

•
$$c: E \mapsto \mathbb{N}^+$$

• $l: F \mapsto \mathbb{Z}$ und $W \in \mathbb{N}$

Ausgabe:
$$f: E \mapsto \mathbb{R}_0^+$$
 mit:

•
$$\forall e : f(e) \leqslant c(e)$$
.

•
$$\forall v \in V \setminus \{s, t\} : \sum_{(a,v) \in E} f((a,v)) = \sum_{(v,a) \in E} f((v,a))$$

Ziel: Bestimme Fluss f w(f) = W und

minimalen
$$I(f) = \sum_{e \in E} f(e) \cdot I(e)$$
.

D.h.
$$I(f) = min\{I(g) \mid g \text{ ist Fluss mit } w(g) = W\}.$$

Walter Unger 6.11.2018 7:05 SS2015 **RWTH**

Beobachtungen

• Falls W = 1, so entspricht das dem kürzesten Wege Problem.

ielle Flüsse (Alternativen)

OOOOOO

Flüsse mit Kostenfunktion

OOOOOOOOOOOOOOOOOOOOOOOOOOOOO

Walter Unger 6.11.2018 7:05 SS2015 RWTH

2:21 Einleitung 2/9 Beobachtungen

- ullet Falls W=1, so entspricht das dem kürzesten Wege Problem.
- Falls I(e) = 0 für alle $e \in E$, dann können die obigen Algorithmen leicht zur Lösung des Problems adaptiert werden:

- ullet Falls W=1, so entspricht das dem kürzesten Wege Problem.
- Falls I(e) = 0 für alle e ∈ E, dann können die obigen Algorithmen leicht zur Lösung des Problems adaptiert werden:
 - Falls erstmalig $w(f) \geqslant W$ gilt, dann breche ab.

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

- Falls W = 1, so entspricht das dem kürzesten Wege Problem.
- Falls I(e) = 0 für alle $e \in E$, dann können die obigen Algorithmen leicht zur Lösung des Problems adaptiert werden:
 - Falls erstmalig $w(f) \geqslant W$ gilt, dann breche ab.
 - Sei p der Wert der letzten Erweiterung.

- ullet Falls W=1, so entspricht das dem kürzesten Wege Problem.
- Falls I(e) = 0 für alle e ∈ E, dann können die obigen Algorithmen leicht zur Lösung des Problems adaptiert werden:
 - Falls erstmalig $w(f) \geqslant W$ gilt, dann breche ab.
 - Sei p der Wert der letzten Erweiterung.
 - **Q** Ersetze den letzten Fluss durch eine Fluss mit dem Wert p (w(f) W).

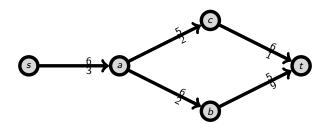
- ullet Falls W=1, so entspricht das dem kürzesten Wege Problem.
- Falls I(e)=0 für alle $e\in E$, dann können die obigen Algorithmen leicht zur Lösung des Problems adaptiert werden:
 - Falls erstmalig $w(f) \geqslant W$ gilt, dann breche ab.
 - Sei p der Wert der letzten Erweiterung.
 - § Ersetze den letzten Fluss durch eine Fluss mit dem Wert p (w(f) W).
- Alternativ kann auch wie folgt vorgegangen werden:

- Falls W = 1, so entspricht das dem kürzesten Wege Problem.
- Falls I(e) = 0 für alle $e \in E$, dann können die obigen Algorithmen leicht zur Lösung des Problems adaptiert werden:
 - Falls erstmalig $w(f) \ge W$ gilt, dann breche ab.
 - Sei p der Wert der letzten Erweiterung.
- **3** Ersetze den letzten Fluss durch eine Fluss mit dem Wert p (w(f) W).
- Alternativ kann auch wie folgt vorgegangen werden:
 - Erzeuge neue Quelle s'.

- ullet Falls W=1, so entspricht das dem kürzesten Wege Problem.
- Falls I(e) = 0 für alle $e \in E$, dann können die obigen Algorithmen leicht zur Lösung des Problems adaptiert werden:
 - Falls erstmalig $w(f) \geqslant W$ gilt, dann breche ab.
 - Sei p der Wert der letzten Erweiterung.
- § Ersetze den letzten Fluss durch eine Fluss mit dem Wert p (w(f) W).
- Alternativ kann auch wie folgt vorgegangen werden:
 - Erzeuge neue Quelle s'.
 - ② Füge Kante e' = (s', s) mit c(e') = W hinzu.

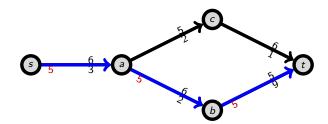
- ullet Falls W=1, so entspricht das dem kürzesten Wege Problem.
- Falls I(e)=0 für alle $e\in E$, dann können die obigen Algorithmen leicht zur Lösung des Problems adaptiert werden:
 - Falls erstmalig $w(f) \geqslant W$ gilt, dann breche ab.
 - Sei p der Wert der letzten Erweiterung.
- **②** Ersetze den letzten Fluss durch eine Fluss mit dem Wert p (w(f) W).
- Alternativ kann auch wie folgt vorgegangen werden:
 - Erzeuge neue Quelle s'.
 - ② Füge Kante e' = (s', s) mit c(e') = W hinzu.
 - Oamit wird der maximale Fluss durche W begrenzt.

Kantenbeschriftung:



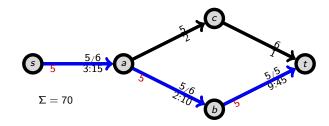
Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

- Kantenbeschriftung:
 - oben: Fluss/MaxFluss und

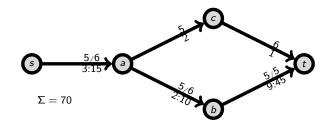


2:22 Einleitung 3/10

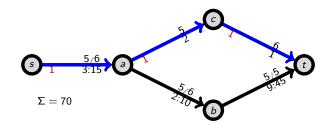
- Kantenbeschriftung:
 - oben: Fluss/MaxFluss und
 - darunter Kosten: Aktuelle Kosten



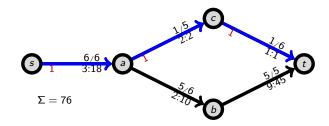
- Kantenbeschriftung:
 - oben: Fluss/MaxFluss und
 - darunter Kosten: Aktuelle Kosten
- Bestimme erst einen Fluss (mit Wert W = 6).



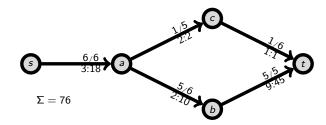
- Kantenbeschriftung:
 - oben: Fluss/MaxFluss und
 - darunter Kosten: Aktuelle Kosten
- Bestimme erst einen Fluss (mit Wert W = 6).



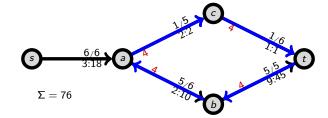
- Kantenbeschriftung:
 - oben: Fluss/MaxFluss und
 - darunter Kosten: Aktuelle Kosten
- Bestimme erst einen Fluss (mit Wert W = 6).



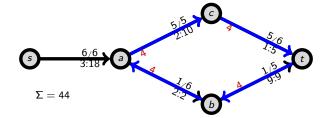
- Kantenbeschriftung:
 - oben: Fluss/MaxFluss und
 - darunter Kosten: Aktuelle Kosten
- Bestimme erst einen Fluss (mit Wert W = 6).
- Der Fluss muss nicht kostenoptimal sein.



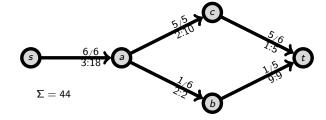
- Kantenbeschriftung:
 - oben: Fluss/MaxFluss und
 - darunter Kosten: Aktuelle Kosten
- Bestimme erst einen Fluss (mit Wert W = 6).
- Der Fluss muss nicht kostenoptimal sein.
- Verbessere die Kosten.



- Kantenbeschriftung:
 - oben: Fluss/MaxFluss und
 - darunter Kosten: Aktuelle Kosten
- Bestimme erst einen Fluss (mit Wert W = 6).
- Der Fluss muss nicht kostenoptimal sein.
- Verbessere die Kosten.

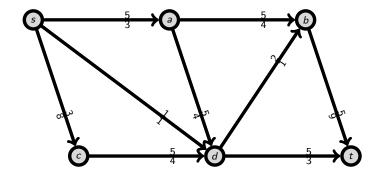


- Kantenbeschriftung:
 - oben: Fluss/MaxFluss und
 - darunter Kosten: Aktuelle Kosten
- Bestimme erst einen Fluss (mit Wert W = 6).
- Der Fluss muss nicht kostenoptimal sein.
- Verbessere die Kosten.



Flüsse mit Kostenfunktion

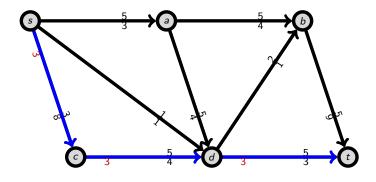
Walter Unger 6.11.2018 7:05 SS2015 RWTH



Spezielle Flüsse (Alternativen)

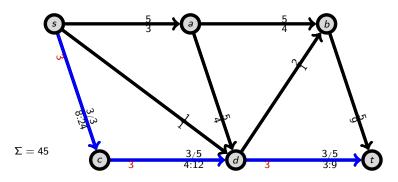
Flüsse mit Kostenfunktion

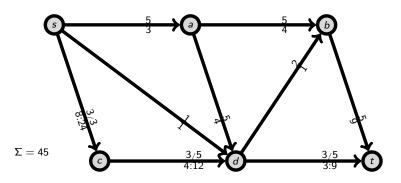
Walter Unger 6.11.2018 7:05 SS2015 RWTH



Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

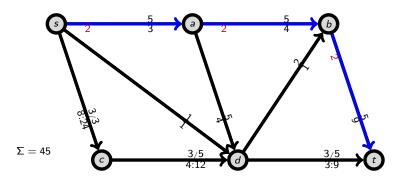




Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion

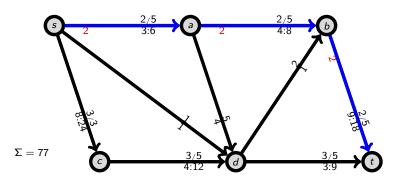
Walter Unger 6.11.2018 7:05 SS2015 RWTH



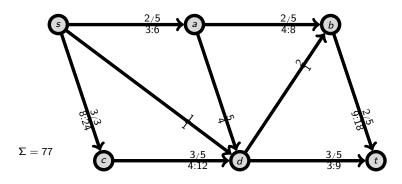
Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 **RWTH**

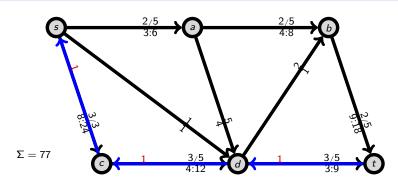


Walter Unger 6.11.2018 7:05 SS2015 RWTH



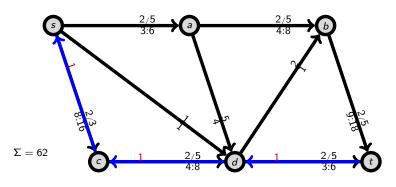
Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH



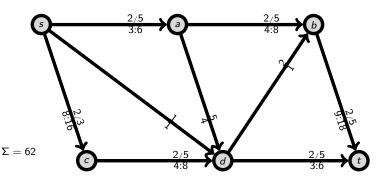
Flüsse mit Kostenfunktion

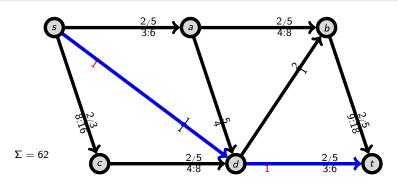
Walter Unger 6.11.2018 7:05 SS2015 RWTH



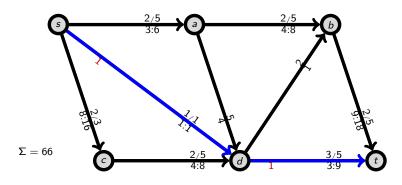
Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

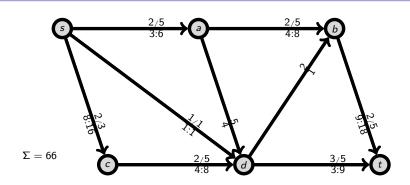




Walter Unger 6.11.2018 7:05 SS2015 RWTH

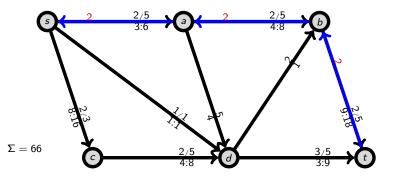


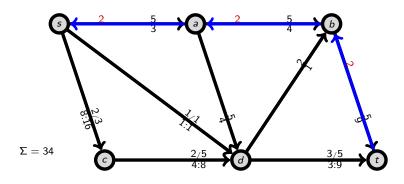
2:23 Einleitung 13/25 Walter Unger 6.11.2018 7:05 SS2015 RWTH Beispiel (W=5)



Flüsse mit Kostenfunktion

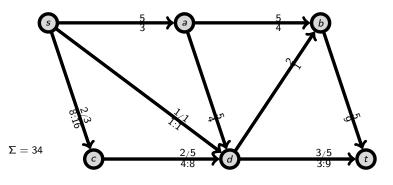
Walter Unger 6.11.2018 7:05 SS2015 RWTH





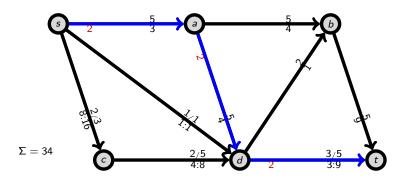
Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH



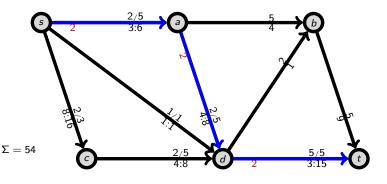
Flüsse mit Kostenfunktion

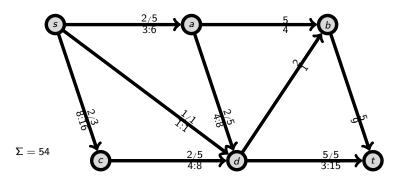
Walter Unger 6.11.2018 7:05 SS2015 RWTH



Flüsse mit Kostenfunktion

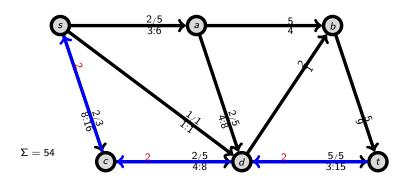
Walter Unger 6.11.2018 7:05 SS2015 RWTH



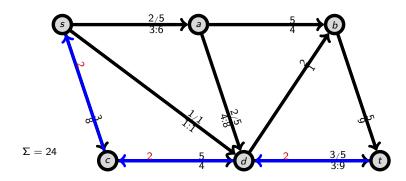


Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 **RWTH**

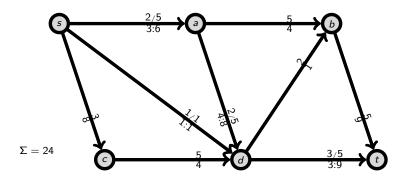


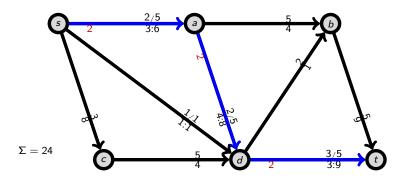
Walter Unger 6.11.2018 7:05 SS2015 **RWTH**

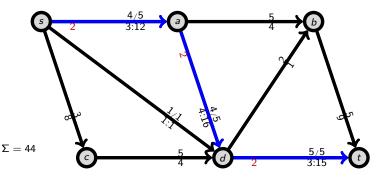


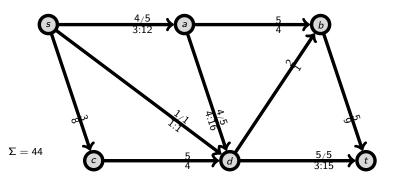
Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 **RWTH**









Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 **RWTH**

Idee

2:24 Idee 1/13

• Bestimme einen beliebigen Fluss f mit w(f) = W.

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Idee

2:24 Idee 2/13

- Bestimme einen beliebigen Fluss f mit w(f) = W.
- Verbessere schrittweise die Kosten des Flusses:

Idee

2:24 Idee 3/13

- Bestimme einen beliebigen Fluss f mit w(f) = W.
- Verbessere schrittweise die Kosten des Flusses:
 - Annahme: es gibt Fluss f' mit I(f') < I(f) und w(f') = w(f).

Idee

2:24 Idee 4/13

- Bestimme einen beliebigen Fluss f mit w(f) = W.
- Verbessere schrittweise die Kosten des Flusses:
 - Annahme: es gibt Fluss f' mit I(f') < I(f) und w(f') = w(f).
 - Dann gibt es einen Unterschied zwischen f und f'.

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 **RWTH**

Idee

- Bestimme einen beliebigen Fluss f mit w(f) = W.
- Verbessere schrittweise die Kosten des Flusses:
 - Annahme: es gibt Fluss f' mit I(f') < I(f) und w(f') = w(f).
 - Dann gibt es einen Unterschied zwischen f und f'.
 - Betrachte diesen Unterschied.

Idee

- Bestimme einen beliebigen Fluss f mit w(f) = W.
- Verbessere schrittweise die Kosten des Flusses:
 - Annahme: es gibt Fluss f' mit I(f') < I(f) und w(f') = w(f).
 - Dann gibt es einen Unterschied zwischen f und f'.
 - Betrachte diesen Unterschied.
 - Das muss ein zyklischer Fluss sein, d.h. ein Fluss ohne Quelle und Senke.

2:24 Idee 7/13

- Bestimme einen beliebigen Fluss f mit w(f) = W.
- Verbessere schrittweise die Kosten des Flusses:
 - Annahme: es gibt Fluss f' mit I(f') < I(f) und w(f') = w(f).
 - Dann gibt es einen Unterschied zwischen f und f'. Betrachte diesen Unterschied.
 - Das muss ein zyklischer Fluss sein, d.h. ein Fluss ohne Quelle und Senke.
 - Für f' und f gilt die Flusserhaltung.

Idee

- Bestimme einen beliebigen Fluss f mit w(f) = W.
- Verbessere schrittweise die Kosten des Flusses:
 - Annahme: es gibt Fluss f' mit I(f') < I(f) und w(f') = w(f).
 - Dann gibt es einen Unterschied zwischen f und f'. Betrachte diesen Unterschied.

 - Das muss ein zyklischer Fluss sein, d.h. ein Fluss ohne Quelle und Senke.
 - Für f' und f gilt die Flusserhaltung.
 - Und $f_{out}(s) = f'_{out}(s) = f_{in}(t) = f'_{in}(t)$.

