J. Appl. Prob. **29**, 667–681 (1992)
Printed in Israel

ON THE OPTIMALITY OF LEPT AND $c\mu$ RULES FOR MACHINES IN PARALLEL

CHENG-SHANG CHANG,* *T. J. Watson Research Center*
XIULI CHAO,** *New Jersey Institute of Technology*
MICHAEL PINEDO,*** *Columbia University*
RICHARD WEBER,**** *University of Cambridge*

Abstract

We consider scheduling problems with m machines in parallel and n jobs. The machines are subject to breakdown and repair. Jobs have exponentially distributed processing times and possibly random release dates. For cost functions that only depend on the set of uncompleted jobs at time t we provide necessary and sufficient conditions for the LEPT rule to minimize the expected cost at all t within the class of preemptive policies. This encompasses results that are known for makespan, and provides new results for the work remaining at time t. An application is that if the $c\mu$ rule has the same priority assignment as the LEPT rule then it minimizes the expected weighted number of jobs in the system for all t. Given appropriate conditions, we also show that the $c\mu$ rule minimizes the expected value of other objective functions, such as weighted sum of job completion times, weighted number of late jobs, or weighted sum of job tardinesses, when jobs have a common random due date.

EXPONENTIAL DISTRIBUTION; MAKESPAN; PRIORITY POLICIES; STOCHASTIC SCHEDULING; WEIGHTED FLOWTIME

AMS 1991 SUBJECT CLASSIFICATION: PRIMARY 90B35

1. Introduction

Consider m machines that operate in parallel and are subject to breakdown and repair. Their up times and down times are arbitrarily distributed. The machines are used to process n jobs, whose release dates, $R_1, \cdots, R_n$, are also arbitrarily distributed. Job j has a processing time P_j, that is an exponentially distributed random variable with rate μ_j.

Received 6 December 1989; revision received 28 June 1991.
* Postal address: IBM Research Division, T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, USA.
** Postal address: Division of Industrial and Management Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA.
*** Postal address: Department of Industrial Engineering and Operations Research, Columbia University, New York, NY 10027, USA.
The research of this author was partially supported by the NSF under grant ECS 86-14689.
**** Postal address: Cambridge University Engineering Department, Mill Lane, Cambridge, CB2 1RX, UK.

Each processing time is distributed independently of all other random variables in the model. Without loss of generality, we make the following assumption.

(A1) $$0 \leqq \mu_1 \leqq \mu_2 \leqq \cdots \leqq \mu_n.$$

Let $Q_{j,t}$ be the indicator function for the event that job j has been released by time t but is not yet complete, i.e. $Q_{j,t}$ is 1 or 0 as job j is or is not in the system at time t. Define $\boldsymbol{Q}_t = (Q_{1,t}, Q_{2,t}, \cdots, Q_{n,t})$ as the state of the jobs at time t. A cost $g(\boldsymbol{Q}_t)$ is associated with the system at time t, where $g: \{0, 1\}^n \mapsto \mathbb{R}$. This cost depends only on the set of jobs in the system at t.

Our main result is the following theorem.

Theorem 1.1. *Suppose* (*A* 1) *holds. Let* π *be the policy that schedules jobs according to priorities that are decreasing in their indices. Then within the class of preemptive policies,* π *minimizes* $Eg(\boldsymbol{Q}_t)$, *the expected cost at time* t, *for all* t, *and all processes of arrivals, machine breakdowns and repairs, if and only if the cost function* $g(\boldsymbol{x})$ *satisfies*

(A2) $$g(\boldsymbol{x}) \geqq g(\boldsymbol{x} - e_\beta),$$

(A3) $$\mu_\alpha g(\boldsymbol{x} - e_\alpha) \geqq \mu_\beta g(\boldsymbol{x} - e_\beta) + (\mu_\alpha - \mu_\beta) g(\boldsymbol{x}) \qquad \textit{for all } \alpha > \beta,$$

where e_j *is a row vector of* n *components, whose* j*th component is* 1 *and other components are* 0.

Note that since (A1) is assumed to hold, π is equivalent to the Longest Expected job Processing Time (LEPT) first rule. This rule is well known to stochastically minimize the *makespan* of jobs with exponentially distributed processing times that are processed on parallel machines; moreover, it is known to hold for an arbitrary arrival process (see Van Der Heyden (1981), Frederickson et al. (1981), Weiss (1982) and Weber (1982), (1983)). This result reappears as an application of Theorem 1.1 at the start of Section 2. Our proof of Theorem 1.1 is based on certain coupling techniques and is similar to the approach of Van Der Heyden. In Section 2 we shed some light on conditions (A2)–(A3), by providing further examples of cost functions for which they hold.

In Section 3, we consider the special case $g(\boldsymbol{x}) = \sum_{j=1}^n c_j x_j$. This is the case of weighted holding cost, in which a cost c_j is incurred for each unit time job j remaining in the system. It is easy to check that this cost function satisfies (A2)–(A3) *if and only if*

(A4) $$\mu_1 c_1 \geqq \mu_2 c_2 \geqq \cdots \geqq \mu_n c_n \geqq 0.$$

Condition (A4) says that the order of increasing expected processing times is the same as the order of increasing values of $c\mu$. So if both (A1) and (A4) hold, π and LEPT are the same and equivalent to the $c\mu$ rule, which is defined as the preemptive rule that always processes a set of jobs whose values of $c\mu$ are greatest. In the literature, a combination of conditions such as (A1) and (A4) is often called an 'agreeability condition'. In Section 3 we show that the $c\mu$ rule minimizes various objective functions, including

(i) the expected weighted number of jobs in the system at an arbitrary time t;

(ii) the expected weighted sum of job completion times;

(iii) the expected weighted number of late jobs when the jobs have a common random due date;

(iv) the expected weighted sum of job tardinesses when the jobs have a common due date.

The LEPT and $c\mu$ rules have received a great deal of attention in the stochastic scheduling literature. If there is only a *single machine*, then there is no need for an agreeability condition. Various authors have considered the single-machine case and shown that the $c\mu$ rule minimizes objective functions such as (i)–(iv) above; see Pinedo (1983), Baras et al. (1985), Buyukkoc et al. (1985) and Shanthikumar and Yao (1991).

