

Übungen zur Vorlesung Approximation-Algorithmen

Blatt 2

Aufgabe 4: Für eine Instanz x von MAXIMUM KNAPSACK bezeichne $m_H(x)$ den Wert der vom modifizierten Greedy-Algorithmus gefundenen Lösung,

$$m_H(x) = \max(m_{\text{Greedy}}(x), p_{\max}) .$$

Konstruieren Sie für jedes $\epsilon > 0$ eine Instanz x , für die $m^*(x) > (2 - \epsilon)m_H(x)$ ist. Die in der Vorlesung gegebene Analyse des Algorithmus kann also nicht verbessert werden.

Aufgabe 5: Zeigen Sie, dass die in der Vorlesung gezeigte Schranke für die Regel *LPT* beim Problem MINIMUM SCHEDULING ON IDENTICAL MACHINES,

$$m_{LPT}(x) \leq \left(\frac{4}{3} - \frac{1}{3p}\right)m^*(x)$$

für Instanzen x mit $p = 2$ scharf ist.

Aufgabe 6: Betrachten Sie die folgende Variante *LPT'* der Regel *LPT*: die Aufträge t_1, \dots, t_n seien sortiert, so dass $\ell_1 \geq \ell_2 \geq \dots \geq \ell_n$. Zunächst werden t_1 und t_2 dem Prozessor 1 zugewiesen, danach wird *List Scheduling* mit der Regel *LPT* auf die verbleibenden Aufträge t_3, \dots, t_n angewandt. Die so gefundene Lösung wird mit der durch *LPT* gefundenen verglichen und die bessere ausgewählt.

Bezeichne $m_{LPT'}(x)$ die Makespan der Lösung, die auf diese Weise gefunden wird. Geben Sie eine obere Schranke für $m_{LPT'}(x)$ für Instanzen x mit $p = 2$ an, die besser ist als die für *LPT* angegebene ($m_{LPT}(x) \leq \frac{7}{6}m^*(x)$).

Aufgabe 7: Betrachten Sie den Algorithmus *Best Fit Decreasing* für das Problem MINIMUM BIN PACKING.

Sortiere A so dass $a_1 \geq a_2 \geq \dots \geq a_n$

$k := 1$; $B_1 := \emptyset$

Für $i := 1$ bis n

finde $1 \leq j \leq k$ mit $\Sigma B_j + a_i \leq 1$ and ΣB_j maximal

falls ein solches j existiert, setze $B_j := B_j \cup \{a_i\}$

sonst setze $k := k + 1$; $B_k := \{a_i\}$

Es gibt Beispiele, die zeigen, dass dieser Algorithmus für manche Instanzen besser ist als *First Fit Decreasing*, z.B.:

$$a_1 = \dots = a_n = 7/10$$

$$a_{n+1} = \dots = a_{3n} = 2/5$$

$$a_{3n+1} = \dots = a_{4n} = 1/5$$

$$a_{4n+1} = \dots = a_{6n} = 3/20$$

Finden Sie eine Instanz, für die *First Fit Decreasing* weniger Bins verwendet als *Best Fit Decreasing*.