



5 Maximale Flüsse

5.1 Transport-Problem

Ein Unternehmer möchte täglich möglichst viele seiner Waren von einem seiner Standorte zu einem anderen Standort transportieren. Er nutzt dazu LKW's anderer Unternehmen, die zwischen verschiedenen Städten hin und her fahren. Jeder hat eine maximale Kapazität, die er von einer zu einer andern Stadt transportieren kann. Jeder dieser LKW's hat allerdings eine maximale Kapazität an Waren, die er transportieren kann. Da es in den einzelnen Städten keine Lagerhäuser gibt, können die Waren nicht zwischengelagert werden, sondern müssen sofort weiter transportiert werden.

Der Ausgangsstandort in diesem Szenario stellt den Startknoten s und der zweite Standort den Endknoten t dar. Die LKW 's fahren entlang der Kanten und ihre Kapazität entspricht der Kapazität der jeweiligen Kante.

Frage: Was ist der maximale Fluss von Waren von s nach t ?

5.1.1 Formale Definition

Angelehnt an das Buch „Network Flows“ von Ahuja, Magnanti und Orlin.

Gegeben sei

- ein gerichteter Graph $G = (V, E)$
- eine Kapazitätsfunktion $u : E \rightarrow \mathbb{R}_0^+$ und
- zwei Knoten $s, t \in V$ mit $s \neq t$.

$u((v, w))$ heißt Kapazität von (v, w) . s heißt Quelle (source) und t Senke (target).

Gesucht ist eine Flussfunktion $x : E \rightarrow \mathbb{R}_0^+$ mit folgenden Eigenschaften:

- i) Kapazitätsbedingung:
 $\forall e \in E : 0 \leq x(e) \leq u(e)$



ii) Massenbalance-Bedingung:

Sei $v \in V$ (beliebig aber fest!)

$$\underbrace{\sum_{(v,u) \in E} x(v,u)}_{\substack{\text{gehe über alle} \\ \text{ausgehenden} \\ \text{Kanten von } v}} - \underbrace{\sum_{(w,v) \in E} x(w,v)}_{\substack{\text{gehe über alle} \\ \text{eingehenden} \\ \text{Kanten von } v}} = \begin{cases} F, & v = s \\ 0, & \forall v \in s, t \\ -F, & v = t \end{cases}$$

$$F \geq 0$$

ii) Optimalitätsbedingung:

F soll maximal sein.

Erklärungen:

- i) Für alle Kanten muss gelten, dass ihr Flusswert zwischen Null und ihrer maximalen Kapazität liegt.
- ii) Für alle Knoten die nicht s oder t sind, muss gelten, dass alles was in sie rein fließt auch wieder abfließt. Es kann also nichts in diesen Knoten gelagert werden. F entspricht dem Fluss der fließt. Aus dem Knoten s fließt der komplette Fluss F heraus, aber nichts herein. Beim Knoten t ist es genau umgekehrt.
- iii) Ohne diese Bedingung wäre das Problem trivial, da der Null-Fluss ($x = 0$) die anderen beiden Bedingungen immer erfüllt.

Notation:

Knoten: i, j

Die Kapazität wird anstatt mit $u((i, j))$ jetzt nur noch mit u_{ij} und die Flussfunktion anstatt mit $x((i, j))$ mit x_{ij} bezeichnet.

5.1.2 Das Restnetzwerk

(residual network)

Ein Restnetzwerk beschreibt mögliche Flussrichtungen.



Definition

Sei x eine aktuelle Flussfunktion, dh. sie erfüllt die Kapazitätsbedingung (z.B. der Nullfluss ($\forall i, j \ x_{ij} = 0$)).

Das Restnetzwerk $G(x)$ besteht aus allen Kanten von G und deren Gegenkante. Es existieren also für jede Kante (i, j) aus G zwei Kanten in $G(x)$, nämlich die Kante (i, j) und ihre Gegenkante (j, i) . Sie besitzen die Restkapazitäten $r_{ij} = u_{ij} - x_{ij}$ und $r_{ji} = x_{ij}$.

