Einführung in die lineare und kombinatorische Optimierung
Serie 6

Sven-Maurice Althoff (FU 4745454)
Michael R. Jung (HU 502133)
Felix Völker (TU 331834)

28. November 2014

Aufgabe 20

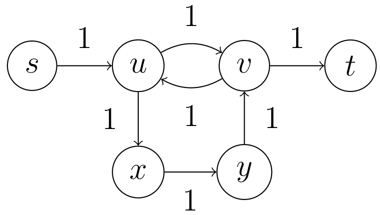
Zeigen wir zunächst "$\Leftarrow$":
Zu prüfen ist, ob der so definierte Fluss x zulässig ist.
Per Voraussetzung gilt: $\forall uv \in A : 0 \leq x_{uv} \leq c_{uv}$.✓
Betrachte beliebigen Knoten $u \in V \setminus \{s,t\}$:
Dieser liege auf den Wegen $P_{i_1}, \ldots, P_{i_p}$ und den Kreisen $C_{j_1}, \ldots C_{j_q}$. Seien $x_{i_1}, \ldots, P_{i_x}$ seine Vorgänger und $y_{i_1}, \ldots, y_{i_p}$ seine Nachfolger auf den Pfaden sowie $v_{j_1}, \ldots v_{j_q}$ seine Vorgänger und $w_{j_1}, \ldots w_{j_q}$ seine Nachfolger auf den Kreisen.
Dann ist

$$\begin{aligned}\sum_{z \in \delta^-(u)} x_{zu} &= \sum_{i \in \{i_1, \ldots, i_p\}} x_{x_i u} + \sum_{j \in \{j_1, \ldots, j_q\}} x_{v_j u} \\ &= \sum_{i \in \{i_1, \ldots, i_p\}} \lambda_i + \sum_{j \in \{j_1, \ldots, j_q\}} \mu_j \\ &= \sum_{i \in \{i_1, \ldots, i_p\}} x_{u y_i} + \sum_{j \in \{j_1, \ldots, j_q\}} x_{u w_j} \\ &= \sum_{z \in \delta^+(u)} x_{uz}\end{aligned}$$

Somit ist der Fluss zulässig.

Aufgabe 21

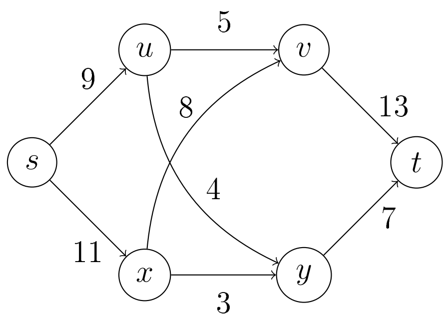
a) Diese Aussage ist falsch. Betrachte folgendes Gegenbeispiel:



Hier können wir alle Bögen saturieren (d.h. $x_a = c_a \, \forall a \in A$) und erhalten einen maximalen Fluss. Insbesondere ist aber $x_{uv} = x_{vu} = 1$.

b) Diese Aussage ist wahr. Seien ein Netzwerk $((V, A), c, s, t)$ und ein maximaler Fluss $x_a, a \in A$ gegeben. Seien $u, v \in V$ mit $x_{uv} \neq 0 \neq x_{vu}$. O.B.d.A. sei $x_{uv} \geq x_{vu}$ (sonst vertausche u und v). Dann ist auch $x'_a, a \in A$ ein maximaler Fluss mit $x'_a = x_a \, \forall a \in A \setminus \{(u, v), (v, u)\}$ und $x'_{uv} = x_{uv} - x_{vu}, x'_{v,u} = 0$.
Man sieht, dass für jeden Knoten v gilt: $\sum\limits_{a \in \delta^+(v)} x'_a - \sum\limits_{a \in \delta^-(v)} x'_a = \sum\limits_{a \in \delta^+(v)} x_a - \sum\limits_{a \in \delta^-(v)} x_a$.
Dies gilt, da sich für alle Knoten außer u und v nichts ändert und für diese beiden sich beide Summen um den gleichen Betrag ändern. Außerdem bleibt der Fluss auch maximal, da sich die Bilanz auch für s nicht geändert hat. Dieses Verfahren kann man sooft wiederholen, bis die geforderte Bedingung für alle Knotenpaare erfüllt ist, da zwei solche Veränderungen sich gegenseitig nicht beeinflussen.

