Datenstrukturen und Algorithmen

Vorlesung 19: Maximaler Fluss

Joost-Pieter Katoen

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

http://www-i2.informatik.rwth-aachen.de/i2/dsal10/

9. Juli 2010

RWTH AACHEN UNIVERSITY

Übersicht

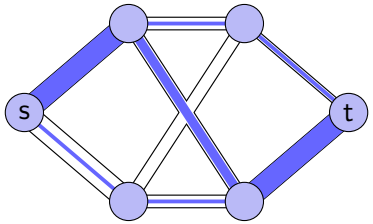
Übersicht

Flussnetzwerk

Flussnetzwerk

Ein Flussnetzwerk G ist ein Tripel $G = (V, E, c)$, wobei:

- (V, E) ein gerichteter Graph ist,
- $c : E \longrightarrow \mathbb{R}^{\geqslant 0}$ die Kapazitätsfunktion und
- $s, t \in V$ Quelle s und Senke t des Flussnetzwerkes.



- Kanten können als Wasserrohre interpretiert werden.
- Die Kapazität gibt die maximale Durchsatzrate (l/s) an.

Fluss in einem Flussnetzwerk

Definition (Fluss)

Ein Fluss ist eine Funktion $f\colon V \times V \to \mathbb{R}$, mit folgenden Eigenschaften:

Beschränkung: Für $u, v \in V$ gilt $f(u,v) \leqslant c(u,v)$.

Asymmetrie: Für $u, v \in V$ gilt $f(u,v) = -f(v,u)$.

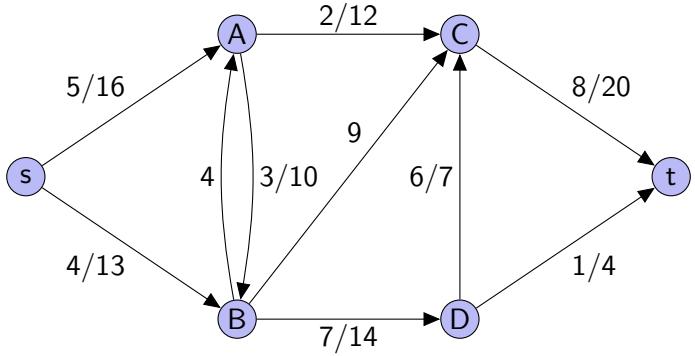
Flusserhaltung: Für $u \in V - \{s,t\}$ gilt: $\sum_{v \in V} f(u,v) = 0$.

Definition (Wert eines Flusses)

Der Wert $|f|$ eines Flusses ist der Gesamtfluss aus der Quelle s:

$$|f| = \sum_{u \in V} f(s,u).$$

Darstellung von Flüssen



- Jeder Knoten liegt auf einem Pfad von der Quelle s zur Senke t.
- Für $(u,v) \notin E$ ist $c(u,v) = 0$.
- Wir beschriften Kanten mit $f(u,v)/c(u,v)$, falls $f(u,v) > 0$.
- Negative Flüsse $f(u,v) < 0$ werden nicht explizit angegeben.
- Der eingezeichnete Fluss f hat den Wert $|f| = 9$.

Maximale Flüsse

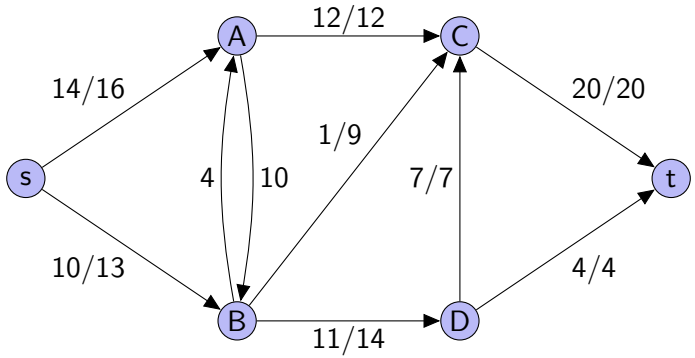
Problem (Maximaler Fluss)

Finde einen maximalen Fluss in einem gegebenen Flussnetzwerk.

