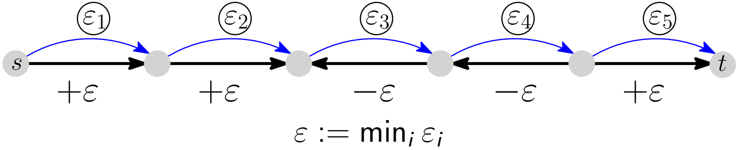
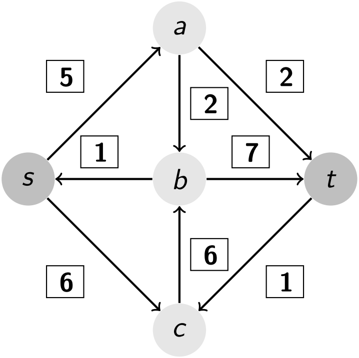
Angelika Steger & Emo Welzl

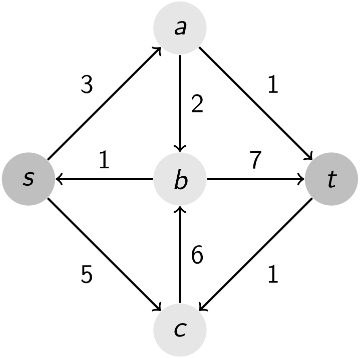
Flüsse in Netzwerken: Algorithmen



Netzwerke und Flüsse

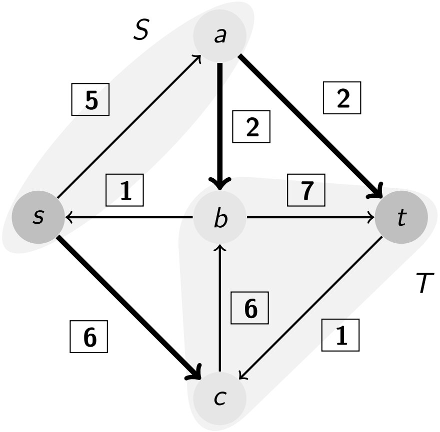


Netzwerk $N = (V, A, c, s, t)$.



Fluss f mit Wert $3 - 1 + 5 = 7$.

Schnitt



s-t-Schnitt (S, T) mit Kapazität $6 + 2 + 2 = 10$.

Schnitt vs. Fluss

Lemma

Ist f ein Fluss und (S, T) ein s-t-Schnitt in einem Netzwerk, so gilt

$$\mathrm{val}(f) \leq \mathrm{cap}(S, T) \ .$$

(bewiesen)

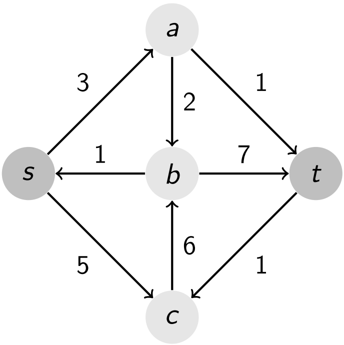
Schnitt vs. Fluss

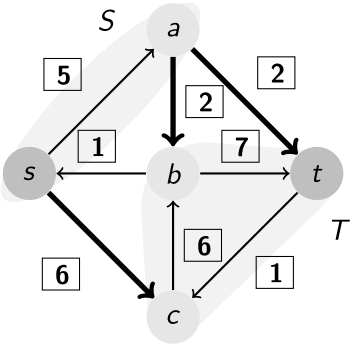
Lemma

Ist f ein Fluss und (S, T) ein s-t-Schnitt in einem Netzwerk, so gilt

$$\mathrm{val}(f) \leq \mathrm{cap}(S, T)\ .$$

(bewiesen)





Fluss mit Wert 7 $\boxed{7 \leq 10}$ Schnitt mit Kapaziät 10

Schnitt vs. Fluss

Lemma

Ist f ein Fluss und (S,T) ein s-t-Schnitt in einem Netzwerk, so gilt

$$\mathrm{val}(f) \leq \mathrm{cap}(S,T)\ .$$

(bewiesen)

Satz („**Maxflow-Mincut Theorem**“)

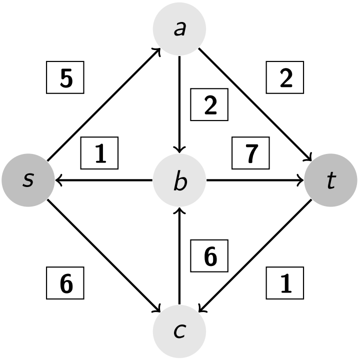
Jedes Netzwerk erfüllt

$$\max_{f\ \mathit{Fluss}} \mathrm{val}(f) = \min_{(S,T)\ \mathit{s\text{-}t\text{-}Schnitt}} \mathrm{cap}(S,T)\ .$$

