Exam Sequencing and scheduling, 2P450,
May 11 2006, 14:00-17:00

This test contains five questions on two pages, with 11 sub-items, for which a total of 50 points can be scored. 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.

Each good answer scores 5 points except for sub-questions 2(a): 2 points, 2(b): 4 points, and 2(c): 4 points.

Question 1. The BALANCING-problem is an NP-complete problem used in class as an example of an intrinsically 'hard' problem. It is defined as follows: we are given a natural number n, n natural numbers $a_1, \ldots, a_n$, and a natural number b with the property that $\sum_{i=1}^{n} a_i = 2b$. Does there exist a subset S of $\{1, \ldots, n\}$, such that $\sum_{i \in S} a_i = b$?

(a) Formulate the minimum makespan problem on three parallel identical machines (P3).

(b) Prove that P3 is also NP-complete ('hard'), by showing that one of the problems is a special case of the other problem.

Question 2.

(a) For minimization of the makespan (length) of a schedule for n jobs $J_1, \ldots, J_n$ on m parallel identical machines $M_1, \ldots, M_m$ we can use the so-called LPT-rule. Explain this rule.

(b) This LPT-rule does not always provide an optimal schedule! Prove that for two machines, the LPT-result is never worse than 7/6 times the best solution. Give an instance for which LPT may give such a bad result.

(c) Now suppose that in addition, each job J_j has a weight $w_j > 0$, and we really would like to minimize total sum of weighted completion times $\sum_j w_j C_j$. Starting from the solution obtained by LPT, how should we rearrange jobs on each machine?

Question 3.

(a) For minimization of the makespan (length) of a schedule for n jobs $J_1, \ldots, J_n$ on parallel machines $M_1, \ldots, M_m$, where each machine has its own speed, and where we allow preemption, give a closed form expression for the optimal makespan in terms of the processing requirements p_j, and the speeds s_i.

(b) For the following instance with six jobs, with processing requirements 40, 60, 80, 100, 370 and 400, and with four machines with speeds 1,2,3,4 respectively, compute the minimum makespan, and a schedule with this makespan.

Question 4.

(a) Explain how the **two**-machine flowshop can be optimally solved in polynomial time. Give the number of elementary calculations to solve the problem, as a function of the number of jobs n.

(b) Compute a schedule of minimum makespan for the following two-machine flowshop

	machine	
job	M_1	M_2
J_1	5	6
J_2	5	4
J_3	4	5
J_4	2	1
J_5	4	3

Why should I believe it is optimal?

Question 5.

(a) Consider a single machine scheduling problem, with job J_j having processing time $p_j > 0$, and where we want to minimize *the sum of squares of completion times* $\sum_{j=1}^{n}(C_j)^2$. Explain how to find an optimal schedule in polynomial time. Prove that your algorithm returns a truely optimal solution.

(b) Explain how, for the problem of minimizing makespan for n jobs on two parallel machines, a schedule can be found in polynomial time, that is guaranteed to be within 10% of the optimal solution.