Spezielle Flüsse (Mindestfluss)

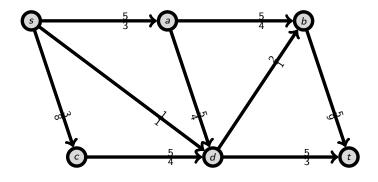
- Bestimme einen beliebigen Fluss f mit w(f) = W.
- Verbessere schrittweise die Kosten des Flusses:
 - Annahme: es gibt Fluss f' mit I(f') < I(f) und w(f') = w(f).
 - Dann gibt es einen Unterschied zwischen f und f'.
 - Betrachte diesen Unterschied. • Das muss ein zyklischer Fluss sein, d.h. ein Fluss ohne Quelle und Senke.
 - Für f' und f gilt die Flusserhaltung.
 - Und $f_{out}(s) = f'_{out}(s) = f_{in}(t) = f'_{in}(t)$.
 - Damit gilt: Falls $f(a, b) \neq f'(a, b)$, dann gibt es $c \in V \setminus \{a, b\}$ mit: $f(a,c) \neq f'(a,c)$.

- Bestimme einen beliebigen Fluss f mit w(f) = W.
- Verbessere schrittweise die Kosten des Flusses:
 - Annahme: es gibt Fluss f' mit I(f') < I(f) und w(f') = w(f).
 - Dann gibt es einen Unterschied zwischen f und f'. Betrachte diesen Unterschied.
 - Das muss ein zyklischer Fluss sein, d.h. ein Fluss ohne Quelle und Senke.
 - Für f' und f gilt die Flusserhaltung.
 - Und $f_{out}(s) = f'_{out}(s) = f_{in}(t) = f'_{in}(t)$.
 - Damit gilt: Falls $f(a, b) \neq f'(a, b)$, dann gibt es $c \in V \setminus \{a, b\}$ mit: $f(a,c) \neq f'(a,c)$.
 - Damit gibt es mindestens einen Kreis mit Fluss g über Kanten e mit $f(e) \neq f'(e)$.

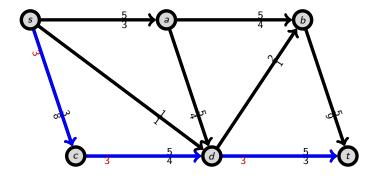
- Bestimme einen beliebigen Fluss f mit w(f) = W.
- Verbessere schrittweise die Kosten des Flusses:
 - Annahme: es gibt Fluss f' mit I(f') < I(f) und w(f') = w(f).
 - Dann gibt es einen Unterschied zwischen f und f'.
 Betrachte diesen Unterschied.
 - Das muss ein zyklischer Fluss sein, d.h. ein Fluss ohne Quelle und Senke.
 - Für f' und f gilt die Flusserhaltung.
 - Und $f_{out}(s) = f'_{out}(s) = f_{in}(t) = f'_{in}(t)$.
 - Damit gilt: Falls $f(a, b) \neq f'(a, b)$, dann gibt es $c \in V \setminus \{a, b\}$ mit: $f(a, c) \neq f'(a, c)$.
 - Damit gibt es mindestens einen Kreis mit Fluss g über Kanten e mit $f(e) \neq f'(e)$.
 - Dieser zyklische Fluss besteht aus einer Summe von Kreisen.

- Bestimme einen beliebigen Fluss f mit w(f) = W.
- Verbessere schrittweise die Kosten des Flusses:
 - Annahme: es gibt Fluss f' mit I(f') < I(f) und w(f') = w(f).
 - Dann gibt es einen Unterschied zwischen f und f'. Betrachte diesen Unterschied.
 - Das muss ein zyklischer Fluss sein, d.h. ein Fluss ohne Quelle und Senke.
 - Für f' und f gilt die Flusserhaltung.
 - Und $f_{out}(s) = f'_{out}(s) = f_{in}(t) = f'_{in}(t)$.
 - Damit gilt: Falls $f(a, b) \neq f'(a, b)$, dann gibt es $c \in V \setminus \{a, b\}$ mit: $f(a,c) \neq f'(a,c)$.
 - Damit gibt es mindestens einen Kreis mit Fluss g über Kanten e mit $f(e) \neq f'(e)$.
 - Dieser zyklische Fluss besteht aus einer Summe von Kreisen.
 - Einer dieser Kreise muss die Kosten f
 ür f' verbessern.

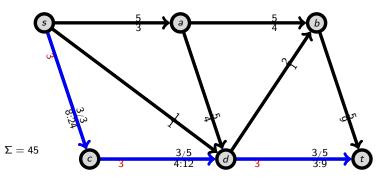
- Bestimme einen beliebigen Fluss f mit w(f) = W.
- Verbessere schrittweise die Kosten des Flusses:
 - Annahme: es gibt Fluss f' mit I(f') < I(f) und w(f') = w(f).
 - Dann gibt es einen Unterschied zwischen f und f'.
 Betrachte diesen Unterschied.
 - Das muss ein zyklischer Fluss sein, d.h. ein Fluss ohne Quelle und Senke.
 - Für f' und f gilt die Flusserhaltung.
 - Und $f_{out}(s) = f'_{out}(s) = f_{in}(t) = f'_{in}(t)$.
 - Damit gilt: Falls $f(a, b) \neq f'(a, b)$, dann gibt es $c \in V \setminus \{a, b\}$ mit: $f(a, c) \neq f'(a, c)$.
 - Damit gibt es mindestens einen Kreis mit Fluss g über Kanten e mit $f(e) \neq f'(e)$.
 - Dieser zyklische Fluss besteht aus einer Summe von Kreisen.
 - Einer dieser Kreise muss die Kosten für f' verbessern.
- Idee: suche diese verbessernden Kreise.

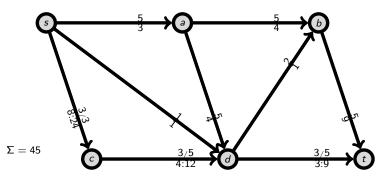


Walter Unger 6.11.2018 7:05 SS2015 RWTH

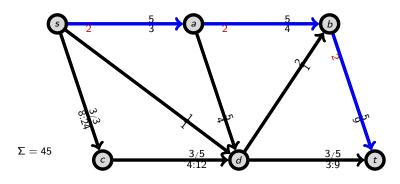


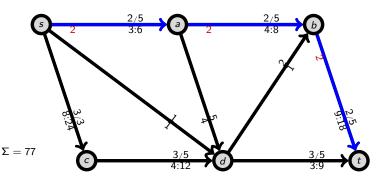
Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH



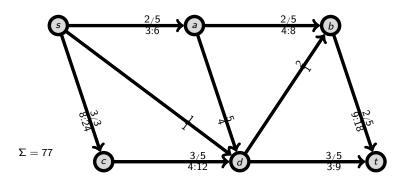


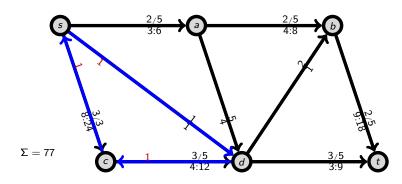
Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

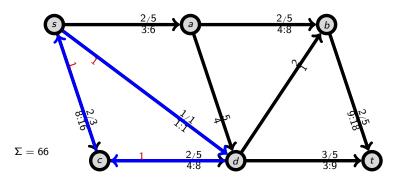




Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

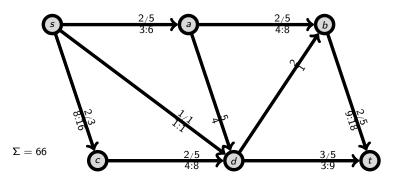




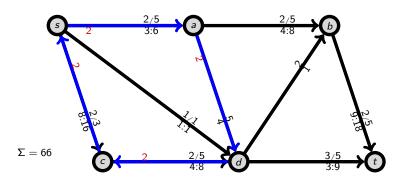


Flüsse mit Kostenfunktion

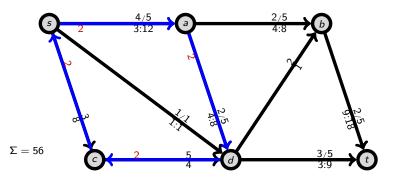
Walter Unger 6.11.2018 7:05 SS2015 **RWTH**

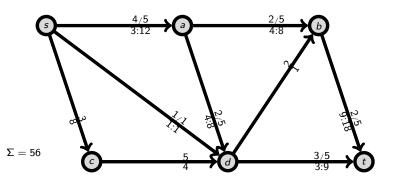


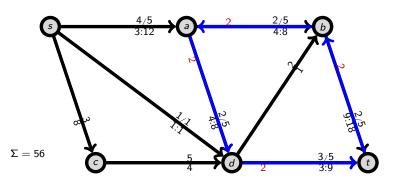
Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH



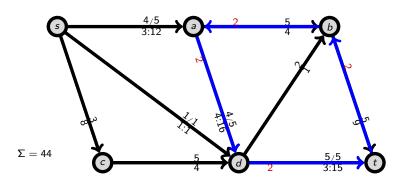
Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

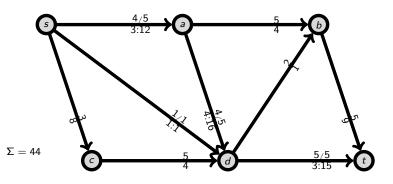






2:25 Idee 15/16 Beispiel mit Kreisen (W = 5)





Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Frage: Warum eine Suche nach Kreisen?

• Ein verbessernder Kreis entspricht zwei Wegen:

Walter Unger 6.11.2018 7:05 SS2015 RWTH

- Ein verbessernder Kreis entspricht zwei Wegen:
 - Ein Weg, der gelöscht wird,

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 **RWTH**

- Ein verbessernder Kreis entspricht zwei Wegen:
 - Ein Weg, der gelöscht wird,
 - Ein Weg, der hinzugefügt wird.

2:26 Idee 4/12

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

- Ein verbessernder Kreis entspricht zwei Wegen:
 - Ein Weg, der gelöscht wird,
 - Ein Weg, der hinzugefügt wird.
 - Die Differenz der Wege ist der Kreis.

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

- Ein verbessernder Kreis entspricht zwei Wegen:
 - Ein Weg, der gelöscht wird,
 - Ein Weg, der hinzugefügt wird.
 - Die Differenz der Wege ist der Kreis.
- Man könnte also auch nach Wegen suchen.

- Ein verbessernder Kreis entspricht zwei Wegen:
 - Ein Weg, der gelöscht wird,
 - Ein Weg, der hinzugefügt wird.
- Die Differenz der Wege ist der Kreis.
- Man könnte also auch nach Wegen suchen.
- Vorteil bei Kreisen:

- Ein verbessernder Kreis entspricht zwei Wegen:
 - Ein Weg, der gelöscht wird,
 - Ein Weg, der hinzugefügt wird.
 - Die Differenz der Wege ist der Kreis.
- Man könnte also auch nach Wegen suchen.
- Vorteil bei Kreisen:
 - Gewinn entspricht direkt den Kosten des Kreises.

- Ein verbessernder Kreis entspricht zwei Wegen:
 - Ein Weg, der gelöscht wird,
 - Ein Weg, der hinzugefügt wird.
- Die Differenz der Wege ist der Kreis.
- Man könnte also auch nach Wegen suchen.
- Vorteil bei Kreisen:
 - Gewinn entspricht direkt den Kosten des Kreises.
 - Einfachere Algorithmen bei sich ändernden Kosten.

- Ein verbessernder Kreis entspricht zwei Wegen:
 - Ein Weg, der gelöscht wird,
 - Ein Weg, der hinzugefügt wird.
- Die Differenz der Wege ist der Kreis.
- Man könnte also auch nach Wegen suchen.
- Vorteil bei Kreisen:
 - Gewinn entspricht direkt den Kosten des Kreises.
 - Einfachere Algorithmen bei sich ändernden Kosten.
 - Optimale Lösung bekannt.

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

- Ein verbessernder Kreis entspricht zwei Wegen:
 - Ein Weg, der gelöscht wird,
 - Ein Weg, der hinzugefügt wird.
- Die Differenz der Wege ist der Kreis.
- Man könnte also auch nach Wegen suchen.
- Vorteil bei Kreisen:
 - Gewinn entspricht direkt den Kosten des Kreises.
 - Einfachere Algorithmen bei sich ändernden Kosten.
 - Optimale Lösung bekannt.
 - "Anbieter" verändert die Kosten einer Kante

- Ein verbessernder Kreis entspricht zwei Wegen:
 - Ein Weg, der gelöscht wird,
 - Ein Weg, der hinzugefügt wird.
- Die Differenz der Wege ist der Kreis.
- Man könnte also auch nach Wegen suchen.
- Vorteil bei Kreisen:
 - Gewinn entspricht direkt den Kosten des Kreises.
 - Einfachere Algorithmen bei sich ändernden Kosten.
 - Optimale Lösung bekannt.
 - "Anbieter" verändert die Kosten einer Kante.
 - Passe bisherige Lösung durch das Suchen von Kreisen an.

- Ein verbessernder Kreis entspricht zwei Wegen:
 - Ein Weg, der gelöscht wird,
 - Ein Weg, der hinzugefügt wird.
- Die Differenz der Wege ist der Kreis.
- Man könnte also auch nach Wegen suchen.
- Vorteil bei Kreisen:
 - Gewinn entspricht direkt den Kosten des Kreises.
 - Einfachere Algorithmen bei sich ändernden Kosten.
 - Optimale Lösung bekannt.
 - "Anbieter" verändert die Kosten einer Kante.
 - Passe bisherige Lösung durch das Suchen von Kreisen an.
 - Die folgenden Beweise werden dadurch einfacher.

Flüsse mit Kostenfunktion

2:27 Idee 1/10 Walter Unger 6.11.2018 7:05 SS2015 RWTH

Definitionen

 $\bullet \ \, \mathsf{Gegeben} \ \, \textit{G} = (\textit{V},\textit{E},\textit{s},\textit{t},\textit{c},\textit{l}) \ \, \mathsf{und} \ \, \mathsf{Restnetzwerk} \ \, \textit{G}_{\textit{f}} = (\textit{V}',\textit{E}',\textit{s},\textit{t},\textit{c}',\textit{l}').$

- ullet Gegeben G=(V,E,s,t,c,l) und Restnetzwerk $G_f=(V',E',s,t,c',l').$
- c'(e) = c(e) für $e \in E \cap E'$

- ullet Gegeben G=(V,E,s,t,c,l) und Restnetzwerk $G_f=(V',E',s,t,c',l')$.
- c'(e) = c(e) für $e \in E \cap E'$
- c'(a,b) = c(b,a) und l'(a,b) = -l(b,a) für $(a,b) \in E' \setminus E$

- $\bullet \ \ \mathsf{Gegeben} \ \ \mathsf{G} = (V, E, s, t, c, \mathit{I}) \ \ \mathsf{und} \ \ \mathsf{Restnetzwerk} \ \ \mathsf{G_f} = (V', E', s, t, c', \mathit{I}').$
- c'(e) = c(e) für $e \in E \cap E'$
- c'(a,b) = c(b,a) und l'(a,b) = -l(b,a) für $(a,b) \in E' \setminus E$
- f' ist eine Zirkulation, gdw.: $\forall v \in V : f'_{in}(v) = f'_{out}(v)$.

- Gegeben G = (V, E, s, t, c, I) und Restnetzwerk $G_f = (V', E', s, t, c', I')$.
- c'(e) = c(e) für $e \in E \cap E'$
- c'(a, b) = c(b, a) und l'(a, b) = -l(b, a) für $(a, b) \in E' \setminus E$
- f' ist eine Zirkulation, gdw.: $\forall v \in V : f'_{in}(v) = f'_{out}(v)$.
- Wert einer Zirkulation f' über Schnitt (S, T):

$$f'(S,T) = \sum_{(v,w)\in E, v\in S, w\in T} f'(v,w) - \sum_{(w,v)\in E, v\in S, w\in T} f'(w,v).$$

- Gegeben G = (V, E, s, t, c, I) und Restnetzwerk $G_f = (V', E', s, t, c', I')$.
- c'(e) = c(e) für $e \in E \cap E'$
- c'(a, b) = c(b, a) und l'(a, b) = -l(b, a) für $(a, b) \in E' \setminus E$
- f' ist eine Zirkulation, gdw.: $\forall v \in V : f'_{in}(v) = f'_{out}(v)$.
- Wert einer Zirkulation f' über Schnitt (S, T):

$$f'(S,T) = \sum_{(v,w)\in E, v\in S, w\in T} f'(v,w) - \sum_{(w,v)\in E, v\in S, w\in T} f'(w,v).$$

• $w(f') = f'(\{s\}, V \setminus \{s\}).$

- Gegeben G = (V, E, s, t, c, I) und Restnetzwerk $G_f = (V', E', s, t, c', I')$.
- c'(e) = c(e) für $e \in E \cap E'$
- c'(a, b) = c(b, a) und l'(a, b) = -l(b, a) für $(a, b) \in E' \setminus E$
- f' ist eine Zirkulation, gdw.: $\forall v \in V : f'_{in}(v) = f'_{out}(v)$.
- Wert einer Zirkulation f' über Schnitt (S, T):

$$f'(S,T) = \sum_{(v,w)\in E, v\in S, w\in T} f'(v,w) - \sum_{(w,v)\in E, v\in S, w\in T} f'(w,v).$$

• $w(f') = f'(\{s\}, V \setminus \{s\}).$

- Gegeben G = (V, E, s, t, c, I) und Restnetzwerk $G_f = (V', E', s, t, c', I')$.
- c'(e) = c(e) für $e \in E \cap E'$
- c'(a,b) = c(b,a) und l'(a,b) = -l(b,a) für $(a,b) \in E' \setminus E$
- f' ist eine Zirkulation, gdw.: $\forall v \in V : f'_{in}(v) = f'_{out}(v)$.
- Wert einer Zirkulation f' über Schnitt (S, T):

$$f'(S,T) = \sum_{(v,w)\in E, v\in S, w\in T} f'(v,w) - \sum_{(w,v)\in E, v\in S, w\in T} f'(w,v).$$

•
$$w(f') = f'(\{s\}, V \setminus \{s\}).$$

Lemma

Es gilt:

Beweis: Flusserhaltung und Induktion über Größe von S.

- Gegeben G = (V, E, s, t, c, l) und Restnetzwerk $G_f = (V', E', s, t, c', l')$.
- c'(e) = c(e) für $e \in E \cap E'$
- c'(a,b) = c(b,a) und l'(a,b) = -l(b,a) für $(a,b) \in E' \setminus E$
- f' ist eine Zirkulation, gdw.: $\forall v \in V : f'_{in}(v) = f'_{out}(v)$.
- Wert einer Zirkulation f' über Schnitt (S, T):

$$f'(S,T) = \sum_{(v,w)\in E, v\in S, w\in T} f'(v,w) - \sum_{(w,v)\in E, v\in S, w\in T} f'(w,v).$$

• $w(f') = f'(\{s\}, V \setminus \{s\}).$

Lemma

Es gilt:

•
$$w(f') = 0$$
 und

Beweis: Flusserhaltung und Induktion über Größe von S.