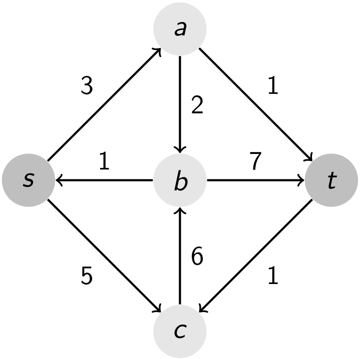
(noch nicht bewiesen)

Ziel: Algorithmus und Beweis des Maxflow-Mincut Theorem.

Verbessern eines gegebenen Flusses (1)

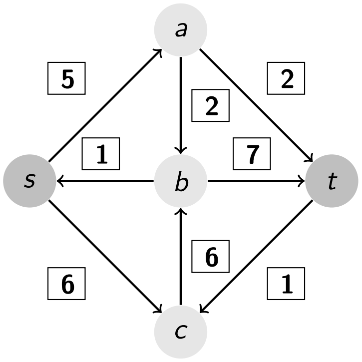


Netzwerk $N = (V, A, c, s, t)$

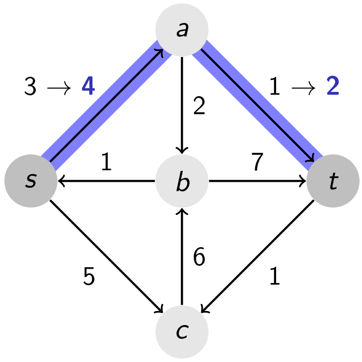


Fluss mit Wert $3 - 1 + 5 = 7$

Verbessern eines gegebenen Flusses (1)

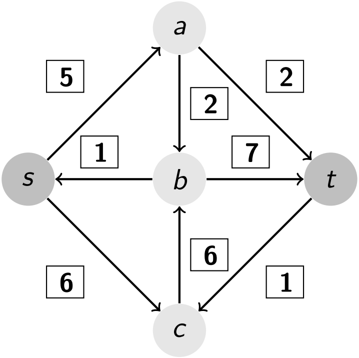


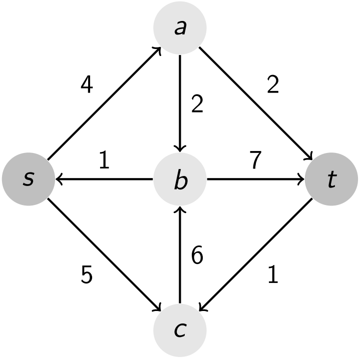
Netzwerk $N = (V, A, c, s, t)$



Fluss mit Wert $\mathbf{4} - 1 + 5 = \mathbf{8}$

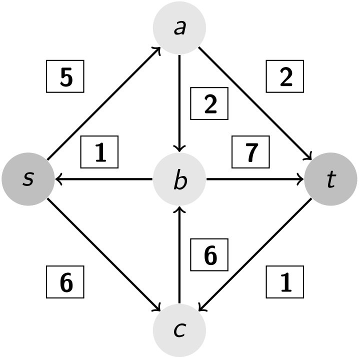
Verbessern eines gegebenen Flusses (2)

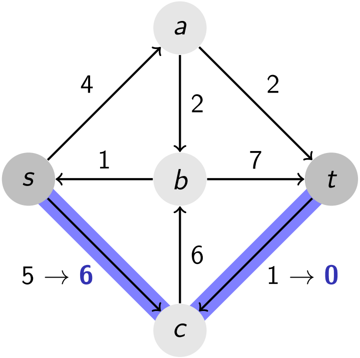




Fluss mit Wert $4 - 1 + 5 = 8$

Verbessern eines gegebenen Flusses (2)

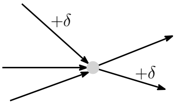




Fluss mit Wert $4 - 1 + \mathbf{6} = \mathbf{9}$

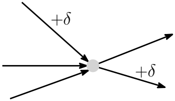
Flusserhaltungserhaltung

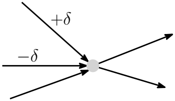
Lokale Veränderungen des Flusses, die die Flusserhaltung erhalten:



Flusserhaltungserhaltung

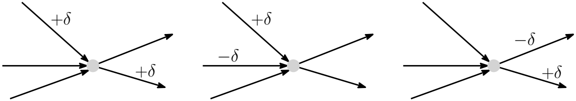
Lokale Veränderungen des Flusses, die die Flusserhaltung erhalten:





