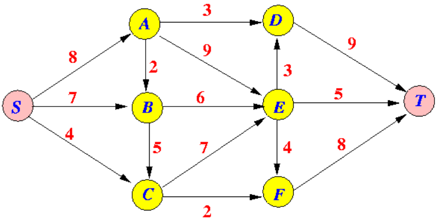
Kapitel 6

Flüsse und Zuordnungen



Inhalt

Flussprobleme

- In diesem Kapitel werden Bewertungen von Kanten als Transportkapazitäten pro Zeiteinheit interpretiert.
- Wir können uns einen Graphen als Versorgungsnetzwerk vorstellen, z.B. als Datennetz.
- Fragen:
  - Welchen Durchsatz können wir erreichen?
  - Wie viele Einheiten können wir von einem Knoten zu einem anderen pro Zeiteinheit transportieren?
  - Welche Kanten bilden dabei einen Engpass?

Flussnetzwerk

Definition 6.1

Ein Flussnetzwerk N ist ein Tupel $N = (G, c, s, t)$ bestehend aus:

- $G = (V, A)$, einem gerichteten Graphen,
- $c : A \longrightarrow \mathbb{R}_+$, einer Kapazitätsfunktion auf den gerichteten Kanten mit nichtnegativen Werten und
- $s, t \in V$, zwei ausgezeichneten Knoten, der Quelle s und der Senke t mit $s \neq t$.

Fluss

Definition 6.2

Es sei $N = (G, c, s, t)$ ein Flussnetzwerk. Für einen Knoten $v \in V$ sei $A_{in}(v) := \{(u, v) \in A\}$ und $A_{out}(v) := \{(v, u) \in A\}$.

Eine Abbildung $f : A \longrightarrow \mathbb{R}$ heißt Fluss auf N, wenn die folgenden Bedingungen erfüllt sind:

1. $0 \leq f(e) \leq c(e)$ für alle $e \in A$,

   d. h. die Kapazität wird für keine Kante überschritten und

2. $\displaystyle\sum_{e \in A_{in}(v)} f(e) = \sum_{e \in A_{out}(v)} f(e)$ für alle $v \in V \setminus \{s, t\}$,

   d. h. aus jedem Knoten fließt genausoviel heraus wie hinein, mit Ausnahme der Quelle s und der Senke t.

Flusswert

Lemma 6.3

Für einen Fluss f eines Flussnetzwerks $N = (G, c, s, t)$ gilt

$$\Phi(f) := \sum_{e \in A_{out}(s)} f(e) - \sum_{e \in A_{in}(s)} f(e) = \sum_{e \in A_{in}(t)} f(e) - \sum_{e \in A_{out}(t)} f(e)$$

Definition 6.4

Der Wert $\Phi(f)$ aus Lemma 6.3 heißt Wert des Flusses f auf N.

Ein Fluss f mit $\Phi(f) \geq \Phi(f')$ für alle Flüsse f' auf N heißt Maximalfluss auf N.

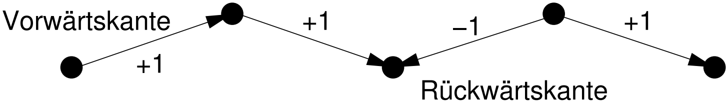
Das Maximalflussproblem besteht darin, zu einem gegebenen Flussnetzwerk N einen Maximalfluss zu bestimmen.

Zunehmender Weg

Definition 6.5

Gegeben sei ein Flussnetzwerk $N = (G, c, s, t)$ mit einem Fluss f. Eine Folge $(v_0, \ldots, v_k)$ heißt zunehmender Weg bzgl. f gdw. für jedes $i = 1, \ldots, k$ eine der folgenden Bedingungen erfüllt ist:

1. $(v_{i-1}, v_i) \in A$ und $f(v_{i-1}, v_i) < c(v_{i-1}, v_i)$ (Vorwärtskante)
2. $(v_i, v_{i-1}) \in A$ und $f(v_i, v_{i-1}) > 0$ (Rückwärtskante)



Kriterium für Maximalfluss

- Offensichtlich können wir den Fluss erhöhen, wenn wir einen zunehmenden Weg gefunden haben.
- Die Existenz eines zunehmenden Weges ist also hinreichend für eine Flusserhöhung.
- Der folgende Satz zeigt, dass dieses Kriterium auch notwendig ist.