Optimality of the $c\mu$ rule for parallel machines requires an agreeability condition. Ross (1983) considered two machines in parallel, n jobs available at time 0 and no arrivals afterwards. For this setup he showed that under the agreeability conditions (A1) and (A4), the $c\mu$ rule minimizes the expected weighted sum of completion times. Kämpke (1987) extended Ross's result to m machines, and weakened the agreeability conditions to $c_1 \geqq \cdots \geqq c_n$ and (A4). Under the agreeability conditions $c_1 \geqq \cdots \geqq c_n$ and $\rho_1(t_1)c_1 \geqq \cdots \rho_n(t_n)c_n$ for all $t_1, \cdots, t_n$, Weber (1988) generalized Kämpke's result to models in which the job processing times have hazard rates, $\rho_1(t), \cdots, \rho_n(t)$. Corollary 3.1 in this paper generalizes Ross's result to m machines and an arbitrary arrival process.

For more general cost functions, conditions (A2) and (A3) arise quite naturally. They have been considered by Weiss and Pinedo (1980), who investigated similar scheduling problems to those in this paper, but under the assumption that all jobs are present at the start. They showed that a policy minimizes the total holding cost incurred by time t if the expected total cost under that policy, say $G(\boldsymbol{x})$, satisfies (A2)–(A3), with g replaced by G. They stated conditions on g that would be sufficient to guarantee optimality of the LEPT rule; however, these were incomplete and corrected by Kämpke (1987), (1989) through the addition of (A3). (We note that Kämpke was concerned with general list policies, not only LEPT. He therefore also required a submodularity condition that is not needed in Theorem 1.1.) The contribution of the present paper is to show that conditions (A2)–(A3) are sufficient to guarantee optimality of LEPT when there are arrivals and machine breakdowns and repairs. Also, conditions (A2)–(A3) are necessary, in the sense that they must hold if LEPT is to be optimal for all t and for every possible process of machine breakdown and repair. Our results also apply to models in which machines have different speeds, since this scenario can be approximated by rapidly alternating periods of breakdown and repair. This generalization has previously been made by Weiss and Pinedo, and also Kämpke.

2. The optimality of the LEPT rule

To shed some light on the conditions (A2)–(A3) in Theorem 1.1, we first provide some examples that satisfy these conditions. We then prove Theorem 1.1.

Example 2.1. Consider the indicator function

$$g_j(\boldsymbol{x}) = \mathbf{1}_{\{\sum_{i=1}^{j} x_i > 0\}}.$$

It is clear that this function satisfies (A2)–(A3). It then follows from Theorem 1.1 that the LEPT rule minimizes $P(\sum_{i=1}^{j} Q_{i,t} > 0)$. Now let Z_j be the completion time of job j. Conditioning on the event $\{R_j = r_j, j = 1, 2, \cdots, n\}$, we note that

$$P\left(\sum_{i=1}^{j} Q_{i,t} > 0\right) = 1 - P(Q_{i,t} = 0, i = 1, 2, \cdots, j)$$
$$= 1 - P\left(\max_{1 \leqq i \leqq j} (Z_i \mathbf{1}_{\{r_i \leqq t\}}) \leqq t\right),$$

where $\mathbf{1}_{\{r_i \leqq t\}}$ equals 1 or 0 as the event $[r_i \leqq t]$ does or does not occur. Considering the case $j = n$, we have that the LEPT rule stochastically minimizes the makespan of jobs arriving before t. Note that since the makespan is larger than t if there is a job arriving after t,

$$P\left(\max_{1 \leqq i \leqq n} (Z_i) \leqq t\right) = P\left(\max_{1 \leqq i \leqq n} (Z_i) \leqq t, t \geqq \max_{1 \leqq i \leqq n} (r_i)\right).$$

Unconditioning the event $\{R_j = r_j, j = 1, 2, \cdots, n\}$ yields that the LEPT rule stochastically minimizes the makespan.

Example 2.2. Let $V_1, \cdots, V_n$, be independent exponential random variables with mean $1/\mu_1, \cdots, 1/\mu_n$. Consider the function $g(\boldsymbol{x}) = \boldsymbol{E}f(V_1 x_1, V_2 x_2, \cdots, V_n x_n)$, where $f: \mathbb{R}_+^n \cup \{0\} \mapsto \mathbb{R}$.

Theorem 2.3. *If f is*

(i) *increasing, i.e. $f(\boldsymbol{z}^1) \leqq f(\boldsymbol{z}^2)$ for all $\boldsymbol{z}^1 \leqq \boldsymbol{z}^2$ componentwise;*

(ii) *arrangement increasing, i.e. $f(\cdots, z_i, \cdots z_j, \cdots) \geqq f(\cdots, z_j, \cdots, z_i, \cdots)$ for all i, j and $z_i \geqq z_j$;*

(iii) *convex in each variable, i.e. $f(\boldsymbol{z} + (\delta_1 + \delta_2)e_i) + f(\boldsymbol{z}) \geqq f(\boldsymbol{z} + \delta_1 e_i) + f(\boldsymbol{z} + \delta_2 e_i)$ for all i and $\delta_1, \delta_2 \geqq 0$;*

(iv) *submodular, i.e. $f(\boldsymbol{z} + \delta_1 e_i + \delta_2 e_j) + f(\boldsymbol{z}) \leqq f(\boldsymbol{z} + \delta_1 e_i) + f(\boldsymbol{z} + \delta_2 e_j)$ for all $i \neq j$ and $\delta_1, \delta_2 \geqq 0$;*

then $g(\boldsymbol{x})$ satisfies (A2)–(A3).

Proof. It is clear that the increasing property of f implies condition (A2). Chang (1992) has shown that (i)–(iv) imply (A3) when f is symmetric. His proof is easily adapted to the case that f is arrangement increasing.

Let $V_{i,t}$ be the remaining work of job i at time t. Since the job processing times are memoryless, $V_{i,t}$ has the same distribution as the quantity $V_i Q_{i,t}$. Using Theorem 2.3, we have the following corollary for the work remaining at time t.

Corollary 2.4. *Suppose (A1) holds and f satisfies (i)–(iv) of Theorem 2.3. Then the LEPT rule minimizes $\boldsymbol{E}f(V_{1,t}, V_{2,t}, \cdots, V_{n,t})$.*

Examples of functions that satisfy these four conditions are (i) $f(\boldsymbol{z}) = \max(a_1 z_n, \cdots, a_n z_n)$, with $a_1 \geqq a_2 \geqq \cdots \geqq a_n$, and $z_i \geqq 0$ for all i, and (ii) $f(\boldsymbol{z}) =$

$\Sigma_{i=1}^{n} h_i(z_i)$, with $h_i(z) \geqq_s h_{i+1}(z)$ for all z (see Definition 3.4) and $h_i(z)$ increasing convex for all i. For more examples, see Chang (1992) and Chang and Yao (1990).