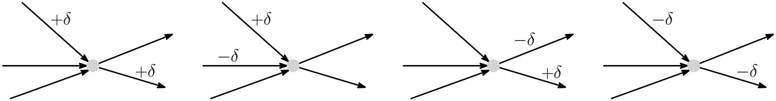
Flusserhaltungserhaltung

Lokale Veränderungen des Flusses, die die Flusserhaltung erhalten:



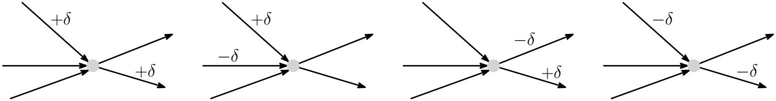
Flusserhaltungserhaltung

Lokale Veränderungen des Flusses, die die Flusserhaltung erhalten:



Flusserhaltungserhaltung

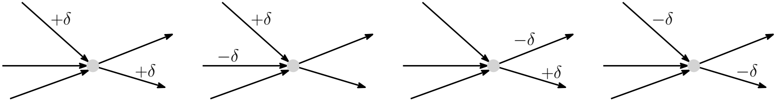
Lokale Veränderungen des Flusses, die die Flusserhaltung erhalten:



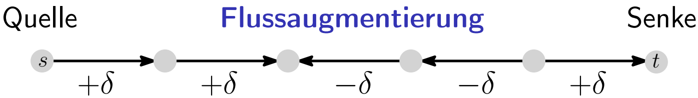
$$\text{Unter Beachtung} \begin{cases} \text{der Kapazität} & \text{bei „}+\delta\text{“, und} \\ \text{des aktuellen Flusses} & \text{bei „}-\delta\text{“.} \end{cases}$$

Flusserhaltungserhaltung

Lokale Veränderungen des Flusses, die die Flusserhaltung erhalten:

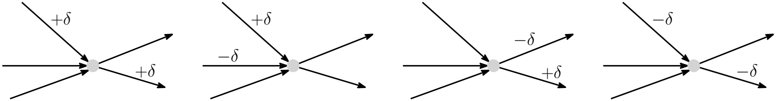


$$\text{Unter Beachtung} \begin{cases} \text{der Kapazität} & \text{bei „}+\delta\text{“, und} \\ \text{des aktuellen Flusses} & \text{bei „}-\delta\text{“.} \end{cases}$$



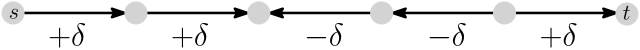
Flusserhaltungserhaltung

Lokale Veränderungen des Flusses, die die Flusserhaltung erhalten:



$$\text{Unter Beachtung} \begin{cases} \text{der Kapazität} & \text{bei „}+\delta\text{“, und} \\ \text{des aktuellen Flusses} & \text{bei „}-\delta\text{“.} \end{cases}$$

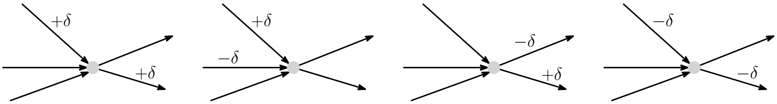
Quelle **Flussaugmentierung** Senke



augmentierender Pfad (ungerichteter Pfad!)

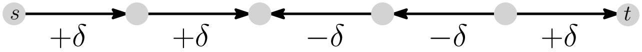
Flusserhaltungserhaltung

Lokale Veränderungen des Flusses, die die Flusserhaltung erhalten:



Unter Beachtung $\begin{cases} \text{der Kapazität} & \text{bei „}+\delta\text{“, und} \\ \text{des aktuellen Flusses} & \text{bei „}-\delta\text{“.} \end{cases}$

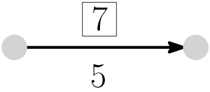
Quelle **Flussaugmentierung** Senke



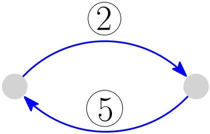
augmentierender Pfad (ungerichteter Pfad!)

Wie finden wir augmentierende Pfade?

Verwaltung des potentiellen Extraflusses/Spielraum

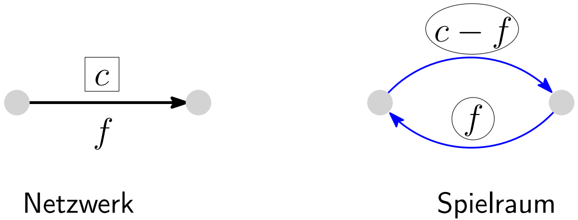


Netzwerk



Spielraum

Verwaltung des potentiellen Extraflusses/Spielraum



Restnetzwerk

Für $e = (u, v)$, sei $e^{\text{opp}} := (v, u)$ (entgegen gerichtete Kante).

