RWTH Aachen **WS 2016**
Lehrgebiet Theoretische Informatik Lösungsvorschlag 3
Kuinke–Rossmanith–Sanchez 11.11.2016

Übung zur Vorlesung Effiziente Algorithmen

Tutoraufgabe 7

Wir definieren eine neue Kostenfunktion $c' \colon E \to (\mathbf{R}_0^+ \times \mathbf{N})$ mit $c'(e) = (c(e), 1)$ und erweitern dies auf die natürliche Weise für Pfade indem wir komponentenweise addieren.

Desweiteren setzen $(a, b) <_{c'} (a', b')$ falls $a < a'$ oder $a = a'$ und $b < b'$.

Mit diesem Maß können wir nun leicht den Abstand bezüglich c' zweier Knoten s, t definieren, als die Kosten des Pfades P von s nach t mit gerinstem $c'(P)$.

Offensichtlich ist eine Lösung der Aufgabe genau ein Pfad P von s nach t mit geringstem $c'(P)$. Er kann nun leicht durch Anwenden von Dijstra's Algorithmus mit Kostenfunktion c' und Vergleichsoperation $<_{c'}$ gefunden werden.

Tutoraufgabe 8

Wir zeigen mittels einer Reduktion von Vertex Cover, daß dieses Problem NP-schwer ist.

Vertex Cover $\leq$ Flat-Rate

Sei $(G = (V, E), k)$ eine Eingabe für Vertex Cover. Wir konstruieren eine Eingabe $G', a = |E|, b = k$ für Flat-Rate wie folgt:

Als Knotenmenge V' wählen wir $\{s, t\} \cup V \cup E$. Dann fügen wir folgende Kanten ein

- (s, v) mit Kapazität $|V|$ und Kosten 1 für alle $v \in V$
- (v, e) mit $v \in V$, $e \in E$, e inzident zu v mit Kapazität 1 und Kosten 0
- (e, t) mit Kapazität 1 und Kosten 0 für alle $e \in E$.

Wir zeigen nun daß G genau dann ein Vertex Cover der Größe k hat, wenn es in G' einen Fluß der Größe $|E|$ mit Kosten höchstens k gibt.

Sei $V' \subseteq V$ ein Vertex Cover. Sei $E_v \subseteq E$ die Menge der Kanten in E, die durch v gecovert werden. Falls eine Kante durch zwei Knoten abgedeckt wird, wird sie nur zu einem E_v hinzugefügt.

Dann konstruieren wir den Fluß f in G' indem von s zu jedem $v \in V'$ genau $|E_v|$ fließen lassen. Weiter von jedem v zu allen Knoten in E_v jeweils 1 mit Kosten 0 und von jedem e zu t ebenfalls 1 mit Kosten 0. Offensichtlich ist dies eine gültige Lösung der Größe $|E|$ mit Kosten k.

Sei nun f ein Fluß der Größe $|E|$ mit Kosten höchstens k in G'. Offensichtlich fließt in jeden Knoten $e \in E$ genau 1, da sonst kein Fluß der Größe $|E|$ möglich wäre. Sei nun V' die Menge aller Knoten, über die ein Fluß zu Knoten in E fließt. Offensichtlich ist jeder Knoten in E adjazent zu einem Knoten in V', also ist V' ein Vertex Cover. Da die Kosten des Flusses höchstens k betragen, folgt außerdem $|V'| \leq k$.

Tutoraufgabe 9

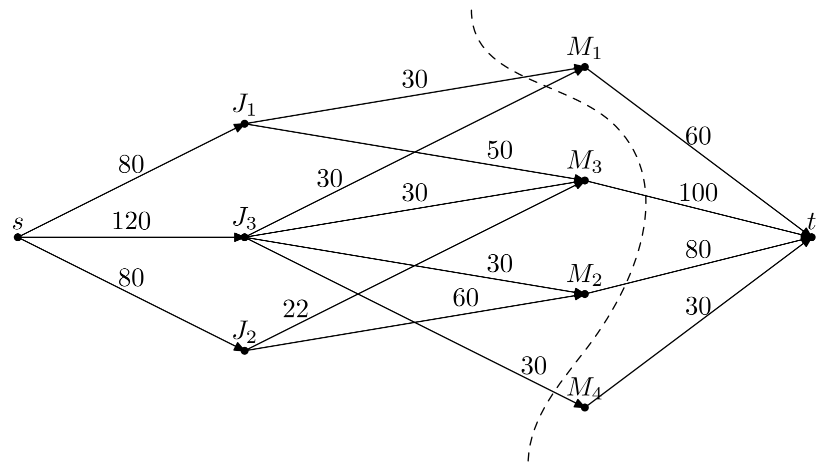
Die Knotenmenge des Flußnetzwerks ist $\{s,t\} \cup \{J_1,\ldots,J_n\} \cup \{M_1,\ldots,M_m\}$. Die Kanten des Netzwerks sind wie folgt:

