TECHNISCHE UNIVERSITÄT WIEN
Vienna | Austria

Maximal flows in networks

Bachelorarbeit aus Diskreter Mathematik

Florian Schager

TU Wien, Vienna, Austria

19. April 2021

Problemstellung

Problem

Wir wollen in einem Flussnetzwerk die maximale Transportkapazität von einem ausgezeichnetem Knoten, der Quelle s, zum Abflussknoten t bestimmen.
Ursprünglich wurde das Problem in den 50ern zur Modellierung des sowjetischen Schienenverkehrs gestellt.
Anwendungen reichen von der Fluglinienplanung bis hin zur Segmentierung in der Bildverarbeitung.

Definitionen (1/2)

Definition (Flussnetzwerk)

Ein Flussnetzwerk $N = (G, c, s, t)$ besteht aus einem gerichteter Graph $G = (V, E)$ mit einer Kapazitätsfunktion $c : E \to \mathbb{R}^+$ mit zwei ausgezeichneten Knoten s und t, wobei t von s aus erreichbar sein soll.

Definitionen (1/2)

Definition (Flussnetzwerk)

Ein Flussnetzwerk $N = (G, c, s, t)$ besteht aus einem gerichteter Graph $G = (V, E)$ mit einer Kapazitätsfunktion $c : E \to \mathbb{R}^+$ mit zwei ausgezeichneten Knoten s und t, wobei t von s aus erreichbar sein soll.

Definition (Fluss)

Ein zulässiger Fluss durch N ist eine Funktion $f : E \to \mathbb{R}_0^+$, welche den folgenden Bedingungen genügt:

Definitionen (1/2)

Definition (Flussnetzwerk)

Ein Flussnetzwerk $N = (G, c, s, t)$ besteht aus einem gerichteter Graph $G = (V, E)$ mit einer Kapazitätsfunktion $c : E \to \mathbb{R}^+$ mit zwei ausgezeichneten Knoten s und t, wobei t von s aus erreichbar sein soll.

Definition (Fluss)

Ein zulässiger Fluss durch N ist eine Funktion $f : E \to \mathbb{R}_0^+$, welche den folgenden Bedingungen genügt:

1. $0 \leq f(e) \leq c(e)$ für jede Kante e; (Kapazitätsbeschränkung)

Definitionen (1/2)

Definition (Flussnetzwerk)

Ein Flussnetzwerk $N = (G, c, s, t)$ besteht aus einem gerichteter Graph $G = (V, E)$ mit einer Kapazitätsfunktion $c : E \to \mathbb{R}^+$ mit zwei ausgezeichneten Knoten s und t, wobei t von s aus erreichbar sein soll.

Definition (Fluss)

Ein zulässiger Fluss durch N ist eine Funktion $f : E \to \mathbb{R}_0^+$, welche den folgenden Bedingungen genügt:

1. $0 \leq f(e) \leq c(e)$ für jede Kante e; (Kapazitätsbeschränkung)
2. $\sum_{e^+=v} f(e) = \sum_{e^-=v} f(e)$ für jeden Knoten $v \neq s, t$.
   Dabei bezeichnen e^- und e^+ jeweils den Start- und Endpunkt der Kante e.
   (Flusserhaltung)

Definitionen (2/2)

Definition (Wert eines Flusses)

Die Größe

$$w(f) := \sum_{e^-=s} f(e) - \sum_{e^+=s} f(e) = \sum_{e^+=t} f(e) - \sum_{e^-=t} f(e)$$

nennen wir den Wert eines Flusses.

Definitionen (2/2)

Definition (Wert eines Flusses)

Die Größe

$$w(f) := \sum_{e^-=s} f(e) - \sum_{e^+=s} f(e) = \sum_{e^+=t} f(e) - \sum_{e^-=t} f(e)$$

nennen wir den Wert eines Flusses.

Definition (Cut)

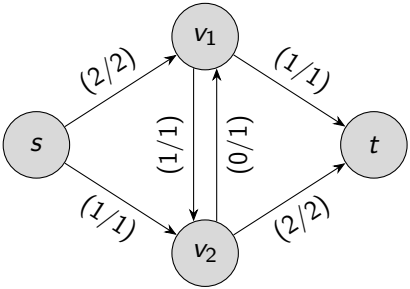
Ein Cut ist eine Partition $V = S \dot{\cup} T$ eines Flussnetzwerks $N = (G, c, s, t)$ mit $s \in S$ und $t \in T$. Die Kapazität eines Cuts ist gegeben durch

$$c(S, T) = \sum_{e^- \in S, s^+ \in T} c(e).$$

Ein Cut heißt minimal, wenn $c(S, T) \leq c(S', T')$ für alle Cuts (S', T').

