Planning & Scheduling
Assignment 3

Lavinia Pulcinella
Student ID: i6233926

May 2020

1 Graded Exercise 3.3

Consider the scheduling problem of n jobs on m parallel identical machines, where preemptions are allowed, with the goal of minimizing the sum of completions times, i.e.,$\sum_j C_j$.

A parallel identical machines setup implies identical machines such that every job j requires p_j units of processing time when processed on any machine. Allowing for preemptions implies that a job can be processed in several time intervals, on different machines, provided that the given job, say job j, doesn't overlap with any of its preempted intervals.

Formally, for every job j we can have a set of intervals $[I_j^{(1)}, ..., I_j^{(n)}]$ such that their total length is p_j and such intervals don't overlap.

Show that if $p_i < p_j$, then job is finished before job j in an optimum schedule. Use an exchange argument. Be aware that the optimum may use preemptions.

The statement is essentially asking us to prove that a preempted optimal schedule for minimum total completion time can be obtained by scheduling jobs according to the shortest processing time (SPT) rule. We know that the SPT rule for minimum total completion time in one machine is optimal. Since we are in the context of m identical and parallel machines, which allow for preemption, we should prove that the SPT rule can be extended to the case of m machines with preemptions.

Suppose that for the two jobs i and j there exist an optimal schedule such that$p_i < p_j$ and that job i finishes after job j i.e. $C_i > C_j$. Consider now the job partition of job i such that $[I_i^{(1)}, ..., I_i^{(n)}]$ and we divide it into two parts such that intervals $[I_i^{(1)}, ..., I_i^{(k)}]$ precede C_j and $[I_i^{(k+1)}..I_i^{(n)}]$ follow C_j.

Since $p_j > p_i$ (by assumption) we should be able to exchange $[I_i^{(k+1)}..I_i^{(n)}]$ with some of job j intervals, say $[I_j^{(l+1)}...I_j^{(n)}]$ making sure there is no overlap between the two parts of job i, namely $[I_i^{(1)}, ..., I_i^{(k)}]$ and $[I_i^{(k+1)}..I_i^{(n)}]$.

It follows that the completion time for job i is then decreased by $C_i - C_j$ while job j's is increased by the same amount and thus resulting in a schedule in which job i is completed before job j. Note that all other jobs remain with the same completion time. This is due to the fact that the exchange involves jobs i and j only and we only need to observe the condition such that the intervals of the same job never overlap since two machines cannot process the same job at the same time.