University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
P. Bamberger

Algorithm Theory
Exercise Sheet 7

Exercise 1: Load Balancing *(2+6+5+5+4 Points)*

Recall the load balancing problem from the lecture: Given m machines, n jobs and for each job i a processing time t_i, we want to assign each job to a machine such that the makespan (largest total processing time of any machine) is minimized. We have seen that the *modified greedy* algorithm, in which we go through the jobs by decreasing length and assign each job to the machine that currently has the smallest load, has an approximation ratio of 3/2.
In this exercise, we want to prove that the algorithm has an even better approximation ratio.

Assume we have n jobs with lengths $t_1 \geq t_2 \geq \cdots \geq t_n$. Let T be the makespan of the greedy solution and let i be a machine with load T. Further, let $\hat{n}$ be the last job that is scheduled on machine i.

(a) Shortly argue why it is sufficient to ignore jobs $\hat{n}+1, \ldots, n$ and instead prove the desired ratio between greedy and optimal for jobs $1, \ldots, \hat{n}$.

(b) Show that if an optimal solution for jobs $1, \ldots, \hat{n}$ assigns at most two jobs to each machine, the algorithm computes an optimal solution.

Hint: Think of a "canonical" way to assign at most two jobs to each machine and show that $T \leq T_{canonical} \leq T_{opt}$.

(c) Show that therefore, either $t_{\hat{n}} \leq T_{opt}/3$ or the greedy algorithm computes an optimal solution.

(d) Conclude that the algorithm has an approximation ratio of at most 4/3.

(e) Show that the 4/3 bound is tight, i.e., there is a sequence of instances for which the ratio between Greedy and OPT converges to 4/3.
Hint: Consider $2m+1$ *jobs for* m *machines, three jobs with processing time* m *and two jobs with processing times* $m+1, m+2, \ldots, 2m-1$ *each.*

Exercise 2: Two Knapsacks *(2+6 Points)*

Consider the following variation of the knapsack problem: Given items $1, \ldots, n$ where each item i has a positive integer *weight* $w_i \in \mathbb{N}$ and a positive *value* $v_i > 0$ and **two** knapsacks of capacities W_1 and W_2, we want to pack the items into the knapsacks such that

- for $j \in \{1, 2\}$, the *total weight* of the items in knapsack j is at most W_j.
- The *total value* of the items that are packed in either knapsack is maximized.

(a) Prove that this problem is not equivalent to the standard knapsack problem with one knapsack of capacity W_1+W_2 by showing that the total value that can be packed into one knapsack of capacity W_1+W_2 can be *arbitrarily* larger than the total value that can be packed into two knapsacks of capacities W_1 and W_2.

(b) Assume that $W_1 \geq W_2$. A simple strategy would be to first compute an optimal solution for a knapsack of capacity W_1 and afterwards, with the remaining elements, an optimal solution for a knapsack of capacity W_2. Show that this algorithm always computes at least a 2-approximation for the problem.

Exercise 3: Vertex Cover Approximation *(4 Points)*

Show that taking all nodes is a 2-approximation algorithm for the vertex cover problem in regular graphs (graphs where all nodes have the same degree)