Beispiel - Flüsse und Cuts

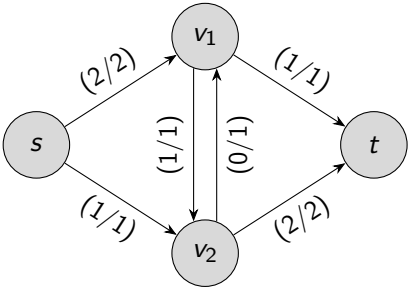
Maximaler Fluss f



(a) $w(f) = 3$

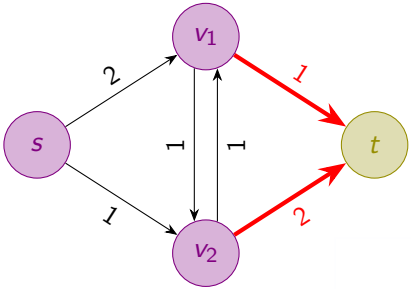
Beispiel - Flüsse und Cuts

Maximaler Fluss f



(c) $w(f) = 3$

Minminaler Cut (S, T)



(d) $c(S, T) = 3$

Erweiternde Pfade

Definition (Pfad)

Wir definieren einen Pfad in einem gerichteten Graphen als eine Folge $(e_1, \ldots, e_n)$ von Kanten, sodass die zugehörige Folge an ungerichteten Kanten ein Pfad in dem zugehörigen ungerichteten Graph ist.

Erweiternde Pfade

Definition (Pfad)

Wir definieren einen Pfad in einem gerichteten Graphen als eine Folge $(e_1, \ldots, e_n)$ von Kanten, sodass die zugehörige Folge an ungerichteten Kanten ein Pfad in dem zugehörigen ungerichteten Graph ist.
Sei $(v_0, \ldots, v_n)$ die zum Pfad zugehörige Knotenfolge. Dann heißt jede Kante der Form $v_{i-1}v_i$ eine Vorwärts-Kante und jede Kante der Form $v_i v_{i-1}$ eine Rückwärts-Kante.

Erweiternde Pfade

Definition (Pfad)

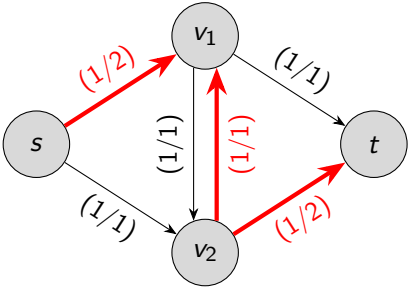
Wir definieren einen Pfad in einem gerichteten Graphen als eine Folge $(e_1, \ldots, e_n)$ von Kanten, sodass die zugehörige Folge an ungerichteten Kanten ein Pfad in dem zugehörigen ungerichteten Graph ist.
Sei $(v_0, \ldots, v_n)$ die zum Pfad zugehörige Knotenfolge. Dann heißt jede Kante der Form $v_{i-1}v_i$ eine Vorwärts-Kante und jede Kante der Form $v_i v_{i-1}$ eine Rückwärts-Kante.

Definition (Erweiternder Pfad)

Ein erweiternder Pfad bezüglich einem Fluss f in einem Flussnetzwerk $N = (G, c, s, t)$ ist ein Pfad P von s nach t, sodass $f(e) < c(e)$ für alle Vorwärts-Kanten $e \in P$ und $f(e) > 0$ für alle Rückwärts-Kanten $e \in P$ gilt.