In fact, the LEPT rule not only minimizes the total amount of work at time t, $\Sigma_{i=1}^{n} V_{i,t}$, in *expectation*, but also in the sense of *stochastic ordering*. The argument goes as follows. For a given problem, consider an auxiliary problem in which the number of available machines is truncated to 1 past t. Let M be the makespan of the auxiliary problem. Clearly, the total amounts of work are the same at time t, if in both problems the same scheduling policy has been used up to time t. Moreover, the total amount of work remaining at time t is $\max(M-t, 0)$. Since the LEPT rule stochastically minimizes the makespan for the auxiliary problem, it also stochastically minimizes the total amount of work remaining at time t for the original problem, taking into account that the maximum function is increasing.

The rest of this section is devoted to the proof for Theorem 1.1. The proof takes the same approach as Van Der Heyden. It uses the uniformization technique and an inductive method that has been called 'forward induction' (see Walrand (1988), Section 8.3). Using the well-known uniformization technique, our continuous-time optimization problem is transformed into a discrete-time one. We first show that π is optimal in one step and then take as an inductive hypothesis that π is optimal in $k-1$ steps. A problem with k steps has k decision epochs. Denote the policies applied between these epochs as $(\sigma_0, \sigma_1, \cdots, \sigma_{k-1})$. If we are able to show that $(\pi, \pi, \cdots, \pi)$ is better than $(\sigma, \pi, \cdots, \pi)$, for any admissible policy σ and any initial state, it follows from the induction hypothesis that π is optimal over k steps. As a result of applying different policies π and σ at time 0, the states at step 1 are different. To show that $(\pi, \pi, \cdots, \pi)$ is better than $(\sigma, \pi, \cdots, \pi)$, we must show that starting from time 0 the states reached after applying π are better than those reached after applying σ. Thus, the proof requires a partial ordering amongst the states. The appropriate partial ordering is contained in the following definition. Using it, we can establish inequalities that are similar to (3.3) and (3.4) in Van Der Heyden's paper.

Definition 2.5 (*partial sum ordering*). *Let* $\boldsymbol{x}^i = (x_1^i, x_2^i, \cdots, x_n^i)$, $i = 1, 2$, *be two vectors. We say that* $\boldsymbol{x}^1$ *is smaller than* $\boldsymbol{x}^2$ *under partial sum, and denote this* $\boldsymbol{x}^1 \leqq_{\mathrm{ps}} \boldsymbol{x}^2$, *if* $\Sigma_{j=1}^{l} x_j^1 \leqq \Sigma_{j=1}^{l} x_j^2$, *for all* $l = 1, 2, \cdots, n$.

The partial sum ordering is very similar to the weak majorization ordering (Marshall and Olkin (1979)). However, the weak majorization ordering of x^1 and x^2 requires their components to be in decreasing order. The following property of the partial sum ordering is easy to prove (see Ross (1983), Lemma 3.4). Note that it does not require x_j to be either 0 or 1.

Lemma 2.6. *If* $\boldsymbol{x}^1 \leqq_{\mathrm{ps}} \boldsymbol{x}^2$, $\Sigma_{j=1}^{n} x_j^1 = \Sigma_{j=1}^{n} x_j^2$ *and* $\mu_1 \leqq \mu_2 \leqq \cdots \leqq \mu_n$, *then* $\Sigma_{j=1}^{n} \mu_j x_j^1 \geqq \Sigma_{j=1}^{n} \mu_j x_j^2$.

The next lemma gives a constructive characterization of the partial sum ordering. Consider two integer-valued vectors that are partial sum ordered. We show that the greater of the two vectors can be transformed into the lesser by a number of canonical steps, each of which either reduces some component by 1, or makes a transfer of 1 from a lesser to a greater component. In the first case, we think of 1 as being 'transferred out' of the vector; in the second case 1 is transferred between two components. Informally, we call this the 'transfer property' of the partial sum ordering. A similar property holds for the weak majorization ordering (Muirhead (1903)).

Lemma 2.7. *If $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$ are two integer-valued vectors and $\boldsymbol{x}^1 \leqq_{\mathrm{ps}} \boldsymbol{x}^2$, then $\boldsymbol{x}^2$ can be transformed into $\boldsymbol{x}^1$ by a finite number of applications of the following two types of transfers*:

(i) $\boldsymbol{x} \mapsto \boldsymbol{x} - e_\beta$, *where* $x_\beta > 0$;

(ii) $\boldsymbol{x} \mapsto \boldsymbol{x} + e_\alpha - e_\beta$, *where* $\alpha > \beta$ *and* $x_\beta > 0$.

Moreover, if T is the combination of a finite number of transfers of the forms in (i) and (ii), then $T(\boldsymbol{x}) \leqq_{\mathrm{ps}} \boldsymbol{x}$.

Proof. That $T(\boldsymbol{x}) \leqq_{\mathrm{ps}} \boldsymbol{x}$ is clear. Conversely, the transfer property is clear for $n = 1$, since the partial sum ordering in one dimension means $x_1^1 \leqq x_1^2$, and we need only make reductions of 1 to x_1^2 to obtain x_1^1. Thus, we can take as our induction hypothesis that $\boldsymbol{x}^2$ can be transformed into $\boldsymbol{x}^1$ when these are vectors of $n-1$ components. If $\Sigma_{i=1}^n x_i^1 < \Sigma_{i=1}^n x_i^2$, then there is a constant l such that $\Sigma_{i=1}^{l-1} x_i^2 \leqq \Sigma_{i=1}^n x_i^1 < \Sigma_{i=1}^l x_i^2$. We can use the transfer $\boldsymbol{x} \mapsto \boldsymbol{x} - e_\beta$, $\beta \geqq l$ until $\Sigma_{i=1}^n x_i^1 = \Sigma_{i=1}^l x_i^2$. Thus, we only need to consider the case that $\Sigma_{i=1}^n x_i^1 = \Sigma_{i=1}^n x_i^2$. Note that $\boldsymbol{x}^1 \leqq_{\mathrm{ps}} \boldsymbol{x}^2$ implies $x_n^1 \geqq x_n^2$. If $x_n^1 = x_n^2$, then the induction hypothesis can be applied to the first $n-1$ components. If $x_n^1 > x_n^2$, then we can find a constant l such that $x_l^2 > 0$ and $x_i^2 = 0$, $i = l+1, \cdots, n$ and repeatedly apply the transfer $\boldsymbol{x}^2 \mapsto \boldsymbol{x}^2 + e_n - e_l$ until the nth component of the resulting vector equals x_n^1. Once again, this leaves a case to which the induction hypothesis for $n-1$ applies. This completes the proof.