Beispiel (Anwendungen)

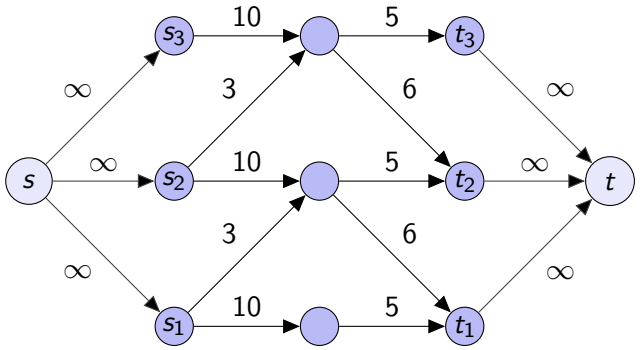
- Wie groß ist der maximale Datendurchsatz zwischen zwei Computern in einem Netzwerk?
- Wie kann der Verkehr in einem Straßennetz so geleitet werden, dass möglichst viele Autos in einer gegeben Zeitspanne ein Ziel erreichen?
- Wie viele Leitungen müssen zerstört sein, damit zwei Computer nicht mehr miteinander kommunizieren können?

Ein maximaler Fluss



- Ein maximaler Fluss in diesem Beispiel hat den Wert $|f| = 24$.
- Es kann mehrere maximale Flüsse geben.

Mehrere Quellen und Senken



- Es kann auch Flussnetzwerke mit mehrere Quellen oder Senken geben.
- Sie können durch eine neue „Superquelle“ und „Supersenke“ in ein Standard-Flussnetzwerk überführt werden.

Flüsse zwischen Knotenmengen

Notationen

$$
\begin{aligned}
f(x, Y) &= \sum_{y \in Y} f(x, y) && \text{für } Y \subseteq V \\
f(X, y) &= \sum_{x \in X} f(x, y) && \text{für } X \subseteq V \\
f(X, Y) &= \sum_{x \in X} \sum_{y \in Y} f(x, y) && \text{für } X, Y \subseteq V
\end{aligned}
$$

Eigenschaften von Flüssen zwischen Mengen

Falls f ein Fluss für $G = (V, E, c)$ ist, dann gilt:

1. $f(X, X) = 0$ für $X \subseteq V$
2. $f(X, Y) = -f(Y, X)$ für $X, Y \subseteq V$
3. $f(X \cup Y, Z) = f(X, Z) + f(Y, Z)$ für $X, Y, Z \subseteq V : X \cap Y = \varnothing$
4. $f(Z, X \cup Y) = f(Z, X) + f(Z, Y)$ für $X, Y, Z \subseteq V : X \cap Y = \varnothing$

Beweis: $f(X, X) = 0$

Beh.: $f(X, X) = 0$

$$
\begin{aligned}
f(X, X) &= \frac{1}{2}\Big(\sum_{x_1 \in X} \sum_{x_2 \in X} f(x_1, x_2) + \sum_{x_1 \in X} \sum_{x_2 \in X} f(x_1, x_2) \Big) \\
&= \frac{1}{2}\Big(\sum_{x_1 \in X} \sum_{x_2 \in X} f(x_1, x_2) + \sum_{x_1 \in X} \sum_{x_2 \in X} f(x_2, x_1) \Big) \\
&= \frac{1}{2} \sum_{x_1 \in X} \sum_{x_2 \in X} \Big(f(x_1, x_2) + f(x_2, x_1) \Big) \\
&= 0
\end{aligned}
$$

□

Für den Beweis benötigen wir lediglich die Eigenschaft der Asymmetrie.

Eingehender Fluss in der Senke

Wie groß ist der an der Senke eingehende Fluss?