Sei $N = (V, A, c, s, t)$ ein Netzwerk ohne entgegen gerichtete Kanten[1] und sei f ein Fluss in N. Das **Restnetzwerk** $N_f := (V, A_f, r_f, s, t)$ ist wie folgt definiert:

1. Ist $e \in A$ mit $f(e) < c(e)$, dann ist e eine Kante in A_f, mit
$$r_f(e) := c(e) - f(e).$$
2. Ist $e \in A$ mit $f(e) > 0$, dann ist e^{opp} in A_f, mit
$$r_f(e^{\text{opp}}) = f(e).$$
3. A_f enthält nur Kanten wie in (1) und (2).

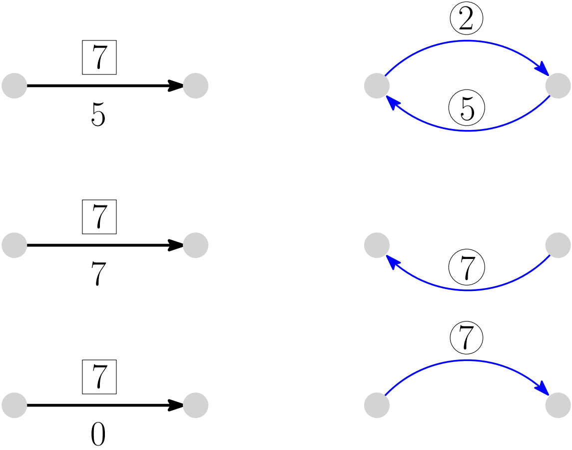
$r_f(e)$, $e \in A_f$, nennen wir die **Restkapazität** der Kante e.

Restkapazität = „Spielraum“

[1] Vereinfachende Annahme, ist aber nicht essentiell.

Restnetzwerk

Netzwerk	Restnetzwerk Restkapazität



Charakterisierung maximaler Fluss

Satz

Sei N ein Netzwerk (ohne entgegegen gerichtete Kanten).

Ein Fluss f ist maximaler Fluss

$$\Leftrightarrow$$

es im Restnetzwerk N_f keinen gerichteten s-t-Pfad gibt.

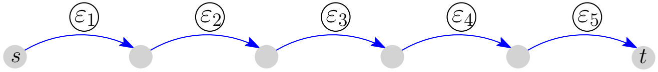
Für jeden maximalen Fluss f

gibt es einen s-t-Schnitt (S, T) mit $\mathrm{val}(f) = \mathrm{cap}(S, T)$.

Beweis

Es gibt im Restnetzwerk N_f einen gerichteten s-t-Pfad
$\Rightarrow$ f kann augmentiert werden ($\Rightarrow f$ ist nicht maximal)

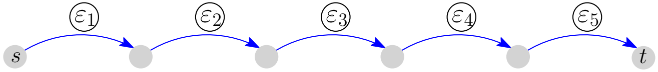
Wir betrachten einen gerichteten s-t-Pfad in N_f:



Beweis

Es gibt im Restnetzwerk N_f einen gerichteten s-t-Pfad
$\Rightarrow$ f kann augmentiert werden ($\Rightarrow$ f ist nicht maximal)

Wir betrachten einen gerichteten s-t-Pfad in N_f:

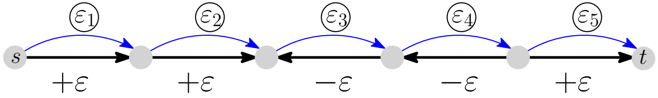


Bestimme die kleinste Restkapazität $\varepsilon := \min_i \varepsilon_i$

Beweis

Es gibt im Restnetzwerk N_f einen gerichteten s-t-Pfad
$\Rightarrow$ f kann augmentiert werden ($\Rightarrow f$ ist nicht maximal)

Wir betrachten einen gerichteten s-t-Pfad in N_f:



Bestimme die kleinste Restkapazität $\varepsilon := \min_i \varepsilon_i$

Augmentiere f entlang des Pfades um ε.

Beweis

Es gibt im Restnetzwerk N_f keinen gerichteten s-t-Pfad
$\Rightarrow\ \exists$ s-t-Schnitt (S, T) mit $\mathrm{cap}(S, T) = \mathrm{val}(f)$ ($\Rightarrow f$ ist maximal)

Beweis

Es gibt im Restnetzwerk N_f **k**einen gerichteten s-t-Pfad
$\Rightarrow\ \exists$ s-t-Schnitt (S, T) mit $\mathrm{cap}(S, T) = \mathrm{val}(f)$ ($\Rightarrow f$ ist maximal)

$S :=$ in N_f von s aus erreichbare Knoten; $T := V \setminus S$.