1. Von der Quelle s läuft zu jedem Knoten J_i eine Kante mit Kapazität j_i, wenn j_i die für einen angenommenen Auftrag J_i erhaltene Zahlung ist.
2. Von einem Auftrag J_i läuft eine Kante zur Maschine M_j, wenn M_j für den Auftrag J_i benötigt wird. Die Kapazität auf dieser Kante sind die Kosten, die für eine einmalige Anmietung der Maschine M_j für den Auftrag J_i entstehen (Tabelle 3).
3. Von jedem Knoten M_i läuft eine Kante mit Kapazität m_i in die Senke t, wenn m_i die Kosten sind, um die Maschine M_i zu kaufen.

Ein Min-Cut (S,T) für dieses Netzwerk ist natürlich endlich und ganzzahlig und kann effizient mit der Methode von Ford und Fulkerson berechnet werden. Wir wählen nun diejenigen Aufträge aus, die in S enthalten sind. Ebenso kaufen wir alle Maschinen in S.

Zur Begründung, warum dieses Verfahren korrekt ist: Die Kanten im Schnitt entsprechen anschaulich einem *Verzicht* auf einen Geldbetrag, welcher der Kapazität dieser Kante entspricht. Wird etwa eine Kante $s \to J_i$ für einen Auftrag J_i vom Schnitt (S,T) geschnitten, dann verzichtet man auf die Einnahmen in Höhe von j_i. Der Verzicht auf die Einnahmen kann günstiger sein (und zu einem kleineren Schnitt führen), als wenn man den Job annimmt und dafür viele andere Kanten (für Aufträge oder Maschinen) schneiden muß. Schneidet man analog eine Kante $M_i \to t$, dann muß man den entsprechenden Kaufpreis für eine Maschine bezahlen. Auch dieses kann günstiger sein, als die Maschine für jeden angenommen Job zu mieten.

Liegt andererseits eine Maschine M_j in T, dann ist es offenbar günstiger, M_j für jeden angenommenen Auftrag J_i anzumieten; andernfalls könnte man im leicht einen kleineren Schnitt konstruieren, indem man M_j nach S verschiebt (und dem Fall entspricht, M_j zu kaufen), was im Widerspruch zur Minimalität des Schnittes steht.



Hausaufgabe 4 (8 Punkte)

Wir können dies durch das Berechnen eines Flußes mit minimalen Kosten berechnen. Wir bilden das übliche Flußnetzwerk für das zugrundeliegende Matchingproblem und suchen nach einem Fluß mit Wert n, wenn n die Zahl der Studierenden ist. Um die Obergrenzen zu respektieren geben wir den Kanten, welche von den Seminaren zu t führen die entsprechende Kapazität (während die anderen Kanten eine Kapazität von eins erhalten). Jeder Kante geben wir Kosten 0 außer den Matchingkanten, welchen wir die Kosten $4-p$ geben, wenn p die Priorität dieser Wahl ist. Die Gesamtkosten sind dann um so kleiner, je höher das Gesamtglücksgefühl ist.

Hausaufgabe 5 (10 Punkte)

$1+8=9$	$21+28=49$	$43+38=81$	$49+72=121$	$55+66=121$	$77+92=169$
$3+6=9$	$27+22=49$	$35+46=81$	$53+68=121$	$57+64=121$	$79+90=169$
$5+4=9$	$29+20=49$	$47+34=81$	$59+62=121$	$65+56=121$	$81+88=169$
$7+2=9$	$19+30=49$	$33+48=81$	$67+54=121$	$61+60=121$	$83+86=169$
$9+16=25$	$23+26=49$	$39+42=81$	$51+70=121$	$87+82=169$	$93+76=169$
$11+14=25$	$17+32=49$	$41+40=81$	$69+52=121$	$89+80=169$	$91+78=169$
$13+12=25$	$31+18=49$	$37+44=81$	$71+50=121$	$73+96=169$	$95+74=169$
$15+10=25$	$25+24=49$	$45+36=81$	$63+58=121$	$75+94=169$	$85+84=169$