Satz 6.6

Ein Fluss f in einem Flussnetzwerk N ist genau dann ein Maximalfluss, wenn kein zunehmender Weg von s nach t existiert.

Beweis.

"$\Rightarrow$": Wenn ein zunehmender Weg W von s nach t existiert, dann kann $\Phi(f)$ um das Minimum der Werte $c(e) - f(e)$ für Vorwärtskanten von W bzw. $f(e)$ für Rückwärtskanten von W erhöht werden.

"$\Leftarrow$": Es gebe keinen zunehmenden Weg von s nach t.

- Es sei S die Menge der Knoten, die von s aus mit einem zunehmenden Weg erreichbar sind, und sei $T := V \setminus S$.
- Für jede Kante $(v, w), v \in S, w \in T$ gilt: $f(v, w) = c(v, w)$
- Für jede Kante $(w, v), w \in T, v \in S$ gilt: $f(w, v) = 0$
- Anschaulich: Die Kanten zwischen S und T bilden einen Engpass, der eine Flusserhöhung verhindert.

Berechnung eines Maximalflusses

Satz 6.6 liefert die Basis zur Berechnung eines Maximalflusses.

1. Wir starten mit einem beliebigen Fluss, z.B. $f(e) = 0$ für alle $e \in A$.
2. Wenn es keinen zunehmenden Weg bzgl. f gibt, dann STOP.
3. Sei $W = (s = v_0, v_1, \ldots, v_k = t)$ ein zunehmender Weg von s nach t bzgl. f und sei

$$\begin{aligned} z \quad := \quad & \min(\{c(v_{i-1}, v_i) - f(v_{i-1}, v_i) | (v_{i-1}, v_i) \text{ Vorwärtskante von } W\} \\ & \cup \{f(v_i, v_{i-1}) | (v_i, v_{i-1}) \text{ Rückwärtskante von } W\}). \end{aligned}$$

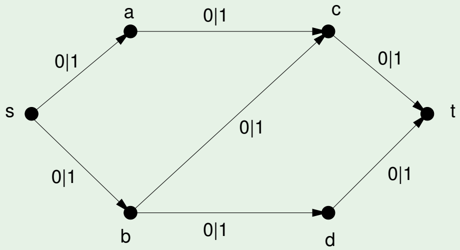
   Setze $f(v_{i-1}, v_i) := f(v_{i-1}, v_i) + z$ für jede Vorwärtskante (v_{i-1}, v_i).

   Setze $f(v_i, v_{i-1}) := f(v_i, v_{i-1}) - z$ für jede Rückwärtskante (v_i, v_{i-1}).

   Weiter mit 2.

Beispiel 6.7

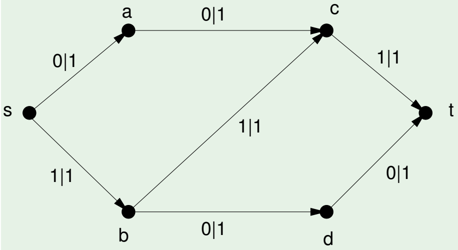
Wir betrachten das folgende Flussnetzwerk. Die Kapazität ist für alle Kanten 1. Der Fluss ist zunächst auf allen Kanten 0.



(s, b, c, t) ist ein zunehmender Weg mit $z = 1$. Alle Kanten des Weges sind Vorwärtskanten.

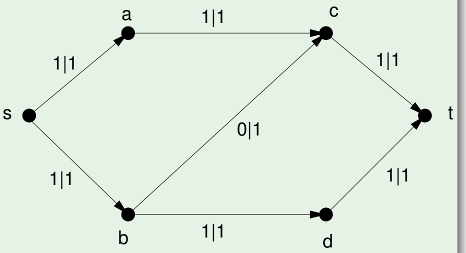
Fortsetzung Beispiel.

Flusserhöhung auf dem zunehmenden Weg ergibt den Graphen:



mit $\Phi(f) = 1$.
(s, a, c, b, d, t) ist nun ein zunehmender Weg mit $z = 1$, wobei (b, c) eine Rückwärtskante ist.

