Prof. Dr. Volker Kaibel
Dipl.-Comp.-Math. Matthias Walter
Wintersemester 2012/2013

7. Übung Kombinatorische Optimierung

www.math.uni-magdeburg.de/institute/imo/teaching/kombopt2012/

Präsentation in den Übungen am 29.11.2012

Aufgabe 1

Für $b \in \mathbb{Z}^V$ ist ein b-*Fluss* in einem Netzwerk $(D = (V, A), u)$ mit $u \in \mathbb{Z}_+^A$ ein Fluss $f \in \mathbb{R}^A$ mit $\mathbb{O}_A \leq f \leq u$ und $\text{ex}_f(v) = b_v$ für alle $v \in V$.
Wir betrachten folgenden Algorithmus für das *Min-Cost-Flow Problem*, für konservative Bogengewichte $c \in \mathbb{R}^A$ einen b-Fluss minimaler c-Kosten $\sum\limits_{a \in A} c_a f_a$ zu finden.

Algorithmus : SuccessiveShortestPath
Eingabe : $D = (V, A)$, $u \in \mathbb{Z}_+^A$, $b \in \mathbb{Z}^V$ mit $\sum\limits_{v \in V} b_v = 0$, $c \in \mathbb{R}^A$ konservativ.
Ausgabe : c-minimaler b-Fluss in (D, u).

1 $b^0 \longleftarrow b$, $f^0 \longleftarrow \mathbb{O}_A$, $k \longleftarrow 0$
2 **while** $b^k \neq \mathbb{O}_V$ **do**
3 Wähle Knoten $s \in V$ mit $b^k(s) < 0$.
4 **if** $\exists t \in V$ *mit* $b^k(t) > 0$, t *von* s *aus in* D_{f^k} *erreichbar* **then**
5 Finde c-kürzesten s-t-Weg P in D_{f^k}.
6 Sei $\gamma := \min\left\{-b^k(s), b^k(t), \min\limits_{a \in P} \overline{u}_a\right\}$. `// ū sind Residualkapazitäten`
7 $f^{k+1} \longleftarrow f^k$ augmentiert entlang von P um Flusswert γ
8 $b^{k+1} \longleftarrow b^k$, $b^{k+1}(s) \longleftarrow b^{k+1}(s) + \gamma$, $b^{k+1}(t) \longleftarrow b^{k+1}(t) - \gamma$, $k \longleftarrow k + 1$
9 **end**
10 **else return** *"Es gibt keinen b-Fluss"*.
11 **end**
12 **return** f^k

Beweisen Sie folgende Aussagen:

(a) Der Algorithmus terminiert nach höchstens $B := \frac{1}{2} \sum\limits_{v \in V} |b_v|$ Augmentierungen.

(b) Der Algorithmus in Zeile 10 terminiert genau dann, wenn es keinen b-Fluss in (D, u) gibt. Setzen Sie dazu f^k mit $b - b^k$ in Relation.

(c) Sei f ein c-minimaler b-Fluss und P ein c-minimaler s-t-Weg in D_f und sei f' durch Augmentieren von f entlang von P um die kleinste auf P auftretende Residualkapazität γ entstanden. Dann ist f' ein c-minimaler b'-Fluss.

Tipp: Wenn f' nicht minimal ist, gibt es einen negativen Kreis. Betrachten Sie diesen und die Bögen von P, um einen kürzeren s-t-Weg zu konstruieren.

(d) Wenn der Algorithmus in Zeile 11 terminiert, dann berechnet er einen c-minimalen b-Fluss.

Bitte wenden!

