max planck institut informatik

SAARLAND UNIVERSITY

COMPUTER SCIENCE

Mayank Goswami and Andreas Wiese **Winter 2015/2016**

Exercises for Approximation Algorithms

www.mpi-inf.mpg.de/departments/algorithms-complexity/teaching/winter15/approx

Tutorials: Andreas Schmid

Exercise Sheet 2 Due: **25.11.2015**

Your homework must be handed in on Wednesday at the beginning of the tutorial.

You need to collect at least 50% of all points over all exercise sheets. You are allowed to work on these exercises in groups but every student has to hand in his/her own write-up.

Exercise 1 (*10+5 points*)

Show that the following scheduling problems can be solved optimal in polynomial time. In both problems the objective is to minimize the makespan of the computed schedule.

a) $P|r_j, p_j = 1|C_{max}$: We want to schedule n jobs on m identical machines. All jobs have unit processing time ($p_j = 1, \forall j \in J$) and come with a positive integer release date r_j before which they can not be processed.

b) **BONUS** $R|p_j \in \{1, \infty\}|C_{max}$: We now want to schedule n jobs on m unrelated machines, where a job j takes time $p_{i,j} \in \{1, \infty\}$ if it is processed on machine $i = 1, \dots, m$.

Exercise 2 (*10 points*)

In this exercise we revisit $P|r_j, p_j = 1|C_{max}$. The problem imminently becomes harder, if the processing times are allowed to take arbitrary values. Give a 3-approximation algorithm for the makespan minimization problem when for any job j, p_j is a positive integer.

Hint: Try to think of a lower bound for OPT using the given release dates.

Exercise 3 (*10 points*) [Shmoys-Williamson Exercise 3.3]

Consider the following scheduling problem: there are n jobs to be scheduled on a single machine, where each job j has a processing time p_j, a weight w_j, and a due date d_j,$j = 1 \dots n$. The objective is to schedule the jobs so as to maximize the total weight of the jobs that complete by their due date.

a) Prove that there always exists an optimal schedule in which all on-time jobs complete before all late jobs, and the on-time jobs complete in an earliest due date order; use this structural result to show how to solve this problem using dynamic programming in $O(nW)$ time, where $W = \sum_j w_j$.

b) Use the result of a) to derive a fully polynomial-time approximation scheme.

Exercise 4 (*10 points*)

In the lecture we have seen a 2-approximation algorithm for the makespan minimization problem on m unrelated machines and shown that it is NP-hard to approximate the problem to a factor of $(\frac{3}{2} - \epsilon)$ for any $\epsilon > 0$. The problem becomes easier if we assume that the number of machines m is a constant. Give a $PTAS$ for the makespan minimization problem on a constant number of unrelated machines.