- Gegeben G = (V, E, s, t, c, I) und Restnetzwerk $G_f = (V', E', s, t, c', I')$.
- c'(e) = c(e) für $e \in E \cap E'$
- c'(a,b) = c(b,a) und l'(a,b) = -l(b,a) für $(a,b) \in E' \setminus E$
- f' ist eine Zirkulation, gdw.: $\forall v \in V : f'_{in}(v) = f'_{out}(v)$.
- Wert einer Zirkulation f' über Schnitt (S, T):

$$f'(S,T) = \sum_{(v,w)\in E, v\in S, w\in T} f'(v,w) - \sum_{(w,v)\in E, v\in S, w\in T} f'(w,v).$$

• $w(f') = f'(\{s\}, V \setminus \{s\}).$

Lemma

Es gilt:

- w(f') = 0 und
- für jeden Schnitt (S, T) gilt: f'(S, T) = 0.

Beweis: Flusserhaltung und Induktion über Größe von S.

Spezielle Flüsse (Mindestfluss)

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Zirkulation und Kostenminimalität

Lemma

2:28 Idee 1/9

Falls f nicht kostenminimal ist, dann gibt es einen Kreis C in G_f und f hat auf C negative Kosten.

Beweis:

Spezielle Flüsse (Mindestfluss)

Spezielle Flüsse (A

Spezielle Flüsse (Alternativen) Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Zirkulation und Kostenminimalität

Lemma

2:28 Idee 2/9

Falls f nicht kostenminimal ist, dann gibt es einen Kreis C in G_f und f hat auf C negative Kosten.

Beweis:

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Zirkulation und Kostenminimalität

Lemma

2:28 Idee 3/9

Falls f nicht kostenminimal ist, dann gibt es einen Kreis C in Gf und f hat auf C negative Kosten.

Beweis:

• Sei f^* Fluss auf G mit $I(f^*) < I(f)$.

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Zirkulation und Kostenminimalität

Lemma

Idee 4/9

Falls f nicht kostenminimal ist, dann gibt es einen Kreis C in Gf und f hat auf C negative Kosten.

- Sei f^* Fluss auf G mit $I(f^*) < I(f)$.
- Damit unterscheiden sich f* und f.

Zirkulation und Kostenminimalität

Lemma

Idee 5/9

Falls f nicht kostenminimal ist, dann gibt es einen Kreis C in Gf und f hat auf C negative Kosten.

- Sei f^* Fluss auf G mit $I(f^*) < I(f)$.
- Damit unterscheiden sich f* und f.
- Für alle $e \in E$ setze $f'(e) = f^*(e) f(e)$.

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Zirkulation und Kostenminimalität

Lemma

Falls f nicht kostenminimal ist, dann gibt es einen Kreis C in G_f und f hat auf C negative Kosten.

- Sei f^* Fluss auf G mit $I(f^*) < I(f)$.
- Damit unterscheiden sich f^* und f.
- Für alle $e \in E$ setze $f'(e) = f^*(e) f(e)$.
- f' ist dann zyklischer Fluss.

Spezielle Flüsse (Alternativen)

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Flüsse mit Kostenfunktion

Zirkulation und Kostenminimalität

Lemma

Idee 7/9

Falls f nicht kostenminimal ist, dann gibt es einen Kreis C in Gf und f hat auf C negative Kosten.

- Sei f^* Fluss auf G mit $I(f^*) < I(f)$.
- Damit unterscheiden sich f* und f.
- Für alle $e \in E$ setze $f'(e) = f^*(e) f(e)$.
- f' ist dann zyklischer Fluss.
- f' wird durch höchstens m' < m viele Kreisflüsse f'_i $(1 \le i \le m')$ gebildet.

Flüsse mit Kostenfunktion

Lemma

Idee 8/9

Falls f nicht kostenminimal ist, dann gibt es einen Kreis C in Gf und f hat auf C negative Kosten.

- Sei f^* Fluss auf G mit $I(f^*) < I(f)$.
- Damit unterscheiden sich f* und f.
- Für alle $e \in E$ setze $f'(e) = f^*(e) f(e)$.
- f' ist dann zyklischer Fluss.
- f' wird durch höchstens m' < m viele Kreisflüsse f'_i $(1 \le i \le m')$ gebildet.
- Damit gilt: $I(f') = I(f^*) I(f) = \sum_{1 \le i \le m'} I(f_i')$

Zirkulation und Kostenminimalität

Lemma

Idee 9/9

Falls f nicht kostenminimal ist, dann gibt es einen Kreis C in Gf und f hat auf C negative Kosten.

- Sei f^* Fluss auf G mit $I(f^*) < I(f)$.
- Damit unterscheiden sich f* und f.
- Für alle $e \in E$ setze $f'(e) = f^*(e) f(e)$.
- f' ist dann zyklischer Fluss.
- f' wird durch höchstens m' < m viele Kreisflüsse f'_i $(1 \le i \le m')$ gebildet.
- Damit gilt: $I(f') = I(f^*) I(f) = \sum_{1 \le i \le m'} I(f_i')$
- Damit existiert j mit $I(f'_i) < 0$.

Spezielle Flüsse (Mindestfluss)
Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Algorithmus (Min-Cost-Flow)

Theorem

2:29 Algorithmus 1/13

Spezielle Flüsse (Mindestfluss)
Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Algorithmus (Min-Cost-Flow)

Theorem

2:29 Algorithmus 2/13

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Algorithmus (Min-Cost-Flow)

Theorem

2:29 Algorithmus 3/13

Ein Fluss f ist kostenminimal, wenn G_f keinen Kreis mit negativen Kosten enthält.

1 Eingabe: G = (V, E, s, t, c, I), W.

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Algorithmus (Min-Cost-Flow)

Theorem

2:29 Algorithmus 4/13

- **1** Eingabe: G = (V, E, s, t, c, I), W.
- **2** Bestimme Fluss f' mit w(f') = W.

zzielle Flüsse (Alternativen) Flüsse mit Kostenfunktion

Algorithmus (Min-Cost-Flow)

Theorem

2:29 Algorithmus 5/13

- **1** Eingabe: G = (V, E, s, t, c, I), W.
- **2** Bestimme Fluss f' mit w(f') = W.
- \odot Solange es in G_f einen negativen Kreis C gibt:

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion

Algorithmus 6/13 Walter Unger 6.11.2018 7:05 SS2015 RWTH

Algorithmus (Min-Cost-Flow)

Theorem

- **1** Eingabe: G = (V, E, s, t, c, I), W.
- **2** Bestimme Fluss f' mit w(f') = W.
- \odot Solange es in G_f einen negativen Kreis C gibt:
 - f = f + f'(C).

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Algorithmus (Min-Cost-Flow)

Theorem

Algorithmus 7/13

- **1** Eingabe: G = (V, E, s, t, c, I), W.
- **2** Bestimme Fluss f' mit w(f') = W.
- \odot Solange es in G_f einen negativen Kreis C gibt:
 - f = f + f'(C).

Theorem

- **1** Eingabe: G = (V, E, s, t, c, I), W.
- **2** Bestimme Fluss f' mit w(f') = W.
- \odot Solange es in G_f einen negativen Kreis C gibt:

$$f = f + f'(C)$$
.

• Es gilt:
$$w(f + f'(C)) = w(f) + w(f'(C)) = w(f) = W$$
.

Theorem

- **1** Eingabe: G = (V, E, s, t, c, I), W.
- **2** Bestimme Fluss f' mit w(f') = W.
- \odot Solange es in G_f einen negativen Kreis C gibt:

$$f = f + f'(C)$$
.

- Es gilt: w(f + f'(C)) = w(f) + w(f'(C)) = w(f) = W.
 - D.h. der Wert des Flusses bleibt gleich.

Theorem

- **1** Eingabe: G = (V, E, s, t, c, I), W.
- **2** Bestimme Fluss f' mit w(f') = W.
- \odot Solange es in G_f einen negativen Kreis C gibt:

$$f = f + f'(C)$$
.

- Es gilt: w(f + f'(C)) = w(f) + w(f'(C)) = w(f) = W.
 - D.h. der Wert des Flusses bleibt gleich.
- Es gilt: I(f + f'(C)) = I(f) + I(f'(C)) < I(f).

Theorem

- **1** Eingabe: G = (V, E, s, t, c, I), W.
- **2** Bestimme Fluss f' mit w(f') = W.
- \odot Solange es in G_f einen negativen Kreis C gibt:

$$f = f + f'(C)$$
.

- Es gilt: w(f + f'(C)) = w(f) + w(f'(C)) = w(f) = W.
 - D.h. der Wert des Flusses bleibt gleich.
- Es gilt: I(f + f'(C)) = I(f) + I(f'(C)) < I(f).
 - D.h. die Kosten verringern sich.

Theorem

- **1** Eingabe: G = (V, E, s, t, c, I), W.
- **2** Bestimme Fluss f' mit w(f') = W.
- \odot Solange es in G_f einen negativen Kreis C gibt:

$$f = f + f'(C)$$
.

- Es gilt: w(f + f'(C)) = w(f) + w(f'(C)) = w(f) = W.
 - D.h. der Wert des Flusses bleibt gleich.
- Es gilt: I(f + f'(C)) = I(f) + I(f'(C)) < I(f).
 - D.h. die Kosten verringern sich.

$\mathsf{Theorem}$

Ein Fluss f ist kostenminimal, wenn G_f keinen Kreis mit negativen Kosten enthält.

- **1** Eingabe: G = (V, E, s, t, c, l), W.
- **2** Bestimme Fluss f' mit w(f') = W.
- \odot Solange es in G_f einen negativen Kreis C gibt:

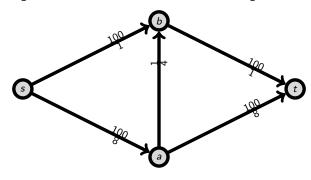
$$f = f + f'(C).$$

- Es gilt: w(f + f'(C)) = w(f) + w(f'(C)) = w(f) = W.
 - D.h. der Wert des Flusses bleibt gleich.
- Es gilt: I(f + f'(C)) = I(f) + I(f'(C)) < I(f).
 - D.h. die Kosten verringern sich.

$\mathsf{Theorem}$

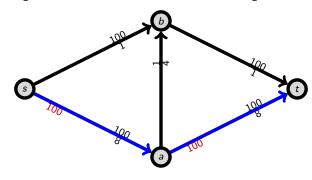
Der obige Algorithmus bestimmt einen kostenminimalen Fluss.

Beispiel zur Laufzeit (W = 100)

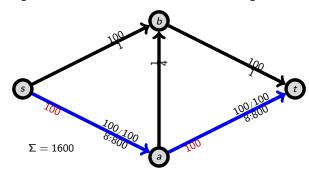


Spezielle Flüsse (Mindestfluss) 2:30 Algorithmus 2/46

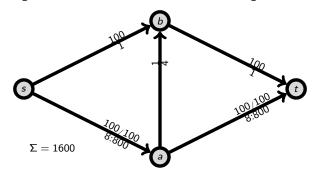
Beispiel zur Laufzeit (W = 100)



Beispiel zur Laufzeit (W = 100)

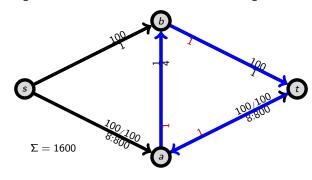


Beispiel zur Laufzeit (W = 100)

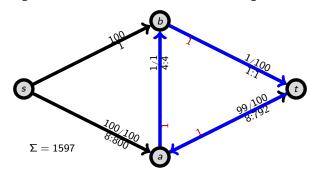


Spezielle Flüsse (Mindestfluss)

Beispiel zur Laufzeit (W = 100)



Beispiel zur Laufzeit (W = 100)

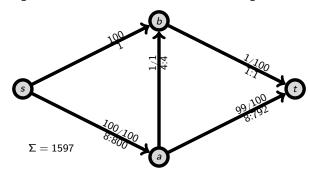


Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Beispiel zur Laufzeit (W = 100)

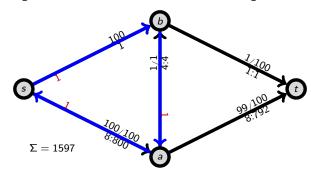
Spezielle Flüsse (Mindestfluss)

2:30 Algorithmus 7/46

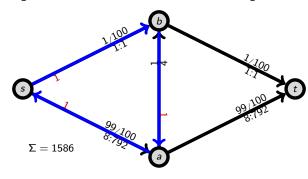


Spezielle Flüsse (Mindestfluss) 2:30 Algorithmus 8/46

Beispiel zur Laufzeit (W = 100)

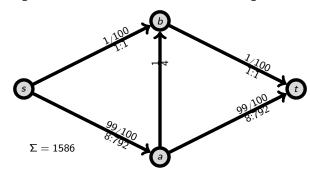


Beispiel zur Laufzeit (W = 100)

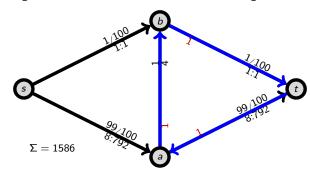


Walter Unger 6.11.2018 7:05 SS2015 RWTH

Beispiel zur Laufzeit (W = 100)



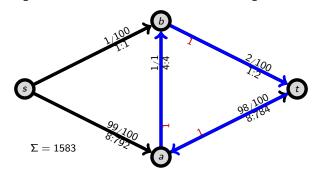
Beispiel zur Laufzeit (W = 100)



Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

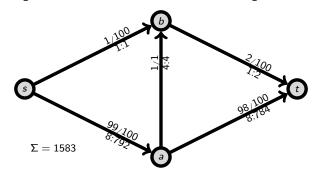
Spezielle Flüsse (Mindestfluss) 2:30 Algorithmus 12/46

Beispiel zur Laufzeit (W = 100)



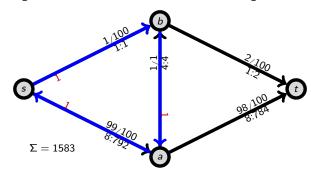
2:30 Algorithmus 13/46

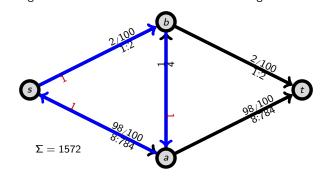
Beispiel zur Laufzeit (W = 100)



Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

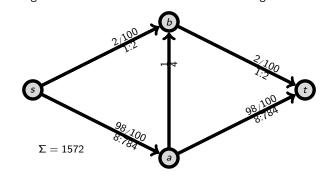
Beispiel zur Laufzeit (W = 100)





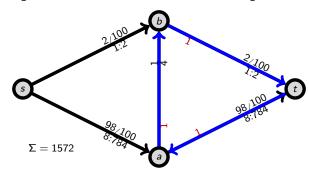
Flüsse mit Kostenfunktion

Spezielle Flüsse (Mindestfluss)



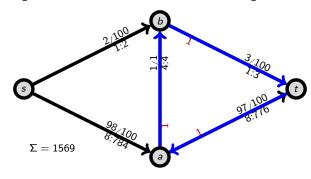
Spezielle Flüsse (Mindestfluss) 2:30 Algorithmus 17/46

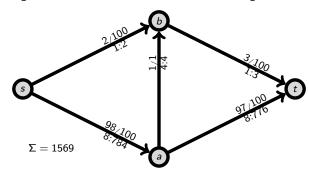
Beispiel zur Laufzeit (W = 100)

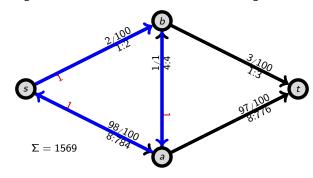


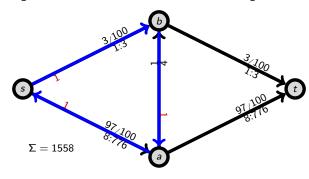
2:30 Algorithmus 18/46

Beispiel zur Laufzeit (W = 100)







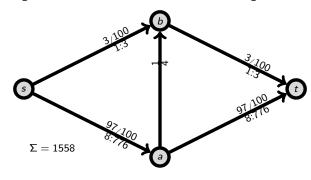


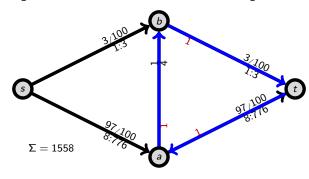
Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

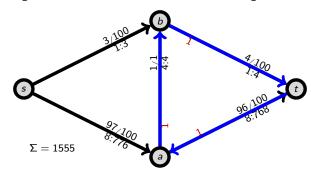
Beispiel zur Laufzeit (W = 100)

Spezielle Flüsse (Mindestfluss)

2:30 Algorithmus 22/46

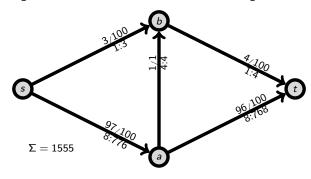


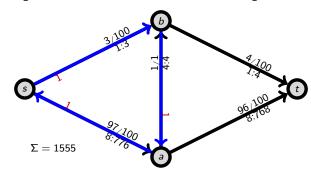


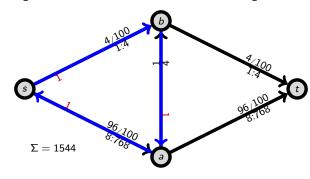


Walter Unger 6.11.2018 7:05 SS2015 RWTH

Beispiel zur Laufzeit (W = 100)

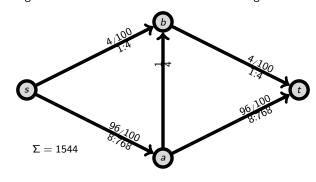


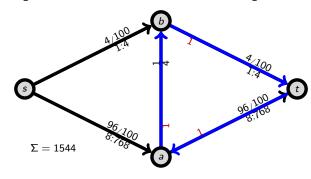


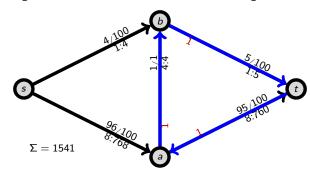


Spezielle Flüsse (Mindestfluss)

2:30 Algorithmus 28/46

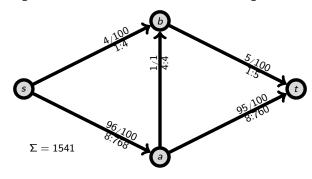


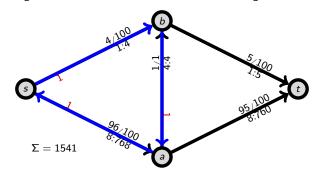


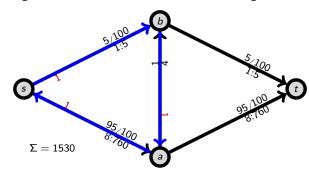


Flüsse mit Kostenfunktion

Beispiel zur Laufzeit (W = 100)

