Exam Sequencing and scheduling, 2P450,
January 24, 2014, 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/or on the web.

Question 1. Consider the use of the LPT-rule for solving the problem of minimizing makespan on m parallel identical machines processing n jobs. We observe the following

LEMMA: for any instance I of $P||C_{\max}$: if there exists an optimal schedule with at most two operations per machine, then LPT when applied on I yields a schedule that has optimal length.

You do not have to prove this lemma!! Give the worst case ratio between the makespans $C_{\max}^{\mathrm{LPT}}(I)$ and $C_{\max}^{\mathrm{OPT}}(I)$ in terms of m and prove that ratio making use of the above Lemma. The answer should contain the following elements:

(a) State the LPT-rule, and state the worst case ratio in terms of a function of m and/or n,

(b) Prove that this expression is an upper bound on the ratio, and provide a family of examples from which it is clear that the analysis is as tight as possible.

Question 2. Consider the problem $1|r_j|\sum_j C_j$ of minimizing total completion time on a single machine under release dates. In particular consider the instance given by

jobs j	1	2	3	4	5	6	7
p_j	8	18	12	10	10	17	16
r_j	0	6	13	14	25	25	50

(a) Formulate and apply an appropriate rule ρ for solving the PREEMPTIVE case. Compute the optimal *preemptive* schedule, and list the results in a table with the following entries:

jobs j	1	2	3	4	5	6	7
C_j^ρ							

(b) Formulate the concept of an α-schedule and compute it for $\alpha = 0.5$. Store the completion times in a table as

jobs j	1	2	3	4	5	6	7
C_j^α							

(c) State, for this particular instance, the optimal non-preemptive schedule and give an (ad-hoc) reasoning why it should be optimal.

Question 3. Consider the problem $(1||\sum_j w_j U_j)$ of minimizing the weighted number of late jobs.

(a) Describe for the general case, how to find the optimal solution, by the use of *Dynamic Programming.*

(b) In particular we may consider the instance given by the following parameters

jobs j	1	2	3	4	5	6	7
p_j	6	18	12	10	10	17	16
d_j	10	26	13	14	25	25	50
w_j	1	2	3	4	5	6	7

Apply the method described in (a) to find the optimal schedule for this instance.

(c) If in the above setting all weights are equal to one, what would then be an easy rule for the general case, and what would be the optimal schedule for the instance with modified weights:

jobs j	1	2	3	4	5	6	7
p_j	6	18	12	10	10	17	16
d_j	10	26	13	14	25	25	50
w_j	1	1	1	1	1	1	1

Question 4. Consider the problem of scheduling jobs on m machines, where each job consists of m subtasks, and where for each job j its i-th sub-task has to be processed on machine i. Here the processing time required is p_{ij}. We consider as objective function $\sum_j C_{mj}$, where C_{jm} denotes the completion time of the last subtask of job j (hence on machine m).

(a) Prove that there exists an optimal schedule in which the jobs are processed on machine 2 in the same order as on machine 1.

(b) Prove or disprove: there always exists an optimal schedule in which the jobs are processed on machine m in the same order as on machine $m-1$.