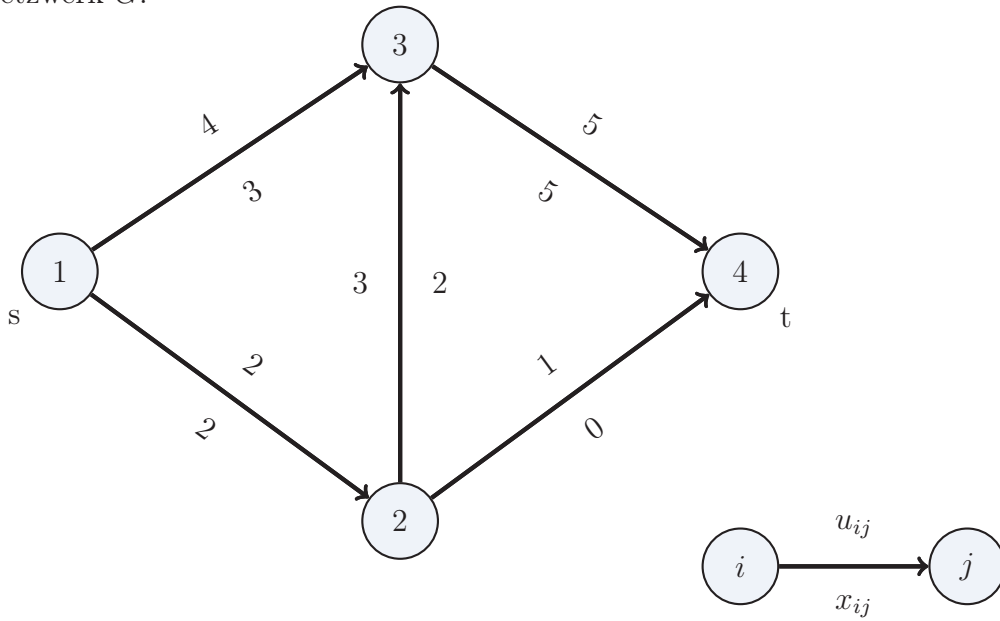


Die Restkapazitäten beschreiben mögliche Änderungen an der Flussfunktion. Eine Erhöhung geschieht in Richtung der Kante (i, j) ($\max r_{ij} = u_{ij} - x_{ij}$) und eine Verminderung in die Gegenrichtung ($\max r_{ji} = x_{ij}$).

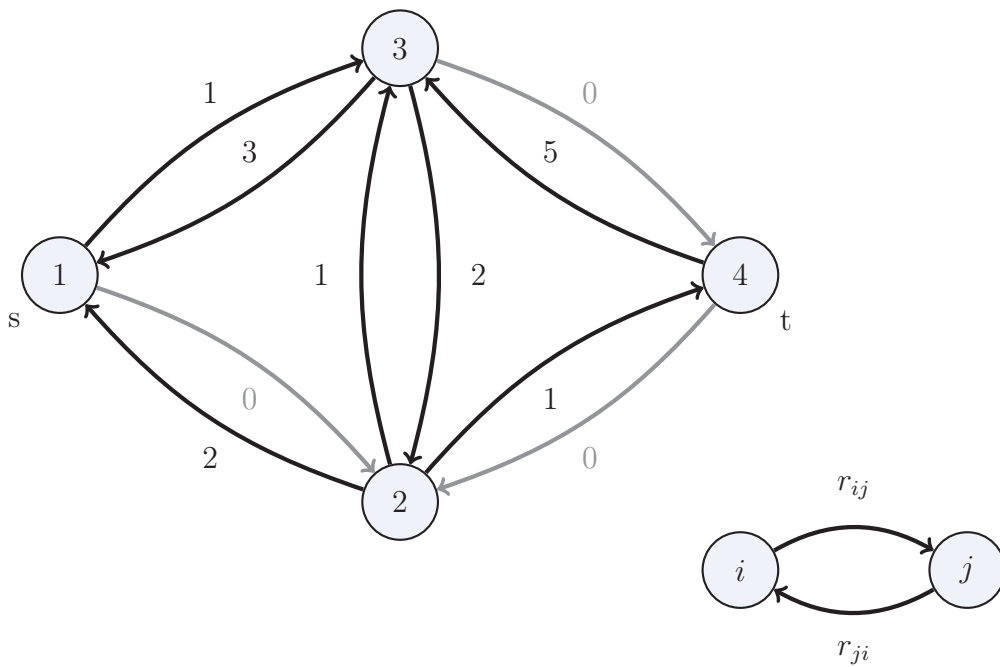


Beispiel 5.1

Netzwerk G :