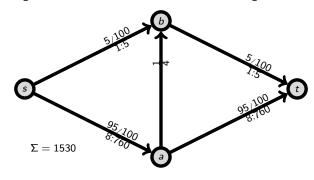


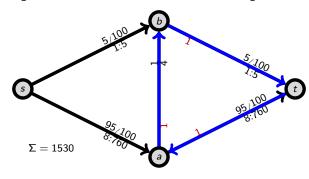


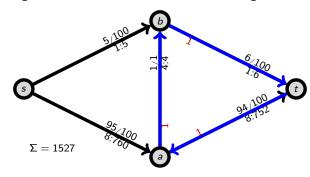


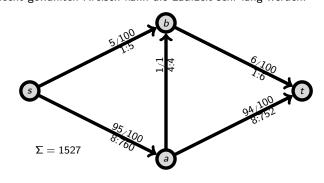
Spezielle Flüsse (Mindestfluss)

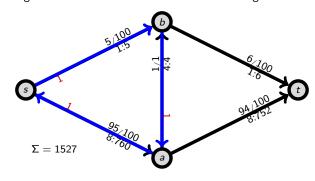
2:30 Algorithmus 34/46

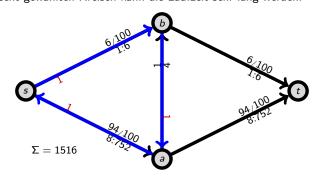


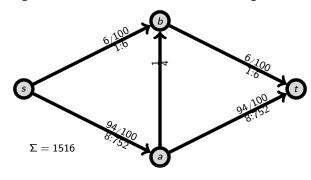


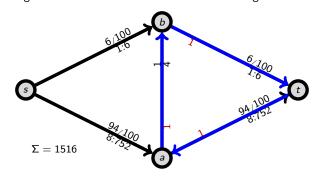


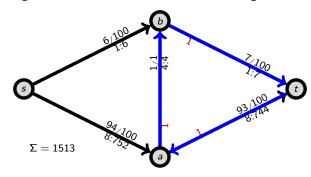


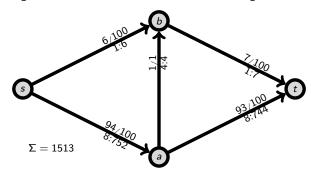


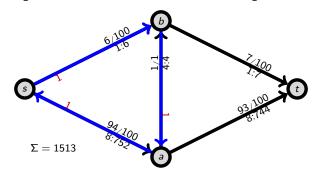


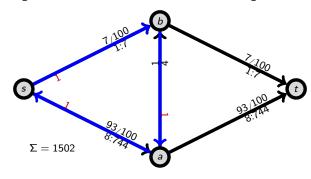






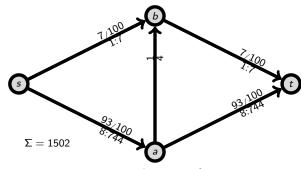






Spezielle Flüsse (Mindestfluss)

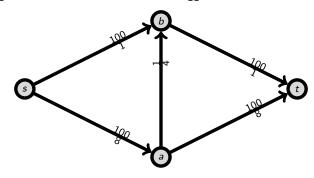
2:30 Algorithmus 46/46



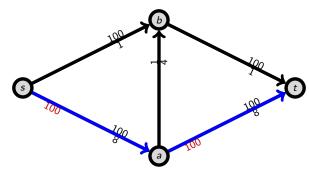
Hier geben wir auf.

Flüsse mit Kostenfunktion

Nochmal Beispiel zur Laufzeit (W = 100)

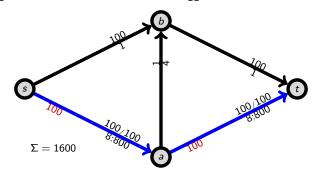


Nochmal Beispiel zur Laufzeit (W = 100)



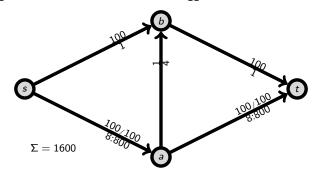
Flüsse mit Kostenfunktion

Nochmal Beispiel zur Laufzeit (W = 100)



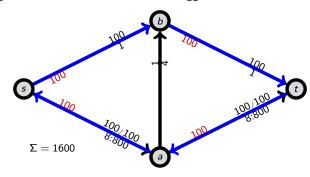
Walter Unger 6.11.2018 7:05 SS2015 RWTH

Nochmal Beispiel zur Laufzeit (W = 100)

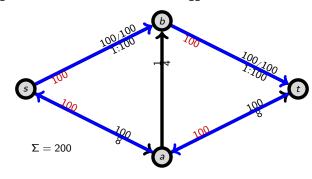


Spezielle Flüsse (Mindestfluss) 2:31 Algorithmus 5/7

Nochmal Beispiel zur Laufzeit (W = 100)



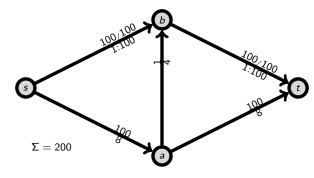
Nochmal Beispiel zur Laufzeit (W = 100)



Spezielle Flüsse (Mindestfluss) 2:31 Algorithmus 7/7

Nochmal Beispiel zur Laufzeit (W = 100)

Bei gut gewählten Kreisen kann die Laufzeit ggf. besser sein.



Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Laufzeit und Verbesserung selbiger

2:32 Verbesserung der Laufzeit 1/10

• Der bisherige Algorithmus hat in obiger Form eine pseudopolynomielle Laufzeit.

tiven) Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Laufzeit und Verbesserung selbiger

2:32 Verbesserung der Laufzeit 2/10

- Der bisherige Algorithmus hat in obiger Form eine pseudopolynomielle Laufzeit.
- Dies kann aber verbessert werden:

Laufzeit und Verbesserung selbiger

2:32 Verbesserung der Laufzeit 3/10

- Der bisherige Algorithmus hat in obiger Form eine pseudopolynomielle Laufzeit.
- Dies kann aber verbessert werden:
 - Wähle kostengünstige Kreise.

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Laufzeit und Verbesserung selbiger

2:32 Verbesserung der Laufzeit 4/10

- Der bisherige Algorithmus hat in obiger Form eine pseudopolynomielle Laufzeit.
- Dies kann aber verbessert werden:
 - Wähle kostengünstige Kreise.
 - Wähle Kreise über kostengünstige Kanten.

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Laufzeit und Verbesserung selbiger

2:32 Verbesserung der Laufzeit 5/10

- Der bisherige Algorithmus hat in obiger Form eine pseudopolynomielle Laufzeit.
- Dies kann aber verbessert werden:
 - Wähle kostengünstige Kreise.
 - Wähle Kreise über kostengünstige Kanten.
 - Also unabhängig von der Kreislänge.

Laufzeit und Verbesserung selbiger

2:32 Verbesserung der Laufzeit 6/10

- Der bisherige Algorithmus hat in obiger Form eine pseudopolynomielle Laufzeit.
- Dies kann aber verbessert werden:
 - Wähle kostengünstige Kreise.
 - Wähle Kreise über kostengünstige Kanten.
 - Also unabhängig von der Kreislänge.
- Dazu werden die Kreise gewählt, die die durchschnittlichen Kantenkosten minimieren.

- Der bisherige Algorithmus hat in obiger Form eine pseudopolynomielle Laufzeit.
- Dies kann aber verbessert werden:
 - Wähle kostengünstige Kreise.
 - Wähle Kreise über kostengünstige Kanten.
 - Also unabhängig von der Kreislänge.
- Dazu werden die Kreise gewählt, die die durchschnittlichen Kantenkosten minimieren.
- Setze:

$$\bar{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

- Der bisherige Algorithmus hat in obiger Form eine pseudopolynomielle Laufzeit.
- Dies kann aber verbessert werden:
 - Wähle kostengünstige Kreise.
 - Wähle Kreise über kostengünstige Kanten.
 - Also unabhängig von der Kreislänge.
- Dazu werden die Kreise gewählt, die die durchschnittlichen Kantenkosten minimieren.
- Setze:

$$\bar{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

• Setze weiter: $\mu(f) = -\bar{I}(C)$.

- Der bisherige Algorithmus hat in obiger Form eine pseudopolynomielle Laufzeit.
- Dies kann aber verbessert werden:
 - Wähle kostengünstige Kreise.
 - Wähle Kreise über kostengünstige Kanten.
 - Also unabhängig von der Kreislänge.
- Dazu werden die Kreise gewählt, die die durchschnittlichen Kantenkosten minimieren.
- Setze:

$$\bar{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

- Setze weiter: $\mu(f) = -\overline{l}(C)$.
- Falls C negative Kosten hat, so ist $\mu(C)$ positiv.

- Der bisherige Algorithmus hat in obiger Form eine pseudopolynomielle Laufzeit.
- Dies kann aber verbessert werden:
 - Wähle kostengünstige Kreise.
 - Wähle Kreise über kostengünstige Kanten.
 - Also unabhängig von der Kreislänge.
- Dazu werden die Kreise gewählt, die die durchschnittlichen Kantenkosten minimieren.
- Setze:

$$\bar{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

- Setze weiter: $\mu(f) = -\overline{I}(C)$.
- Falls C negative Kosten hat, so ist $\mu(C)$ positiv.
- $\mu(f)$ gibt also die Verbesserung eines Kreisflusses an.

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Mean-Algorithmus (Min-Cost-Flow)

2:33 Verbesserung der Laufzeit 1/8

$$\bar{l}(c) = \frac{l(c)}{|c|} = \frac{\sum_{e \in C} l(e)}{|c|}$$

$$\mu(f) = -\bar{l}(e)$$

1 Eingabe: G = (V, E, s, t, c, I), W.

2:33 Verbesserung der Laufzeit 2/8 Mean-Algorithmus (Min-Cost-Flow)

- Eingabe: G = (V, E, s, t, c, I), W.
- **2** Bestimme Fluss f mit w(f) = W.

$$I(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

2:33 Verbesserung der Laufzeit 3/8 Walter Unger 6.11.2018 7:05 SS2015 RWTH

Mean-Algorithmus (Min-Cost-Flow)
$$\bar{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{c \in C} I(c)}{|C|}$$

- **1** Eingabe: G = (V, E, s, t, c, I), W.
- **2** Bestimme Fluss f mit w(f) = W.
- \odot Solange es in G_f einen negativen Kreis C gibt,

2:33 Verbesserung der Laufzeit 4/8

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Mean-Algorithmus (Min-Cost-Flow)

$$\bar{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

$$\mu(f) = -\bar{I}(C)$$

- **1** Eingabe: G = (V, E, s, t, c, I), W.
- **2** Bestimme Fluss f mit w(f) = W.
- \odot Solange es in G_f einen negativen Kreis C gibt,
 - Wähle C mit $\overline{I}(C)$ minimal.

2:33 Verbesserung der Laufzeit 5/8

Mean-Algorithmus (Min-Cost-Flow)

$$\bar{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

$$\mu(f) = -\bar{I}(C)$$

- **1** Eingabe: G = (V, E, s, t, c, I), W.
- **2** Bestimme Fluss f mit w(f) = W.
- \odot Solange es in G_f einen negativen Kreis C gibt,
 - Wähle C mit $\overline{I}(C)$ minimal.
 - $oldsymbol{0}$ Bestimme maximalen zyklischen Fluss f' auf C.

2:33 Verbesserung der Laufzeit 6/8

Mean-Algorithmus (Min-Cost-Flow)

$$I(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

- Eingabe: G = (V, E, s, t, c, I), W.
- **2** Bestimme Fluss f mit w(f) = W.
- \odot Solange es in G_f einen negativen Kreis C gibt,
 - Wähle C mit $\overline{I}(C)$ minimal.
 - Bestimme maximalen zyklischen Fluss f' auf C.
 - f = f + f'(C).

2:33 Verbesserung der Laufzeit 7/8 Mean-Algorithmus (Min-Cost-Flow)

$$I(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

- **1** Eingabe: G = (V, E, s, t, c, I), W.
- **2** Bestimme Fluss f mit w(f) = W.
- \odot Solange es in G_f einen negativen Kreis C gibt,
 - Wähle C mit $\overline{I}(C)$ minimal.
 - $oldsymbol{0}$ Bestimme maximalen zyklischen Fluss f' auf C.
 - f = f + f'(C).
 - Es gilt: w(f + f'(C)) = w(f) + w(f'(C)) = w(f) = W.

2:33 Verbesserung der Laufzeit 8/8 Mean-Algorithmus (Min-Cost-Flow)

$$I(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

- **1** Eingabe: G = (V, E, s, t, c, I), W.
- **2** Bestimme Fluss f mit w(f) = W.
- \odot Solange es in G_f einen negativen Kreis C gibt.
 - Wähle C mit $\overline{I}(C)$ minimal.
 - **1** Bestimme maximalen zyklischen Fluss f' auf C.
 - f = f + f'(C).
 - Es gilt: w(f + f'(C)) = w(f) + w(f'(C)) = w(f) = W.
 - Es gilt: I(f + f'(C)) = I(f) + I(f'(C)) < I(f).

Walter Unger 6.11.2018 7:05 SS2015 RWTH

2:34 Verbesserung der Laufzeit 1/11 Überblick zum Beweis

$$\bar{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$
$$\mu(f) = -\bar{I}(C)$$
$$\mathbf{1} - x \leqslant e^{-x}, x = \mathbf{1}/n$$

• Teile Iterationen in Phasen auf.

 $\bar{l}(C) = \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|}$ $\mu(f) = -\bar{l}(C)$ $\mathbf{1} - x \leqslant e^{-x}, x = \mathbf{1}/n$

2:34 Verbesserung der Laufzeit 2/11 Überblick zum Beweis

- Teile Iterationen in Phasen auf.
- Bestimme die Anzahl der Phasen.

Walter Unger 6.11.2018 7:05 SS2015 RWTH

2:34 Verbesserung der Laufzeit 3/11 Überblick zum Beweis

Title Iterationen in Phasen auf.
$$\overline{l}(c) = \frac{l(c)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|}$$
• Teile Iterationen in Phasen auf.
$$\mathbf{1} - x \leqslant e^{-x}, x = \mathbf{1}/n$$

- Teile Iterationen in Phasen auf.
- Bestimme die Anzahl der Phasen.
- Bestimme die Anzahl der Iterationen pro Phase.

2:34 Verbesserung der Laufzeit 4/11

$$\overline{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

$$\mu(f) = -\overline{I}(C)$$

$$\mathbf{1} - \mathbf{x} \leq e^{-X}, \mathbf{x} = \mathbf{1}/n$$

- Teile Iterationen in Phasen auf.
- Bestimme die Anzahl der Phasen.
- Bestimme die Anzahl der Iterationen pro Phase.
 - Zeige dies unter Verwendung einer besonderen Annahme.

2:34 Verbesserung der Laufzeit 5/11 Überblick zum Beweis

Teile Iterationen in Phasen auf.

$$\bar{l}(C) = \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|}$$

$$\mu(f) = -\bar{l}(C)$$

$$\mathbf{1} - x \leqslant e^{-x}, x = \mathbf{1}/n$$

- Bestimme die Anzahl der Phasen.
- Bestimme die Anzahl der Iterationen pro Phase.
 - Zeige dies unter Verwendung einer besonderen Annahme.
 - Verändere I so, dass Annahme immer gilt und Algorithmus analog vorgeht.

2:34 Verbesserung der Laufzeit 6/11

- $\bar{l}(C) = \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|}$ $\mu(f) = -\bar{l}(C)$ $\mathbf{1} x \leqslant e^{-x}, x = \mathbf{1}/n$ Teile Iterationen in Phasen auf.
- Bestimme die Anzahl der Phasen.
- Bestimme die Anzahl der Iterationen pro Phase.
 - Zeige dies unter Verwendung einer besonderen Annahme.
 - Verändere I so, dass Annahme immer gilt und Algorithmus analog vorgeht.
 - Bestimme dazu Potential der Knoten, und addiere dies zu den Kantenkosten

$$\bar{l}(C) = \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|}$$

$$\mu(f) = -\bar{l}(C)$$

$$\mathbf{1} - x \leqslant e^{-x}, x = \mathbf{1}/n$$

- Teile Iterationen in Phasen auf.
- Bestimme die Anzahl der Phasen.
- Bestimme die Anzahl der Iterationen pro Phase.
 - Zeige dies unter Verwendung einer besonderen Annahme.
 - Verändere I so, dass Annahme immer gilt und Algorithmus analog vorgeht.
 - Bestimme dazu Potential der Knoten, und addiere dies zu den Kantenkosten
- Bestimme Laufzeit f
 ür eine Iteration.

 $1-x\leqslant e^{-x}, x=1/n$

2:34 Verbesserung der Laufzeit 8/11

- Teile Iterationen in Phasen auf.
- Bestimme die Anzahl der Phasen.
- Bestimme die Anzahl der Iterationen pro Phase.
 - Zeige dies unter Verwendung einer besonderen Annahme.
 - Verändere I so, dass Annahme immer gilt und Algorithmus analog vorgeht. Bestimme dazu Potential der Knoten, und addiere dies zu den
 - Kantenkosten
- Bestimme Laufzeit f
 ür eine Iteration.
- Zeige, wie man einfach einen Min-Mean-Kreis C findet (Dynamisches Programmieren).

2:34 Verbesserung der Laufzeit 9/11

- $1-x\leqslant e^{-x}, x=1/n$
- Teile Iterationen in Phasen auf.
- Bestimme die Anzahl der Phasen.
- Bestimme die Anzahl der Iterationen pro Phase.
 - Zeige dies unter Verwendung einer besonderen Annahme.
 - Verändere I so, dass Annahme immer gilt und Algorithmus analog vorgeht.
 - Bestimme dazu Potential der Knoten, und addiere dies zu den Kantenkosten
- Bestimme Laufzeit f
 ür eine Iteration.
- Zeige, wie man einfach einen Min-Mean-Kreis C findet (Dynamisches Programmieren).
 - Bestimme dazu vorah den Wert von C

2:34 Verbesserung der Laufzeit 10/11 Überblick zum Beweis

$$\overline{l}(c) = \frac{\overline{l}(c)}{|c|} = \frac{\sum_{e \in C} \overline{l}(e)}{|c|}$$

$$\mu(f) = -\overline{l}(c)$$
• Teile Iterationen in Phasen auf.
$$1 - x \leqslant e^{-x}, x = 1/n$$

- Teile Iterationen in Phasen auf.
- Bestimme die Anzahl der Phasen
- Bestimme die Anzahl der Iterationen pro Phase.
 - Zeige dies unter Verwendung einer besonderen Annahme.
 - Verändere I so, dass Annahme immer gilt und Algorithmus analog vorgeht. Bestimme dazu Potential der Knoten, und addiere dies zu den
 - Kantenkosten
- Bestimme Laufzeit f
 ür eine Iteration.
- Zeige, wie man einfach einen Min-Mean-Kreis C findet (Dynamisches Programmieren).
 - Bestimme dazu vorah den Wert von C
 - Dann wird Suche nach C einfach (passe Kosten an).

