Moore-Hodgson: Minimizing the number of late jobs

The following algorithm due to Moore and Hodgson schedules jobs on a single machine minimizing the number of late jobs:

1. sort jobs in order of increasing due date: \( d_j \);
2. start with scheduled job set \( J_0 = \emptyset \), load \( \lambda = 0 \);
3. for \( j = 1, \ldots, n \), if \( \lambda + p_j \leq d_j \), then \( J_j = J_{j-1} \cup \{ j \} \); \( \lambda = \lambda + p_j \); otherwise, let \( j_{\text{max}} \in J_{j-1} \cup \{ j \} \) have largest processing time; set \( J_j = J_{j-1} \cup \{ j \} \setminus \{ j_{\text{max}} \} \); \( \lambda = \lambda + p_j - p_{j_{\text{max}}} \).
4. Schedule jobs in \( J_n \) in order of due date; discard jobs not in \( J_n \) or schedule them in any order after the jobs in \( J_n \).

Claim: Moore-Hodgson yields a schedule with a minimum number of late jobs.

Proof: assume the jobs are already in due date order. Then the claim is equivalent to: (*) for each \( k = 1, \ldots, n \) \( J_k \) is a maximum cardinality feasible subset of \( S(k) := \{ 1, \ldots, k \} \). We prove a slightly stronger statement: (***) for each \( k = 1, \ldots, n \) \( J_k \) is a maximum cardinality feasible subset of \( S(k) := \{ 1, \ldots, k \} \), and among all maximum cardinality feasible sets, \( J_k \) has smallest total length.

To prove (***) let \( N(k) \) denote the true maximum cardinality of a feasible subset of \( S(k) \), and let \( F_k \) denote such a maximum cardinality set of minimum total length. Here feasible means that the subset can be scheduled in time, which can be tested by checking the EDD-schedule for \( F_k \). Assuming that (***) is not true, consider the smallest counter-example. Evidently, \( n > 1 \), since for a single job, \( J_1 = \emptyset \) if and only if \( N(1) = 0 \) if and only if \( p_1 > d_1 \).

By minimality we have that \( |J_{n-1}| = N(n-1) \) and \( p(J_{n-1}) = p(F_{n-1}) \). Note that
\[
|J_{n-1}| \leq |J_n| \leq |J_{n-1}| + 1,
\]
and similarly, \( N(n-1) \leq N(n) \leq N(n-1) + 1 \), and furthermore \( |J_n| \leq N(n) \). As (***) is not true we must have

(a) \( |J_{n-1}| = |J_n| \) and \( N(n) = N(n-1) + 1 \), or
(b) \( |J_n| = N(n) \) but \( J_n \) is not of minimum total length.

If we are in case (a), then there set \( F_n \) is of size \( N(n) \) and contains job \( n \). But then \( F_n \setminus \{ n \} \) has size \( N(n-1) \) and has total length at least that of \( J_{n-1} \). \( F_n \) is feasible, hence its EDD-schedule is feasible. It ends with job \( n \), which means that \( J_{n-1} \cup \{ n \} \) is also feasible. Hence \( J_n = J_{n-1} \cup \{ n \} \) contradicting (a).

If we are in case (b) and moreover \( N(n) = N(n-1) + 1 \), then \( n \in J_n \), and \( n \in F_n \) with \( |F_n| = |J_n| \), and \( p(F_n) < p(J_n) \). But then \( p(F_n \setminus \{ n \}) < p(J_{n-1}) \), contradicting the minimum length of \( J_{n-1} \).

If we are in case (b) and \( N(n) = N(n-1) \), then \( |J_n| \leq |F(n)| = |J_{n-1}| \). So insertion of \( n \) was followed by deletion of some job \( k \), and so \( p(J_n) \leq p(J_{n-1}) = p(F_{n-1}) \). Now it follows from \( p(F_n) < p(J_n) \), that \( n \in F_n \). Now, let \( j_{\text{max}} = \arg \max \{ p(j) \mid j \in J_{n-1} \cup \{ n \} \} \), and let \( j_1 = \max \{ j \mid j \in J_{n-1} \setminus F_n \} \). Then \( p(F_n \cup \{ j_1 \} \setminus \{ n \}) = p(F_n) + p(j_1) - p(n) < p(J_n) + p(j_1) - p(n) \leq p(J_n) + p(j_{\text{max}}) - p(n) = p(L_{n-1}) = p(F_{n-1}) \).

Note that by definition of \( j_1 \), the set \( J' \) of all jobs in \( J_{n-1} \) higher than \( j_1 \) belongs to \( F_n \) as well. From the schedule for \( F_n \) remove job \( n \) and jobs \( J' \), process remaining jobs as early as
possible, next process job $j_1$ and then jobs $J'$ in EDD order. Then the latter jobs complete earlier than they do in the schedule for $J_{n-1}$, as $p(F_n) + p(j_1) - p(n) < p(J_{n-1})$. So the set $F_n \cup \{j_1\} \setminus \{n\}$ is feasible, contradicting the minimality of $p(F_{n-1})$. 