Beweis

Es gibt im Restnetzwerk N_f keinen gerichteten s-t-Pfad
$\Rightarrow\ \exists$ s-t-Schnitt (S, T) mit $\text{cap}(S, T) = \text{val}(f)$ ($\Rightarrow f$ ist maximal)

$S :=$ in N_f von s aus erreichbare Knoten; $T := V \setminus S$.

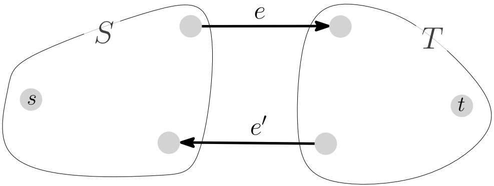
$$\left.\begin{array}{r} s \text{ von } s \text{ aus in } N_f \text{ erreichbar} \Rightarrow s \in S \\ t \text{ von } s \text{ aus nicht erreichbar} \Rightarrow t \notin S \end{array}\right\} \Rightarrow (S, T) \text{ ist } s\text{-}t\text{-Schnitt.}$$

Beweis

Es gibt im Restnetzwerk N_f keinen gerichteten s-t-Pfad
$\Rightarrow\ \exists$ s-t-Schnitt (S, T) mit $\operatorname{cap}(S, T) = \operatorname{val}(f)$ ($\Rightarrow f$ ist maximal)

$S :=$ in N_f von s aus erreichbare Knoten; $T := V \setminus S$.

$$\left.\begin{array}{r} s \text{ von } s \text{ aus in } N_f \text{ erreichbar} \Rightarrow s \in S \\ t \text{ von } s \text{ aus nicht erreichbar} \Rightarrow t \notin S \end{array}\right\} \Rightarrow (S, T) \text{ ist } s\text{-}t\text{-Schnitt.}$$

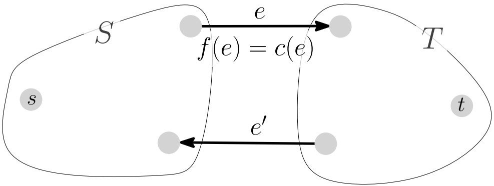


Beweis

Es gibt im Restnetzwerk N_f <u>**k**einen</u> gerichteten s-t-Pfad
$\Rightarrow\ \exists$ s-t-Schnitt (S, T) mit $\mathrm{cap}(S, T) = \mathrm{val}(f)$ ($\Rightarrow f$ ist <u>maximal</u>)

$S :=$ in N_f von s aus erreichbare Knoten; $T := V \setminus S$.

$$\left.\begin{array}{r} s \text{ von } s \text{ aus in } N_f \text{ erreichbar} \Rightarrow s \in S \\ t \text{ von } s \text{ aus nicht erreichbar} \Rightarrow t \notin S \end{array}\right\} \Rightarrow (S, T) \text{ ist } s\text{-}t\text{-Schnitt.}$$

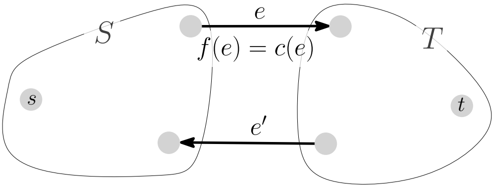


Beweis

Es gibt im Restnetzwerk N_f keinen gerichteten s-t-Pfad
$\Rightarrow\ \exists$ s-t-Schnitt (S, T) mit $\mathrm{cap}(S, T) = \mathrm{val}(f)$ ($\Rightarrow f$ ist maximal)

$S :=$ in N_f von s aus erreichbare Knoten; $T := V \setminus S$.

$$\left.\begin{array}{r} s \text{ von } s \text{ aus in } N_f \text{ erreichbar} \Rightarrow s \in S \\ t \text{ von } s \text{ aus nicht erreichbar} \Rightarrow t \notin S \end{array}\right\} \Rightarrow (S, T) \text{ ist } s\text{-}t\text{-Schnitt.}$$



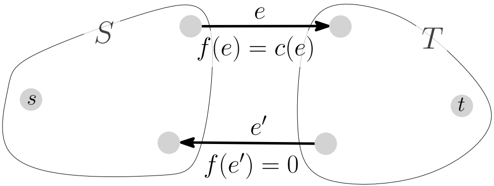
$f(S, T) = \mathrm{cap}(S, T)$

Beweis

Es gibt im Restnetzwerk N_f keinen gerichteten s-t-Pfad
$\Rightarrow\ \exists$ s-t-Schnitt (S, T) mit $\text{cap}(S, T) = \text{val}(f)$ ($\Rightarrow f$ ist maximal)