In the following lemma, we show that the partial sum ordering is preserved under π (the LEPT rule).

Lemma 2.8 (*π-monotone*). *Consider two systems with different initial conditions. Let $Q_{j,t}^i$ be the indicator function for the event that job j is in the system i at time t, and $\boldsymbol{Q}_t^i = (Q_{1,t}^i, Q_{2,t}^i, \cdots, Q_{n,t}^i)$, $i = 1, 2$. Assume (A1) and $\boldsymbol{Q}_0^1 \leqq_{\mathrm{ps}} \boldsymbol{Q}_0^2$. Then there exist two random vectors $\hat{\boldsymbol{Q}}_t^1$ and $\hat{\boldsymbol{Q}}_t^2$ such that under π we have (i) $\hat{\boldsymbol{Q}}_t^i =_{\mathrm{st}} \boldsymbol{Q}_t^i$, $i = 1, 2$, and (ii) $\hat{\boldsymbol{Q}}_t^1 \leqq_{ps} \hat{\boldsymbol{Q}}_t^2$.*

Proof. Let $m(t)$ denote the number of machines available at time t and let $\{M_k, k \geqq 1\}$ be the set of epochs that $dm(t) \neq 0$. At each time M_k, one machine breaks down or one machine becomes available again following its repair. Consider the event $\{R_j = r_j, j = 1, \cdots, n \text{ and } M_k = m_k, k \geqq 1\}$. Let $t_0 = 0$ and $\{t_k, k \geqq 1\}$ be the sequence of $\{r_j, j = 1, \cdots, n$ and m_k, $k \geqq 1\}$ after sorting in time. Clearly, between t_k and t_{k+1} there are no arrivals and the number of machines is constant. First, we show that we can construct two processes between t_0 and t_1 such that they satisfy conditions (i) and (ii) of this lemma. Using the standard coupling and uniformization technique (Keilson (1979)), we generate a Poisson process with rate $\Delta = m(t_0)\mu_n$. Let $\{\tau_k, k \geqq 1\}$ be its arrival epochs and define $\tau_0 = 0$. Generate a sequence of independent and identically distributed (i.i.d.) random variables $\{U_k, k \geqq 1\}$ uniformly distributed over $[0, 1]$. Construct the uniformized Markov chains as follows:

$$\hat{Q}_0^i = Q_0^i, \qquad i = 1, 2,$$

and

$$\hat{Q}_{j,\tau_{k+1}}^i = \hat{Q}_{j,\tau_k}^i - \delta_{j,k}^i \qquad \text{for all } \tau_{k+1} < t_1$$

where

$$\delta_{j,k}^i = \mathbf{1}\left\{\frac{\Sigma_{l=1}^{j-1} \mu_l X_{l,k}^i}{\Delta} \leqq U_k < \frac{\Sigma_{l=1}^{j} \mu_l X_{l,k}^i}{\Delta}\right\},$$

and $X_{j,k}^i = 1$ if job j is served in system i at time τ_k and $X_{j,k}^i = 0$ otherwise. As desired, it is clear that $\hat{Q}_t^i =_{\rm st} Q_t^i$, $i = 1, 2$. Under π, $X_{j,k}^i = \hat{Q}_{j,\tau_k}^i$ if $\Sigma_{l=1}^j \hat{Q}_{l,\tau_k}^i \leqq m(t_0)$ and $X_{j,k}^i = 0$ otherwise. Thus, $\Sigma_{l=1}^j X_{l,k}^2 = \min\{m(t_0), \Sigma_{l=1}^j \hat{Q}_{l,\tau_k}^2\}$. From the induction hypothesis $\Sigma_{l=1}^j \hat{Q}_{l,\tau_k}^1 \leqq \Sigma_{l=1}^j \hat{Q}_{l,\tau_k}^2$, it follows that $\Sigma_{l=1}^j X_{l,k}^1 \leqq \Sigma_{l=1}^j X_{l,k}^2$, $j = 1, \cdots, n$. Since there is at most one departure at τ_{k+1}, $\Sigma_{l=1}^j \hat{Q}_{l,\tau_{k+1}}^1 \leqq \Sigma_{l=1}^j \hat{Q}_{l,\tau_{k+1}}^2$ if $\Sigma_{l=1}^j \hat{Q}_{l,\tau_k}^1 < \Sigma_{l=1}^j \hat{Q}_{l,\tau_k}^2$. Therefore, we need only consider the case that $\Sigma_{l=1}^j \hat{Q}_{l,\tau_k}^1 = \Sigma_{l=1}^j \hat{Q}_{l,\tau_k}^2$. In such case, $\Sigma_{l=1}^j X_{l,k}^1 = \Sigma_{l=1}^j X_{l,k}^2$ and $\Sigma_{l=1}^p X_{l,k}^1 \leqq \Sigma_{l=1}^p X_{l,k}^2$, $p = 1, \cdots, j-1$. From Lemma 2.6, it follows that $\Sigma_{l=1}^j \mu_l X_{l,k}^1 \geqq \Sigma_{l=1}^j \mu_l X_{l,k}^2$ and thus $\Sigma_{l=1}^j \delta_{l,k}^1 \geqq \Sigma_{l=1}^j \delta_{l,k}^2$. Therefore, we conclude that $\hat{Q}_{\tau_{k+1}}^1 \leqq_{\rm ps} \hat{Q}_{\tau_{k+1}}^2$. Observe that the partial sum ordering is also preserved when there is an arrival of the same type at both systems. By induction, we can construct two processes such that $\hat{Q}_t^1 \leqq_{\rm ps} \hat{Q}_t^2$.

Let $J_t(Q_0) \stackrel{\rm def}{=} E(g(Q_t) \mid Q_0)$ be the expected cost at time t from the initial state Q_0 under the policy π.