Aufgrund der Flusserhaltung ist zu erwarten, dass er dem austretenden Fluss an der Quelle entspricht:

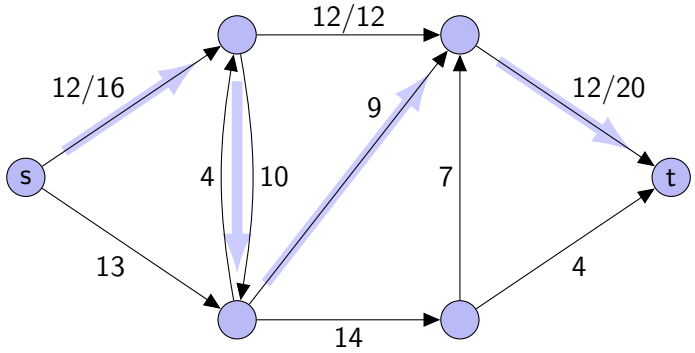
$$f(s, V) = f(V, t)$$

Beweis:

$$
\begin{aligned}
f(s, V) &= f(V, V) - f(V - \{s\}, V) \\
&= -f(V - \{s\}, V) \\
&= f(V, V - \{s\}) \\
&= f(V, t) + f(V, V - \{s, t\}) && \mid \text{Flusserhaltung} \\
&= f(V, t)
\end{aligned}
$$

Übersicht

Ford-Fulkerson-Methode – Idee



1. Suche einen Pfad p von s nach t.
2. Setze den Fluss der Kanten in p um die kleinste Kapazität in p.
3. Suche einen Pfad p von s nach t, aus Kanten mit freier Kapazität.
4. Ergänze den Fluss der Kanten in p um die kleinste Restkapazität in p.
5. Wiederhole 3. und 4. bis es keinen Pfad p mehr gibt.

Restnetzwerke

„Netzwerk minus Fluss = Restnetzwerk“

Definition (Restnetzwerk G_f)

Gegeben sei das Flussnetzwerk $G = (V, E, c)$ und ein Fluss f, dann ist $G_f = (V, E_f, c_f)$ das Restnetzwerk (auch: Residualnetzwerk) zu G und f mit:

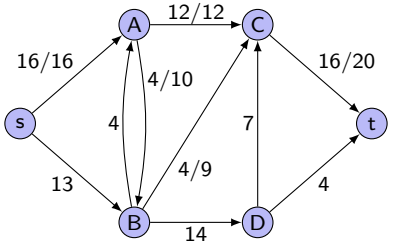
$$E_f = \{ (u, v) \in V \times V \mid c_f(u, v) > 0 \},$$

und

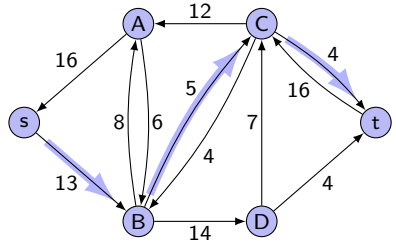
$$c_f(u, v) = c(u, v) - f(u, v),$$

c_f die Restkapazität von G.

Augmentierende Pfade



Flussnetzwerk G

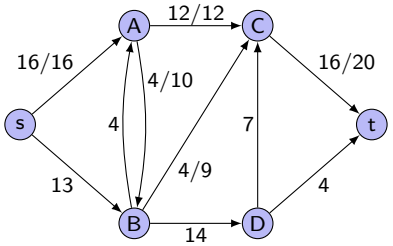


Restnetzwerk G_f

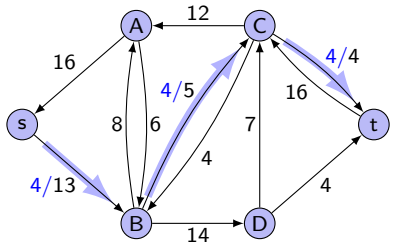
- Ein s-t-Pfad p in G_f heißt augmentierender Pfad (vergrößernder Pfad).
- $c_f(p) = \min\{ c_f(u, v) \mid (u, v) \in p \}$ heißt Restkapazität von p.