Beispiel - Erweiternde Pfade

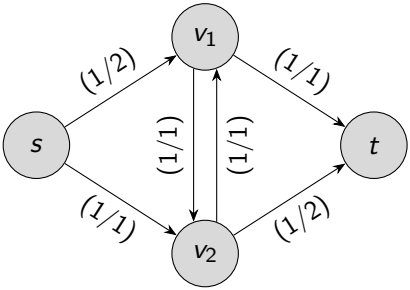
Erweiternder Pfad (s, v_1, v_2, t)



(e) $w(f_0) = 2$

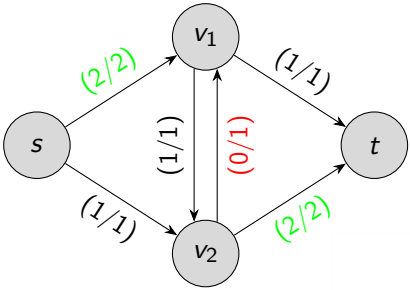
Beispiel - Erweiternde Pfade

Erweiternder Pfad (s, v_1, v_2, t)



(g) $w(f_0) = 2$

Erweiterter Fluss f_1



(h) $w(f_1) = 3$

Wichtige Resultate

Satz (Augmenting Path Theorem)

Ein Fluss f in einem Flussnetzwerk $N = (G, c, s, t)$ ist genau dann maximal, wenn es keine erweiternden Pfade bezüglich f gibt.

Wichtige Resultate

Satz (Augmenting Path Theorem)

Ein Fluss f in einem Flussnetzwerk $N = (G, c, s, t)$ ist genau dann maximal, wenn es keine erweiternden Pfade bezüglich f gibt.

Satz (Integral Flow Theorem)

Sei $N = (G, c, s, t)$ ein Flussnetzwerk mit ausschließlich ganzzahligen Kapazitäten. Dann existiert ein maximaler Fluss, der ebenso nur aus ganzzahligen Werten $f(e)$ besteht.

Wichtige Resultate

Satz (Augmenting Path Theorem)

Ein Fluss f in einem Flussnetzwerk $N = (G, c, s, t)$ ist genau dann maximal, wenn es keine erweiternden Pfade bezüglich f gibt.

Satz (Integral Flow Theorem)

Sei $N = (G, c, s, t)$ ein Flussnetzwerk mit ausschließlich ganzzahligen Kapazitäten. Dann existiert ein maximaler Fluss, der ebenso nur aus ganzzahligen Werten $f(e)$ besteht.

Satz (Max-Flow Min-Cut Theorem)

Der maximale Wert eines Flusses in einem Flussnetzwerk N entspricht der minimalen Kapazität eines Cuts in N.

Algorithmus: Grundidee

Algorithmus

- Starte mit dem trivialen Fluss: $f(e) = 0, e \in E$.

Algorithmus: Grundidee

Algorithmus

- Starte mit dem trivialen Fluss: $f(e) = 0, e \in E$.
- Finde einen erweiternden Pfad P.

Algorithmus: Grundidee

Algorithmus

- Starte mit dem trivialen Fluss: $f(e) = 0, e \in E$.
- Finde einen erweiternden Pfad P.
- Berechne

$$d := \min[\{c(e) - f(e) : e \text{ Vorwärts-Kante } \in P\} \cup \{f(e) : e \text{ Rückwärts-Kante } \in P\}].$$

Algorithmus: Grundidee

Algorithmus

- Starte mit dem trivialen Fluss: $f(e) = 0, e \in E$.
- Finde einen erweiternden Pfad P.
- Berechne

$$d := \min[\{c(e) - f(e) : e \text{ Vorwärts-Kante } \in P\} \cup \{f(e) : e \text{ Rückwärts-Kante } \in P\}].$$

- Konstruiere erweiterten Fluss f' mit $w(f') = w(f) + d$:

$$f'(e) = \begin{cases} f(e) + d, & e \text{ ist Vorwärts-Kante } \in P \\ f(e) - d, & e \text{ ist Rückwärts-Kante } \in P \\ f(e), & \text{sonst} \end{cases}$$

Algorithmus: Grundidee

Algorithmus

- Starte mit dem trivialen Fluss: $f(e) = 0, e \in E$.
- Finde einen erweiternden Pfad P.
- Berechne

$$d := \min[\{c(e) - f(e) : e \text{ Vorwärts-Kante } \in P\} \cup \{f(e) : e \text{ Rückwärts-Kante } \in P\}].$$

- Konstruiere erweiterten Fluss f' mit $w(f') = w(f) + d$:

$$f'(e) = \begin{cases} f(e) + d, & e \text{ ist Vorwärts-Kante } \in P \\ f(e) - d, & e \text{ ist Rückwärts-Kante } \in P \\ f(e), & \text{sonst} \end{cases}$$

- Wiederhole solange, bis kein erweiternder Pfad mehr gefunden werden kann.