Corollary 2.9. *Assume (A1)–(A3) hold, and $Q_0^1 \leqq_{\rm ps} Q_0^2$. Then scheduling under π we have $J_t(Q_0^1) \leqq J_t(Q_0^2)$ for all $t \geqq 0$.*

Proof. From (A2) and (A3), it follows that $g(x - e_\alpha) \geqq g(x - e_\beta)$ for all $\alpha > \beta$. Using the transfer property in Lemma 2.7 yields that $g(x^1) \leqq g(x^2)$ if $x^1 \leqq_{\rm ps} x^2$. It then follows from the π-monotone property in Lemma 2.8 that $J_t(Q_0^1) \leqq J_t(Q_0^2)$ if $Q_0^1 \leqq_{\rm ps} Q_0^2$.

Lemma 2.10. *Under (A1)–(A3) and π,*

$$\text{(1)} \qquad \mu_\alpha J_t(Q_0 - e_\alpha) \geqq \mu_\beta J_t(Q_0 - e_\beta) + (\mu_\alpha - \mu_\beta) J_t(Q_0),$$

for any initial state Q_0 with $Q_{\alpha,0} = Q_{\beta,0} = 1$ and $\alpha > \beta$.

Proof. To simplify the notations in the proof, we denote $e_{i,j} = e_i + e_j$. We use the same construction as in the proof of Lemma 2.8. First, we show that (1) holds for all $t \in [t_0, t_1)$. From (A3), it follows that (1) is satisfied at $\tau_0 = 0$. Now take as an induction hypothesis that (1) holds for $t \in [\tau_0, \tau_k]$. Let S (or S_α, S_β) be the set of jobs served at time 0 under π given the initial state Q_0 (or $Q_0 - e_\alpha$, $Q_0 - e_\beta$). Note that $\{\tau_{k+1} - \tau_k\}$ is a sequence of i.i.d. exponential random variables. Analogous to Kolmogorov's backward equation, we have

$$J_{\tau_{k+1}}(Q_0)=\sum_{j\in S}J_{\tau_k}(Q_0-e_j)\frac{\mu_j}{\Delta}+J_{\tau_k}(Q_0)\frac{\Delta-\sum_{j\in S}\mu_j}{\Delta}.$$

Similarly,

$$J_{\tau_{k+1}}(Q_0-e_\alpha)=\sum_{j\in S_\alpha}J_{\tau_k}(Q_0-e_{\alpha,j})\frac{\mu_j}{\Delta}+J_{\tau_k}(Q_0-e_\alpha)\frac{\Delta-\sum_{j\in S_\alpha}\mu_j}{\Delta},$$

and

$$J_{\tau_{k+1}}(Q_0-e_\beta)=\sum_{j\in S_\beta}J_{\tau_k}(Q_0-e_{\beta,j})\frac{\mu_j}{\Delta}+J_{\tau_k}(Q_0-e_\beta)\frac{\Delta-\sum_{j\in S_\beta}\mu_j}{\Delta}.$$

Consider the following four cases.

Case 1: $\Sigma_{j=1}^{\beta} Q_{j,0}>m(t_0)$. In this case, $S=S_\alpha=S_\beta$. Thus,

$$\begin{aligned}&\mu_\alpha J_{\tau_{k+1}}(Q_0-e_\alpha)-\mu_\beta J_{\tau_{k+1}}(Q_0-e_\beta)-(\mu_\alpha-\mu_\beta)J_{\tau_{k+1}}(Q_0)\\&=\sum_{j\in S}\frac{\mu_j}{\Delta}(\mu_\alpha J_{\tau_k}(Q_0-e_{\alpha,j})-\mu_\beta J_{\tau_k}(Q_0-e_{\beta,j})-(\mu_\alpha-\mu_\beta)J_{\tau_k}(Q_0-e_j))\\&\quad+\frac{\Delta-\sum_{j\in S}\mu_j}{\Delta}(\mu_\alpha J_{\tau_k}(Q_0-e_\alpha)-\mu_\beta J_{\tau_k}(Q_0-e_\beta)-(\mu_\alpha-\mu_\beta)J_{\tau_k}(Q_0))\\&\geqq 0.\end{aligned}$$

Case 2: $\Sigma_{j=1}^{\beta} Q_{j,0}\leqq m(t_0)$ and $\Sigma_{j=1}^{\alpha} Q_{j,0}>m(t_0)$. In this case, there exists a γ such that $\beta\leqq\gamma<\alpha$ and $\Sigma_{j=1}^{\gamma} Q_{j,0}\leqq m(t_0)$, $\Sigma_{j=1}^{\gamma+1} Q_{j,0}>m(t_0)$. Clearly, $S=S_\alpha=S_\beta\cup\{\beta\}\setminus\{\gamma+1\}$. Thus,

$$\begin{aligned}&\mu_\alpha J_{\tau_{k+1}}(Q_0-e_\alpha)-\mu_\beta J_{\tau_{k+1}}(Q_0-e_\beta)-(\mu_\alpha-\mu_\beta)J_{\tau_{k+1}}(Q_0)\\&=\sum_{j\in S\setminus\{\beta\}}\frac{\mu_j}{\Delta}(\mu_\alpha J_{\tau_k}(Q_0-e_{\alpha,j})-\mu_\beta J_{\tau_k}(Q_0-e_{\beta,j})-(\mu_\alpha-\mu_\beta)J_{\tau_k}(Q_0-e_j))\\&\quad+\frac{\Delta-\sum_{j\in S}\mu_j}{\Delta}(\mu_\alpha J_{\tau_k}(Q_0-e_\alpha)-\mu_\beta J_{\tau_k}(Q_0-e_\beta)-(\mu_\alpha-\mu_\beta)J_{\tau_k}(Q_0))\\&\quad+\frac{\mu_\beta}{\Delta}(\mu_\alpha J_{\tau_k}(Q_0-e_{\alpha,\beta})-\mu_{\gamma+1}J_{\tau_k}(Q_0-e_{\beta,\gamma+1})-(\mu_\alpha-\mu_{\gamma+1})J_{\tau_k}(Q_0-e_\beta))\\&\geqq 0.\end{aligned}$$

