Approximationsalgorithmen

Wintersemester 2013/2014

Prof. Dr. Georg Schnitger
Dipl.-Inf. Bert Besser

Arbeitsgruppe Theoretische Informatik, Institut für Informatik

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Übungsblatt 9

Ausgabe: 20.01.2014
Abgabe : 27.01.2014 **vor** Vorlesungsbeginn

9.1. Aufgabe (6) — *Prize Collecting Vertex Cover*

Im Price Collecting Vertex Cover Problem ist ein Graph (V, E) mit nicht negativen Knoten- und Kantengewichten $w(v), v \in V$ bzw. $w(e), e \in E$ gegeben. Gesucht ist eine Teilmenge der Knoten $V' \subseteq V$, so dass $\sum_{v \in V'} w(v) + \sum_{u,v \notin V'} w(\{u, v\})$ minimal ist. Im Unterschied zum herkömmlichen Vertex Cover Problem werden nicht überdeckte Kanten nicht verboten aber bestraft.

Zeige, dass der folgende Algorithmus 2-approximativ ist (Algorithmus 5.3 aus dem Skript):

> Solange eine Kante $\{u, v\} \in E$ mit $c := \min\{w(u), w(v), w(\{u, v\})\} > 0$ existiert:
>   Setze $w(u) -= c$, $w(v) -= c$ und $w(\{u, v\}) -= c$
> Gib $\{u : w(u) = 0\}$ aus.

9.2. Aufgabe (10) — *Kompatibilität*

Aus den Mengen L, R von Objekten und den Kompatibilitäten $K \subseteq L \times R$ sollen billig Teilmengen $L' \subseteq L, R' \subseteq R$ der Objekte mit den neuen Kompatibilitäten $K' = L' \times R'$ erzeugt werden. Wir dürfen die Kompatibilität zwischen zwei beliebigen Objekten $l \in L, r \in R$ herstellen – dabei sind die Kosten $w(\{l, r\})$ von den beteiligten Objekten abhängig – oder ein Objekt x entfernen – wir bezahlen dann mit dem Wert $w(x)$ des Objekts x und der mit x nun nicht mehr bestehenden Kompatibilitäten. Das Objekt x ist ein und für alle Mal verschwunden, zu x kann keine Kompatibilität mehr hergestellt werden.

Wir verallgemeinern das Problem. Die Eingabe besteht aus einem k-partiten Graphen $G = (V_1, \ldots, V_k, E)$ mit $E \subseteq \mathcal{E} := \cup_{1 \leq i,j \leq k, i \neq j} \{\{u, v\} : u \in V_i, v \in V_j\}$ und den Knoten- und Paargewichten

$$w(v) \geq 0, v \in \cup_{1 \leq i \leq k} V_i \quad \text{bzw.} \quad w(\{u, v\}) \geq 0, \{u, v\} \in \mathcal{E}.$$

Die Ausgabe ist ein vollständiger k-partiter Graph $G' = (V_1', \ldots, V_k', \mathcal{E}')$ mit $V_i' \subseteq V_i$ und $\mathcal{E}' = \cup_{1 \leq i,j \leq k, i \neq j} \{\{u, v\} : u \in V_i', v \in V_j'\}$, wobei G' aus G durch Knotenlöschungen und Kanteneinfügungen entsteht. Das Einfügen einer neuen Kante zwischen zwei nicht gelöschten Knoten u, v verursacht die Kosten $w(\{u, v\})$, das Löschen eines Knotens x verursacht die Kosten

$$W(x) = w(x) + \sum_{\{x,y\} \in E} w(\{x, y\}).$$

b. w.

Gesucht ist eine Menge von Lösch- und Einfügeoperationen mit minimalen Kosten. Entwickle einen Local-Ratio-Algorithmus und zeige, dass er 3-approximativ ist.

Hinweis: Die Kante $\{u, v\} \notin E$ wird eingefügt, genau dann wenn keiner von u, v gelöscht wird. Die Kante $\{u, v\} \in E$ wird gelöscht, genau dann wenn mindestens einer von u, v gelöscht wird. Eine Lösung entspricht also eineindeutig einer Menge von gelöschten Knoten. Es ist daher ausreichend eine Menge von gelöschten Knoten zu errechnen.