Algorithmus: Ford-Fulkerson - Beispiel

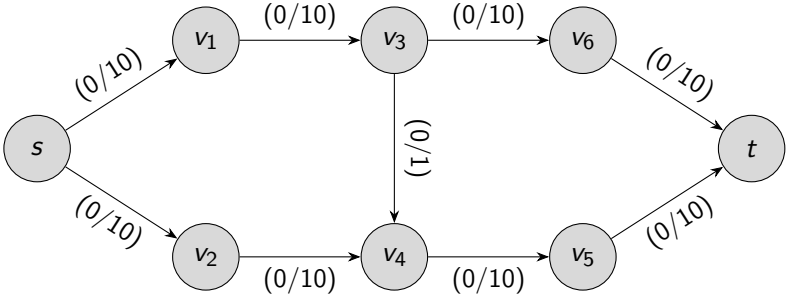


Abbildung: Flussnetzwerk, Kanten mit $(f(e)/c(e))$ markiert

Algorithmus: Ford-Fulkerson - Beispiel

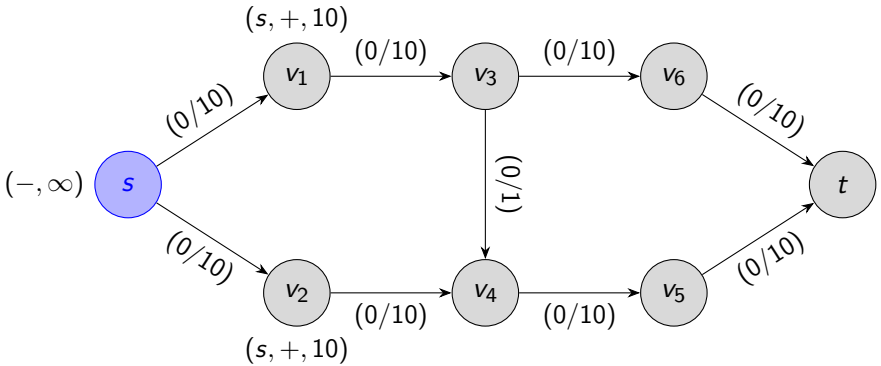


Abbildung: $w(f_0) = 0$

Algorithmus: Ford-Fulkerson - Beispiel

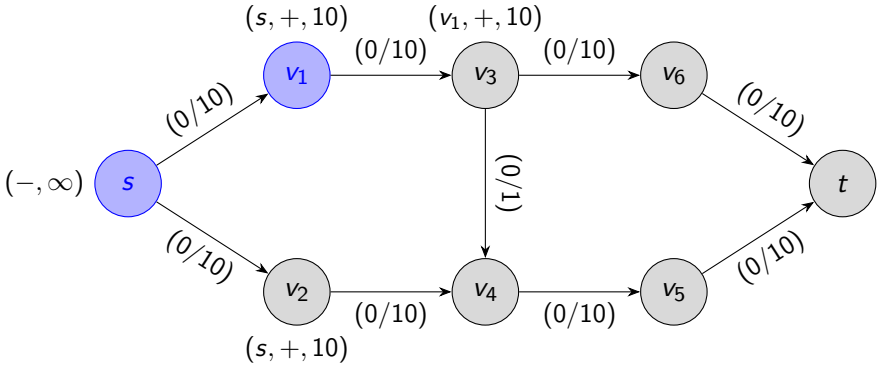


Abbildung: $w(f_0) = 0$

Algorithmus: Ford-Fulkerson - Beispiel

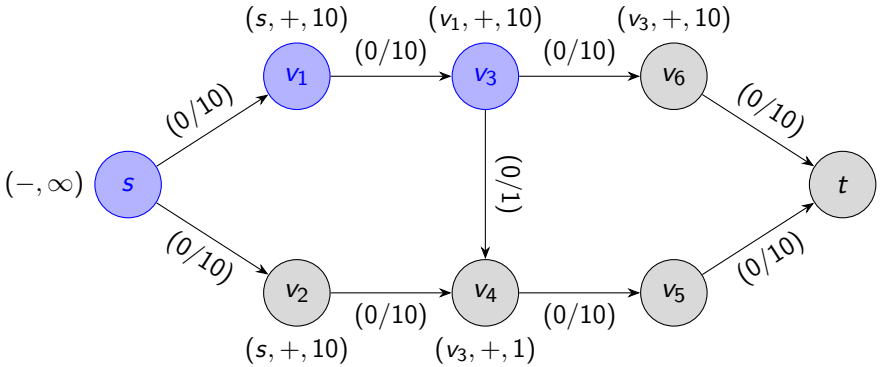


Abbildung: $w(f_0) = 0$

Algorithmus: Ford-Fulkerson - Beispiel

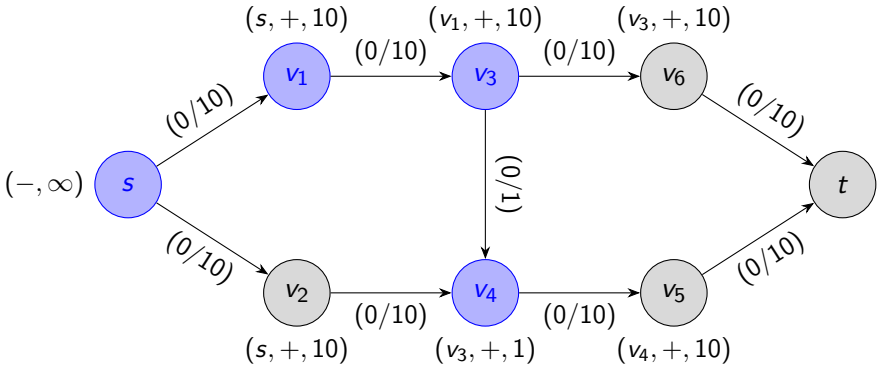


Abbildung: $w(f_0) = 0$

Algorithmus: Ford-Fulkerson - Beispiel

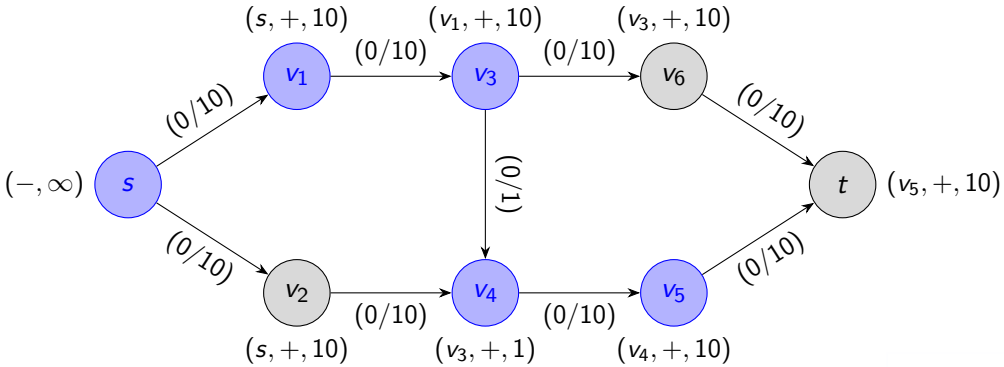


Abbildung: $w(f_0) = 0$

Algorithmus: Ford-Fulkerson - Beispiel