Case 3: $\Sigma_{j=1}^{\alpha} Q_{j,0}\leqq m(t_0)$ and $\Sigma_{j=1}^{n} Q_{j,0}>m(t_0)$. Analogous to the proof of Case 2, there exists a γ such that $\alpha\leqq\gamma<n$ and $\Sigma_{j=1}^{\gamma} Q_{j,0}\leqq m(t_0)$, $\Sigma_{j=1}^{\gamma+1} Q_{j,0}>m(t_0)$. Clearly, $S=S_\alpha\cup\{\alpha\}\setminus\{\gamma+1\}=S_\beta\cup\{\beta\}\setminus\{\gamma+1\}$. Thus,

$$\mu_\alpha J_{\tau_{k+1}}(Q_0-e_\alpha)-\mu_\beta J_{\tau_{k+1}}(Q_0-e_\beta)-(\mu_\alpha-\mu_\beta)J_{\tau_{k+1}}(Q_0)$$

$$=\sum_{j\in S\setminus\{\alpha,\beta\}}\frac{\mu_j}{\Delta}(\mu_\alpha J_{\tau_k}(Q_0-e_{\alpha,j})-\mu_\beta J_{\tau_k}(Q_0-e_{\beta,j})-(\mu_\alpha-\mu_\beta)J_{\tau_k}(Q_0-e_j))$$

$$+\frac{\Delta-\sum_{j\in S_\beta}\mu_j}{\Delta}(\mu_\alpha J_{\tau_k}(Q_0-e_\alpha)-\mu_\beta J_{\tau_k}(Q_0-e_\beta)-(\mu_\alpha-\mu_\beta)J_{\tau_k}(Q_0))$$

$$+\frac{\mu_{\gamma+1}}{\Delta}(\mu_\alpha J_{\tau_k}(Q_0-e_{\alpha,\gamma+1})-\mu_\beta J_{\tau_k}(Q_0-e_{\beta,\gamma+1})-(\mu_\alpha-\mu_\beta)J_{\tau_k}(Q_0-e_{\gamma+1}))$$

$$+\frac{(\mu_\alpha-\mu_\beta)}{\Delta}(\mu_{\gamma+1}J_{\tau_k}(Q_0-e_{\gamma+1})-\mu_\beta J_{\tau_k}(Q_0-e_\beta)-(\mu_{\gamma+1}-\mu_\beta)J_{\tau_k}(Q_0))$$

$$\geqq 0.$$

Case 4: $\sum_{j=1}^{n} Q_{j,0}\leqq m(t_0)$. Clearly, all the jobs are served and $S=S_\alpha\cup\{\alpha\}=S_\beta\cup\{\beta\}$. Thus,

$$\mu_\alpha J_{\tau_{k+1}}(Q_0-e_\alpha)-\mu_\beta J_{\tau_{k+1}}(Q_0-e_\beta)-(\mu_\alpha-\mu_\beta)J_{\tau_{k+1}}(Q_0)$$

$$=\sum_{j\in S\setminus\{\alpha,\beta\}}\frac{\mu_j}{\Delta}(\mu_\alpha J_{\tau_k}(Q_0-e_{\alpha,j})-\mu_\beta J_{\tau_k}(Q_0-e_{\beta,j})-(\mu_\alpha-\mu_\beta)J_{\tau_k}(Q_0-e_j))$$

$$+\frac{\Delta-\sum_{j\in S}\mu_j}{\Delta}(\mu_\alpha J_{\tau_k}(Q_0-e_\alpha)-\mu_\beta J_{\tau_k}(Q_0-e_\beta)-(\mu_\alpha-\mu_\beta)J_{\tau_k}(Q_0))$$

$$+\frac{\mu_\alpha\mu_\beta}{\Delta}(J_{\tau_k}(Q_0-e_\alpha)-J_{\tau_k}(Q_0-e_\beta)).$$

Since $Q_0-e_\beta\leqq_{\mathrm{ps}}Q_0-e_\alpha$, $J_{\tau_k}(Q_0-e_\alpha)\geqq J_{\tau_k}(Q_0-e_\beta)$ from Corollary 2.9. It then follows from these four cases that (1) holds for $t\in[t_0,t_1)$.

Now take as the induction hypothesis that (1) holds for all $t\in[t_0,t_k)$. If t_k is an epoch where $dm(t)\neq 0$, then it follows from the same argument, with $m(t_0)$ being replaced by $m(t_k)$, that (1) holds for all $t\in[t_0,t_{k+1})$. If t_k is the arrival epoch of job j, then it follows from the same argument, with Q_0 replaced by Q_0+e_j, that (1) holds for all $t\in[t_0,t_{k+1})$. Thus, (1) holds for any $t\geqq 0$.

Proof of Theorem 1.1 (sufficient condition). Again, we transform the continuous-time problem into a discrete-time one by using uniformization. Let $\{t_k,k\geqq 0\}$ be defined as in the proof of Lemmas 2.8 and 2.10. Recall that t_k is either an arrival epoch or an epoch when a machine breaks down or becomes available. Generate a sequence of independent Poisson processes, $N_l(t)$, $l\geqq 0$ with rates $\Delta_l(y)=ym(t_l)\mu_n$ for some $y\geqq 1$. Let $\{\tau_{l,k},k\geqq 1\}$ be the arrival epochs of the lth Poisson processes and define $\tau_{l,0}=0$. Observe that the $\tau_{l,k}$'s are epochs when a job completion may occur in the coupling scheme we presented in the proof of Lemmas 2.8 and 2.10. Analogous to the proof of

Pinedo (1983), Theorem 1, we first show that π is optimal within the class of policies $\mathscr{G}(y)$, where $\mathscr{G}(y)$ consists of all policies that allow preemption only at time epochs $\{t_l + \tau_{l,k}, k \geqq 0, l \geqq 0\}$ with $t_l + \tau_{l,k} < t_{l+1}$. Then it follows from a continuity argument as $y \to \infty$ that π is optimal among all the policies.