Aktueller Flusswert: $F = 5$
Restnetzwerk $G(x)$:





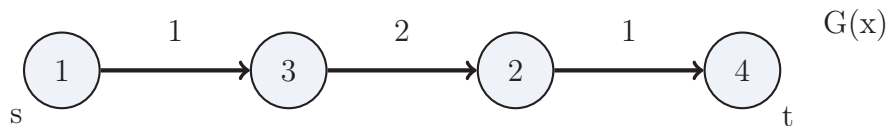
Da über Kanten mit einer Restkapazität von Null, nichts mehr fließen kann, werden sie weggelassen. Dadurch ist die folgende Beobachtung möglich.

Beobachtung: (ohne 0-Kanten)

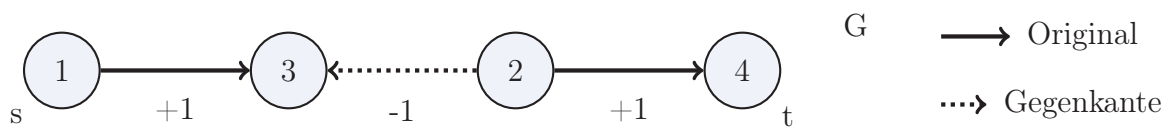
Wenn alle Null-Kanten im Graphen weggelassen werden, entspricht die Frage ob ein Pfad existiert der Frage ob ein erhöhender Pfad existiert.

Jeder Pfad in $G(x)$ von s nach t beschreibt eine mögliche Erhöhung des Gesamtflusses F . (\rightarrow Pfaderhöhung)

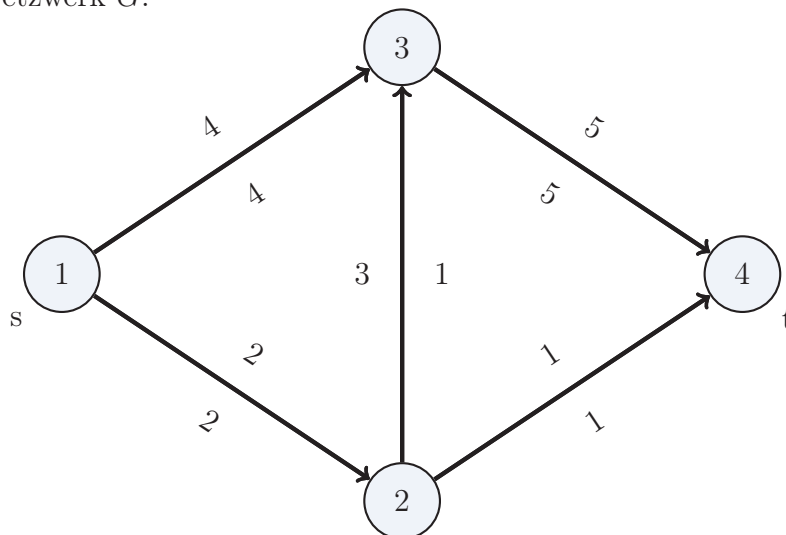
Im Beispiel 5.1 existiert nur ein solcher Pfad:



Eine Erhöhung um $\delta = \min\{v_{ij} | (i, j) \in P\} = 1$ ist möglich.