Der Pfad im obigen Beispiel hat die Restkapazität 4.

Augmentierende Pfade



Flussnetzwerk G



Restnetzwerk G_f

Entlang p lässt sich in G_f der Fluss $f_p(u, v)$ konstruieren mit:

$$f_p(u,v) = \begin{cases} c_f(p) & \text{falls } (u,v) \text{ auf } p \\ -c_f(p) & \text{falls } (v,u) \text{ auf } p \\ 0 & \text{sonst} \end{cases}$$

Ford-Fulkerson-Theorem

Theorem (Ford-Fulkerson)

Sei $G = (V, E, c)$ ein Flussnetzwerk und f ein Fluss in G, sowie f' ein Fluss in G_f.

Dann ist $f + f'$ ein Fluss in G.

Beweis.

Wir müssen zeigen, dass $f + f'$ beschränkt, asymmetrisch und flusserhaltend ist (nächste Folie). □

1. Asymmetrie:

$$\begin{aligned} (f+f')(u,v) &= f(u,v) + f'(u,v) \\ &= -f(v,u) - f'(v,u) \\ &= -(f(v,u) + f'(v,u)) \\ &= -(f+f')(v,u) \end{aligned}$$

2. Flusserhaltung:

$$(f+f')(u,V) = f(u,V) + f'(u,V) = 0 \qquad | \; \forall u \in V - \{s,t\}$$

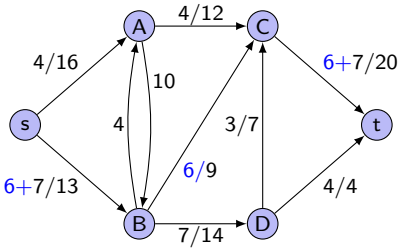
3. Beschränkung:

$$\begin{aligned} (f+f')(u,v) &= f(u,v) + f'(u,v) \\ &\leqslant f(u,v) + c_f(u,v) \\ &= f(u,v) + (c(u,v) - f(u,v)) \\ &= c(u,v) \end{aligned}$$

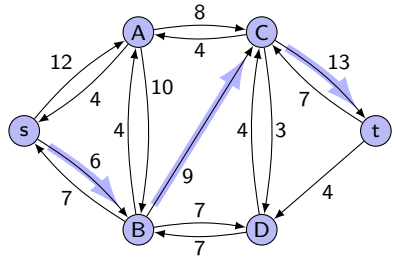
Die Ford-Fulkerson-Methode

Algorithmus

Initialisiere Fluss f zu 0
while es gibt einen augmentierenden Pfad p
  do augmentiere f entlang p // $f := f + f_p$
return f



Flussnetzwerk G



Restnetzwerk G_f

Implementierung Ford-Fulkerson-Methode

```
1  int[n,n] maxFlow(List adjLst[n], int n, int s, int t) {
2    int flow[n,n] = 0, path[];
3    int cfp; // Restkapazität des Pfades
4
5    while (true) {
6      // Finde augmentierenden Pfad und dessen Restkapazität
7      (path, cfp) = augmentPfad(adjLst, flow, s, t);
8      if (cfp == 0) { // kein Pfad gefunden
9        return flow;
10     }
11
12     // addiere Restkapazität entlang des Pfades zum Fluss
13     for (int i = 1; i < pfad.length; i++) {
14       int u = pfad[i-1], v = pfad[i];
15       f[u,v] = f[u,v] + cfp;
16       f[v,u] = -f[u,v];
17     }
18   }
19 }
```


Laufzeit der Ford-Fulkerson-Methode

Ein Flussproblem ist integral, wenn alle Kapazitäten ganzzahlig sind.

Theorem

Sei f^* der durch die Ford-Fulkerson-Methode bestimmte Fluss zu einem integralen Flussproblem, so benötigt die Methode $|f^*|$ Iterationen und es ergibt sich eine Laufzeit von $O(E \cdot |f^*|)$.

Beweis.