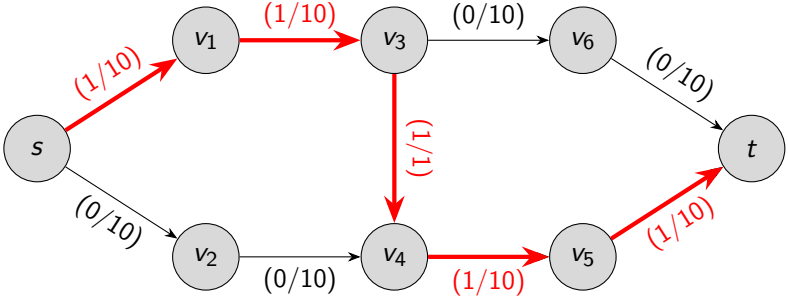


Abbildung: $w(f_1) = 1$

Algorithmus: Ford-Fulkerson - Beispiel

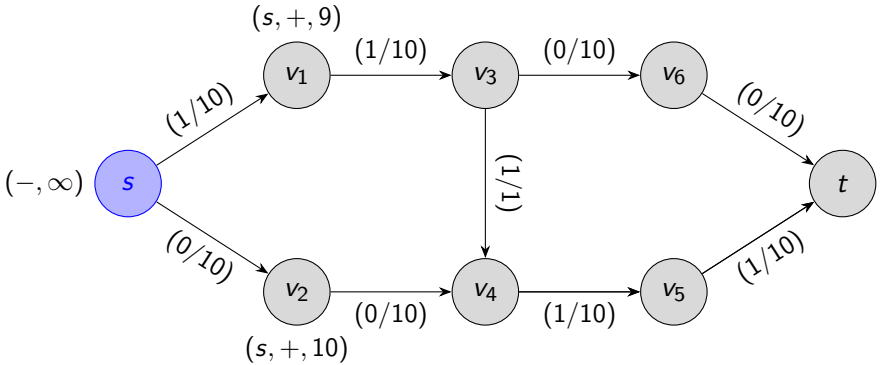


Abbildung: $w(f_1) = 1$

Algorithmus: Ford-Fulkerson - Beispiel

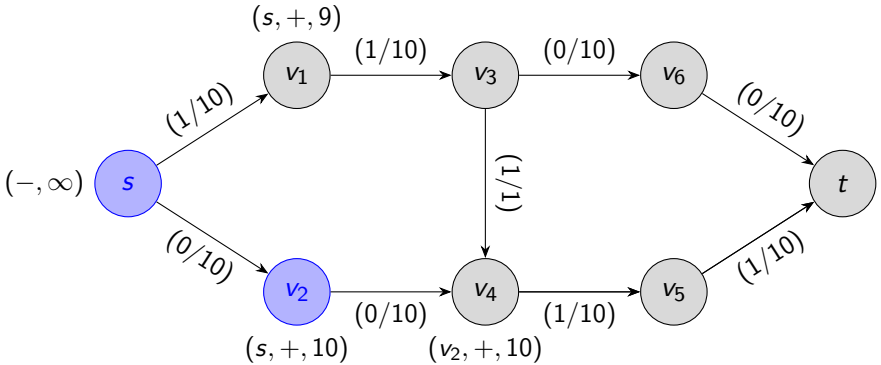


Abbildung: $w(f_1) = 1$

Algorithmus: Ford-Fulkerson - Beispiel

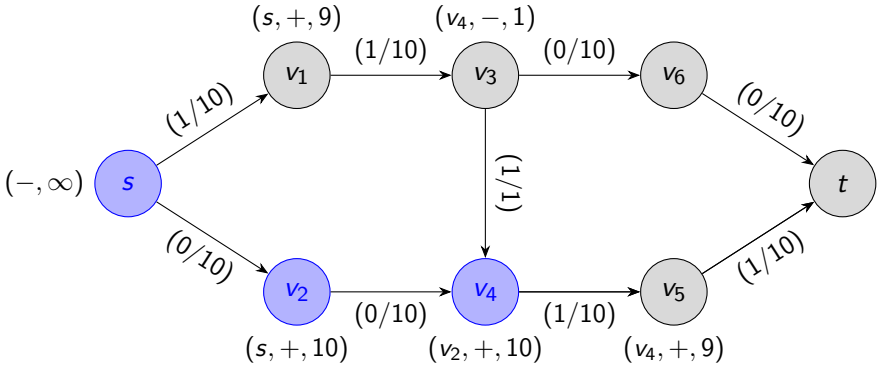


Abbildung: $w(f_1) = 1$

Algorithmus: Ford-Fulkerson - Beispiel

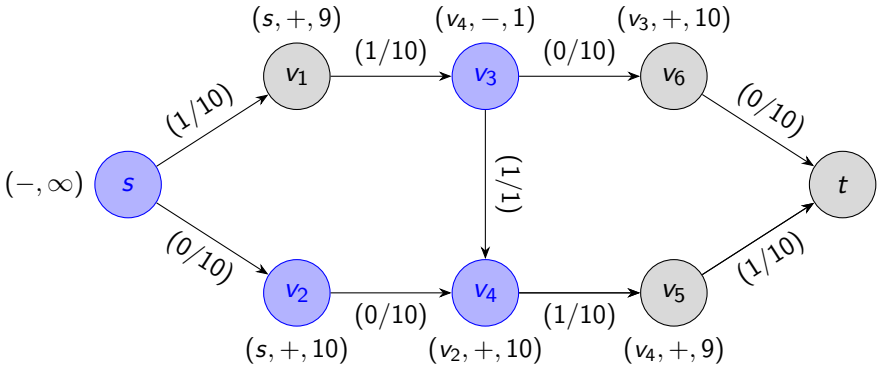


Abbildung: $w(f_1) = 1$

Algorithmus: Ford-Fulkerson - Beispiel

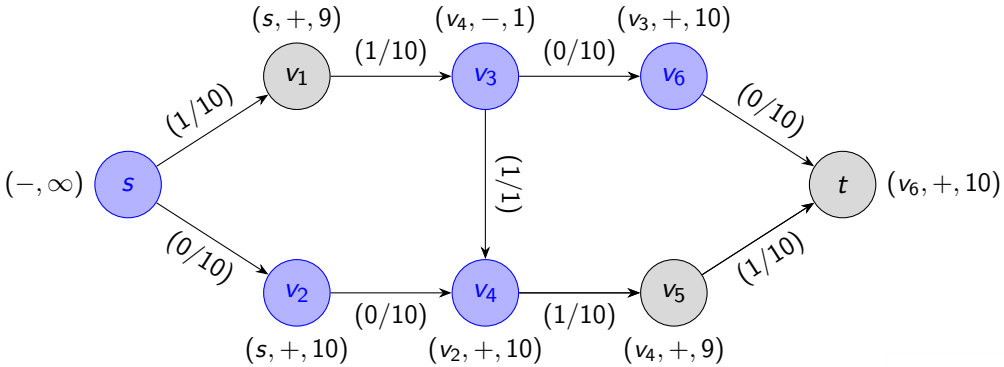


Abbildung: $w(f_1) = 1$

Algorithmus: Ford-Fulkerson - Beispiel

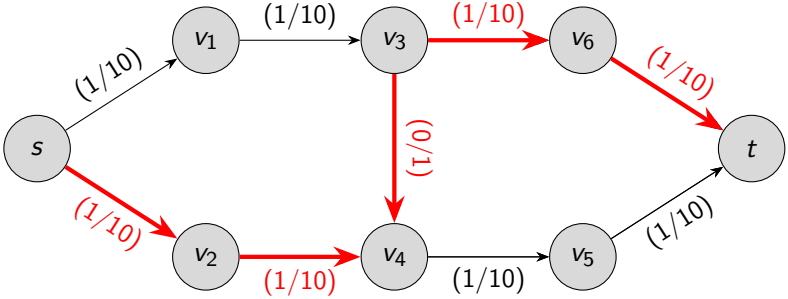


Abbildung: $w(f_2) = 2$

Algorithmus: Ford-Fulkerson - Analyse