$S :=$ in N_f von s aus erreichbare Knoten; $T := V \setminus S$.

$$\left.\begin{array}{r} s \text{ von } s \text{ aus in } N_f \text{ erreichbar} \Rightarrow s \in S \\ t \text{ von } s \text{ aus nicht erreichbar} \Rightarrow t \notin S \end{array}\right\} \Rightarrow (S, T) \text{ ist } s\text{-}t\text{-Schnitt.}$$



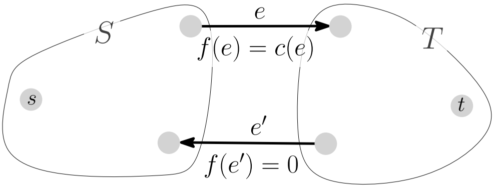
$f(S, T) = \text{cap}(S, T)$

Beweis

Es gibt im Restnetzwerk N_f keinen gerichteten s-t-Pfad
$\Rightarrow\ \exists\ s$-t-Schnitt (S, T) mit $\mathrm{cap}(S, T) = \mathrm{val}(f)$ ($\Rightarrow f$ ist maximal)

$S :=$ in N_f von s aus erreichbare Knoten; $T := V \setminus S$.

$$\left.\begin{array}{r} s \text{ von } s \text{ aus in } N_f \text{ erreichbar} \Rightarrow s \in S \\ t \text{ von } s \text{ aus nicht erreichbar} \Rightarrow t \notin S \end{array}\right\} \Rightarrow (S, T) \text{ ist } s\text{-}t\text{-Schnitt.}$$



$f(S, T) = \mathrm{cap}(S, T)$

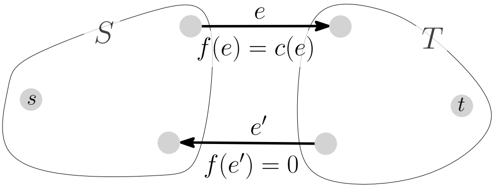
$f(T, S) = 0$

Beweis

Es gibt im Restnetzwerk N_f keinen gerichteten s-t-Pfad
$\Rightarrow\ \exists$ s-t-Schnitt (S, T) mit $\mathrm{cap}(S, T) = \mathrm{val}(f)$ ($\Rightarrow f$ ist maximal)

$S :=$ in N_f von s aus erreichbare Knoten; $T := V \setminus S$.

$$\left.\begin{array}{r} s \text{ von } s \text{ aus in } N_f \text{ erreichbar} \Rightarrow s \in S \\ t \text{ von } s \text{ aus nicht erreichbar} \Rightarrow t \notin S \end{array}\right\} \Rightarrow (S, T) \text{ ist } s\text{-}t\text{-Schnitt.}$$

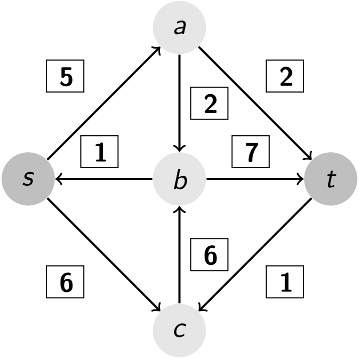


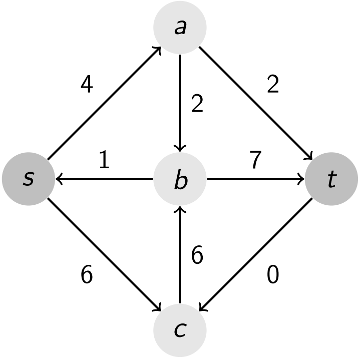
$f(S, T) = \mathrm{cap}(S, T)$

$f(T, S) = 0$

$$\mathrm{val}(f) = \underbrace{f(S, T)}_{=\mathrm{cap}(S,T)} - \underbrace{f(T, S)}_{=0} = \mathrm{cap}(S, T) \qquad \square$$

Beweis – Beispiel „Finde den Schnitt“





Fluss mit Wert $4 - 1 + 6 = 9$

Charakterisierung maximaler Fluss

Satz

Sei N ein Netzwerk (ohne entgegegen gerichtete Kanten).

Ein Fluss f ist maximaler Fluss

$\Leftrightarrow$

es im Restnetzwerk N_f keinen gerichteten s-t-Pfad gibt.