In jeder Iteration wird der Wert des Flusses um $c_f(p) \geqslant 1$ erhöht.
Er ist anfangs 0 und am Ende f^*. □

Korollar

Bei rationalen Kapazitäten terminiert die Ford–Fulkerson–Methode.
Brüche können durch Multiplikation aufgehoben werden.

- Für ein integrales Flussproblem bestimmt die Ford-Fulkerson-Methode einen Fluss f, sodass jedes $f(u, v)$ ganzzahlig ist.

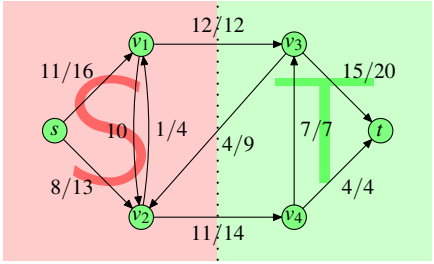
Schnitte in Flussnetzwerken

Wir zeigen mittels Schnitten, dass ein maximaler Fluss berechnet wird:

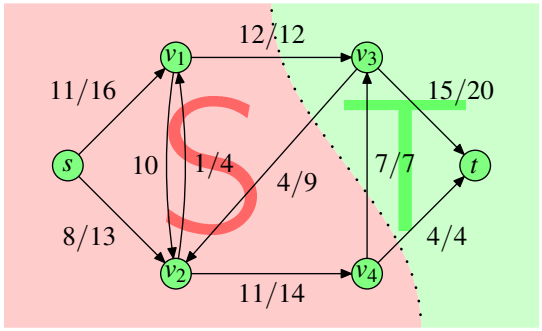
Definition

Ein Schnitt (S, T) in einem Flussnetzwerk $G = (V, E, c)$ ist eine Partition $S \cup T = V$, $S \cap T = \varnothing$ mit $s \in S$ und $t \in T$.

- Wenn f ein Fluss in G ist, dann ist $f(S, T)$ der Fluss über (S, T).
- Die Kapazität von (S, T) ist $c(S, T)$.
- Ein minimaler Schnitt ist ein Schnitt mit minimaler Kapazität.



Schnitte in Netzwerken



S	:	$\{s, v_1, v_2\}$	$\{s\}$	$\{s, v_1, v_2, v_4\}$
T	:	$\{t, v_3, v_4\}$	$\{t, v_1, v_2, v_3, v_4\}$	$\{t, v_3\}$
Fluss	:	19	19	19
Kapazität	:	26	29	23

- Für den Fluss über einen Schnitt gilt: $f(S, T) = |f| \leqslant c(S, T)$.

Max-flow Min-cut Theorem

Theorem (Max-flow Min-cut)

Sei f ein Fluss im Flussnetzwerk $G = (V, E, c)$, dann sind äquivalent:

1. *f ist ein maximaler Fluss.*
2. *In G_f gibt es keinen augmentierenden Pfad.*
3. *$|f| = c(S, T)$ für einen Schnitt (S, T), d. h. (S,T) ist minimal.*

Folgerungen

1. Die Kapazität eines minimalen Schnittes ist gleich dem Wert eines maximalen Flusses.
2. Falls die Ford–Fulkerson–Methode terminiert, berechnet sie einen maximalen Fluss.

Maximaler Fluss | Ford-Fulkerson-Methode

Max-flow Min-cut Theorem

Theorem (Max-flow Min-cut)

Sei f ein Fluss im Flussnetzwerk $G = (V, E, c)$, dann sind äquivalent:

1. *f ist ein maximaler Fluss.*
2. *In G_f gibt es keinen augmentierenden Pfad.*
3. *$|f| = c(S, T)$ für einen Schnitt (S, T), d. h. (S,T) ist minimal.*

1. ⇒ 2. (Widerspruchsbeweis).

Sei f ein maximaler Fluss und p einen augmentierender Pfad.

⇒ $f + f_p$ ist ein Fluss in G mit $|f + f_p| > |f|$.

