Fernau/Raible
Universität Trier

Wintersemester 2008/2009

Übungen zur Vorlesung

Näherungsalgorithmen

Aufgabenblatt 10

Aufgabe 1
Sei P ein Maximierungsproblem, dessen Zielfunktion ganzzahlig ist: $\sum_{i=1}^{n} c_i r_i \cdot x_i$, wobei $r_1, \ldots, r_n \in \mathbb{N}$ als Instanz zu sehen ist und x_i als Indikatorvariable ($x_i = 1 \iff$ Element x_i ist in Lösung). Solch ein Problem ist *monoton* falls für zwei Instanzen $r'_1, \ldots, r'_n$ und $r_1, \ldots, r_n$ mit $r'_i \leq r_i$ für alle i gilt:

$$OPT(r'_1, \ldots, r'_n) \leq OPT(r_1, \ldots, r_n)$$

Zeige, dass wenn P monoton ist und einen Pseudopolynomiellen Algorithmus A besitzt, dann hat es einen $FPTAS$.
(Tipp: Teile ähnlich wie bei Knapsack die r_i durch eine geeignete Konstante. Ist $c_i = 1$ und $r_i = p_i$ so lässt sich damit Knapsack ausdrücken)

Aufgabe 2
Zeigen Sie, dass es für (ungerichtetes) TSP mit Dreiecksungleichung keinen FPTAS geben kann, falls $P \neq NP$.
Zeigen Sie zuerst, dass das Problem stark NP-hart ist. Das heisst, dass Sie die Kantengewichte durch ein Polynom beschränken müssen. Gleichzeitig soll aber auch die Dreiecksungleichung erfüllt sein. Betrachten Sie den Standard NP-Härte Beweis und modifizieren Sie ihn.

Aufgabe 3
Zeigen Sie, dass folgende Probleme keinen FPTAS besitzen, falls $P \neq NP$.

- Variable-weighted 2-Sat
  Eingabe: Eine logische Formel F mit maximaler Klauselläge zwei über den Variablen $V(F) = \{x_1, \ldots, x_n\}$ und $w : V(F) \to \mathbb{N}$.
  Ausgabe: Eine Belegung die F erfüllt, sodass $\sum_{\substack{x_i \in V(F) \\ x_i = \text{true}}} w(x_i)$ kleinstmöglich.

  In obiger Summe werden also nur die Variablen gezählt die auf true gesetzt werden.

- Leaf-weighted Spanning Tree
  Eingabe: $G(V, E)$ und $\alpha : V \to \mathbb{N}$.

Ausgabe: Ein Spannbaum $T \subseteq E$, sodass $\sum_{x \in L(T)} \alpha(x)$ kleinstmöglich ist (wobei $L(T)$ die Blätter von T sind).

Aufgabe 4

MaxCoverage

Eingabe: Ein Universum $\mathcal{U} = \{u_1, \ldots, u_n\}$, $\mathcal{S} = \{S_1, \ldots, S_m\}$ mit $S_i \subseteq \mathcal{U}$ und $n \geq k > 0$.

Gesucht: $\mathcal{C} \subseteq \mathcal{S}$ mit $|\mathcal{C}| = k$ und $|\bigcup_{S \in \mathcal{C}} S|$ größtmöglich.

Betrachte nun den folgenden Greedy-Algorithmus:

1: $\mathcal{C} \leftarrow \emptyset$.
2: **for** $i = 1 : k$ **do**
3: Wähle $C_i \in \mathcal{S}$, sodass $|\mathcal{C} \cup C_i|$ größtmöglich.
4: $\mathcal{C} \leftarrow \mathcal{C} \cup C_i$.
5: **end for**
6: **return** $\mathcal{C}$.

Zeigen Sie:

1. $|\bigcup_{i=1}^{l} C_i| - |\bigcup_{i=1}^{l-1} C_i| \geq \frac{1}{k}(OPT - |\bigcup_{i=1}^{l-1} C_i|)$
   Tipp: Benutzen Sie das Taubenschlagprinzip (Pigeon-hole-principle). Sei Q eine Menge aus der optimalen Lösung. Was kann man über $|Q \setminus \bigcup_{i=1}^{l-1} C_i|$ sagen?
2. $|\bigcup_{i=1}^{l} C_i| \geq [1 - (1 - \frac{1}{k})^l]OPT$ mit $l = 1, \ldots, k$.
   Tipp: Induktion und Teil 1).
3. $|\bigcup_{S \in \mathcal{C}} S| \leq (1 - e^{-1})OPT$