 $1-x \leqslant e^{-x}, x=1/n$

Überblick zum Beweis

Verbesserung der Laufzeit 11/11

$\bar{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$ $\mu(f) = -\bar{I}(C)$

- Teile Iterationen in Phasen auf.
- Bestimme die Anzahl der Phasen
- Bestimme die Anzahl der Iterationen pro Phase.
 - Zeige dies unter Verwendung einer besonderen Annahme.
 - Verändere I so, dass Annahme immer gilt und Algorithmus analog vorgeht.
 - Bestimme dazu Potential der Knoten, und addiere dies zu den Kantenkosten
- Bestimme Laufzeit f
 ür eine Iteration.
- Zeige, wie man einfach einen Min-Mean-Kreis C findet (Dynamisches Programmieren).
 - Bestimme dazu vorah den Wert von C
 - Dann wird Suche nach C einfach (passe Kosten an).
- Beachte für verbessernde Kreise C gilt: $\bar{l}(C) \leqslant -1/n$.

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Laufzeit (Aufteilung in Phasen)

2:35 Verbesserung der Laufzeit 1/10

• Sei f der Fluss zu Beginn der i-ten Phase.

$$\bar{l}(C) = \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|}$$
$$\mu(f) = -\bar{l}(C)$$
$$\mathbf{1} - x \leqslant e^{-x}, x = \mathbf{1}/n$$

Laufzeit (Aufteilung in Phasen)

Verbesserung der Laufzeit 2/10

- Sei f der Fluss zu Beginn der i-ten Phase.
- Die Phase i endet, falls ein Fluss g gefunden wird mit:

$$\mu(g) \leqslant (1 - 1/n) \cdot \mu(f)$$
 oder $\mu(g) \leqslant 0$.

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Laufzeit (Aufteilung in Phasen)

Verbesserung der Laufzeit 3/10

- Sei f der Fluss zu Beginn der i-ten Phase.
- Die Phase i endet, falls ein Fluss g gefunden wird mit:

$$\mu(g) \leqslant (1 - 1/n) \cdot \mu(f) \text{ oder } \mu(g) \leqslant 0.$$

• Falls $\mu(g) \leq 0$ terminiert der Algorithmus.

Laufzeit (Aufteilung in Phasen)

Verbesserung der Laufzeit 4/10

$$\bar{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

- Sei f der Fluss zu Beginn der i-ten Phase.
- Die Phase i endet, falls ein Fluss g gefunden wird mit:

$$\mu(g)\leqslant (1-1/n)\cdot \mu(f)$$
 oder $\mu(g)\leqslant 0.$

- Falls $\mu(g) \leq 0$ terminiert der Algorithmus.
- Sei μ_0 der Wert von μ zu Beginn der ersten Phase.

Verbesserung der Laufzeit 5/10 Laufzeit (Aufteilung in Phasen)

- Sei f der Fluss zu Beginn der i-ten Phase.
- Die Phase i endet, falls ein Fluss g gefunden wird mit:

$$\mu(g)\leqslant (1-1/n)\cdot \mu(f)$$
 oder $\mu(g)\leqslant 0$.

- Falls $\mu(g) \leq 0$ terminiert der Algorithmus.
- Sei μ_0 der Wert von μ zu Beginn der ersten Phase.
- Sei μ_i der Wert von μ am Ende der *i*-ten Phase $(1 \le i \le T)$.

Verbesserung der Laufzeit 6/10

Laufzeit (Aufteilung in Phasen)

- Sei f der Fluss zu Beginn der i-ten Phase.
- Die Phase i endet, falls ein Fluss g gefunden wird mit:

$$\mu(g)\leqslant (1-1/n)\cdot \mu(f)$$
 oder $\mu(g)\leqslant 0$.

- Falls $\mu(g) \leq 0$ terminiert der Algorithmus.
- Sei μ_0 der Wert von μ zu Beginn der ersten Phase.
- Sei μ_i der Wert von μ am Ende der *i*-ten Phase $(1 \le i \le T)$.
- Damit gilt:

$$\mu_i \leqslant (1-1/n) \cdot \mu_{i-1} \leqslant \frac{\mu_{i-1}}{e^{1/n}}.$$

Walter Unger 6.11.2018 7:05 SS2015 RWIH

Laufzeit (Aufteilung in Phasen)

Verbesserung der Laufzeit 7/10

- Sei f der Fluss zu Beginn der i-ten Phase.
- Die Phase i endet, falls ein Fluss g gefunden wird mit:

$$\mu(g)\leqslant (1-1/n)\cdot \mu(f)$$
 oder $\mu(g)\leqslant 0$.

- Falls $\mu(g) \leq 0$ terminiert der Algorithmus.
- Sei μ_0 der Wert von μ zu Beginn der ersten Phase.
- Sei μ_i der Wert von μ am Ende der *i*-ten Phase $(1 \le i \le T)$.
- Damit gilt:

$$\mu_i \leqslant (1-1/n) \cdot \mu_{i-1} \leqslant \frac{\mu_{i-1}}{e^{1/n}}.$$

• Weiter gilt: $\mu_0 \leqslant L = \sum_{e \in F} |I(e)|$

 $\overline{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} \overline{I}(e)}{|C|}$ $\mu(f) = -\overline{I}(C)$ $\mathbf{1} - x \leqslant e^{-x}, x = \mathbf{1}/n$

Laufzeit (Aufteilung in Phasen)

Verbesserung der Laufzeit 8/10

- Sei f der Fluss zu Beginn der i-ten Phase.
- Die Phase i endet, falls ein Fluss g gefunden wird mit:

$$\mu(\mathsf{g})\leqslant (1-1/\mathsf{n})\cdot \mu(\mathsf{f}) \;\mathsf{oder}\; \mu(\mathsf{g})\leqslant 0.$$

- Falls $\mu(g) \leq 0$ terminiert der Algorithmus.
- Sei μ_0 der Wert von μ zu Beginn der ersten Phase.
- Sei μ_i der Wert von μ am Ende der *i*-ten Phase $(1 \le i \le T)$.
- Damit gilt:

$$\mu_i \leqslant (1-1/n) \cdot \mu_{i-1} \leqslant \frac{\mu_{i-1}}{e^{1/n}}.$$

- Weiter gilt: $\mu_0 \leqslant L = \sum_{e \in F} |I(e)|$
- Wegen der Ganzzahligkeit gilt: $\mu_{T-1} \ge 1/n$.

 $\bar{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$ $\mu(f) = -\bar{I}(C)$ $\mathbf{1} - x \leqslant e^{-x}, x = \mathbf{1}/n$

Laufzeit (Aufteilung in Phasen)

Verbesserung der Laufzeit 9/10

- Sei f der Fluss zu Beginn der i-ten Phase.
- Die Phase i endet, falls ein Fluss g gefunden wird mit:

$$\mu(g)\leqslant (1-1/n)\cdot \mu(f) ext{ oder } \mu(g)\leqslant 0.$$

- Falls $\mu(g) \leq 0$ terminiert der Algorithmus.
- Sei μ_0 der Wert von μ zu Beginn der ersten Phase.
- Sei μ_i der Wert von μ am Ende der *i*-ten Phase $(1 \le i \le T)$.
- Damit gilt:

$$\mu_i \leqslant (1-1/n) \cdot \mu_{i-1} \leqslant \frac{\mu_{i-1}}{e^{1/n}}.$$

- Weiter gilt: $\mu_0 \leqslant L = \sum_{e \in F} |I(e)|$
- Wegen der Ganzzahligkeit gilt: $\mu_{T-1} \ge 1/n$.
- Es folgt:

$$T-1\leqslant \log_{\mathrm{e}^{1/n}}(nL)=\frac{\ln(nL)}{\ln(e^{1/n})}=n\ln(nL)$$

Laufzeit (Aufteilung in Phasen)

$$\bar{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$
$$\mu(f) = -\bar{I}(C)$$
$$\mathbf{1} - x \leqslant e^{-x}, x = \mathbf{1}/n$$

- Sei f der Fluss zu Beginn der i-ten Phase.
- Die Phase i endet, falls ein Fluss g gefunden wird mit:

$$\mu(g) \leqslant (1 - 1/n) \cdot \mu(f) \text{ oder } \mu(g) \leqslant 0.$$

- Falls $\mu(g) \leq 0$ terminiert der Algorithmus.
- Sei μ_0 der Wert von μ zu Beginn der ersten Phase.
- Sei μ_i der Wert von μ am Ende der *i*-ten Phase $(1 \le i \le T)$.
- Damit gilt:

$$\mu_i \leqslant (1-1/n) \cdot \mu_{i-1} \leqslant \frac{\mu_{i-1}}{e^{1/n}}.$$

- Weiter gilt: $\mu_0 \leqslant L = \sum_{e \in F} |I(e)|$
- Wegen der Ganzzahligkeit gilt: $\mu_{T-1} \ge 1/n$.
- Es folgt:

$$T-1 \leqslant \log_{e^{1/n}}(nL) = \frac{\ln(nL)}{\ln(e^{1/n})} = n\ln(nL)$$

• Und: $T \leq n \ln(nL) + 1$.

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Laufzeit (Iterationen pro Phase)

• Zeige im Folgenden:

2:36 Verbesserung der Laufzeit 1/9

2:36 Verbesserung der Laufzeit 2/9

- Zeige im Folgenden:
 - Die Phase (und der Algorithmus) terminiert nach spätestens m Iterationen, oder

2:36 Verbesserung der Laufzeit 3/9

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion

$$\bar{l}(c) = \frac{l(c)}{|c|} = \frac{\sum_{e \in C} l(e)}{|c|}$$
Zeige im Folgenden:
$$\mu(f) = -\bar{l}(c)$$

- Zeige im Folgenden:
 - Die Phase (und der Algorithmus) terminiert nach spätestens m Iterationen, oder
 - Die nächste Phase startet nach spätestens m-1 Iterationen.

2:36 Verbesserung der Laufzeit 4/9

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Zeige im Folgenden:
$$\vec{l}(c) = \frac{\vec{l}(c)}{|c|} = \frac{\sum_{e \in C} \vec{l}(e)}{|c|}$$

$$\mu(f) = -\vec{l}(c)$$

- Zeige im Folgenden:
 - Die Phase (und der Algorithmus) terminiert nach spätestens m Iterationen, oder
 - Die nächste Phase startet nach spätestens m-1 Iterationen.
 - D.h. war der initiale Fluss f zu Beginn der Phase,

2:36 Verbesserung der Laufzeit 5/9

Walter Unger 6.11.2018 7:05 SS2015 **RWTH**

$$\bar{l}(c) = \frac{l(c)}{|c|} = \frac{\sum_{e \in C} l(e)}{|c|}$$
Zeige im Folgenden:
$$\mu(f) = -\bar{l}(C)$$

- Zeige im Folgenden:
 - Die Phase (und der Algorithmus) terminiert nach spätestens m Iterationen, oder
 - ullet Die nächste Phase startet nach spätestens m-1 Iterationen.
 - D.h. war der initiale Fluss f zu Beginn der Phase,
 - ullet dann ist ein Fluss g nach spätestens m-1 Iterationen erreicht mit:

2:36 Verbesserung der Laufzeit 6/9

Walter Unger 6.11.2018 7:05 SS2015 RWTH

$$\overline{l}(c) = \frac{l(c)}{|c|} = \frac{\sum_{e \in C} l(e)}{|c|}$$
Zeige im Folgenden:
$$\mu(f) = -\overline{l}(c)$$

- Zeige im Folgenden:
 - Die Phase (und der Algorithmus) terminiert nach spätestens m Iterationen, oder
 - Die nächste Phase startet nach spätestens m-1 Iterationen.
 - D.h. war der initiale Fluss f zu Beginn der Phase,
 - dann ist ein Fluss g nach spätestens m-1 Iterationen erreicht mit:
 - $\mu(g) \leq (1 1/n) \cdot \mu(f)$.

2:36 Verbesserung der Laufzeit 7/9 Walter Unger 6.11.2018 7:05 SS2015 RWTH

$$C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

$$\mu(f) = -\overline{I}(C)$$

- Zeige im Folgenden:
 - Die Phase (und der Algorithmus) terminiert nach spätestens m Iterationen, oder
 - Die nächste Phase startet nach spätestens m-1 Iterationen.
 - D.h. war der initiale Fluss f zu Beginn der Phase,
 - dann ist ein Fluss g nach spätestens m-1 Iterationen erreicht mit:
 - $\mu(g) \leq (1 1/n) \cdot \mu(f)$.
- Vorgehen:

Walter Unger 6.11.2018 7:05 SS2015 RWTH

$$\overline{I}(C) = \frac{|C|}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$
Zeige im Folgenden:

- Zeige im Folgenden:
 - Die Phase (und der Algorithmus) terminiert nach spätestens m Iterationen, oder
 - Die nächste Phase startet nach spätestens m-1 Iterationen.
 - D.h. war der initiale Fluss f zu Beginn der Phase.
 - dann ist ein Fluss g nach spätestens m-1 Iterationen erreicht mit:
 - $\mu(g) \leq (1 1/n) \cdot \mu(f)$.
- Vorgehen:
 - Zeige Behauptung unter der Annahme: $\forall e \in E(G_f) : I(e) \ge -\mu(f)$.

Walter Unger 6.11.2018 7:05 SS2015 **RWTH**

$$C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

$$u(f) = -\overline{I}(C)$$

- Zeige im Folgenden:
 - Die Phase (und der Algorithmus) terminiert nach spätestens m Iterationen, oder
 - ullet Die nächste Phase startet nach spätestens m-1 Iterationen.
 - D.h. war der initiale Fluss f zu Beginn der Phase,
 - ullet dann ist ein Fluss g nach spätestens m-1 Iterationen erreicht mit:
 - $\mu(g) \leqslant (1 1/n) \cdot \mu(f)$.
- Vorgehen:
 - Zeige Behauptung unter der Annahme: $\forall e \in E(G_f) : I(e) \geqslant -\mu(f)$.
 - Zeige Behauptung: Verändere dann I, so dass die Annahme immer gilt.

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Laufzeit (Iterationen pro Phase)

2:37 Verbesserung der Laufzeit 1/7

 $\overline{\mathit{f}(c)} = \frac{\mathit{l(c)}}{|c|} = \frac{\sum_{e \in C} \mathit{l(e)}}{|c|}$ • Typ 1 Iteration: Kreis C enthält nur Kanten mit negativen Kosten $\varepsilon(G_F)$: $\mathit{l(e)} \geqslant -\mu(f)$

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Laufzeit (Iterationen pro Phase)

 $\overline{I(c)} = \frac{I(c)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$ • Typ 1 Iteration: Kreis C enthält nur Kanten mit negativen Kosten $\varepsilon(G_f): I(e) \geqslant -\mu(f)$

- Typ 2 Iteration: Kreis C enthält mindestens eine Kante mit positiven
- Kosten.

2:37 Verbesserung der Laufzeit 3/7

$$\overline{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

- Typ 1 Iteration: Kreis C enthält nur Kanten mit negativen Kosten $E(G_f): \mathcal{I}(e) \geqslant -\mu(f)$
- ullet Typ 2 Iteration: Kreis C enthält mindestens eine Kante mit positiven Kosten.
- Bei jeder Typ 1 Iteration wird mindestens eine Kante saturiert und entfernt.

2:37 Verbesserung der Laufzeit 4/7

$$\bar{I}(C) = \frac{I(C)}{I(C)} = \frac{\sum_{e \in C} I(e)}{\sum_{e \in C} I(e)}$$

- Typ 1 Iteration: Kreis C enthält nur Kanten mit negativen Kosten $\varepsilon(G_f): \ell(e) \geqslant -\mu(f)$
- ullet Typ 2 Iteration: Kreis C enthält mindestens eine Kante mit positiven Kosten.
- Bei jeder Typ 1 Iteration wird mindestens eine Kante saturiert und entfernt.
- Alle dabei neu entstehenden Kanten haben positive Kosten (andere Richtung).

$$\bar{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

- Typ 1 Iteration: Kreis C enthält nur Kanten mit negativen Kosten $\mathcal{E}(G_f): \ell(e) \geqslant -\mu(f)$
- Typ 2 Iteration: Kreis *C* enthält mindestens eine Kante mit positiven Kosten.
- Bei jeder Typ 1 Iteration wird mindestens eine Kante saturiert und entfernt.
- Alle dabei neu entstehenden Kanten haben positive Kosten (andere Richtung).
- Nach spätestens m konsekutiven Typ 1 Iterationen terminiert das Verfahren.

2:37 Verbesserung der Laufzeit 6/7

$$\bar{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

- Typ 1 Iteration: Kreis C enthält nur Kanten mit negativen Kosten $\varepsilon(G_f): \ell(e) \geqslant -\mu(f)$
- Typ 2 Iteration: Kreis C enthält mindestens eine Kante mit positiven Kosten.
- Bei jeder Typ 1 Iteration wird mindestens eine Kante saturiert und entfernt.
- Alle dabei neu entstehenden Kanten haben positive Kosten (andere Richtung).
- Nach spätestens m konsekutiven Typ 1 Iterationen terminiert das Verfahren.
- Wenn also die Phase mehr als m Iterationen hat, folgt nach spätestens m-1 Iterationen eine Typ 2 Iteration.

2:37 Verbesserung der Laufzeit 7/7

$$\bar{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

$$u(f) = -\bar{I}(C)$$

- Typ 1 Iteration: Kreis C enthält nur Kanten mit negativen Kosten $\mathcal{E}(G_f): l(e) \geqslant -\mu(f)$
- Typ 2 Iteration: Kreis C enthält mindestens eine Kante mit positiven Kosten.
- Bei jeder Typ 1 Iteration wird mindestens eine Kante saturiert und entfernt.
- Alle dabei neu entstehenden Kanten haben positive Kosten (andere Richtung).
- Nach spätestens m konsekutiven Typ 1 Iterationen terminiert das Verfahren.
- Wenn also die Phase mehr als m Iterationen hat, folgt nach spätestens m-1 Iterationen eine Typ 2 Iteration.
- Wir zeigen nun, dass dieser Fall unter der Annahme nicht auftritt.

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Laufzeit mit Annahmen (Iterationen pro Phase)

 $\overline{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}, \ \mu(f) = -\overline{I}(C), \ \text{Annahme:} \ \forall e \in E(G_f) : I(e) \geqslant -\mu(f)$ • Sei g der Fluss vor der ersten Typ 2 Iteration.