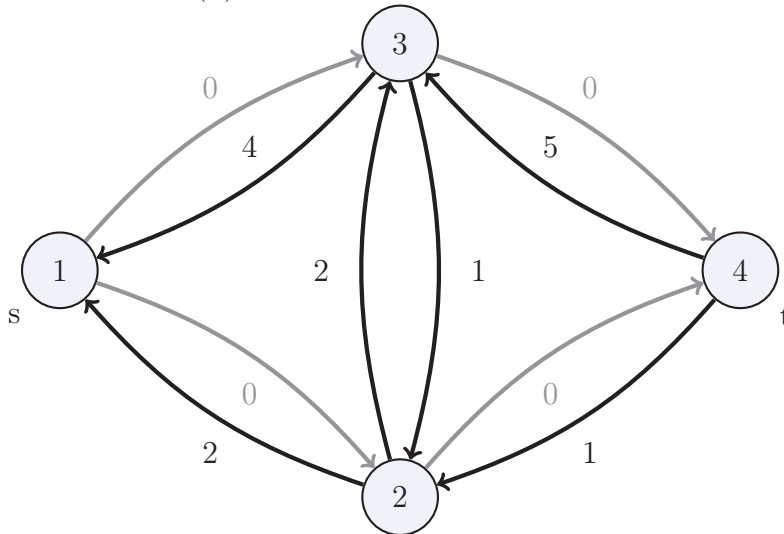
Netzwerk G :



Flusswert $F = 6$



Restnetzwerk $G(x)$:

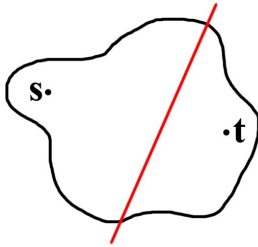


Hier existiert in $G(x)$ kein Pfad mehr von s nach t . Daher ist keine weitere Erhöhung mehr möglich.

Anmerkungen:

- Für den Nullfluss ($x = 0$) gilt $G(x) = G$.
 - Wird eine Kante (i, j) in G saturiert (dh. $x_{ij} = u_{ij}$), so wird die Kante (i, j) aus $G(x)$ entfernt.
 - Setzt man $x_{ij} = 0$, so wird die Gegenkante (j, i) aus $G(x)$ entfernt.
- ⇒ Änderungen an der Flussfunktion führen zu Änderungen in $G(x)$.

5.2 Schnitte



Betrachte einen Schnitt der s und t trennt. Alles was von s nach t fließt, fließt auch über den Schnitt.

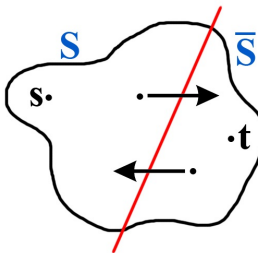
⇒ Kapazität des Schnittes ist obere Schranke für den maximalen Fluss.

Das gilt für alle Schnitte und damit auch für den minimalen Schnitt (*MinCut*).

5.2.1 Definition des (s,t) -Schnittes

Ein (s,t) -Schnitt $[S, \bar{S}]$ ist eine Partitionierung der Knoten V in zwei disjunkte Teilmengen S und \bar{S} ($= V \setminus S$) mit $s \in S$ und $t \in \bar{S}$.

Notation:



$(i, j) \in (S, \bar{S})$ oder $(i, j) \in (\bar{S}, S)$.

Der Fluss von s nach t muss über die Kanten

(S, \bar{S}) ($= E \cap S \times \bar{S}$)

Vorwärtskanten

(\bar{S}, S) sind die Rückwärtskanten.

Kapazität des (s,t) -Schnittes

Die Kapazität des (s,t) -Schnittes ist die Summe der Kapazitäten der Kanten, deren source-Knoten in S und deren target-Knoten in \bar{S} liegen.

$$u[S, \bar{S}] = \sum_{(i,j) \in (S, \bar{S})} u_{ij}$$

Restkapazität des (s,t) -Schnittes

Die Restkapazität des (s,t) -Schnittes ist die Summe der Restkapazitäten der Kanten, deren source-Knoten in S und deren target-Knoten in \bar{S} liegen.

$$r[S, \bar{S}] = \sum_{(i,j) \in (S, \bar{S})} r_{ij}$$

Fluss über den Schnitt

Der Wert des Flusses über den Schnitt ist die Differenz zwischen der Summe der Flusswerte der Kanten, die von S nach \bar{S} laufen und der, die von \bar{S} nach S laufen. Da der Fluss von \bar{S} nach S zurück fließender Fluss ist, muss er abgezogen werden.

$$F = \underbrace{\sum_{(i,j) \in (S, \bar{S})} x_{ij}}_{\text{maximal}} - \underbrace{\sum_{(i,j) \in (\bar{S}, S)} x_{ij}}_{\geq 0}$$

$$\Rightarrow F \leq u[S, \bar{S}]$$

Dies gilt für alle (s, t) -Schnitte. Insbesondere für den mit minimaler Kapazität.