Auf der Rückwärtskante wird der Fluss verringert, ansonsten erhöht.



Wir haben $\Phi(f) = 2$ und es existiert kein zunehmender Weg. Damit ist der angegebene Fluss ein Maximalfluss.

Markierungsalgorithmus

- Der Markierungsalgorithmus von Ford und Fulkerson (1956) konkretisiert das Verfahren zur Berechnung maximaler Flüsse.
- Man markiert sukzessive die Knoten w auf einem zunehmenden Weg mit drei Werten $v(w), r(w), z(w)$.
-
  - $v(w)$ ist der Vorgänger von w in dem zunehmenden Weg.
  - $r(w)$ gibt die Richtung der verwendeten Kante an
    ($\rightarrow=$ Vorwärtskante, $\leftarrow=$ Rückwärtskante).
  - $z(w)$ ist der mögliche zusätzliche Fluss auf dem Weg nach w.

Algorithmus 6.8

Gegeben sei ein Flussnetzwerk $N = (G, c, s, t)$ *und ein initialer Fluss* $f(e) \equiv 0$.

1. *Setze* $S := \{s\}, R := \{s\}, z(s) := \infty$.
2. *Wähle einen Knoten* $u \in R$. *Setze* $R := R \setminus \{u\}$.
3. *Für alle* $w \in V \setminus S$ *mit* $(u, w) \in A$ *und* $f(u, w) < c(u, w)$:

$$S := S \cup \{w\}, R := R \cup \{w\},$$
$$v(w) := u, r(w) := \rightarrow, z(w) := \min\{z(u), c(u, w) - f(u, w)\}$$

4. *Für alle* $w \in V \setminus S$ *mit* $(w, u) \in A$ *und* $f(w, u) > 0$:

$$S := S \cup \{w\}, R := R \cup \{w\},$$
$$v(w) := u, r(w) := \leftarrow, z(w) := \min\{z(u), f(w, u)\}$$

Fortsetzung Algorithmus.

5. *Falls* $R = \emptyset$*, dann STOP. Falls* $t \in S$*, dann weiter mit 6, ansonsten weiter mit 2.*
6. $z := z(t); w := t.$
7. *Falls* $r(w) = \rightarrow$: $u := v(w), f(u, w) := f(u, w) + z$
   Falls $r(w) = \leftarrow$: $u := v(w), f(w, u) := f(w, u) - z$
8. $w := v(w)$*. Falls* $w = s$*, dann weiter mit 1, ansonsten weiter mit 7.*

Satz 6.9

Sei $N = (G, c, s, t)$ *ein Flussnetzwerk mit rationaler Kapazitätsfunktion* c*. Dann berechnet Algorithmus 6.8 einen maximalen Fluss* f *auf* N*.*

- Bei irrationalen Kapazitäten kann es vorkommen, dass der Markierungsalgorithmus immer weitere Verbesserungen des Flusswertes findet, ohne jemals zu terminieren.
- Auch bei ganzzahligen Kapazitäten ist die Laufzeit des Markierungsalgorithmus nicht polynomial, da die Anzahl der Schritte von c abhängen kann.
- Eine polynomiale Laufzeit erhält man aber, wenn man für die Suche nach einem zunehmenden Weg die Breitensuche einsetzt (Edmonds und Karp, 1972).

Algorithmus von Edmonds und Karp

Satz 6.10

Ersetzt man in Algorithmus 6.8 den Schritt 2 durch

2a. *Wähle den Knoten $u \in R$, der zuerst in R eingefügt wurde. Setze $R := R \setminus \{u\}$.*

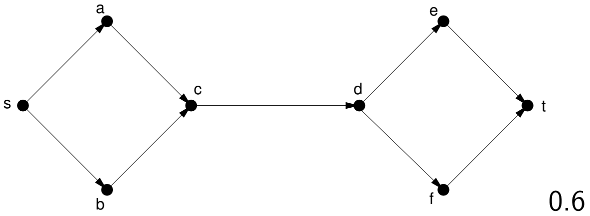
dann berechnet der Markierungsalgorithmus für beliebige Kapazitätsfunktionen in Zeit $O(|V||E|^2)$ einen Maximalfluss.

