Exam Sequencing and scheduling, 2P450,
January 10, 2008, 09:00-12:00

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. The problem of minimizing late jobs is defined as follows. We are given a set of jobs J, with, for job $j \in J$ a processing time $p_j \in \mathbb{N}$ and a due date $d_j \in \mathbb{N}$. We consider a schedule of these jobs on a single machine, and call a job *late* if its completion time C_j exceeds the due date: $C_j > d_j$. The objective is to minimize the number of late jobs.

(a) Formulate the EDD-rule and prove that if J is *feasible*, that is, if there is exists a schedule σ with $C_j^\sigma \leq d_j$, for all $j \in J$, then each EDD-schedule also schedules each job in time.

(b) Now consider a subset of jobs $S \subseteq J$ such that $\max_{j \in S} p_j \leq \min_{j \notin S} p_j$, and such that S is feasible. Prove that if K is a maximum feasible subset of J, then there exists a feasible subset K' of J of equal size ($|K| = |K'|$) that moreover includes S, i.e. $S \subseteq K'$.

Question 2. We consider the on-line version of the problem of minimizing the makespan (length) of a schedule for n jobs $J_1, \ldots, J_n$ on m parallel identical machines $M_1, \ldots, M_m$, where information about a job j (existence and processing time) is known at its release time r_j. We use the following approximation algorithm. At time 0, let $J(0)$ be the set of jobs known at time zero, that is those jobs with release time $r_j \leq 0$. Schedule the jobs in $J(0)$ optimally with respect to makespan. Let $C(0)$ denote the (optimal) makespan for $J(0)$. Now although new jobs become available during the interval $(0, C(0)]$ we stick to the schedule for $J(0)$. Then, at time $C(0)$, let $J(1)$ denote the set of jobs that have become known. If at time $C(0)$ no jobs have arrived, we wait until the first moment that a new job is being released. We now schedule optimally the jobs in $J(1)$, starting from time $C(0)$, and so on.
Let $J(k)$ denote the kth batch of jobs that we schedule in this way, and let $L(k)$ denote the optimal makespan for a schedule for jobs $J(k)$ only, for $k = 0, \ldots, K$, when released at time 0.

(a) Let OPT denote the overall (off-line) optimum makespan (for all jobs $J_1, \ldots, J_n$). Let $T(k)$ denote the completion time of batch $J(k)$. Show that $\text{OPT} \geq T_{k-2} + L_k$, for all $k = 2, \ldots, K$.

(b) Further argue that $\text{OPT} \geq L_{k-1}$, for all $k = 2, \ldots, K$, and that $\text{APP} \leq 2\text{OPT}$, where APP denotes the makespan resulting from the approximation.

Question 3. Consider the following 4 by 4 job shop problem on four machines, where each of four jobs consists of four operations with processing times and dedicated machines given by the following tables:

job	operations			
	1	2	3	4
1	23	17	17	5
2	17	8	8	11
3	14	14	5	29
4	17	14	14	8

proces times

job	operations			
	1	2	3	4
1	1	3	4	2
2	2	3	1	4
3	3	4	1	2
4	4	1	3	2

machines

(a) Let us assume that in an initial schedule machine 1 handles the jobs in order $1, 2, 3, 4$; machine 2 processes them in order $2, 3, 1, 4$; machine 3 in order $3, 2, 1, 4$; and machine 4 processes the jobs in order $4, 3, 1, 2$. Describe which operations form the *critical path* and give the length of the critical path.

(b) If in the general case we want to (try to) improve an existing schedule, which pairs of operations may be good candidates for swapping? Argue that swapping the pair of your choice does not lead to inconsistent operation orders (that is to cycles in the resulting precedence graph). Swap the first pair along the critical path. Does it lead to an improvement or to a worse solution?

(c) Provide four lower bounds on the optimal makespan by solving a (preemptive) head-body-tail problem per machine.

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	6	7
J_2	8	5
J_3	4	5
J_4	2	1
J_5	4	3
J_6	1	3

Why should I believe it is optimal?

(c) Give an example of a **four**-machine flow shop problem in which the best *permutation schedule* has a larger makespan than the best arbitrary schedule.