Für jeden maximalen Fluss f

gibt es einen s-t-Schnitt (S, T) mit $\mathrm{val}(f) = \mathrm{cap}(S, T)$.

Charakterisierung maximaler Fluss

Satz

Sei N ein Netzwerk (ohne entgegegen gerichtete Kanten).

Ein Fluss f ist maximaler Fluss

$\Leftrightarrow$

es im Restnetzwerk N_f keinen gerichteten s-t-Pfad gibt.

Für jeden maximalen Fluss f

gibt es einen s-t-Schnitt (S, T) mit $\mathrm{val}(f) = \mathrm{cap}(S, T)$.

- Zeigt noch nicht, dass es immer einen maximalen Fluss gibt.

Ford-Fulkerson Algorithmus

Ford-Fulkerson(V, A, c, s, t)

$f \leftarrow \mathbf{0}$ $\qquad\triangleright$ Fluss konstant 0
while $\exists$ s-t-Pfad P in N_f **do** $\qquad\triangleright$ augmentierender Pfad
    Augmentiere den Fluss entlang P
return f $\qquad\triangleright$ maximaler Fluss

Ford-Fulkerson Algorithmus

Ford-Fulkerson(V, A, c, s, t)

$f \leftarrow \mathbf{0}$ ▷ Fluss konstant 0
while $\exists$ s-t-Pfad P in N_f **do** ▷ augmentierender Pfad
  Augmentiere den Fluss entlang P
return f ▷ maximaler Fluss

- Wir können nicht garantieren, dass der Algorithmus terminiert.

Ford-Fulkerson Algorithmus

Ford-Fulkerson(V, A, c, s, t)

$f \leftarrow \mathbf{0}$ ▷ Fluss konstant 0
while $\exists$ s-t-Pfad P in N_f **do** ▷ augmentierender Pfad
    Augmentiere den Fluss entlang P
return f ▷ maximaler Fluss

- Wir können nicht garantieren, dass der Algorithmus terminiert.
- Der Algorithmus kann bei Kapazitäten aus $\mathbb{R}$ unendlich laufen.

Ford-Fulkerson Algorithmus

Ford-Fulkerson(V, A, c, s, t)

$f \leftarrow \mathbf{0}$	▷ Fluss konstant 0
while $\exists$ s-t-Pfad P in N_f **do**	▷ augmentierender Pfad
Augmentiere den Fluss entlang P	
return f	▷ maximaler Fluss

- Wir können nicht garantieren, dass der Algorithmus terminiert.
- Der Algorithmus kann bei Kapazitäten aus $\mathbb{R}$ unendlich laufen.
- Bei Kapazitäten aus $\mathbb{N}_0$ bleiben im Algorithmus Flüsse und Restkapazitäten ganzzahlig. In jedem Augmentierungsschritt wird der Fluss ganzzahlig ≥ 1 verbessert. D.h. insbesondere auch, dass das Ergebnis **ganzzahlig** ($A \to \mathbb{N}_0$) ist.

Analyse

Sei $n := |V|$ und $m := |A|$ für Netzwerk $N = (V, A, c, s, t)$.

[2] Vereinfachende Annahme, ist aber nicht essentiell.

Sei $n := |V|$ und $m := |A|$ für Netzwerk $N = (V, A, c, s, t)$.

- Angenommen $c\colon A \to \mathbb{N}_0$ und $U := \max_{e \in A} c(e)$. Dann gilt
$$\mathrm{val}(f) \leq \mathrm{cap}(\{s\}, V \setminus \{s\}) \leq (n-1)U$$
und es gibt **höchstens $(n-1)U$ Augmentierungsschritte**.

[2] Vereinfachende Annahme, ist aber nicht essentiell.

Analyse

Sei $n := |V|$ und $m := |A|$ für Netzwerk $N = (V, A, c, s, t)$.

- Angenommen $c\colon A \to \mathbb{N}_0$ und $U := \max_{e \in A} c(e)$. Dann gilt
$$\mathrm{val}(f) \leq \mathrm{cap}(\{s\}, V \setminus \{s\}) \leq (n-1)U$$
und es gibt **höchstens $(n-1)U$ Augmentierungsschritte**.
- **Ein Augmentierungsschritt**
  Suche s-t-Pfad in N_f, Augmentieren, Aktualisierung von N_f
  benötigt $O(m)$ Zeit.

[2] Vereinfachende Annahme, ist aber nicht essentiell.

Analyse

Sei $n := |V|$ und $m := |A|$ für Netzwerk $N = (V, A, c, s, t)$.

