Fernau/Raible
Universität Trier

Wintersemester 2008/2009

Übungen zur Vorlesung

Näherungsalgorithmen

Aufgabenblatt 3

Aufgabe 1 (Extended Local-Ratio-Theorem + Kruskal's MST-Algorithmus)

1. **Extended Local-Ratio-Theorem**

   - Def.: Überdeckung C ist w-minimal, falls $\forall_{x \in C\,,\, w(x)>0} : C \setminus \{x\}$ ist keine Überdeckung.
   - δ ist minimal-r-effektiv, falls
     (a) $\forall x \in X : 0 \leq \delta(x) \leq w(x)$
     (b) $\delta(C) \leq r \cdot OPT(\delta)$ für alle δ-minimalen Überdeckungen C
   - Betrachte nun folgenden generischen Algorithmus $A(X, f, w)$:
     (a) Nehme minimal-r-effektives δ
     (b) $C \leftarrow B(X, f, w - \delta)$
     (c) Für alle $x \in C$: Falls $\delta(x) > 0$ und $C \setminus \{x\}$ ist Überdeckung, dann $C \leftarrow C \setminus \{x\}$
     (d) **return** C
   - Zeige nun das **Extended Local-Ratio-Theorem**: Falls B ein C mit $(w - \delta)(C) \leq r \cdot OPT(w - \delta)$ liefert, dann $w(C) \leq r \cdot OPT(w)$.

2. Es gelte Q ist Überdeckung $\iff$ $f(Q) = 1$ $\iff$ Q ist spannender zusammenhängender Subgraph, der alle mit 0 gewichteten Kanten enthält.

   Betrachte nun folgenden Algorithmus $A(E, f, w)$ mit

   (a) Falls $E_{\geq 1} := \{e \in E \mid w(e) > 0\} = \emptyset$, dann **return** E
   (b) Nehme aus der Menge $E_{\geq 1}$ das Element e_i mit dem kleinsten Gewicht.
   (c) Sei $\delta(e) = \begin{cases} w(e_i) : w(e) > 0 \\ 0 \quad : \quad \text{sonst} \end{cases}$
   (d) $C \leftarrow A(E, f, w - \delta)$
   (e) Für alle $x \in C$: Falls $\delta(x) > 0$ und $f(C \setminus \{x\}) = 1$ (ist Überdeckung!), dann $C \leftarrow C \setminus \{x\}$
   (f) **return** C

Was berechnet der Algorithmus? Benutze nun das **Extended Local-Ratio-Theorem** um die Korrektheit des Algorithmus A zu beweisen. Zeige also, dass A eine optimale Lösung liefert dadurch, dass δ minimal-1-effektiv ist. Vor dem ersten Aufruf von A soll gelten $w : E \to \mathbb{N}_{>0}$.

Aufgabe 2 (Job-Scheduling)
Gegeben seien m Maschinen $M_1, \ldots, M_m$ und Jobs $J_1, \ldots, J_n$ sowie zu jedem Job J_i seine Laufzeit $p_i \in \mathbb{N}$.
Wir suchen nun eine Zuweisung der Jobs auf Maschinen ($z : \{1, \ldots, n\} \to \{1, \ldots, m\}$), so dass die maximale Durchlaufzeit D_{max} minimiert wird:

$$D_{max} = \max_{1 \leq j \leq m} \sum_{1 \leq i \leq n, z(i)=j} p_i$$

Für eine Maschine M_j sei die aktuelle Last unter z $L_z(M_j) = \sum_{1 \leq i \leq n, z(i)=j} p_i$.
Betrachte nun folgenden Greedy-Algorithmus:

1: $z(i) = \infty$ für $1 \leq i \leq n$
2: **for** $i = 1 : n$ **do**
3: Wähle M_j so, dass $L_z(M_j)$ minimal ist.
4: Setze $z(i) = j$.
5: **end for**

1. Sei M_ℓ eine Maschine mit $L_z(M_\ell) = D_{max}$. Sei J_ℓ der Job der als letztes M_ℓ zugeteilt wurde. Betrachte nun die partielle Zuteilung z' vor dieser Zuteilung von J_ℓ auf M_ℓ. Zeige nun:

$$L_{z'}(M_\ell) \leq \frac{1}{m} \sum_{1 \leq i \leq n, i \neq \ell} p_i$$

2. Zeige, dass $D_{\max} \leq \frac{1}{m} \sum_{1 \leq k \leq n} p_k + (1 - \frac{1}{m}) p_\ell$.

3. Sei Opt die Durchlaufzeit eines optimalen Schedule.
Zeige nun $D_{\max} \leq 2Opt$.

4. Zeige, dass wenn $m = 2$ sogar $D_{max} \leq 1.5Opt$ gilt.

Aufgabe 3 (Rucksack)
Betrachten Sie, dass Rucksackproblem (Maximum Knapsack) auf Seite 21 VL 3.
Nehmen Sie an Sie besitzen einen Algorithmus A um die Entscheidungsvariante dieses Problems in Laufzeit $t(n)$ zu lösen (n = Anzahl der Objekte).
Fur gegebenes $x \in I_{\mathcal{P}}$ und $k \in \mathbb{N}$ können Sie nun die Frage $m^*_{\mathcal{P}}(x) \geq k$? entscheiden.
Nutze nun A um ein $y \in S_{\mathcal{P}}$ zu finden, so dass $m_{\mathcal{P}}(x, y) \geq k$ falls eines existiert.
Dazu sollte nicht mehr als $O(n \cdot (t(n))$ Schritte verwendet werden