To simplify the notation, let $\{\tau'_k, k \geqq 0\}$ be the sequence of $\{t_l + \tau_{l,k}, k \geqq 0, l \geqq 0\}$ after sorting. We use induction on $\{\tau'_k\}$. First, we show π is optimal in a single step for all initial states $\boldsymbol{Q}_0$. If τ'_1 is the arrival epoch of job j, then the state of jobs at τ'_1 is $\boldsymbol{Q}_0 + e_j$ for any policy. This implies the costs at τ'_1 are the same for any policy. If τ'_1 is an epoch that a machine breaks down or becomes available, i.e. $dm(t) \neq 0$, then the state of jobs at τ'_1 is $\boldsymbol{Q}_0$ for any policy. Again, this implies the costs at τ'_1 are the same for any policy. Thus, it suffices to consider the case that τ'_1 is not a point of $\{t_k, k \geqq 1\}$. Consider a policy $\sigma \in \mathscr{G}(y)$. Let $X^{\sigma}_{j,0}$ be the indicator function of the event that job j is served at time 0 under a policy σ and let $\boldsymbol{X}^{\sigma}_0 = (X^{\sigma}_{1,0}, \cdots, X^{\sigma}_{n,0})$. Let $\boldsymbol{X}^{\pi}_0$ be the corresponding quantity under π. Since the policy π schedules jobs according to priority of the lowest index, we have that $\boldsymbol{X}^{\sigma}_0 \leqq_{\mathrm{ps}} \boldsymbol{X}^{\pi}_0$ for any policy σ in $\mathscr{G}(y)$. From Lemma 2.7, it follows that $\boldsymbol{X}^{\sigma}_0$ can be reached from $\boldsymbol{X}^{\pi}_0$ through a finite number of applications of the following two types of transfers: (i) $\boldsymbol{x} \mapsto \boldsymbol{x} - e_\beta$ and (ii) $\boldsymbol{x} \mapsto \boldsymbol{x} + e_\alpha - e_\beta, \alpha > \beta$. Thus, we only need to compare the expected cost at τ'_1 for two policies σ^1 and σ^2 with $\boldsymbol{X}^{\sigma^1}_0 = T(\boldsymbol{X}^{\sigma^2}_0)$, where T is of the form (i) or (ii). Now let $\boldsymbol{Q}^{\sigma^i}_{\tau'_1}$ be the states of jobs at τ'_1 under the policy σ^i, $i = 1, 2$. It suffices to show that

$$E(g(\boldsymbol{Q}^{\sigma^1}_{\tau'_1}) \mid \boldsymbol{Q}_0) \geqq E(g(\boldsymbol{Q}^{\sigma^2}_{\tau'_1}) \mid \boldsymbol{Q}_0). \tag{2}$$

Case 1: The first type of transfer. In this case, $\boldsymbol{X}^{\sigma^1}_0 = \boldsymbol{X}^{\sigma^2}_0 - e_\beta$ for some β. This is equivalent to inserting idle time into a machine. Clearly, we have $\boldsymbol{Q}^{\sigma^1}_{\tau'_1} \geqq \boldsymbol{Q}^{\sigma^2}_{\tau'_1}$. It then follows from (A2) that (2) holds.

Case 2: The second type of transfer. In this case, $\boldsymbol{X}^{\sigma^1}_0 = \boldsymbol{X}^{\sigma^2}_0 + e_\alpha - e_\beta$ for some $\alpha > \beta$. (Clearly, we assume $X^{\sigma^1}_{\alpha,0} = 1$ and $X^{\sigma^1}_{\beta,0} = 0$.) The only difference between σ^1 and σ^2 is that σ^2 puts the low-index job β into service instead of the high-index job α. Now define $S^i = \{j : X^{\sigma^i}_{j,0} = 1\}$ as the set of jobs served at time 0 under policy σ^i. Then for $i = 1, 2$,

$$P(\boldsymbol{Q}^{\sigma^i}_{\tau'_1} = \boldsymbol{Q}_0 - e_j \mid \boldsymbol{Q}_0) = \frac{\mu_j}{\Delta_0(y)}, \qquad j \in S^i \tag{3}$$

$$P(\boldsymbol{Q}^{\sigma^i}_{\tau'_1} = \boldsymbol{Q}_0 \mid \boldsymbol{Q}_0) = \frac{\Delta_0(y) - \sum_{j \in S^i} \mu_j}{\Delta_0(y)}, \qquad j \notin S^i. \tag{4}$$

Thus, we have from (A3) that

$$\begin{aligned} &E(g(\boldsymbol{Q}^{\sigma^1}_{\tau'_1}) \mid \boldsymbol{Q}_0) - E(g(\boldsymbol{Q}^{\sigma^2}_{\tau'_1}) \mid \boldsymbol{Q}_0) \\ &= \frac{1}{\Delta_0(y)} (\mu_\alpha g(\boldsymbol{Q}_0 - e_\alpha) - \mu_\beta g(\boldsymbol{Q}_0 - e_\beta) - (\mu_\alpha - \mu_\beta) g(\boldsymbol{Q}_0)) \\ &\geqq 0. \end{aligned}$$

Therefore, π is optimal in a single step.

Take as the induction hypothesis that π is optimal over $k-1$ steps. Now we would like to show that the policy that applies σ^2 at time 0 and then applies π after τ_1' is better than the policy that applies σ^1 at time 0 and then applies π after τ_1'. Now let $\boldsymbol{Q}_{\tau_k'}^{\sigma^i}$, $i=1,2$, be the states of jobs at τ_k' under these two policies. It suffices to show that

$$E(g(\boldsymbol{Q}_{\tau_k'}^{\sigma^1}) \mid \boldsymbol{Q}_0) \geqq E(g(\boldsymbol{Q}_{\tau_k'}^{\sigma^2}) \mid \boldsymbol{Q}_0). \tag{5}$$

Again, we only have to consider the case that τ_1' is not a point of $\{t_k, k \geqq 1\}$. For the first type of transfer, we have that $\boldsymbol{Q}_{\tau_1'}^{\sigma^1} \geqq \boldsymbol{Q}_{\tau_1'}^{\sigma^2}$. This implies $\boldsymbol{Q}_{\tau_1'}^{\sigma^1} \geqq_{\mathrm{ps}} \boldsymbol{Q}_{\tau_1'}^{\sigma^2}$. It then follows from Corollary 2.9 that (5) holds. Analogous to (4), we have from the transition probabilities of the uniformized Markov chains that for the second type of transfer,

$$\begin{aligned}
&E(g(\boldsymbol{Q}_{\tau_k'}^{\sigma^1}) \mid \boldsymbol{Q}_0) - E(g(\boldsymbol{Q}_{\tau_k'}^{\sigma^2}) \mid \boldsymbol{Q}_0) \\
&\quad = \frac{1}{\Delta_0(y)}(\mu_\alpha J_{\tau_1', \tau_k'}(\boldsymbol{Q}_0 - e_\alpha) - \mu_\beta J_{\tau_k', \tau_1'}(\boldsymbol{Q}_0 - e_\beta) - (\mu_\alpha - \mu_\beta) J_{\tau_1', \tau_k'}(\boldsymbol{Q}_0))
\end{aligned}$$

where $J_{\tau_1', \tau_k'}(\boldsymbol{Q}_0) = E(g(\boldsymbol{Q}_{\tau_k'}) \mid \boldsymbol{Q}_{\tau_1'})$. From Lemma 2.10 and the induction hypothesis, it follows that π is optimal within $\mathscr{G}(y)$. A continuity argument then completes the proof that the conditions in Theorem 1.1 are sufficient for optimality of π.

