

Frank Hoffmann
Klaus Kriegel
Romain Grunert
Ludmilla Scharf
André Schulz

Ws 2007/08

18. Januar 2008
Abgabe: 28. Januar 2008
vor der Vorlesung

12. Übung zur Vorlesung Höhere Algorithmik

1. Aufgabe: Load Balancing (5 Punkte).

Aus der Vorlesung kennen Sie das Load Balancing Problem und den Greedy-Balance Approximationsalgorithmus dafür. Betrachten Sie folgenden Spezialfall des Problems: alle Zeiten t_1, \dots, t_n sind ganze Zahlen aus dem Intervall $[1, 100]$, die Anzahl der Maschinen ist $m = 20$, und die Gesamtlaufzeit aller Arbeitsaufträge ist mindestens 10 000, d.h., $S = \sum_{j=1}^n t_j \geq 10\,000$.

Zeigen Sie, dass in diesem Fall der Algorithmus Greedy-Balance mit dem Approximationsfaktor 1,2 arbeitet.

2. Aufgabe: Unabhängige Knotenmenge (5 Punkte).

Gegeben sei ein schlichter ungerichteter Graph $G = (V, E)$. Eine Teilmenge der Knoten $I \subset V$ heißt *unabhängige Knotenmenge*, wenn keine zwei Knoten aus I in G durch eine Kante verbunden sind: $\forall u, v \in I : \{u, v\} \notin E$. Zusätzlich sei eine Gewichtsfunktion $w : V \rightarrow \mathbb{R}^+$ gegeben. Gesucht ist eine unabhängige Knotenmenge mit maximalem Gewicht, wobei das Gewicht einer Knotenmenge als Summe der Gewichte aller Knoten der Menge definiert ist.

Betrachten Sie folgenden Greedy-Algorithmus, wobei $N(v)$ die Menge der Nachbarknoten von v bezeichnet und $\overline{N(v)} = N(v) \cup \{v\}$.

- 1 $I = \emptyset, V' = V$
- 2 **while** $V' \neq \emptyset$ **do**
- 3 wähle v aus V' mit maximalem Gewicht
- 4 $I = I \cup \{v\}$
- 5 $V' = V' \setminus \overline{N(v)}$

- (a) Sei I die von dem Greedy-Algorithmus gefundene unabhängige Knotenmenge und I^* eine optimale unabhängige Knotenmenge. Zeigen Sie, dass für jeden Knoten $v \in I^*$ gilt: entweder ist $v \in I$, oder es gibt einen Knoten $u \in I$ mit $w(u) \geq w(v)$ und $\{u, v\}$ ist eine Kante in E .
- (b) Zeigen Sie, dass der Algorithmus den Approximationsfaktor $\frac{1}{\Delta}$ hat (d.h. $w(I) \geq \frac{1}{\Delta}w(I^*)$), wobei Δ der maximale Knotengrad in G ist.

3. Aufgabe: Center Selection (5 Punkte).

Wir betrachten das Center Selection Problem für Punkte auf der reellen Achse: Gegeben

ist eine Menge von n Orten $S = \{s_1, \dots, s_n\} \subset \mathbb{R}$ und eine Zahl $m \in \mathbb{N}$. Finden Sie eine Menge von m Zahlen $Z = \{z_1, \dots, z_m\}$, die den maximalen Abstand von den Orten s_i zu Z minimiert:

$$r^* = \min_Z \max_{1 \leq i \leq n} \text{dist}(s_i, Z)$$

Geben Sie einen Algorithmus an, der das Problem mit Hilfe der Dynamischen Programmierung exakt löst und analysieren Sie diesen.

4. Aufgabe: SubsetSum (5 Punkte).

Betrachten Sie folgende Variante des SubsetSum Problems: Gegeben sind n Zahlen $x_1, \dots, x_n \in \mathbb{N}$ und eine Zahl $X \in \mathbb{N}$. Finden Sie eine Teilmenge $I \subset \{1, \dots, n\}$, so dass die Summe $S = \sum_{i \in I} x_i \leq X$ und S wird maximiert.

Betrachten Sie folgende zwei Heuristiken:

Best-Fit: Initialisiere $I = \emptyset$. Solange $S \leq X$ suche aus allen x_i mit $i \notin I$ das größte so dass $S + x_i \leq X$ und füge i der Indexmenge I hinzu.

First-Fit: Gehe die Zahlen x_1, \dots, x_n in der Reihenfolge der Indexierung durch und füge die Indizes der Menge I hinzu solange die Summe $S \leq X$ bleibt.

Bestimmen Sie möglichst gute Approximationsfaktoren für beide Heuristiken.