Prof. Dr. Nicole Megow
Dr. Felix Hommelsheim
Dr. Alexander Lindermayr
Bart Zondervan

Summer 2025

Advanced Algorithms

Exercise Sheet 8

Submission: Monday, June 9 at 11:59 am.
This exercise will be discussed Wednesday, June 11

Exercise 8.1 (Single-Machine scheduling with Release Dates) (7 Points)

Consider the preemptive single-machine scheduling problem $1|r_j, \text{pmtn}|\sum C_j$, in which a set of jobs, each with a processing time $p_j \geq 0$ and a release date $r_j \geq 0$ must be scheduled preemptively on a single machine with the objective to minimize the total completion time.

Show that the *Shortest Remaining Processing Time (SRPT)* rule is optimal. SRPT schedules at any moment in time the job with the shortest remaining processing time.

Exercise 8.2 (Makespan Scheduling via LP) (7 Points)

We have seen in class that the scheduling problems $P|\text{pmtn}|C_{\max}$ and $P|\text{pmtn}, r_j|C_{\max}$ can be solved optimally in polynomial time. Recall that in these problems, we are given a set of jobs J, each with processing time $p_j \geq 0$ and possibly a release date $r_j \geq 0$, and m identical parallel machines to execute these jobs. The task is to find a feasible schedule, possibly using preemption, such that the makespan is minimized.

Show how to solve the problem with release dates using linear programming (LP). *Hint:* Formulate the subproblem of assigning a feasible amount of jobs' processing to time intervals via an LP, given that you knew the optimal makespan.

Exercise 8.3 (Assignment Problem) (6 Points)

In class we have reduced the unrelated-machine scheduling problem $R||C_{\max}$ to the problem of finding a perfect matching of minimum cost.

Model the scheduling problem as a integer linear program. Argue that your formulation is correct.

The following exercises are no homework and shall be discussed in presence.

Exercise 8.4 (Maximum Lateness Scheduling)

Consider the following single-machine scheduling problem $1||L_{\max}$, in which a set of jobs, each with a processing time $p_j \geq 0$ and a due date $d_j \geq 0$ must be scheduled non-preemptively on a single machine. The goal is to minimize the maximum *lateness* $L_{\max} := \max_j L_j$, where the lateness of a job j is defined as $L_j = \max\{0, C_j - d_j\}$.

Show that the algorithm that schedules jobs in non-decreasing order of due dates computes an optimal solution.