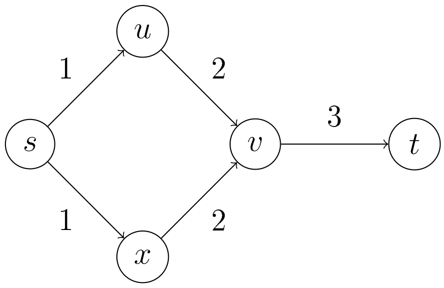
c) Diese Aussage ist falsch. Betrachte folgendes Gegenbeispiel:



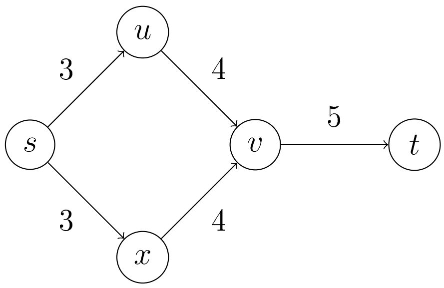
Hier sind alle Kapazitäten paarweise verschieden, aber sowohl $B := \delta^+(\{s\})$ als auch $C := \delta^+(\{s, u, x\})$ minimale (s, t)-Schnitte mit $c(B) = c(C) = 20$. Minimal sind sie, denn es existiert ein Fluss $f_a, a \in A$ mit $f_a = c_a \, \forall a \in A$ und Wert 20. Im Übrigen ist hier sogar jeder (s, t)-Schnitt minimal.

d) Diese Aussage ist wahr. Sei $B \subseteq$ ein beliebiger (s,t)-Schnitt und $c'_a = \lambda c_a \, \forall a \in A$. Dann ist nach der Veränderung $c'(B) = \sum\limits_{a \in B} c'_a = \sum\limits_{a \in B} \lambda c_a = \lambda \sum\limits_{a \in B} c_a$. Nun ist klar, dass ein minimaler Schnitt auch minimal bleibt, da für beliebige (s,t)-Schnitte S_1, S_2 mit $c(S_1) \leq c(S_2)$ gilt: $c'(S_1) \leq c'(S_1) \iff \lambda c(S_1) \leq \lambda c(S_2) \overset{\lambda > 0}{\iff} c(S_1) \leq c(S_2)$.

e) Diese Aussage ist falsch. Betrachte folgendes Gegenbeispiel:



Hier wäre $B := \delta^+(s)$ ein minimaler (s,t)-Schnitt mit $c(B) = 2$. Erhöhen wir aber jede Kapazität um 2, so erhalten wir folgendes Netzwerk:



Hier ist nun $\delta^+(s)$ kein minimaler (s,t)-Schnitt mehr, denn $c(\delta^+(s)) = 6$, aber $c(\delta^+(\{s,u,x,v\})) = 5$.
Das Problem bei dieser Veränderung ist, dass ein (s,t)-Schnitt in der Anzahl der beteiligten Kanten skaliert wird.

Aufgabe 23

Im folgenden nennen wir ein kardinalitätsmaximales Matching ein *größtes* Matching, und einen maximalen, zulässigen (s,t)-Fluss einen *größten* Fluss.
Zunächst zeigen wir, dass der Wert eines größten Matchings dem eines größten Flusses entspricht.
Sei also ein größtes Matching $M \subseteq E$ gegeben. Betrachte nun die gerichtete Variante $M' = \{(u,v) \in V_1 \times V_2 | \{u,v\} \in M\}$. Dann ist

$$x_a = \begin{cases} 1 & a \in \{(s,u) | \exists v \in V_2 : uv \in M'\} \cup M' \cup \{(v,t) | \exists u \in V_1 : uv \in M'\} \\ 0 & \text{sonst} \end{cases}$$

offensichtlich ein zulässiger Fluss, dessen Wert dem von M entspricht.
Andererseits ist aber auch $S := \delta^+(\{s\} \cup V_1 \cup \{v \in V_2 | \exists u \in V_1 : uv \in M'\}$ ein (s,t)-Schnitt, dessen Kapazität dem Wert von M entspricht, denn wenn $c(S)$ größer wäre, so könnten wir ein größeres Matching finden, indem wir alle Kanten aus S in ihrer ungerichteten Form zu M hinzunähmen.
Hier wird schon deutlich, wie wir aus größten Flüssen x in D, deren Werte alle aus $\{0,1\}$ kommen (im folgenden 0-1-Flüsse genannt), in größte Matchings überführen.
Einen beliebigen größten Fluss wandeln wir folgendermaßen in einen größten 0-1-Fluss um:

```
for each su ∈ A
  x_su=0;
for each vt ∈ A
  x_vt=0;
for each uv ∈ A ∩ V1 × V2
if 0<x_uv
  begin
   x_uv=1;
   x_su=1;
   x_vt=1;
   for each uv' ∈ A
      x_uv'=0;
   for each u'v ∈ A
      x_u'v=0;
  end
```

Dieser Algorithmus verändert den Wert des ursprünglichen Flusses nicht, da für jeden solchen Bogen uv entweder su oder vt (oder beide) bereits saturiert war, sonst könnte man ja den Fluss vergrößern. Bauen wir uns also einen Schnitt S auf, der su enthält falls dieser Bogen saturiert war oder im anderen Falle vt, so erhalten wir einen minimalen Schnitt, der sowohl durch den ursprünglichen Fluss als auch durch den veränderten saturiert wird.

Aufgabe 22