Beispiel 6.11

Anwendung des Markierungsalgorithmus. Tafel ✎

Trennender Schnitt

Wie groß kann der Fluss in dem folgenden Flussnetzwerk höchstens sein?



Der Fluss kann nicht größer als die Kapazität der der Kante (c, d) sein, da jeder Weg von s nach t diese Kante enthält.

Die Kante (c, d) ist ein sogenannter trennender Schnitt.

Minimaler Schnitt

Definition 6.12

Es sei $N = (G, c, s, t)$ ein Flussnetzwerk mit $G = (V, A)$.

Für eine Teilmenge $S \subseteq V$ heißt $A_S := \{(v, w) \in A | v \in S, w \in V \setminus S\}$ Schnitt von G.

Falls $s \in S, t \in V \setminus S$, so ist A_S ein trennender Schnitt.

Ein trennender Schnitt A_S mit minimaler Kapazität

$$c(A_S) := \sum_{e \in A_S} c(e)$$

heißt minimaler Schnitt.

Max-Flow-Min-Cut-Theorem

Satz 6.13

In einem Flussnetzwerk $N = (G, c, s, t)$ ist der Wert eines maximalen Flusses gleich der Kapazität eines minimalen Schnittes.

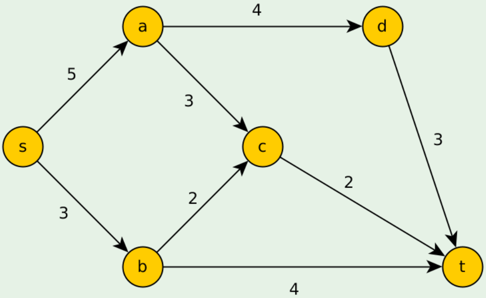
Beweis.

Tafel ✎.

- Kennen wir einen Fluss f und finden wir einen trennenden Schnitt A_S mit $\Phi(f) = c(A_S)$, so ist f ein Maximalfluss und A_S ein minimaler Schnitt.
- Für die Menge S bei Terminierung von Algorithmus 6.8 ist A_S ein minimaler Schnitt (vgl. Beweis zu Satz 6.6).
- Der Markierungsalgorithmus berechnet also nicht nur einen maximalen Fluss sondern auch einen minimalen Schnitt.

Beispiel 6.14

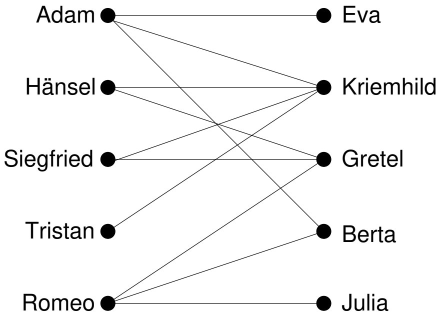
Wie lautet ein Maximalfluss für den folgenden Graphen?



$f(s,b) = f(b,t) = 3,$
$f(s,a) = 5, f(a,c) = f(c,t) = 2,$
$f(a,d) = f(d,t) = 3$ also $\Phi(f) = 8.$
Die Kapazität des trennenden Schnittes $\{(s,b),(c,t),(d,t)\}$ ist 8.
Also ist f ein Maximalfluss.

Heiratsproblem

Das Heiratsproblem lautet: In einer Gruppe von Männer und Frauen kennnen sich einige Männer und Frauen. Ist es möglich, daß jeder mit einer seiner Bekannten verheiratet wird?



Matching

Definition 6.15

Es sei $G = (V, E)$ ein bipartiter Graph.

Ein Menge $M \subseteq E$ von Kanten heißt Zuordnung (Matching) gdw.

$$\text{für alle } e_1, e_2 \in M \text{ mit } e_1 \neq e_2 \text{ gilt} : e_1 \cap e_2 = \emptyset,$$

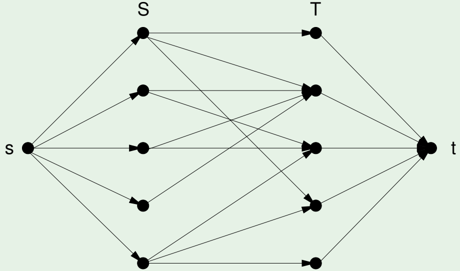
d. h. die Kanten haben keinen gemeinsamen Knoten.