Verbesserung der Laufzeit 2/9

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Laufzeit mit Annahmen (Iterationen pro Phase)

$$\bar{l}(C) = \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|}, \ \mu(f) = -\bar{l}(C), \ \text{Annahme:} \ \forall e \in E(G_f) : l(e) \geqslant -\mu(f)$$

- Sei g der Fluss vor der ersten Typ 2 Iteration.
- Die Annahme gilt weiter (keine neuen Kanten mit negativen Kosten):

$$\forall e \in E(G_g) : I(e) \geqslant -\mu(f)$$

Verbesserung der Laufzeit 3/9

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Laufzeit mit Annahmen (Iterationen pro Phase)

$$\bar{l}(C) = \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|}, \ \mu(f) = -\bar{l}(C), \ \text{Annahme:} \ \forall e \in E(G_f) : l(e) \geqslant -\mu(f)$$

- Sei g der Fluss vor der ersten Typ 2 Iteration.
- Die Annahme gilt weiter (keine neuen Kanten mit negativen Kosten):

$$\forall e \in E(G_g) : I(e) \geqslant -\mu(f)$$

• Sei C der Min-Mean-Kreis in G_g und H die Kanten mit negativen Kosten in C.

$$\bar{l}(C) = \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|}, \ \mu(f) = -\bar{l}(C), \ \text{Annahme:} \ \forall e \in E(G_f) : l(e) \geqslant -\mu(f)$$

- Sei g der Fluss vor der ersten Typ 2 Iteration.
- Die Annahme gilt weiter (keine neuen Kanten mit negativen Kosten):

$$\forall e \in E(G_g) : I(e) \geqslant -\mu(f)$$

- Sei C der Min-Mean-Kreis in G_g und H die Kanten mit negativen Kosten in C.
- Damit gilt:

$$\mu(g) = \sum_{e \in C} \frac{-l(e)}{|C|} \leqslant \sum_{e \in H} \frac{-l(e)}{|C|} \leqslant |H| \cdot \frac{\mu(f)}{|C|}$$

$$\overline{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}, \ \mu(f) = -\overline{I}(C), \ \text{Annahme:} \ \forall e \in E(G_f) : I(e) \geqslant -\mu(f)$$

- Sei g der Fluss vor der ersten Typ 2 Iteration.
- Die Annahme gilt weiter (keine neuen Kanten mit negativen Kosten):

$$\forall e \in E(G_g) : I(e) \geqslant -\mu(f)$$

- Sei C der Min-Mean-Kreis in G_g und H die Kanten mit negativen Kosten in C.
- Damit gilt:

Verbesserung der Laufzeit 5/9

$$\mu(g) = \sum_{e \in C} \frac{-l(e)}{|C|} \leqslant \sum_{e \in H} \frac{-l(e)}{|C|} \leqslant |H| \cdot \frac{\mu(f)}{|C|}$$

• Wegen $|H| \leq |C| - 1$ folgt:

$$|H|/|C| \le 1 - 1/|C| \le 1 - 1/n$$

$$\overline{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}, \ \mu(f) = -\overline{I}(C), \ \text{Annahme:} \ \forall e \in E(G_f) : I(e) \geqslant -\mu(f)$$

- Sei g der Fluss vor der ersten Typ 2 Iteration.
- Die Annahme gilt weiter (keine neuen Kanten mit negativen Kosten):

$$\forall e \in E(G_g) : I(e) \geqslant -\mu(f)$$

- Sei C der Min-Mean-Kreis in G_g und H die Kanten mit negativen Kosten in C.
- Damit gilt:

$$\mu(g) = \sum_{e \in C} \frac{-l(e)}{|C|} \leqslant \sum_{e \in H} \frac{-l(e)}{|C|} \leqslant |H| \cdot \frac{\mu(f)}{|C|}$$

• Wegen $|H| \leqslant |C| - 1$ folgt:

$$|H|/|C| \le 1 - 1/|C| \le 1 - 1/n$$

• Damit: $\mu(g) \leqslant (1 - 1/n) \cdot \mu(f)$.

Verbesserung der Laufzeit 7/9

Laufzeit mit Annahmen (Iterationen pro Phase)

$$\overline{l}(C) = \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|}, \ \mu(f) = -\overline{l}(C), \ \text{Annahme:} \ \forall e \in E(G_f) : l(e) \geqslant -\mu(f)$$

- Sei g der Fluss vor der ersten Typ 2 Iteration.
- Die Annahme gilt weiter (keine neuen Kanten mit negativen Kosten):

$$\forall e \in E(G_g) : I(e) \geqslant -\mu(f)$$

- ullet Sei C der Min-Mean-Kreis in G_g und H die Kanten mit negativen Kosten in C.
- Damit gilt:

$$\mu(g) = \sum_{e \in C} \frac{-l(e)}{|C|} \leqslant \sum_{e \in H} \frac{-l(e)}{|C|} \leqslant |H| \cdot \frac{\mu(f)}{|C|}$$

• Wegen $|H| \leq |C| - 1$ folgt:

$$|H|/|C| \le 1 - 1/|C| \le 1 - 1/n$$

- Damit: $\mu(g) \leq (1 1/n) \cdot \mu(f)$.
- Widerspruch: sind schon am Ende der Phase.

Verbesserung der Laufzeit 8/9

Laufzeit mit Annahmen (Iterationen pro Phase)

$$\overline{l}(C) = \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|}, \quad \mu(f) = -\overline{l}(C), \quad \text{Annahme: } \forall e \in E(G_f) : l(e) \geqslant -\mu(f)$$

- Sei g der Fluss vor der ersten Typ 2 Iteration.
- Die Annahme gilt weiter (keine neuen Kanten mit negativen Kosten):

$$\forall e \in E(G_g) : I(e) \geqslant -\mu(f)$$

- Sei C der Min-Mean-Kreis in G_g und H die Kanten mit negativen Kosten in C.
- Damit gilt:

$$\mu(g) = \sum_{e \in C} \frac{-l(e)}{|C|} \leqslant \sum_{e \in H} \frac{-l(e)}{|C|} \leqslant |H| \cdot \frac{\mu(f)}{|C|}$$

• Wegen $|H| \leq |C| - 1$ folgt:

$$|H|/|C| \le 1 - 1/|C| \le 1 - 1/n$$

- Damit: $\mu(g) \leq (1 1/n) \cdot \mu(f)$.
- Widerspruch: sind schon am Ende der Phase.
- Also gibt es in der Phase keine Typ 2 Iterationen.

$$\bar{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}, \ \mu(f) = -\bar{I}(C), \ \text{Annahme:} \ \forall e \in E(G_f) : I(e) \geqslant -\mu(f)$$

- Sei g der Fluss vor der ersten Typ 2 Iteration.
- Die Annahme gilt weiter (keine neuen Kanten mit negativen Kosten):

$$\forall e \in E(G_g) : I(e) \geqslant -\mu(f)$$

- Sei C der Min-Mean-Kreis in G_g und H die Kanten mit negativen Kosten in C.
- Damit gilt:

$$\mu(g) = \sum_{e \in C} \frac{-l(e)}{|C|} \leqslant \sum_{e \in H} \frac{-l(e)}{|C|} \leqslant |H| \cdot \frac{\mu(f)}{|C|}$$

• Wegen $|H| \leqslant |C| - 1$ folgt:

$$|H|/|C| \le 1 - 1/|C| \le 1 - 1/n$$

- Damit: $\mu(g) \leq (1 1/n) \cdot \mu(f)$.
- Widerspruch: sind schon am Ende der Phase.
- Also gibt es in der Phase keine Typ 2 Iterationen.
- Falls die Annahme gilt, so endet die Phase nach m-1 Typ 1 Iterationen.

se (Alternativen) Flüsse mit Kostenfunktion

Laufzeit (Erzwinge die Annahme)

2:39 Verbesserung der Laufzeit 1/9

$$\overline{l}(C) = \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|}, \ \mu(f) = -\overline{l}(C), \ \text{Annahme: } \forall e \in E(G_f) : l(e) \geqslant -\mu(f)$$

 $l(C) = \frac{|C|}{|C|} = \frac{|D|}{|C|}, \ \mu(f) = -l(C), \ \text{Annahme:} \ \forall e \in E(G_f) : l(e) \geqslant -\mu$ • Wir sorgen dafür, dass die Annahme erfüllt werden kann:

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Laufzeit (Erzwinge die Annahme)

$$\bar{l}(C) = \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|}, \ \mu(f) = -\bar{l}(C), \ \text{Annahme:} \ \forall e \in E(G_f) : l(e) \geqslant -\mu(f)$$

- Wir sorgen dafür, dass die Annahme erfüllt werden kann:
- Wir verändern / geeignet.

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Laufzeit (Erzwinge die Annahme)

2:39 Verbesserung der Laufzeit 3/9

$$\overline{l}(C) = \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|}, \ \mu(f) = -\overline{l}(C), \ \text{Annahme:} \ \forall e \in E(G_f) : l(e) \geqslant -\mu(f)$$

- Wir sorgen dafür, dass die Annahme erfüllt werden kann:
- Wir verändern / geeignet.
- Verhalten des Algorithmus sollte unverändert sein.

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Laufzeit (Erzwinge die Annahme)

$$\overline{l}(C) = \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|}, \ \mu(f) = -\overline{l}(C), \ \text{Annahme:} \ \forall e \in E(G_f) : l(e) \geqslant -\mu(f)$$

- Wir sorgen dafür, dass die Annahme erfüllt werden kann:
- Wir verändern / geeignet.
- Verhalten des Algorithmus sollte unverändert sein.
- Sei $p: V \mapsto \mathbb{Z}$ ein Potential für die Knoten.

Laufzeit (Erzwinge die Annahme)

Verbesserung der Laufzeit 5/9

$$\overline{l}(C) = \frac{\overline{l}(C)}{|C|} = \frac{\sum_{e \in C} \overline{l}(e)}{|C|}, \ \mu(f) = -\overline{l}(C), \ \text{Annahme:} \ \forall e \in E(G_f) : \overline{l}(e) \geqslant -\mu(f)$$

- Wir sorgen dafür, dass die Annahme erfüllt werden kann:
- Wir verändern / geeignet.
- Verhalten des Algorithmus sollte unverändert sein.
- Sei $p: V \mapsto \mathbb{Z}$ ein Potential für die Knoten.
- Setze für alle $e = (v, w) \in E(G_f)$ setze: l'(e) = l(e) + p(v) p(w).

Laufzeit (Erzwinge die Annahme)

Verbesserung der Laufzeit 6/9

$$\overline{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}, \ \mu(f) = -\overline{I}(C), \ \text{Annahme:} \ \forall e \in E(G_f) : I(e) \geqslant -\mu(f)$$

- Wir sorgen dafür, dass die Annahme erfüllt werden kann:
- Wir verändern / geeignet.
- Verhalten des Algorithmus sollte unverändert sein.
- Sei $p: V \mapsto \mathbb{Z}$ ein Potential für die Knoten.
- Setze für alle $e = (v, w) \in E(G_f)$ setze: I'(e) = I(e) + p(v) p(w).
- Für $\overline{e} = (w, v)$ gilt:

$$l'(\overline{e}) = l(\overline{e}) + p(w) - p(v)$$
$$-l(e) + p(w) - p(v) = -l'(e)$$

Laufzeit (Erzwinge die Annahme)

Verbesserung der Laufzeit 7/9

$$\overline{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}, \ \mu(f) = -\overline{I}(C), \ \text{Annahme:} \ \forall e \in E(G_f) : I(e) \geqslant -\mu(f)$$

- Wir sorgen dafür, dass die Annahme erfüllt werden kann:
- Wir verändern / geeignet.
- Verhalten des Algorithmus sollte unverändert sein.
- Sei $p: V \mapsto \mathbb{Z}$ ein Potential für die Knoten.
- Setze für alle $e = (v, w) \in E(G_f)$ setze: I'(e) = I(e) + p(v) p(w).
- Für $\overline{e} = (w, v)$ gilt:

$$l'(\overline{e}) = l(\overline{e}) + p(w) - p(v)$$
$$-l(e) + p(w) - p(v) = -l'(e)$$

• Potentiale ändern die Kostensumme auf Kreisen C nicht: I(C) = I'(C).

Laufzeit (Erzwinge die Annahme)

$$\overline{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}, \ \mu(f) = -\overline{I}(C), \ \text{Annahme:} \ \forall e \in E(G_f) : I(e) \geqslant -\mu(f)$$

- Wir sorgen dafür, dass die Annahme erfüllt werden kann:
- Wir verändern / geeignet.
- Verhalten des Algorithmus sollte unverändert sein.
- Sei $p: V \mapsto \mathbb{Z}$ ein Potential für die Knoten.
- Setze für alle $e = (v, w) \in E(G_f)$ setze: I'(e) = I(e) + p(v) p(w).
- Für $\overline{e} = (w, v)$ gilt:

$$l'(\overline{e}) = l(\overline{e}) + p(w) - p(v)$$

 $-l(e) + p(w) - p(v) = -l'(e)$

- Potentiale ändern die Kostensumme auf Kreisen C nicht: I(C) = I'(C).
- Potentiale ändern damit auch nicht den Ablauf des Verfahrens.

Laufzeit (Erzwinge die Annahme)

$$\overline{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}, \ \mu(f) = -\overline{I}(C), \ \text{Annahme:} \ \forall e \in E(G_f) : I(e) \geqslant -\mu(f)$$

- Wir sorgen dafür, dass die Annahme erfüllt werden kann:
- Wir verändern / geeignet.
- Verhalten des Algorithmus sollte unverändert sein.
- Sei $p: V \mapsto \mathbb{Z}$ ein Potential für die Knoten.
- Setze für alle $e = (v, w) \in E(G_f)$ setze: I'(e) = I(e) + p(v) p(w).
- Für $\overline{e} = (w, v)$ gilt:

$$l'(\overline{e}) = l(\overline{e}) + p(w) - p(v) -l(e) + p(w) - p(v) = -l'(e)$$

- Potentiale ändern die Kostensumme auf Kreisen C nicht: I(C) = I'(C).
- Potentiale ändern damit auch nicht den Ablauf des Verfahrens.
- Wir zeigen nun, dass es Potentiale gibt, die die Annahme erzwingen.

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion

2:40 Verbesserung der Laufzeit 1/6 Walter Unger 6.11.2018 7:05 SS2015 RWTH

Laufzeit (Erzwinge die Annahme durch Potential)

$$\bar{l}(C) = \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|}, \ \mu(f) = -\bar{l}(C), \ \text{Annahme:} \ \forall e \in E(G_f) : l(e) \geqslant -\mu(f)$$

Lemma (Existenz von p)

In $G_f = (V, E_f)$ gelte $\overline{I}(C) \geqslant -\mu$ für jeden Kreis C.

Dann gibt es $p: V \mapsto \mathbb{Z}$ mit:

$$I'(e) = I(e) + p(w) - p(v) \ge -\mu$$
 für jede Kante $e \in E_f$.

2:40 Verbesserung der Laufzeit 2/6 Walter Unger 6.11.2018 7:05 SS2015 **RWTH**<u>Laufzeit (Erzwinge die Annahme durch Potential)</u>

$\bar{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}, \ \mu(f) = -\bar{I}(C), \ \text{Annahme:} \ \forall e \in E(G_f) : I(e) \geqslant -\mu(f)$

Lemma (Existenz von p)

In $G_f = (V, E_f)$ gelte $\overline{I}(C) \geqslant -\mu$ für jeden Kreis C.

Dann gibt es $p: V \mapsto \mathbb{Z}$ mit:

$$I'(e) = I(e) + p(w) - p(v) \geqslant -\mu$$
 für jede Kante $e \in E_f$.

• Für
$$e \in E_f$$
 setze: $I_{\mu}(e) = I(e) + \mu$

Laufzeit (Erzwinge die Annahme durch Potential)

$$\overline{l}(C) = \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|}, \ \mu(f) = -\overline{l}(C), \ \text{Annahme:} \ \forall e \in E(G_f) : l(e) \geqslant -\mu(f)$$

Lemma (Existenz von p)

Verbesserung der Laufzeit 3/6

In $G_f = (V, E_f)$ gelte $\overline{I}(C) \geqslant -\mu$ für jeden Kreis C.

Dann gibt es $p: V \mapsto \mathbb{Z}$ mit:

$$I'(e) = I(e) + p(w) - p(v) \ge -\mu$$
 für jede Kante $e \in E_f$.

- Für $e \in E_f$ setze: $I_{\mu}(e) = I(e) + \mu$
- Für v ∈ V bestimme Kantenzug P in G_f von beliebigen Knoten w zu v mit minimalen Gewicht. Setze dann p(v) = I_μ(P).

Laufzeit (Erzwinge die Annahme durch Potential)

$$\overline{l}(C) = \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|}, \ \mu(f) = -\overline{l}(C), \ \text{Annahme:} \ \forall e \in E(G_f) : l(e) \geqslant -\mu(f)$$

Lemma (Existenz von p)

Verbesserung der Laufzeit 4/6

In $G_f = (V, E_f)$ gelte $\overline{I}(C) \geqslant -\mu$ für jeden Kreis C.

Dann gibt es $p: V \mapsto \mathbb{Z}$ mit:

$$I'(e) = I(e) + p(w) - p(v) \geqslant -\mu$$
 für jede Kante $e \in E_f$.

- Für $e \in E_f$ setze: $I_{\mu}(e) = I(e) + \mu$
- Für v ∈ V bestimme Kantenzug P in G_f von beliebigen Knoten w zu v mit minimalen Gewicht. Setze dann p(v) = I_μ(P).
- Beachte: Das ist wohldefinert, denn aus $\bar{l}(C) \geqslant -\mu$ folgt $l_{\mu}(C) \geqslant 0$ für beliebigen Kreis C.

Laufzeit (Erzwinge die Annahme durch Potential)

$$\overline{l}(C) = \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|}, \ \mu(f) = -\overline{l}(C), \ \text{Annahme:} \ \forall e \in E(G_f) : l(e) \geqslant -\mu(f)$$

Lemma (Existenz von p)

In
$$G_f = (V, E_f)$$
 gelte $\overline{I}(C) \geqslant -\mu$ für jeden Kreis C.

Dann gibt es $p: V \mapsto \mathbb{Z}$ mit:

$$I'(e) = I(e) + p(w) - p(v) \geqslant -\mu$$
 für jede Kante $e \in E_f$.

- Für $e \in E_f$ setze: $I_{\mu}(e) = I(e) + \mu$
- Für v ∈ V bestimme Kantenzug P in G_f von beliebigen Knoten w zu v mit minimalen Gewicht. Setze dann p(v) = I_μ(P).
- Beachte: Das ist wohldefinert, denn aus $\bar{l}(C) \geqslant -\mu$ folgt $l_{\mu}(C) \geqslant 0$ für beliebigen Kreis C.
- Damit gilt für jede Kante $e = (v, w) \in E_f : p(w) \leq p(v) + I_{\mu}(e)$.

