Exam Sequencing and scheduling, 2P450,
January 18, 2010, 14:00-17:00

This test contains four questions on two pages, with 10 sub-items, for which a total of 50 points can be scored. Each good answer scores 5 points The final grade is obtained by dividing the amount of points by five. Questions may be answered in English and/or in Dutch. Please explain your answers. You are NOT allowed to use the handouts distributed in class.

Question 1. Consider a problem of processing 100 jobs using 10 identical machines, minimizing makespan ($P10||C_{\max}$). We choose to build a schedule by use of the LPT-rule.

(a) Give a short description of the rule, and give the best worst-case ratio on the performance of this algorithm (no proof required);

(b) Now we are given the following additional information: the processing times of the jobs add up $500'000'000'000$, and the largest and smallest job lengths are $9'833'673'362$ and $5'681'271$ respectively. Provide a tighter bound on the outcome of the LPT-rule when applied to this instance.

Question 2. Consider the following flowshop-like problem: we have m machines, and n jobs that need to be processed. Job j consists of subtasks T_{ij}, for $i = 1, \ldots, m$, needing processing on machine M_i for a period of time p_{ij}. The flowshop characteristic entails that T_{ij} must be finished before $T_{(i+1)j}$ can be processed. A machine can handle one job at a time, and tasks cannot be preempted. The objective is to minimize the *average* job-completion $\frac{1}{n}\sum_{j=1}^{n} C_{mj}$.

(a) Prove that when looking for an optimal schedule, we may restrict ourselves to schedules in which the job order on the first machine is equal to the job order on the second machine;

(b) Prove — or — disprove by an counterexample, that when looking for an optimal schedule, we may restrict ourselves to schedules in which the job order on the last machine is equal to the order on the one but last machine;

(c) Consider the flow shop on four machines, with the regular objective of minimizing makespan. Is there always an optimal solution in which the job-orders on all machines are equal to the order on the first machine? If yes, give a proof; if not, give a counter-example.

Question 3. Consider the following instance of the problem of minimizing makespan in an OPEN shop environment. We have four jobs and four machines, with processing times given by the following table:

$j \setminus m$	1	2	3	4
1	8	3	12	4
2	4	10	8	2
3	8	5	7	10
4	11	7	9	4

(a) Let us allow for *preemption*. Describe briefly how to find, in general, for this type of problem a schedule in which makespan is minimized; for an instance with n jobs and m machines, give a non-trivial upper bound on the number of major iterations.

(b) Find using this method an optimal schedule, for the four by four case, allowing for preemption.

(c) Now let us forbid preemption, and only schedule the operations on machines 1 and 2, so that the data becomes:

$j \setminus m$	1	2
1	8	3
2	4	10
3	8	5
4	11	7

Find an optimal schedule minimizing makespan.

Question 4. In class we used the problem called three-partition as an example of a hard problem: *Given a number b and a sequence of numbers $a_1, a_2, \ldots, a_{3k} \in \mathbb{N}$ with properties $\frac{b}{4} < a_i < \frac{b}{2}$, for all i, and $\sum_{i=1}^{3k} a_i = kb$; decide whether or not there exists a partition of the a_i into k sets each holding three a_i's that sum up to b.*

Use this to prove that the problem of deciding whether for a single machine, and a set of jobs with processing times p_j, release dates r_j and due dates d_j, there exists a schedule with maximum lateness $L_{\max} \leq 0$. Here all numbers involved are positive integers.

(a) For an arbitrary instance of three-partition define an appropriate instance of the $L_{\max}$ problem.

(b) Prove that either both instances have a YES-answer, or both have a NO-answer.