Exam Sequencing and scheduling, 2P450,
April 12, 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 n jobs using m identical machines, minimizing makespan ($Pm||C_{\max}$). We choose to build a schedule by use of the LS-rule (List-Schedule).

(a) Give a short description of the rule, and give the best worst-case ratio on the performance of this algorithm ;

(b) Prove that the above analysis is best possible by providing, for every $m > 1$, an instance I_m, for which the LS-approach yields a result matching the upper bound on the ratio.

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 $i = 1, \dots, m$, 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 maximum job-completion-time $\max_{j=1}^{n} C_{mj}$.

(a) Suppose that $m = 3$. 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 and third machine; i.e. we my restrict he search for an optimal schedule to so-called *permutation*-schedules.

(b) Suppose that $m = 2$. Explain Johnson's algorithm to find the optimal schedule.

(c) Next suppose $m = 4$. 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 *lateness* in a single machine environment. Here, processing times, release dates and due dates are given by p_j, r_j and d_j, respectively.

j	1	2	3	4	5	6
p_j	8	3	12	4	5	3
r_j	4	10	8	2	14	18
d_j	-1	-11	-8	-27	-13	-15

(a) Let us allow for *preemption*. Describe briefly how to find, in general, for this type of problem a schedule in which lateness is minimized;

(b) Find using this method an optimal schedule, for this instance, allowing for preemption.

(c) Explain why this type of scheduling problem (especially with $d_j < 0$) is of importance when trying to solve the general job-shop problem $J||C_{\max}$.

Question 4. In class we used the problem called the balance-problem as an example of a hard problem: *Given a set of items of weight $w_1, \ldots, w_n$, with $w_i \in \mathbb{N}$; decide whether there is a subset $S \subset \{1, \ldots, n\}$, with the property that the total weight of items in S equals the total weight of items outside S.*

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 the balance-problem 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.