Aufgabe 2

Seien $G = (V, E)$ ein ungerichteter Graph, $u \in \mathbb{R}_+^E$ eine Kapazitätsfunktion und K_V der vollständige Graph auf der Knotenmenge V. Sei T ein aufspannender Baum in K_V. Jedes $e \in T$ zerlegt T in zwei Komponenten mit den Knotenmengen C_e und $V \setminus C_e$. Sei $w_e := u(\delta_G(C_e))$ die Kapazität des dazugehörigen Schnittes. Weiterhin sei für $s, t \in V$ der Wert $\lambda_{s,t}$ die minimale u-Kapazität eines s-t-Schnittes. Man nennt T einen *Gomory-Hu-Baum* für (G, u), wenn er

$$\lambda_{s,t} = \min_{e \in P_{s,t}} w_e \text{ für alle } s, t \in V$$

erfüllt, wobei $P_{s,t}$ der eindeutige s-t-Weg in T ist.

Algorithmus : GomoryHuTree
Eingabe : $G = (V, E)$, $u \in \mathbb{R}_+^E$
Ausgabe : Ein Gomory-Hu-Baum T für (G, u)

1 $V(T) \longleftarrow \{V\}$, $E(T) \longleftarrow \varnothing$
2 **while** $\exists X \in V(T)$ *mit* $|X| \geq 2$ **do**
3   Seien $s, t \in X$ mit $s \neq t$.
4   **for** C *Zusammenhangskomponente von* $T - X$ $\;(= T[V(T) \setminus \{X\}])$ **do**
5         $S_C \longleftarrow \bigcup_{Y \in V(C)} Y$
6   **end**
7   Definiere (G', u') aus (G, u) durch Schrumpfen von S_C zu Knoten v_C für alle C.
8   Finde minimalen s-t-Schnitt $\delta(A')$ in (G', u'). Setze $B' \longleftarrow V(G') \setminus A'$.
9   Setze $A \longleftarrow \bigcup_{v_c \in A'} S_C \cup (A' \cap X)$ und $B \longleftarrow \bigcup_{v_c \in B'} S_C \cup (B' \cap X)$.
10   Setze $V(T) \longleftarrow (V(T) \setminus \{X\}) \cup \{A \cap X, B \cap X\}$.
11   **for** $e = \{X, Y\} \in E(T)$ *zu* X *inzidente Kante* **do**
12   **if** $Y \subseteq A$ **then** $e' \longleftarrow \{A \cap X, Y\}$ **else** $e' \longleftarrow \{B \cap X, Y\}$
13   Setze $E(T) \longleftarrow (E(T) \setminus \{e\}) \cup \{e'\}$ und $w(e') \longleftarrow w(e)$.
14   **end**
15   Setze $E(T) \longleftarrow E(T) \cup \{\{A \cap X, B \cap X\}\}$.
16   Setze $w(\{A \cap X, B \cap X\}) \longleftarrow u'(\delta_{G'}(A'))$.
17 **end**
18 Ersetze alle $\{x\} \in V(T)$ durch x und alle $\{\{x\}, \{y\}\} \in E(T)$ durch $\{x, y\}$.

Beweisen Sie folgende Aussagen:

(a) Für beliebige (G, u) und Knoten $i, j, k \in V$ gilt $\lambda_{i,k} \geq \min\{\lambda_{i,j}, \lambda_{j,k}\}$.

(b) Nach jeder Ausführung von Zeile 8 gilt $A \,\dot\cup\, B = V$.

(c) Nach jeder Ausführung von Zeile 8 ist $E(A, B)$ ein minimaler s-t-Schnitt in (G, u).
Tipp: Benutzen Sie Übung 4, Aufgabe 2 (c).

(d) In jedem Schritt (bis Zeile 18) gilt für alle $e \in E(T)$

$$w(e) = u(\delta_G(\bigcup_{Z \in C_e} Z)) \;,$$

wobei C_e und $V(T) \setminus C_e$ die Zusammenhangskomponenten von $T - e$ sind. Weiterhin gilt für alle $e = \{P, Q\} \in E(T)$, dass es Knoten $p \in P$ und $q \in Q$ gibt mit $\lambda_{p,q} = w(e)$.

(e) Die Ausgabe des Algorithmus ist ein Gomory-Hu-Baum.