Approximationsalgorithmen

Wintersemester 2013/2014

Prof. Dr. Georg Schnitger

Dipl. Inf. Bert Besser

Arbeitsgruppe Theoretische Informatik, Institut für Informatik

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Übungsblatt 3

Ausgabe: 04.11.2013

Abgabe : 11.11.2013 **vor** Vorlesungsbeginn

3.1. Aufgabe (4+4+4) *k-Matroide*

Wir betrachten die folgende Variante des Scheduling-Problems. Beginnend mit Zeitpunkt 0 sollen auf einer Maschine die Jobs $J = \{j_1, \ldots, j_n\}$ ausgeführt werden. Für $1 \leq i \leq n$ hat Job j_i die Länge $|j_i| \in \mathbb{N}$, die Frist $f_i \in \mathbb{N}$ und den Profit $p_i \geq 0$. Jeder Job muss zu einem Zeitpunkt in $\mathbb{N}$ starten. Der Profit für einen Job wird nur dann ausgezahlt, wenn die entsprechende Frist eingehalten wird. Es gelte

$$1 \leq |j_i| \leq k. \tag{1}$$

Gesucht ist eine profitabelste Bearbeitungsreihenfolge der Jobs.

a) Zeige, dass $(\mathcal{P}, J)$ ein k-erweiterbares Mengensystem ist, wobei $\mathcal{P}$ alle Teilmengen von Jobs sind, die unter Einhaltung aller entsprechenden Fristen bearbeitet werden können.

b) Zeige: für jedes $\varepsilon > 0$ ist der Approximationsfaktor des Greedy-Algorithmus höchstens $\frac{1}{k-\varepsilon}$.

c) Welchen Approximationsfaktor hat der Greedy-Algorithms, wenn wir (1) durch

$$\max_i |j_i| \leq k \cdot \min_i |j_i| \tag{2}$$

ersetzen? Was passiert, wenn wir für jeden Job zusätzlich eine Startzeit $s_i \in \mathbb{N}$ fordern, *nach* der der Job ausgeführt werden muss, um den Profit p_i zu erhalten?

> In der Vorlesung haben wir gezeigt, dass der Greedy-Algorithmus optimal ist, wenn alle Jobs Einheitslänge haben. Hier zeigen wir, dass der Approximationsfaktor mit der maximalen Bearbeitungslänge übereinstimmt.

3.2. Aufgabe (6+6) *Sequenzierung*

a) Seien u_1, u_2, v_1, v_2 vier beliebige, nicht notwendigerweise verschiedene Strings und es gelte $ov(u_1, v_1) = \max\{ ov(u_i, v_j) \mid 1 \leq i, j \leq 2\}$. Zeige die *Monge-Ungleichung* (Gaspard Monge 1746-1818):

$$ov(u_1, v_1) + ov(u_2, v_2) \geq ov(u_1, v_2) + ov(u_2, v_1).$$

b) Im zyklischen Überdeckungsproblem sind Strings $s_1, \ldots, s_n$ gegeben und eine möglichst kurze zyklische Überdeckung der Strings ist gesucht. Der Greedy-Algorithmus arbeitet iterativ. In jedem Schritt wird das Paar (s_i, s_j) mit größtem Overlap $ov(s_i, s_j)$ bestimmt – $i = j$ darf gelten – und der String s_j wird maximal über den String s_i geschoben; sodann werden s_i und s_j durch den resultierenden String ersetzt. Der Algorithmus endet, wenn nur noch Kreise existieren.

Zeige, dass der Greedy-Algorithmus eine kürzeste Überdeckung bestimmt.