Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
S. Daum, H. Ghodselahi, O. Saukh February 04, 2014

Algorithm Theory, Winter Term 2013/14
Problem Set 7

hand in by Thursday, February 13, 2014

Exercise 1: Randomized Approximation

The maximum 3-coloring problem asks for assigning one of the colors $\{1, 2, 3\}$ to each node $v \in V$ of graph $G = (V, E)$ such that the number of edges $\{u, v\} \in E$ for which u and v get different colors is maximized. A simple randomized algorithm for the problem would be to assign a uniform random color to each node. What is the expected approximation ration of this algorithm?

Exercise 2: Load Balancing Approximation

In class, we considered the following load balancing problem. There are n jobs and m machines; job i requires t_i time units to be executed. The objective is to assign each job to one of the m machines such that the makespan (the maximum total processing time of a single machine) is minimized. We have seen that if the jobs are sorted such that $t_1 \geq t_2 \geq \cdots \geq t_n$, the following greedy algorithm has an approximation ratio of 3/2. The greedy algorithm goes through the jobs in the given order and always assigns a job to the machine with the least load. In the exercise, we want to understand this algorithm a bit better.
As in the lecture, assume that T is the makespan of the solution constructed by the described algorithm and assume that i is a machine with load T in the greedy solution. Further, assume that $\hat{n}$ is the last job that is scheduled on machine i (in class, we called this job j). Note that even without jobs $\hat{n} + 1, \ldots, n$, the makespan of the algorithm is T and clearly the optimal makespan when only considering jobs $1, \ldots, \hat{n}$ cannot be larger than the optimal makespan for all the jobs.

(a) Show that if an optimal solution for jobs $1, \ldots, \hat{n}$ assigns at most 2 jobs to each machine, the algorithm computes an optimal solution (i.e., $T = T^*$).

(b) Show that therefore, either $t_{\hat{n}} \leq T^*/3$ or the greedy algorithm computes an optimal solution.

(c) Based on (a) and (b), conclude that the algorithm has approximation ratio at most 4/3.

(d) Try to find a bad input for the algorithm, where the ratio T/T^* between the makespan of the solution of the algorithm and the optimal makespan is at least a fixed constant larger than 1.
Hint: There is an example with $m = 2$ and $n = O(1)$.

Exercise 3: Online Bin-Packing

The *Online Bin-Packing* problem is a variant of the knapsack problem. We are given an unlimited number of bins, each of size 1. We get a sequence of items one by one (each of size at most 1), and are required to place them into bins as we receive them. Our goal is to minimize the number of bins we use, subject to the constraint that no bin should be filled to more than its capacity. In this question we will consider a simple online algorithm for this problem called **First-Fit (FF)**. FF orders the bins arbitrarily, and places each item into the first bin that has enough space to hold the item.

(a) Prove that FF has competitive ratio 2.

(b) Give a sequence of item sizes for which the competitive ratio of FF is no better than $\frac{3}{2}$.

Exercise 4: Parallel Algorithms

Give a PRAM algorithm that uses the all-prefix-sums operation to find the position of the first positive entry in an array of n integers. What is the running time of your algorithm if you use p processors for the computation?