Laufzeit (Erzwinge die Annahme durch Potential)

$$\overline{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}, \ \mu(f) = -\overline{I}(C), \ \text{Annahme:} \ \forall e \in E(G_f) : I(e) \geqslant -\mu(f)$$

Lemma (Existenz von p)

In
$$G_f = (V, E_f)$$
 gelte $\overline{I}(C) \geqslant -\mu$ für jeden Kreis C.

Dann gibt es $p: V \mapsto \mathbb{Z}$ mit:

$$I'(e) = I(e) + p(w) - p(v) \geqslant -\mu$$
 für jede Kante $e \in E_f$.

- Für $e \in E_f$ setze: $I_{\mu}(e) = I(e) + \mu$
- Für v ∈ V bestimme Kantenzug P in G_f von beliebigen Knoten w zu v mit minimalen Gewicht. Setze dann p(v) = I_u(P).
- Beachte: Das ist wohldefinert, denn aus $\bar{I}(C) \geqslant -\mu$ folgt $I_{\mu}(C) \geqslant 0$ für beliebigen Kreis C.
- Damit gilt für jede Kante $e = (v, w) \in E_f : p(w) \leq p(v) + l_{\mu}(e)$.
- Es folgt:

$$l'(e) = l(e) + p(v) - p(w)$$

= $l_{\mu}(e) + p(v) - p(w) - \mu \ge -\mu$

2:41 Verbesserung der Laufzeit 1/4 Laufzeit pro Iteration

$$\overline{l}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}, \ \mu(f) = -\overline{l}(C), \ \forall e \in E(G_f) : I(e) \geqslant -\mu(f)$$

Lemma (Karp, 1978)

Ein Min-Mean-Cycle kann in G_f in Zeit O(nm) gefunden werden.

2:41 Verbesserung der Laufzeit 2/4 Laufzeit pro Iteration

$$\overline{l}(C) = \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|}, \ \mu(f) = -\overline{l}(C), \ \forall e \in E(G_f) : l(e) \geqslant -\mu(f)$$

Lemma (Karp, 1978)

Ein Min-Mean-Cycle kann in G_f in Zeit O(nm) gefunden werden.

Beweis:

• Setze für $v \in V$ und $k \in \{0, ..., n\}$: $d_k(v)$ seien die Kosten des günstigsten Kantenzugs nach v mit genau kKanten.

2:41 Verbesserung der Laufzeit 3/4 Laufzeit pro Iteration

$$\overline{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}, \ \mu(f) = -\overline{I}(C), \ \forall e \in E(G_f) : I(e) \geqslant -\mu(f)$$

Lemma (Karp, 1978)

Ein Min-Mean-Cycle kann in G_f in Zeit O(nm) gefunden werden.

- Setze für $v \in V$ und $k \in \{0, ..., n\}$: $d_k(v)$ seien die Kosten des günstigsten Kantenzugs nach v mit genau kKanten.
- Es gilt $d_0(v) = 0$ und

$$d_{k+1}(v) = \min_{e=(w,v)\in F_c} (d_k(w) + I(e)).$$

Laufzeit pro Iteration

$$\overline{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}, \ \mu(f) = -\overline{I}(C), \ \forall e \in E(G_f) : I(e) \geqslant -\mu(f)$$

Lemma (Karp, 1978)

Ein Min-Mean-Cycle kann in G_f in Zeit O(nm) gefunden werden.

Beweis:

- Setze für $v \in V$ und $k \in \{0, ..., n\}$: $d_k(v)$ seien die Kosten des günstigsten Kantenzugs nach v mit genau kKanten.
- Es gilt $d_0(v) = 0$ und

$$d_{k+1}(v) = \min_{e=(w,v)\in E_f} (d_k(w) + I(e)).$$

• Alle $d_k(v)$ Werte können durch dynamische Programmierung in Zeit O(nm) bestimmt werden.

2:42 Verbesserung der Laufzeit 1/7

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Wert des Min-Mean-Kreises

$$\overline{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}, \ \mu(f) = -\overline{I}(C)$$

Lemma

Der Wert Ī(C) des Min-Mean-Kreises ist:

$$\alpha = \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(v) - d_j(v)}{n - j} \right)$$

2:42 Verbesserung der Laufzeit 2/7
Wert des Min-Mean-Kreises

$$\bar{l}(C) = \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|}, \ \mu(f) = -\bar{l}(C)$$

Lemma

Der Wert Ī(C) des Min-Mean-Kreises ist:

$$\alpha = \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(v) - d_j(v)}{n - j} \right)$$

2:42 Verbesserung der Laufzeit 3/7

Wert des Min-Mean-Kreises

$$\overline{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}, \ \mu(f) = -\overline{I}(C)$$

Lemma

Der Wert Ī(C) des Min-Mean-Kreises ist:

$$\alpha = \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(v) - d_j(v)}{n-j} \right)$$

Beweis:

• Sei C ein Min-Mean-Kreis.

2:42 Verbesserung der Laufzeit 4/7

$\bar{l}(C) = \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|}, \ \mu(f) = -\bar{l}(C)$

Lemma

Der Wert Ī(C) des Min-Mean-Kreises ist:

$$\alpha = \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(v) - d_j(v)}{n-j} \right)$$

- Sei C ein Min-Mean-Kreis.
- ullet Wenn alle Kantenkosten um δ erhöht werden, bleibt C Min-Mean-Kreis.

2:42 Verbesserung der Laufzeit 5/7

Wert des Min-Mean-Kreises

$$\overline{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}, \ \mu(f) = -\overline{I}(C)$$

Lemma

Der Wert Ī(C) des Min-Mean-Kreises ist:

$$\alpha = \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(v) - d_j(v)}{n - j} \right)$$

- Sei C ein Min-Mean-Kreis.
- ullet Wenn alle Kantenkosten um δ erhöht werden, bleibt C Min-Mean-Kreis.
- Der Wert von C erhöht sich auch um δ .

2:42 Verbesserung der Laufzeit 6/7
Wert des Min-Mean-Kreises

$$\overline{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}, \ \mu(f) = -\overline{I}(C)$$

Lemma

Der Wert Ī(C) des Min-Mean-Kreises ist:

$$\alpha = \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(v) - d_j(v)}{n - j} \right)$$

- Sei C ein Min-Mean-Kreis.
- ullet Wenn alle Kantenkosten um δ erhöht werden, bleibt C Min-Mean-Kreis.
- Der Wert von C erhöht sich auch um δ .
- Also können wir alle Kantenkosten um δ verschieben bis $\bar{l}(C) = 0$ gilt.

2:42 Verbesserung der Laufzeit 7/7

Wert des Min-Mean-Kreises

$$\bar{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}, \ \mu(f) = -\bar{I}(C)$$

Lemma

Der Wert Ī(C) des Min-Mean-Kreises ist:

$$\alpha = \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(v) - d_j(v)}{n-j} \right)$$

- Sei C ein Min-Mean-Kreis.
- Wenn alle Kantenkosten um δ erhöht werden, bleibt C Min-Mean-Kreis.
- Der Wert von C erh
 öht sich auch um δ.
- Also können wir alle Kantenkosten um δ verschieben bis $\bar{I}(C) = 0$ gilt.
- Zeige nun: $\alpha = 0$ gilt damit auch.

2:43 Verbesserung der Laufzeit 1/8 Wert des Min-Mean-Kreises (Zeige $\alpha \geqslant 0$)

 Sei v beliebig und P Kantenzug, der bei v endet und Kosten $d_n(v)$ hat.

$$\begin{split} \overline{l}(C) &= \frac{l(C)}{|C|} = \frac{\sum_{\mathbf{c} \in C} l(\mathbf{c})}{|C|} \\ &\mu(f) &= -\overline{l}(C) \\ \alpha &= \min_{V \in V} \max_{j=\mathbf{0}}^{n-\mathbf{1}} \left(\frac{d_n(v) - d_j(v)}{n-j} \right) \end{split}$$

2:43 Verbesserung der Laufzeit 2/8 Wert des Min-Mean-Kreises (Zeige $\alpha \geqslant 0$)

- Sei v beliebig und P Kantenzug, der bei v endet und Kosten $d_n(v)$ hat.
- P muss Kreis C beinhalten.

$$\begin{split} \bar{l}(C) &= \frac{l(C)}{|C|} = \frac{\sum_{\mathbf{c} \in C} l(\mathbf{c})}{|C|} \\ \mu(f) &= -\bar{l}(C) \\ \alpha &= \min_{\mathbf{v} \in V} \max_{j=\mathbf{0}}^{n-\mathbf{1}} \left(\frac{d_n(\mathbf{v}) - d_j(\mathbf{v})}{n-j} \right) \end{split}$$

2:43 Verbesserung der Laufzeit 3/8

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Wert des Min-Mean-Kreises (Zeige $\alpha \geq 0$)

- Sei v beliebig und P Kantenzug, der bei v endet und Kosten $d_n(v)$ hat.
- $\bar{l}(C) = \frac{l(C)}{|C|} = \frac{\sum_{\mathbf{c} \in C} l(\mathbf{c})}{|C|}$ $\mu(f) = -\bar{l}(C)$ $\alpha = \min_{\mathbf{v} \in V} \max_{j=\mathbf{0}}^{n-\mathbf{1}} \left(\frac{d_n(\mathbf{v}) d_j(\mathbf{v})}{n-j} \right)$
- P muss Kreis C beinhalten.
- Sei P' der verbleibende Kantenzug mit j Kanten ohne C.

2:43 Verbesserung der Laufzeit 4/8

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Wert des Min-Mean-Kreises (Zeige $\alpha \geq 0$)

- Sei v beliebig und P Kantenzug,
- der bei v endet und Kosten $d_n(v)$ hat.
- P muss Kreis C beinhalten. • Sei P' der verbleibende Kantenzug mit j Kanten ohne C.
- Dann hat C n j Kanten.

$$\bar{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

$$\mu(f) = -\bar{I}(C)$$

$$\alpha = \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(v) - d_j(v)}{n-j}\right)$$

Walter Unger 6.11.2018 7:05 SS2015 RWTH

2:43 Verbesserung der Laufzeit 5/8 Wert des Min-Mean-Kreises (Zeige $\alpha \geq 0$)

- Sei v beliebig und P Kantenzug,
- $\overline{I(C)} = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$ $\mu(f) = -\overline{I(C)}$ $\alpha = \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(v) d_j(v)}{n-j} \right)$ der bei v endet und Kosten $d_n(v)$ hat.
- P muss Kreis C beinhalten.
- Sei P' der verbleibende Kantenzug mit j Kanten ohne C.
- Dann hat C n i Kanten.
- Wegen $I(C) \ge 0$ gilt:

$$d_n(v) = I(P) = I(C) + I(P') \geqslant I(P') \geqslant d_j(v).$$

 $\overline{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$ $\mu(f) = -\overline{I}(C)$ $\alpha = \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(v) - d_j(v)}{n-j} \right)$

Wert des Min-Mean-Kreises (Zeige $\alpha \geqslant 0$)

- Sei v beliebig und P Kantenzug,
- der bei v endet und Kosten $d_n(v)$ hat.
- P muss Kreis C beinhalten. • Sei P' der verbleibende Kantenzug mit j Kanten ohne C.
- Dann hat C n i Kanten.
- Wegen $I(C) \ge 0$ gilt:

$$d_n(v) = I(P) = I(C) + I(P') \geqslant I(P') \geqslant d_j(v).$$

• Für jeden Knoten v gibt es $j \in \{1, \ldots, n-1\}$ mit $d_n(v) \ge d_i(v)$.

2:43 Verbesserung der Laufzeit 7/8 Wert des Min-Mean-Kreises (Zeige $lpha\geqslant 0$)

Sei v beliebig und P Kantenzug,

des will weath weises (Zeige a > 0)

 $\overline{I(C)} = \frac{\underline{I(C)}}{|C|} = \frac{\sum_{e \in C} \underline{I(e)}}{|C|}$ $\mu(f) = -\overline{I(C)}$ $\alpha = \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(v) - d_j(v)}{n-j} \right)$

- der bei v endet und Kosten $d_n(v)$ hat. • P muss Kreis C beinhalten.
- Sei P' der verbleibende Kantenzug mit j Kanten ohne C.
- Dann hat C n i Kanten.
- Wegen $I(C) \ge 0$ gilt:

$$d_n(v) = I(P) = I(C) + I(P') \geqslant I(P') \geqslant d_j(v).$$

- Für jeden Knoten v gibt es $j \in \{1, \ldots, n-1\}$ mit $d_n(v) \geqslant d_j(v)$.
- Damit $\alpha \geqslant 0$.

Wert des Min-Mean-Kreises (Zeige $\alpha \geq 0$)

- Sei v beliebig und P Kantenzug,
- $\overline{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$ $\mu(f) = -\overline{I}(C)$ $\alpha = \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(v) d_j(v)}{n-j} \right)$ der bei v endet und Kosten $d_n(v)$ hat.
- P muss Kreis C beinhalten.
- Sei P' der verbleibende Kantenzug mit j Kanten ohne C.
- Dann hat C n i Kanten.
- Wegen $I(C) \ge 0$ gilt:

$$d_n(v) = I(P) = I(C) + I(P') \geqslant I(P') \geqslant d_j(v).$$

- Für jeden Knoten v gibt es $j \in \{1, ..., n-1\}$ mit $d_n(v) \ge d_i(v)$.
- Damit $\alpha \geqslant 0$.
- Im Folgenden zeigen wir nun noch $\alpha \leq 0$.

2:44 Verbesserung der Laufzeit 1/9

Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Wert des Min-Mean-Kreises (Zeige
$$\alpha \leqslant 0$$
)

$$\alpha \leqslant 0$$

$$\bar{I}(C) =$$

$$\begin{split} \overline{l}(C) &= \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|} \\ \mu(f) &= -\overline{l}(C) \\ \alpha &= \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(v) - d_j(v)}{n-j} \right) \end{split}$$

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

2:44 Verbesserung der Laufzeit 2/9 Wert des Min-Mean-Kreises (Zeige $\alpha \leq 0$)

$$\begin{split} \overline{l}(C) &= \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|} \\ \mu(f) &= -\overline{l}(C) \\ \alpha &= \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(v) - d_j(v)}{n-j} \right) \end{split}$$

Verbesserung der Laufzeit 3/9

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Wert des Min-Mean-Kreises (Zeige $\alpha \leq 0$)

• Zeige: Es gibt Knoten
$$w$$
 mit: $d_n(w) \leqslant d_j(w)$ für alle $\alpha = \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(w)}{d_n(w)}\right)$

 $i \in \{0, \ldots, n-1\}$

$$I(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

$$\mu(f) = -I(C)$$

$$\alpha = \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(v) - d_j(v)}{n-j} \right)$$

Verbesserung der Laufzeit 4/9

Wert des Min-Mean-Kreises (Zeige $\alpha \leq 0$)

$$i(c) = \frac{I(c)}{|c|} = \frac{\sum_{e \in C} I(e)}{|c|}$$
• Zeige: Es gibt Knoten w mit: $d_n(w) \leqslant d_j(w)$ für alle
$$\alpha = \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(v) - d_j(v)}{n-j}\right)$$

$$j \in \{0, \dots, n-1\}$$

• Sei v Knoten des Min-Mean-Kreises C.

2:44 Verbesserung der Laufzeit 5/9

Wert des Min-Mean-Kreises (Zeige $\alpha \leq 0$)

$$\overline{l}(c) = \frac{|C|}{|C|} = \frac{\sum_{e \in C} \overline{l}(e)}{|C|}$$
• Zeige: Es gibt Knoten w mit: $d_n(w) \leqslant d_j(w)$ für alle
$$a = \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(v) - d_j(v)}{n-j}\right)$$

$$i \in \{0, \dots, n-1\}$$

- Sei v Knoten des Min-Mean-Kreises C.
- Sei P kürzester Kantenzug, der bei v endet. O.B.d.A. enthält P keinen Kreis. Sei p < n die Anzahl der Kanten in P

2:44 Verbesserung der Laufzeit 6/9 Wert des Min-Mean-Kreises (Zeige $\alpha \leq 0$)

ert des Min-Mean-Kreises (Zeige
$$lpha \leqslant 0$$
)

$$\overline{I(c)} = \frac{|C|}{|C|} = \frac{\sum_{e \in C} |C|}{|C|}$$
• Zeige: Es gibt Knoten w mit: $d_n(w) \leqslant d_j(w)$ für alle
$$a = \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(v) - d_j(v)}{n-j}\right)$$

$$j \in \{0, \dots, n-1\}$$

- Sei v Knoten des Min-Mean-Kreises C.
- Sei P kürzester Kantenzug, der bei v endet. O.B.d.A. enthält P keinen Kreis. Sei p < n die Anzahl der Kanten in
- Sei w der Knoten, der auf C von V aus nach n-p Kanten liegt. Diesen Kantenzug nennen wir Q.

Verbesserung der Laufzeit 7/9

Wert des Min-Mean-Kreises (Zeige $\alpha \leq 0$)

$$\pi(c) = \frac{I(c)}{|c|} = \frac{\sum_{e \in C} I(e)}{|c|}$$
• Zeige: Es gibt Knoten w mit: $d_n(w) \leqslant d_j(w)$ für alle
$$a = \min_{v \in V} \max_{j=0}^{n-1} \binom{d_n(v) - d_j(v)}{n-j}$$

$$j \in \{0, \dots, n-1\}$$

- Sei v Knoten des Min-Mean-Kreises C.
- Sei P kürzester Kantenzug, der bei v endet. O.B.d.A. enthält P keinen Kreis. Sei p < n die Anzahl der Kanten in P
- Sei w der Knoten, der auf C von v aus nach n − p Kanten liegt. Diesen Kantenzug nennen wir Q.
- Sei R der Weg von w zu v auf C mit r Kanten.

Verbesserung der Laufzeit 8/9 Wert des Min-Mean-Kreises (Zeige $\alpha \leq 0$)

$$\bar{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

• Zeige: Es gibt Knoten w mit: $d_n(w) \leq d_i(w)$ für alle $i \in \{0, \ldots, n-1\}$

$$\bar{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

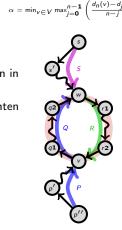
$$\mu(f) = -\bar{I}(C)$$

$$\alpha = \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(v) - d_j(v)}{n-j} \right)$$

- Sei v Knoten des Min-Mean-Kreises C.
- Sei P kürzester Kantenzug, der bei v endet. O.B.d.A. enthält P keinen Kreis. Sei p < n die Anzahl der Kanten in
- Sei w der Knoten, der auf C von V aus nach n-p Kanten liegt. Diesen Kantenzug nennen wir Q.
- Sei R der Weg von w zu v auf C mit r Kanten.
- Sei $j \in \{0, ..., n-1\}$ und S ein Kantenzug minimaler Länge aus *i* Kanten, der an *w* endet.

