Fernau/Raible
Universität Trier

Wintersemester 2006/2007

Übungen zur Vorlesung

Näherungsalgorithmen

Aufgabenblatt 3

Aufgabe 1 (Nearest-Neighbor Heuristik für TSP)

Gegeben sei ein vollständiger Graph $G(V,E)$ (G ist also ein K_n) und eine Gewichtsfunktion $w : E \to \mathbb{N}^{\geq 1}$. Die Nearest-Neighbor Heuristik wählt eine Knoten $s \in V$, setzte $T = s = v_1$ und verfährt wie folgt:

1. Sei $T = v_1 \dots v_\ell$ die aktuelle Teiltour und
   $NN = \{v \in N(v_\ell) \setminus T \mid w(v)$ hat kleinstmögliches Gewicht in $N(v_\ell) \setminus T\}$.
2. Wähle nun ein $u \in NN$ und setzte $T' = Tu$.
3. $T \leftarrow T'$, gehe zu 1..

Geben Sie für jede Konstante $c > 0$ einen Graph G an, so dass die obige Heuristik eine Lösung liefert, die eine relative Abweichung von MINDESTENS c bzgl. des Optimums hat.

Aufgabe 2 (Max-TSP)

Gegeben sei ein vollständiger Graph $G(V,E)$ (G ist also ein K_n) und eine Gewichtsfunktion $w : E \to \mathbb{N}^{\geq 1}$. Das Max-TSP Problem sucht nun nach einer längsten Rundtour durch alle Knoten von G.
Eine *Kreisüberdeckung* ist eine Menge von disjunkten Kreisen $\mathcal{C} = \{C_1, \dots, C_\ell\}$ mit $V = \bigcup_{i=1}^{\ell} C_i$. Die Länge einer Kreisüberdeckung ist $\sum_{i=1}^{\ell} w(E(C_i))$, also die Summe der Gewichte der Kanten, die in den Kreisen $C_1, \dots, C_\ell$ vorkommen. Eine längste Kreisüberdeckung kann in $O(n^3)$ berechnet werden.

- Entwerfe einen Polynomzeit-Algorithmus der ein Tour durch alle Knoten liefert, die mindestens $\frac{2}{3}$-mal so lang ist wie die optimale Tour.
- Hat dies irgendwelche Auswirkungen auf Min-TSP?

Aufgabe 3 (Symmetrisches Δ-TSP)

Gegeben $G(V,E)$ und $w : E \to \mathbb{N}$, gesucht kostenminimale Tour durch alle Knoten.

1. Sei ST eine minimaler spannender Baum und T_{opt} eine optimale Rundreise in G.
   Zeige: $w(ST) < w(T_{opt})$.

2. Betrachten Sie die folgende closest point Heuristik für das symmetrische Traveling Salesman Problem mit Dreiecksungleichung: Starte mit einem Zyklus Z_1 , der aus nur einer (beliebigen) Stadt besteht; erweitere in jedem Schritt den Zyklus Z_i um eine Stadt u, fur die gilt:

$$\exists v \in Z_i : \forall x \in Z_i \text{ und } \forall y \notin Z_i : w(\{x, y\}) \geq w(\{v, u\})$$

Wir wählen also ein $u \notin Z_i$, so dass der Minimalabstand bzgl. aller Städte in Z_i minimiert wird. v sei die Stadt aus Z_i mit dem geringsten Abstand zu u. Dabei soll u unmittelbar nach v und vor dessen Nachfolger in Z_i eingefügt werden um Z_{i+1} zu erhalten. Nach n Schritten ist eine Tour gefunden. Zeigen Sie, daß diese Heuristik eine Tour findet, die höchstens doppelt so lang ist wie die optimale Tour. Hinweis: Vergleichen Sie den Algorithmus mit dem Prim-Algorithmus fur MST's:

(a) $V' \leftarrow v$, wobei v beliebig, $i = 1$, $MST_1 = \emptyset$
(b) Solange $V \neq V'$, tue
   i. Wähle $e \in E$ mit $w(e) = \min\{\{a, b\} \mid a \in V',\ b \notin V'\}$
   ii. $i = i + 1$, $MST_i = MST_{i-1} \cup \{e\}$, $V' = v' \cup \{e\}$
(c) **return** MST_n.

Tipp: Schätze $w(Z_i)$ in jedem Schritt durch $w(MST_i)$ nach oben ab.