$u[S, \bar{S}] \Rightarrow$ Der Wert eines maximalen Flusses ist nicht größer als die Kapazität eines minimalen Schnittes.

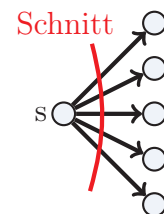
$F_{max} \leq MinCut$ (Schwache Variante des MaxFlow/MinCut-Theorem)

Obere Schranke der Kapazität des (s,t)-Schnittes

Sei \mathcal{U} = maximale Kapazität aller Kanten ($\max\{u_{ij} | (i, j) \in E\}$)

Kapazität eines Schnittes $\leq (n - 1) \cdot \mathcal{U}$

$$\Rightarrow F_{max} \leq n \cdot \mathcal{U}$$



5.3 Algorithmen auf $G(x)$

Es gibt zwei Möglichkeiten auf $G(x)$ zu arbeiten. Explizit darauf zu arbeiten, bedeutet, dass intern ein zweiter Graph aufgebaut wird. Dieser Graph besitzt sowohl die Originalkanten, als auch die Gegenkanten, solange es keine Null-Kanten sind.



Bei diesen Graphen kann z.B. DFS oder BFS zur Pfadsuche eingesetzt werden. Asymptotisch schneller ist es, wenn man implizit arbeitet, also auf dem Originalgraphen. Dabei müssen bei einer Pfaderhöhung Kanten eingefügt bzw. gelöscht werden. Im Gegensatz zum expliziten Restnetzwerk müssen hier nicht nur die ausgehenden, sondern auch die eingehenden Kanten betrachtet werden.

cap: Kapazitäten (u_{ij})

flow: Aktueller Flusswert (x_{ij})

```
1 forall_out_edges(e,v){
2     r = cap[e] - flow[e];
3     if(r == 0)
4         continue; // Ausfiltern aller Kanten, die nicht in
                    //G(x) sind
5     //Rumpf
6 }
7 forall_in_edges(e,v){
8     r = flow[e];
9     if(r == 0)
10        continue;
11    //Rumpf
12 }
```

Listing 5.1: Ausfiltern der Kanten



5.4 Erhöhende-Pfad-Algorithmen

Der Labeling-Algorithmus ist ein Erhöhender-Pfad-Algorithmus (augmenting path).

5.4.1 Der allgemeine Labeling-Algorithmus

Algorithmus 3 : Allgemeiner Labeling-Algorithmus

```
1  $x = 0$  // Nullfunktion (alle  $x_{ij} = 0$ )
2 while  $G(x)$  enthält einen Pfad von  $s$  nach  $t$  do
3   Sei  $P$  ein solcher (erhöhender) Pfad
4    $\delta = \min\{r_{ij} \mid (i, j) \in P\}$ 
5   Erhöhe den Fluss  $x$  entlang von  $P$  um  $\delta$  Einheiten
6   Berechne  $G(x)$  neu
7 end
```

5.4.2 Der Labeling-Algorithmus

(Konkrete Implementation des allgemeinen Algorithmus)

Idee:

(Ähnlich wie bei DFS, BFS, explore from ...)

Finde die Menge S aller Knoten, die von s aus in $G(x)$ erreichbar sind. Diese werden markiert. Wenn t markiert wurde, dann gilt $t \in S$, dh. es existiert ein erhöhender Pfad \rightarrow Pfaderhöhung

Darstellung der Pfade:

Pred-Array (siehe kürzeste Wege)

$pred[v]$ ist die Kante über die v zum ersten Mal erreicht wurde.