Wert des Min-Mean-Kreises (Zeige $\alpha \leq 0$)

- Zeige: Es gibt Knoten w mit: $d_n(w) \leq d_i(w)$ für alle $i \in \{0, \ldots, n-1\}$
- Sei v Knoten des Min-Mean-Kreises C.
- Sei P kürzester Kantenzug, der bei v endet. O.B.d.A. enthält P keinen Kreis. Sei p < n die Anzahl der Kanten in
- Sei w der Knoten, der auf C von V aus nach n-p Kanten liegt. Diesen Kantenzug nennen wir Q.
- Sei R der Weg von w zu v auf C mit r Kanten.
- Sei $j \in \{0, ..., n-1\}$ und S ein Kantenzug minimaler Länge aus j Kanten, der an w endet.



Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

2:45 Verbesserung der Laufzeit 1/8 Wert des Min-Mean-Kreises (Zeige
$$lpha\leqslant 0$$
)

$$\begin{split} \overline{l}(C) &= \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|} \\ \mu(f) &= -\overline{l}(C) \\ \alpha &= \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(v) - d_j(v)}{n-j} \right) \end{split}$$

Iternativen) Flüsse mit Kostenfunktion

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Wert des Min-Mean-Kreises (Zeige $\alpha \leqslant 0$)

2:45 Verbesserung der Laufzeit 2/8

$$\begin{split} \overline{l}(C) &= \frac{l(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|} \\ \mu(f) &= -\overline{l}(C) \\ \alpha &= \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(v) - d_j(v)}{n-j} \right) \end{split}$$

Walter Unger 6.11.2018 7:05 SS2015 RWTH

Verbesserung der Laufzeit 3/8 Wert des Min-Mean-Kreises (Zeige $\alpha \leq 0$)

Dann gilt:

$$d_p(w) \leqslant l(P) + l(Q) = d_p(v) + l(Q).$$

$$\overline{I}(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

$$\mu(f) = -\overline{I}(C)$$

$$\alpha = \min_{v \in V} \max_{j=0}^{n-1} \left(\frac{d_n(v) - d_j(v)}{n-j}\right)$$

Verbesserung der Laufzeit 4/8

Wert des Min-Mean-Kreises (Zeige $\alpha \leq 0$)

Dann gilt:

$$d_n(w) \leqslant I(P) + I(Q) = d_p(v) + I(Q).$$

und

$$d_p(v) \leqslant d_{j+r}(v) \leqslant d_j(w) + l(R).$$

$$I(C) = \frac{I(C)}{|C|} = \frac{\sum_{e \in C} I(e)}{|C|}$$

$$\mu(f) = -I(C)$$

$$\alpha = \min_{v \in V} \max_{j=0}^{n-1} \binom{d_n(v) - d_j(v)}{n-j}$$

$$\begin{split} \widetilde{l}(C) &= \frac{l(C)}{|C|} = \frac{\sum_{\boldsymbol{e} \in C} l(\boldsymbol{e})}{|C|} \\ \mu(f) &= -\widetilde{l}(C) \\ \alpha &= \min_{\boldsymbol{v} \in V} \max_{j=\boldsymbol{0}}^{n-\boldsymbol{1}} \left(\frac{d_n(\boldsymbol{v}) - d_j(\boldsymbol{v})}{n-j} \right) \end{split}$$

Verbesserung der Laufzeit 5/8 Wert des Min-Mean-Kreises (Zeige $\alpha \leq 0$)

Dann gilt:

$$d_n(w) \leqslant I(P) + I(Q) = d_p(v) + I(Q).$$

und

$$d_p(v) \leqslant d_{i+r}(v) \leqslant d_i(w) + I(R).$$

es folgt:

$$d_n(w) \leqslant d_j(w) + I(R) + I(Q).$$

$$\begin{split} \widetilde{l}(C) &= \frac{l(C)}{|C|} = \frac{\sum_{\boldsymbol{e} \in C} l(\boldsymbol{e})}{|C|} \\ \mu(f) &= -\widetilde{l}(C) \\ \alpha &= \min_{\boldsymbol{v} \in V} \max_{j=\boldsymbol{0}}^{n-\boldsymbol{1}} \left(\frac{d_n(\boldsymbol{v}) - d_j(\boldsymbol{v})}{n-j} \right) \end{split}$$

Verbesserung der Laufzeit 6/8 Wert des Min-Mean-Kreises (Zeige $\alpha \leq 0$)

Dann gilt:

$$d_n(w) \leqslant I(P) + I(Q) = d_n(v) + I(Q).$$

und

$$d_p(v) \leqslant d_{i+r}(v) \leqslant d_i(w) + I(R).$$

es folgt:

$$d_n(w) \leqslant d_j(w) + l(R) + l(Q).$$

• Der Kantenzug $Q \odot R$ hat Kosten 0 (wegen I(C) = 0).

$$\begin{split} \widetilde{l}(C) &= \frac{\widetilde{l}(C)}{|C|} = \frac{\sum_{e \in C} l(e)}{|C|} \\ \mu(f) &= -\widetilde{l}(C) \\ \alpha &= \min_{v \in V} \max_{j=\mathbf{0}}^{n-\mathbf{1}} \left(\frac{d_n(v) - d_j(v)}{n-j} \right) \end{split}$$

Verbesserung der Laufzeit 7/8 Wert des Min-Mean-Kreises (Zeige $\alpha \leq 0$)

Dann gilt:

$$d_n(w) \leqslant I(P) + I(Q) = d_p(v) + I(Q).$$

und

$$d_p(v) \leqslant d_{i+r}(v) \leqslant d_i(w) + I(R).$$

es folgt:

$$d_n(w) \leqslant d_i(w) + l(R) + l(Q).$$

- Der Kantenzug $Q \odot R$ hat Kosten 0 (wegen I(C) = 0).
- Damit gilt: $d_n(w) \leq d_i(w)$.

Wert des Min-Mean-Kreises (Zeige $\alpha \leq 0$)

Dann gilt:

$$d_n(w) \leqslant l(P) + l(Q) = d_p(v) + l(Q).$$

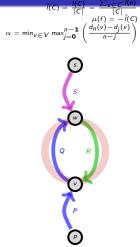
und

$$d_n(v) \leq d_{i+r}(v) \leq d_i(w) + I(R).$$

• es folgt:

$$d_n(w) \leqslant d_j(w) + I(R) + I(Q).$$

- Der Kantenzug $Q \odot R$ hat Kosten 0 (wegen I(C) = 0).
- Damit gilt: $d_n(w) \leqslant d_j(w)$.



Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Gesamtverfahren

2:46 Verbesserung der Laufzeit 1/9

9 Berechne die $d_k(v)$ Werte und bestimme α .

2:46 Verbesserung der Laufzeit 2/9

Spezielle Flüsse (Alternativen)

- **9** Berechne die $d_k(v)$ Werte und bestimme α .
- 2 Addiere zu allen Kantenkosten um α .

2:46 Verbesserung der Laufzeit 3/9

Spezielle Flüsse (Alternativen)

- Berechne die $d_k(v)$ Werte und bestimme α .
- ② Addiere zu allen Kantenkosten um α .
 - Damit ist der Wert des Min-Mean-Kreises 0.

- **9** Berechne die $d_k(v)$ Werte und bestimme α .
- \bigcirc Addiere zu allen Kantenkosten um α .
 - Damit ist der Wert des Min-Mean-Kreises 0.
- \odot Transformiere die Kantenkosten um den Wert α wie in Lemma "Existenz von p" beschrieben.

- **9** Berechne die $d_k(v)$ Werte und bestimme α .
- ② Addiere zu allen Kantenkosten um α .
 - Damit ist der Wert des Min-Mean-Kreises 0.
- **9** Transformiere die Kantenkosten um den Wert α wie in Lemma "Existenz von p" beschrieben.
 - Die Potentiale ergeben sich aus den Werten $d_k(v)$.

- Berechne die $d_k(v)$ Werte und bestimme α .
- 2 Addiere zu allen Kantenkosten um α .
 - Damit ist der Wert des Min-Mean-Kreises 0.
- Transformiere die Kantenkosten um den Wert α wie in Lemma "Existenz von p" beschrieben.
 - Die Potentiale ergeben sich aus den Werten $d_k(v)$.
 - Alle Kantenkosten sind nun nicht negativ.

- **9** Berechne die $d_k(v)$ Werte und bestimme α .
- \bigcirc Addiere zu allen Kantenkosten um α .
 - Damit ist der Wert des Min-Mean-Kreises 0.
- ullet Transformiere die Kantenkosten um den Wert α wie in Lemma "Existenz von p" beschrieben.
 - Die Potentiale ergeben sich aus den Werten $d_k(v)$.
 - Alle Kantenkosten sind nun nicht negativ.
 - Die Kanten des Min-Mean-Cycle haben Wert 0.

- **9** Berechne die $d_k(v)$ Werte und bestimme α .
- ② Addiere zu allen Kantenkosten um α .
 - Damit ist der Wert des Min-Mean-Kreises 0.
- **9** Transformiere die Kantenkosten um den Wert α wie in Lemma "Existenz von p" beschrieben.
 - Die Potentiale ergeben sich aus den Werten $d_k(v)$.
 - Alle Kantenkosten sind nun nicht negativ.
 - Die Kanten des Min-Mean-Cycle haben Wert 0.
- Lösche alle Kanten mit Wert größer 0.

- **9** Berechne die $d_k(v)$ Werte und bestimme α .
- 2 Addiere zu allen Kantenkosten um α .
 - Damit ist der Wert des Min-Mean-Kreises 0.
- **②** Transformiere die Kantenkosten um den Wert α wie in Lemma "Existenz von p" beschrieben.
 - Die Potentiale ergeben sich aus den Werten $d_k(v)$.
 - Alle Kantenkosten sind nun nicht negativ.
 - Die Kanten des Min-Mean-Cycle haben Wert 0.
- 4 Lösche alle Kanten mit Wert größer 0.
- Suche mit Tiefensuche Kreis in den verbleibenden Kanten.

Spezielle Flüsse (Mindestfluss)
00000000
2:47 Verbesserung der Laufzeit 1/6

Spezielle Flüsse (Alternativen)

Gesamtlaufzeit

Theorem

Der Mean-Cycle-Algorithmus hat eine Laufzeit von $O(m^2 \cdot n^2 \cdot \log(nL))$.

Gesamtlaufzeit

Theorem

Der Mean-Cycle-Algorithmus hat eine Laufzeit von $O(m^2 \cdot n^2 \cdot \log(nL))$.

Beweis, siehe obige Überlegungen:

• $n \log(nL) + 1$ Phasen (L Maximum der absoluten Kantenkosten).

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Gesamtlaufzeit

Theorem

Der Mean-Cycle-Algorithmus hat eine Laufzeit von $O(m^2 \cdot n^2 \cdot \log(nL))$.

- $n \log(nL) + 1$ Phasen (L Maximum der absoluten Kantenkosten).
- m Iterationen pro Phase.

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Gesamtlaufzeit

Theorem

Der Mean-Cycle-Algorithmus hat eine Laufzeit von $O(m^2 \cdot n^2 \cdot \log(nL))$.

- $n \log(nL) + 1$ Phasen (L Maximum der absoluten Kantenkosten).
- m Iterationen pro Phase.
- O(nm) Laufzeit pro Iteration.

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Gesamtlaufzeit

Theorem

Der Mean-Cycle-Algorithmus hat eine Laufzeit von $O(m^2 \cdot n^2 \cdot \log(nL))$.

- $n \log(nL) + 1$ Phasen (L Maximum der absoluten Kantenkosten).
- m Iterationen pro Phase.
- O(nm) Laufzeit pro Iteration.

Gesamtlaufzeit

Theorem

Der Mean-Cycle-Algorithmus hat eine Laufzeit von $O(m^2 \cdot n^2 \cdot \log(nL))$.

Beweis, siehe obige Überlegungen:

- $n \log(nL) + 1$ Phasen (L Maximum der absoluten Kantenkosten).
- m Iterationen pro Phase.
- O(nm) Laufzeit pro Iteration.

Theorem

Der Mean-Cycle-Algorithmus hat eine Laufzeit von $O(m^3 \cdot n^2 \cdot \log n)$.

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Literatur

 Ahuja, Magnanti, Orlin: Network Flows: Theory, Algorihms, and Applications, Prentice Hall, 1993.

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Literatur

- Ahuja, Magnanti, Orlin: Network Flows: Theory, Algorihms, and Applications, Prentice Hall, 1993.
- Cormen, Leiserson, Rives: Introduction to Algorithms, First Edition, MIT Press, 1990.

Literatur

- Ahuja, Magnanti, Orlin: Network Flows: Theory, Algorihms, and Applications, Prentice Hall, 1993.
- Cormen, Leiserson, Rives: Introduction to Algorithms, First Edition, MIT Press, 1990.
- Cormen, Leiserson, Rives: Introduction to Algorithms, Second Edition, MIT Press, 2001.

Spezielle Flüsse (Alternativen)

Flüsse mit Kostenfunktion Walter Unger 6.11.2018 7:05 SS2015 RWTH

Literatur

- Ahuja, Magnanti, Orlin: Network Flows: Theory, Algorihms, and Applications, Prentice Hall, 1993.
- Cormen, Leiserson, Rives: Introduction to Algorithms, First Edition, MIT Press. 1990.
- Cormen, Leiserson, Rives: Introduction to Algorithms, Second Edition, MIT Press. 2001.
- Ottmann, Widmayer: Algorithmen und Datenstrukturen. BI-Wiss.-Verl. 1990.

• Was ist die Laufzeit der Ford-Fulkerson Methode?

- Was ist die Laufzeit der Ford-Fulkerson Methode?
- Wann hat die Ford-Fulkerson Methode die maximale Laufzeit?

- Was ist die Laufzeit der Ford-Fulkerson Methode?
- Wann hat die Ford-Fulkerson Methode die maximale Laufzeit?
- Warum liefert die Ford-Fulkerson Methode den maximalen Fluss?

- Was ist die Laufzeit der Ford-Fulkerson Methode?
- Wann hat die Ford-Fulkerson Methode die maximale Laufzeit?
- Warum liefert die Ford-Fulkerson Methode den maximalen Fluss?
- Was ist die Laufzeit der Ford-Fulkerson Methode mit Breitensuche?

- Was ist die Laufzeit der Ford-Fulkerson Methode?
- Wann hat die Ford-Fulkerson Methode die maximale Laufzeit?
- Warum liefert die Ford-Fulkerson Methode den maximalen Fluss?
- Was ist die Laufzeit der Ford-Fulkerson Methode mit Breitensuche?
- Wie ist die Idee des Min-Cut-Max-Flow Theorems?

- Was ist die Laufzeit der Ford-Fulkerson Methode?
- Wann hat die Ford-Fulkerson Methode die maximale Laufzeit?
- Warum liefert die Ford-Fulkerson Methode den maximalen Fluss?
- Was ist die Laufzeit der Ford-Fulkerson Methode mit Breitensuche?
- Wie ist die Idee des Min-Cut-Max-Flow Theorems?
- Wie wird der Schnitt zu dem maximalen Fluss gefunden?

- Was ist die Laufzeit der Ford-Fulkerson Methode?
- Wann hat die Ford-Fulkerson Methode die maximale Laufzeit?
- Warum liefert die Ford-Fulkerson Methode den maximalen Fluss?
- Was ist die Laufzeit der Ford-Fulkerson Methode mit Breitensuche?
- Wie ist die Idee des Min-Cut-Max-Flow Theorems?
- Wie wird der Schnitt zu dem maximalen Fluss gefunden?
- Was ist die Idee des Algorithmus von Dinitz?

- Was ist die Laufzeit der Ford-Fulkerson Methode?
- Wann hat die Ford-Fulkerson Methode die maximale Laufzeit?
- Warum liefert die Ford-Fulkerson Methode den maximalen Fluss?
- Was ist die Laufzeit der Ford-Fulkerson Methode mit Breitensuche?
- Wie ist die Idee des Min-Cut-Max-Flow Theorems?
- Wie wird der Schnitt zu dem maximalen Fluss gefunden?
- Was ist die Idee des Algorithmus von Dinitz?
- Wie funktioniert die Forward-Propagation?

- Was ist die Laufzeit der Ford-Fulkerson Methode?
- Wann hat die Ford-Fulkerson Methode die maximale Laufzeit?
- Warum liefert die Ford-Fulkerson Methode den maximalen Fluss?
- Was ist die Laufzeit der Ford-Fulkerson Methode mit Breitensuche?
- Wie ist die Idee des Min-Cut-Max-Flow Theorems?
- Wie wird der Schnitt zu dem maximalen Fluss gefunden?
- Was ist die Idee des Algorithmus von Dinitz?
- Wie funktioniert die Forward-Propagation?
- Wie ist die Laufzeit vom Algorithmus von Dinitz?

- Was ist die Laufzeit der Ford-Fulkerson Methode?
- Wann hat die Ford-Fulkerson Methode die maximale Laufzeit?
- Warum liefert die Ford-Fulkerson Methode den maximalen Fluss?
- Was ist die Laufzeit der Ford-Fulkerson Methode mit Breitensuche?
- Wie ist die Idee des Min-Cut-Max-Flow Theorems?
- Wie wird der Schnitt zu dem maximalen Fluss gefunden?
- Was ist die Idee des Algorithmus von Dinitz?
- Wie funktioniert die Forward-Propagation?
- Wie ist die Laufzeit vom Algorithmus von Dinitz?
- Wie ist die Begründung zur Laufzeit vom Algorithmus von Dinitz?

• Wie bestimmt man Flüsse mit einem Mindestfluss auf den Kanten?

- Wie bestimmt man Flüsse mit einem Mindestfluss auf den Kanten?
- Wie bestimmt man Flüsse mit einem Mindestfluss auf den Kanten, wo aber auch ein leerer Fluss erlaubt ist?

- Wie bestimmt man Flüsse mit einem Mindestfluss auf den Kanten?
- Wie bestimmt man Flüsse mit einem Mindestfluss auf den Kanten, wo aber auch ein leerer Fluss erlaubt ist?
- Wie ist die Idee der Algorithmen zu kostenminimalen Flüssen?

- Wie bestimmt man Flüsse mit einem Mindestfluss auf den Kanten?
- Wie bestimmt man Flüsse mit einem Mindestfluss auf den Kanten, wo aber auch ein leerer Fluss erlaubt ist?
- Wie ist die Idee der Algorithmen zu kostenminimalen Flüssen?
- Was ist die Laufzeit der Algorithmen zu kostenminimalen Flüssen?

- Wie bestimmt man Flüsse mit einem Mindestfluss auf den Kanten?
- Wie bestimmt man Flüsse mit einem Mindestfluss auf den Kanten, wo aber auch ein leerer Fluss erlaubt ist?
- Wie ist die Idee der Algorithmen zu kostenminimalen Flüssen?
- Was ist die Laufzeit der Algorithmen zu kostenminimalen Flüssen?
- Gebe die Idee zum Beweis der Laufzeit der Algorithmen zu kostenminimalen Flüssen?