- Der Algorithmus garantiert noch nicht effiziente Wahl der erweiternden Pfade und hat nicht polynomielle Laufzeit in Abhängigkeit von $|V|$ und $|E|$, da die Laufzeit auch noch von der Kapazitätsfunktion c abhängt.

Algorithmus: Ford-Fulkerson - Analyse

- Der Algorithmus garantiert noch nicht effiziente Wahl der erweiternden Pfade und hat nicht polynomielle Laufzeit in Abhängigkeit von $|V|$ und $|E|$, da die Laufzeit auch noch von der Kapazitätsfunktion c abhängt.
- Kann für irrationale Kapazitäten scheitern, ist aber in der Praxis irrelevant.

Algorithmus: Ford-Fulkerson - Analyse

- Der Algorithmus garantiert noch nicht effiziente Wahl der erweiternden Pfade und hat nicht polynomielle Laufzeit in Abhängigkeit von $|V|$ und $|E|$, da die Laufzeit auch noch von der Kapazitätsfunktion c abhängt.
- Kann für irrationale Kapazitäten scheitern, ist aber in der Praxis irrelevant.

Modifikation (Edmonds und Karp)

Wird die Reihenfolge, in der die Knoten mit Labeln versehen werden, gemäß einer Breitensuche gewählt, so lässt sich der Aufwand auf $O(|V||E|^2)$ reduzieren.

Vertiefung

Weiterführende Algorithmen

- Blocking Flows
- Push-Relabel-Algorithmus von Goldberg und Tarjan

Vertiefung

Weiterführende Algorithmen

- Blocking Flows
- Push-Relabel-Algorithmus von Goldberg und Tarjan

Anwendungen

Welche Probleme können auf ein Maximum Flow Problem reduziert werden?

- Maximale Anzahl disjunkter Pfade von der Quelle zum Abfluss.
- Knotenüberdeckungsproblem.
- Matrix-Rundungsproblem.