Prof. Dr. Susanne Albers
Dr. Dimitrios Letsios
Dario Frascaria
Lehrstuhl für Theoretische Informatik
Fakultät für Informatik
Technische Universität München

Problem set 11
June 29, 2015
Summer Semester 2015

Online and Approximation Algorithms

Due July 06, 2015 before class!

Exercise 1 (Fractional Knapsack - 10 points)

In the fractional knapsack problem, we are given n items with weights $w_1, w_2, \ldots, w_n$ and values $v_1, v_2, \ldots, v_n$. The objective is to pack items of maximum total value in a knapsack of capacity B and we are allowed to take any fraction of each item. A fraction $x_i \in [0, 1]$ of item i has weight $x_i \cdot w_i$ and value $x_i \cdot v_i$. Propose an optimal algorithm for the fractional knapsack problem and show that it solves the problem optimally in polynomial time.

Exercise 2 (Greedy Knapsack - 10 points)

Consider the greedy algorithm for the knapsack problem which sorts the objects in decreasing ratio of value to weight and packs the items in this order until it finds item k, which is the first item that does not fit. Show that the approximation ratio of this algorithm is not bounded by a constant. Next, consider a modified algorithm which works similarly except that it returns the most valuable among $\{1, 2, \ldots, k-1\}$ and $\{k\}$. Show that, if every item has weight at most B, where B is the knapsack capacity, then this algorithm achieves an approximation ratio of 2.

Exercise 3 (Makespan Minimization - 10 points)

Recall the problem of scheduling n jobs with processing times $p_1, p_2, \ldots, p_n$ on m machines, where the goal is makespan minimization. Consider the algorithm which starts from an arbitrary schedule S and modifies the schedule iteratively as follows. It identifies a job j currently assigned to processor p and it moves j to processor q if the new completion time of q after the move is smaller than the initial completion time of processor p. The algorithm terminates if it is not possible to perform such a move for any job. Show that this algorithm terminates and that it achieves an approximation ratio of 2.

Exercise 4 (Makespan Minimization on Two Machines - 10 points)

We consider the problem of scheduling n jobs with processing times $p_1, p_2, \ldots, p_n$ on two machines with the objective of minimizing the makespan. Our aim is to derive a PTAS (different than the one presented in class).

- Let M be a boolean 2-dimensional matrix. For $i = 0, 1, \ldots, n$ and $j = 0, 1, \ldots, P$, where $P = \sum_{k=1}^{n} p_k$, $M(i, j)$ is true if there is a subset $S \subseteq \{1, 2, \ldots, i\}$ such that

$\sum_{k\in S} p_k = j$. Otherwise, $M(i,j)$ is false. Propose an algorithm for computing M in pseudo-polynomial time. Then, for a given pair (i,j) and if $M(i,j)$ is true, propose an algorithm for computing a subset $S \subseteq \{1,2,\ldots,i\}$ such that $\sum_{k\in S} p_k = j$ in pseudo-polynomial time.

- Propose an algorithm for solving the makespan minimization problem on two machines in pseudo-polynomial time.

- By using an appropriate scaling the processing times of the jobs, show that there exists a PTAS which computes an $(1+\epsilon)$-approximation algorithm in $O(\frac{n^2}{\epsilon})$ time.