- Angenommen $c\colon A \to \mathbb{N}_0$ und $U := \max_{e \in A} c(e)$. Dann gilt
$$\mathrm{val}(f) \leq \mathrm{cap}(\{s\}, V \setminus \{s\}) \leq (n-1)U$$
und es gibt **höchstens $(n-1)U$ Augmentierungsschritte**.
- **Ein Augmentierungsschritt**
Suche s-t-Pfad in N_f, Augmentieren, Aktualisierung von N_f
benötigt $O(m)$ Zeit.

Satz (Ford-Fulkerson mit ganzzahligen Kapazitäten)

Sei $N = (V, A, c, s, t)$ ein Netzwerk mit $c : A \to \mathbb{N}_0^{\leq U}$, $U \in \mathbb{N}$, ohne entgegen gerichtete Kanten.[2] *Dann gibt es einen ganzzahligen maximalen Fluss. Er kann in Zeit $O(mnU)$ berechnet werden.*

[2] Vereinfachende Annahme, ist aber nicht essentiell.

Damit haben wir auch bewiesen.

Satz („Maxflow-Mincut Theorem“, ganzzahlig)

Jedes Netzwerk ohne entgegen gerichtete Kanten mit ganzzahligen Kapazitäten erfüllt

$$\max_{f \text{ Fluss}} \mathrm{val}(f) = \min_{(S,T) \text{ s-t-Schnitt}} \mathrm{cap}(S, T) \ .$$

Damit haben wir auch bewiesen.

Satz („**Maxflow-Mincut Theorem**“, ganzzahlig)

Jedes Netzwerk ohne entgegen gerichtete Kanten mit ganzzahligen Kapazitäten erfüllt

$$\max_{f \text{ Fluss}} \mathrm{val}(f) = \min_{(S,T) \text{ s-t-Schnitt}} \mathrm{cap}(S,T) \ .$$

Der Satz gilt auch, wenn das Netzwerk entgegen gerichtete Kanten hat.

Maxflow-Mincut Theorem

Damit haben wir auch bewiesen.

Satz („Maxflow-Mincut Theorem“, ganzzahlig)

Jedes Netzwerk ohne entgegen gerichtete Kanten mit ganzzahligen Kapazitäten erfüllt

$$\max_{f \text{ Fluss}} \mathrm{val}(f) = \min_{(S,T) \text{ s-t-Schnitt}} \mathrm{cap}(S,T) \ .$$

Der Satz gilt auch, wenn das Netzwerk entgegen gerichtete Kanten hat. Und er gilt auch bei beliebigen reellen Kapazitäten.

Anmerkungen

- **Capacity-Scaling** [Dinitz-Gabow'73] Sind in einem Netzwerk alle Kapazitäten ganzzahlig und höchstens U, so kann ein ganzzahliger maximaler Fluss in Zeit $O(mn(1 + \log U))$ berechnet werden kann.

Anmerkungen

- **Capacity-Scaling** [Dinitz-Gabow'73] Sind in einem Netzwerk alle Kapazitäten ganzzahlig und höchstens U, so kann ein ganzzahliger maximaler Fluss in Zeit $O(mn(1 + \log U))$ berechnet werden kann.
- **Dynamic Trees** [Sleator-Tarjan'83] Der maximale Fluss eines Netzwerks kann in Zeit $O(mn \log n)$ berechnet werden.

Anmerkungen

- **Capacity-Scaling** [Dinitz-Gabow'73] Sind in einem Netzwerk alle Kapazitäten ganzzahlig und höchstens U, so kann ein ganzzahliger maximaler Fluss in Zeit $O(mn(1 + \log U))$ berechnet werden kann.
- **Dynamic Trees** [Sleator-Tarjan'83] Der maximale Fluss eines Netzwerks kann in Zeit $O(mn \log n)$ berechnet werden.
- Alle Schranken gelten nach Maxflow-Mincut auch für die Berechnung eines minimalen s-t-Schnitts.

Anmerkungen

- **Capacity-Scaling** [Dinitz-Gabow'73] Sind in einem Netzwerk alle Kapazitäten ganzzahlig und höchstens U, so kann ein ganzzahliger maximaler Fluss in Zeit $O(mn(1 + \log U))$ berechnet werden kann.
- **Dynamic Trees** [Sleator-Tarjan'83] Der maximale Fluss eines Netzwerks kann in Zeit $O(mn \log n)$ berechnet werden.
- Alle Schranken gelten nach Maxflow-Mincut auch für die Berechnung eines minimalen s-t-Schnitts.

- Wir besprechen als Nächstes weitere Anwendungen (Matchings, Bildsegmentierung).