Dr. Ulf-Peter Schroeder

Paderborn, 31. Mai 2017
Abgabe: Keine!

Übungen zur Vorlesung

Methoden des Algorithmenentwurfs

SS 2017

Blatt 6

Aufgabe 14:
Wir betrachten das Optimierungsproblem 2-Prozessor-Scheduling. Gegeben sind zwei identische Prozessoren, sowie m Jobs mit den Bearbeitsungszeiten $t_1, \ldots, t_m \in \mathbb{N}$. Jeder Job kann auf einem beliebigen Prozessor ausgeführt werden und benötigt auf beiden Prozessoren die gleiche Zeit. Jeder Prozessor kann zu einem Zeitpunkt nur einen Job bearbeiten. Die Jobs sollen so auf beide Prozessoren verteilt werden, dass alle Jobs bearbeitet werden und der letzte Job möglichst früh beendet ist. D.h., zu finden ist eine Partition $\{1, 2, \ldots, m\} = J_1 \cup J_2$, sodass die Bearbeitungszeit

$$\max \left\{ \sum_{i \in J_1} t_i, \sum_{j \in J_2} t_j \right\}$$

minimal ist. Betrachten Sie den folgenden naiven Algorithmus:

NaivesScheduling
  $J_1 \leftarrow \{1, 2, \ldots, m\}$
  $J_2 \leftarrow \emptyset$
  return (J_1, J_2)

Zeigen Sie nun:

a) Der Algorithmus NaivesScheduling gibt einen Schedule zurück, dessen Bearbeitungszeit höchstens doppelt so lang ist wie die Bearbeitungszeit eines optimalen Schedules.

b) Für alle $m \geq 2$ gibt es eine Eingabe $t_1, t_2, \ldots t_m$, sodass NaivesScheduling Mengen J_1 und J_2 berechnet, deren Bearbeitungszeit tatsächlich doppelt so groß ist, wie die Bearbeitungszeit eines optimalen Schedules.

Aufgabe 15:
Suppose you are given a set of positive integers $A = \{a_1, a_2, \ldots, a_n\}$ and a positive integer B. A subset $S \subseteq A$ is called *feasible* if the sum of the numbers in S does not exceed B:

$$\sum_{a_i \in S} a_i \leq B.$$

The sum of the numbers in S will be called the *total sum* of S. You would like to select a feasible subset S of A whose total sum is as large as possible.

a) Consider the following Algorithm:

NaiveSubset
$S \leftarrow \emptyset$
$T \leftarrow 0$
for $i = 1, 2, \ldots, n$ **do**
  if $T + a_i \leq B$ **then**
    $S \leftarrow S \cup \{a_i\}$
    $T \leftarrow T + a_i$
  end if
end for
return S

Give an instance in which the total sum of the set S returned by this algorithm is less than half the total sum of some other feasible subset of A.

b) Give a polynomial-time approximation algorithm for this problem with the following guarantee: It returns a feasible set $S \subseteq A$ whose total sum is at least half as large as the maximum total sum of any feasible set $S' \subseteq A$.

Aufgabe 16:
A *vertex-cover* of an undirected graph $G = (V, E)$ is a subset $V' \subseteq V$ such that if (u, v) is an edge of G, then either $u \in V'$ or $v \in V'$. The size of a vertex cover is the number of vertices in it. The vertex-cover-problem is to find a vertex cover of minimum size in a given undirected graph.

a) Develop an greedy approach to approximate an optimal vertex cover where the size of the greedy solution is guaranteed to be no more than twice the size of an optimal vertex cover.

b) Show that the greedy approach where we repeatedly select a vertex of highest degree, and remove all of its incident edges, does not have a approximation ratio bound of 2.