Heiratsproblem als Flussproblem modellieren

- Das Heiratsproblem ist genau dann lösbar, wenn ein Matching maximaler Mächtigkeit existiert.
- Das wesentliche Problem besteht also darin, zu einem bipartiten Graphen ein Matching mit maximaler Mächtigkeit zu berechnen. Hierzu können wir das Problem als Flussproblem modellieren.
- Alle Kanten des Orginalgraphen werden von S nach T gerichtet.
- Neuer Knoten s und neue Kanten (s, v) für alle $v \in S$
- Neuer Knoten t und neue Kanten (v, t) für alle $v \in T$
- Alle Kanten erhalten die Kapazität 1.

Beispiel 6.16

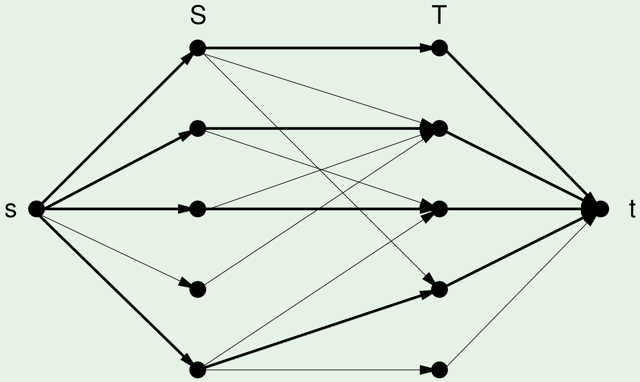
Transformiert man den bipartiten Graphen des gegebenen Heiratsproblem nach obigen Regeln ergibt sich:



- Es sei f ein Maximalfluss, der mit dem Markierungsalgorithmus berechnet wurde. Wegen der Ganzzahligkeit des Problems ist dann auch f ganzzahlig.
- In einen Knoten $v \in T$ kann dann nicht mehr als 1 hineinfließen.
- Aus einem Knoten $v \in S$ kann nicht mehr als 1 herausfließen.
- Daraus folgt, dass die Kanten zwischen S und T mit Fluss 1 ein Matching bilden.
- Die Anzahl dieser Kanten ist gleich dem Flusswert $\Phi(f)$.
- Damit stellen diese Kanten ein Matching maximaler Mächtigkeit dar.

Beispiel 6.17

Für unser spezielles Heiratsproblem erhalten wir als eine Lösung:



Es ist also nicht möglich, alle zu verheiraten. Tristan und Julia bleiben bei dieser Lösung unverheiratet.

Das gewichtete Zuordnungsproblem

Das gewichtete Zuordnungsproblem lautet:

- Gegeben sei ein vollständiger bipartiter Graph, d.h. für alle $s \in S$ und $t \in T$ gibt es eine Kante, die s und t verbindet.
- Auf den Kanten ist eine Gewichtsfunktion definiert.
- Man finde ein Matching maximaler Mächtigkeit mit minimalem Gesamtgewicht.
- Beispiel: Man verbindet im Heiratsproblem alle Männer mit allen Frauen und drückt den Grad der Zuneigung durch ein Antipathiegewicht aus.
- Es sind dann alle zu verheiraten und zwar so, dass die Gesamtantipathie möglichst klein und somit die Gesamtzuneigung möglichst groß ist.
- Zur Lösung von gewichteten Zuordnungsproblemen wendet man ähnliche Verfahren wie bei der Berechnung von Maximalflüssen und wie beim ungewichteten Zuordnungsproblem an.
- Ein sehr bekanntes Verfahren ist der ungarische Algorithmus.

Zusammenfassung

- Flussnetzwerk und Maximalfluss
- Zunehmender Weg als Grundlage zur Berechnung von maximalen Flüssen
- Berechnung maximaler Flüsse mit dem Markierungsalgorithmus
- max flow = min cut
- Anwendung des Maximalflussproblems zur Berechnung maximaler Matchings