⇒ Widerspruch! Denn f ist maximaler Fluss.

□

Joost-Pieter Katoen | Datenstrukturen und Algorithmen | 24/27

Maximaler Fluss | Ford-Fulkerson-Methode

Max-flow Min-cut Theorem

Theorem (Max-flow Min-cut)

Sei f ein Fluss im Flussnetzwerk $G = (V, E, c)$, dann sind äquivalent:

1. *f ist ein maximaler Fluss.*
2. *In G_f gibt es keinen augmentierenden Pfad.*
3. *$|f| = c(S, T)$ für einen Schnitt (S, T), d. h. (S,T) ist minimal.*

2. ⇒ 3.

Es gibt keinen s-t-Pfad in G_f.

Sei $S := \{ v \in V \mid \exists\ s\text{-}v\text{-Pfad in } G_f \}$ und $T := V - S$, dann gilt:

1. $\forall u \in S,\ v \in T$ gilt: $c_f(u, v) = 0 \quad \Rightarrow \quad f(u, v) = c(u, v)$.
2. (S,T) ist ein Schnitt und somit gilt $f(S, T) = |f|$.

⇒ $c(S, T) = f(S, T) = |f|$

□

Joost-Pieter Katoen | Datenstrukturen und Algorithmen | 24/27

Maximaler Fluss | Ford-Fulkerson-Methode

Max-flow Min-cut Theorem

Theorem (Max-flow Min-cut)

Sei f ein Fluss im Flussnetzwerk $G = (V, E, c)$, dann sind äquivalent:

1. *f ist ein maximaler Fluss.*
2. *In G_f gibt es keinen augmentierenden Pfad.*
3. *$|f| = c(S, T)$ für einen Schnitt (S, T), d. h. (S,T) ist minimal.*

3. ⇒ 1.

Sei f' ein beliebiger Fluss dann gilt:

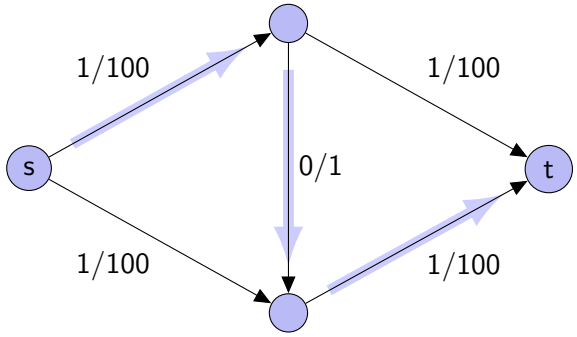
$$|f'| = f(S, T) = \sum_{u \in S} \sum_{v \in T} f'(u, v) \leqslant \sum_{u \in S} \sum_{v \in T} c(u, v) = c(S, T)$$

Da $|f| = c(S, T)$ und $\forall\, f' :\ |f'| \leqslant c(S, T)$, folgt f ist maximal.

□

Joost-Pieter Katoen | Datenstrukturen und Algorithmen | 24/27

Laufzeit der Ford-Fulkerson-Methode



Die Worst-Case-Laufzeit ist abhängig vom Wert eines maximalen Flusses.

Übersicht

Edmonds-Karp-Algorithmus

Edmonds-Karp-Algorithmus

Eine Implementierung der Ford-Fulkerson-Methode, die zur Bestimmung augmentierender Pfade eine Breitensuche nutzt wird als Edmonds-Karp-Algorithmus bezeichnet. ($O(V \cdot E^2)$)

Lemma

Wird der Edmonds-Karp-Algorithmus genutzt, so steigt für alle Knoten $v \in V - \{s, t\}$ der Abstand des kürzesten Pfades $\delta_f(s, v)$ im Restnetzwerk G_f monoton mit jeder Erweiterung des Flusses.

Theorem

Die Gesamtzahl der Iterationen im Edmonds-Karp-Algorithmus für das Flussnetzwerk $G = (V, E, c)$ ist in $O(V \cdot E)$.