(Necessary condition). We would like to show that if the policy π minimizes the expected cost for all t and any realization of machine breakdowns and repairs then (A2) and (A3) must hold for all initial states $\boldsymbol{Q}_0$. This means we may consider a problem with l machines available at time 0, where $l = \sum_{i=1}^{\beta} Q_{i,0}$. Clearly, the policy π begins by assigning to machines all jobs whose index is not larger than β. Call this set of jobs S. Then the expected cost of the policy π after a very small amount of time δt is

$$\sum_{i \in S} g(\boldsymbol{Q}_0 - e_i)\mu_i \delta t + \left(1 - \sum_{i \in S} \mu_i \delta t\right) g(\boldsymbol{Q}_0) + o(\delta t). \tag{6}$$

Now consider a policy σ^1 which schedules jobs having indices smaller than β on to $l-1$ machines and keeps the lth machine idle. Let σ^1 be the policy that is the same as π but schedules job α instead of job β, $\alpha > \beta$. The expected costs of the policies σ^1 and σ^2, after a very small amount of time δt, can be computed similarly. It is easy to see that the difference between the expected costs of the policies π and σ^1 at time δt is

$$g(\boldsymbol{Q}_0 - e_\beta)\mu_\beta \delta t - g(\boldsymbol{Q}_0)\mu_\beta \delta t + o(\delta t). \tag{7}$$

The optimality of π for all t, when there are l machines available at time 0, implies that the quantity in (7) must not be greater than 0. Letting $\delta t \to 0$ yields (A2). Similarly, the difference between the expected costs of the policy π and σ^2 at time δt is

$$g(\boldsymbol{Q}_0 - e_\beta)\mu_\beta \delta t - g(\boldsymbol{Q}_0)\mu_\beta \delta t - g(\boldsymbol{Q}_0 - e_\alpha)\mu_\alpha \delta t + g(\boldsymbol{Q}_0)\mu_\alpha \delta t + o(\delta t). \tag{8}$$

The optimality of π implies that the quantity in (8) must not be greater than 0. Letting $\delta t \to 0$ yields (A3).

3. The optimality of the $c\mu$ rule

In this section, we consider the cost function $g(\boldsymbol{x}) = \Sigma_{j=1}^{n} c_j x_j$. In other words, for each unit time that job j remains in the system, a cost c_j is incurred. Assuming the agreeability condition (A4) of Section 1, we use Theorem 1.1 to establish optimality criteria for the $c\mu$ rule. Recall that the $c\mu$ rule is the preemptive policy that orders jobs in increasing priority according to increasing values of $c\mu$. Under (A4), the $c\mu$ rule results in the same order as the LEPT rule. The objective functions that are considered in this section include the expected values of (i) the weighted number of jobs in the system at an arbitrary time t, (ii) the weighted sum of job completion times, (iii) the weighted number of late jobs when the jobs have a common random due date, and (iv) the weighted sum of job tardinesses when the jobs have a common due date.

Corollary 3.1 (*minimization of the expected weighted number of jobs at arbitrary time t*). *Assume (A1) holds. Let π be the policy that schedules jobs according to priorities that are decreasing in their indices. Then π minimizes $\boldsymbol{E}[\Sigma_{j=1}^{n} c_j Q_{j,t}]$, the expected weighted number of jobs, for all t ($t > 0$), and all processes of arrivals, breakdowns and repairs, if and only if (A4) is satisfied. Preemption need only occur at the release of a new job.*

Proof. For $g(\boldsymbol{x}) = \Sigma_{i=1}^{n} c_i x_i$, it is easy to see that (A2)$\Leftrightarrow c_i \geqq 0$ and that (A3)$\Leftrightarrow c_i\mu_i \geqq c_j\mu_j$, $i < j$.

In fact, Corollary 3.1 also follows immediately from the known result that LEPT minimizes the makespan *stochastically*. Recall, $V_{j,t} = Q_{j,t}/\mu_j$. Observe that the objective function is an arrangement increasing function of the expected work remaining at time t, i.e.

$$\boldsymbol{E}\left[\sum_{j=1}^{n} c_j Q_{j,t}\right] = \boldsymbol{E}\left[\sum_{j=1}^{n} c_j\mu_j V_{j,t}\right] = \sum_{i=1}^{n} (c_i\mu_i - c_{i+1}\mu_{i+1})\boldsymbol{E}\left[\sum_{j=1}^{i} V_{j,t}\right],$$

where we take $c_{n+1}\mu_{n+1} = 0$. Since LEPT minimizes the expected work remaining at every time t, it is clear that the summation on the right-hand side above is minimized by LEPT.

It is clear that without the agreeability condition (A1), the $c\mu$ rule itself cannot be optimal. Counterexamples can be found, even under Kämpke's weaker agreeability condition of $c_1 \geqq \cdots \geqq c_n$ and (A4). Consider the following set of jobs: $c_j\mu_j = 1$, $j = 1, \cdots, n-1$, $\mu_j = n-1$, $j = 1, \cdots, n-1$, $\mu_n = \frac{1}{2}$ and $c_j = 1/(n-1)$, $j = 1, 2, \cdots, n$. There are two machines and all n jobs have a common (fixed) due date at 2. Take n very large. The first $n-1$ jobs require a total amount of processing equal to 1. The variance in this total amount goes to 0 as n tends to ∞. It is clear that job n will start under the $c\mu$ rule at time $\frac{1}{2}$. It is also clear that starting job n (the large job) at time 0 significantly increases the probability that it is completed by time 2, and therefore maximizes the expected number of jobs completed by the due date. For an example in which the $c\mu$ rule does not minimize the expected sum of the weighted completion times, consider again two machines and the same set of jobs as the one described above and all n jobs available at time 0. Now, instead of a due date at time 2, a second batch of jobs arrives at time 2. It is clear that the $c\mu$ rule does not minimize the expected sum of the weighted completion times. It is necessary to start the long job at time 0 in order to maximize the expected machine utilization before time 2.

Corollary 3.2 (*minimization of the expected weighted number of late jobs*). *Assume* $(A1)$ *and* $(A4)$ *hold, and the jobs have a common due date,* D. *Let* Z_j *be the completion time of job* j. *Let* U_j *be the indicator function for the event* $[Z_j \geqq D_j]$. *Then* π *minimizes* $\boldsymbol{E}[\sum_{j=1}^{n} c_j U_j]$.

Proof. Consider the event $\{R_j = r_j, j = 1, \cdots, n, D = d\}$. On this event, $U_j = Q_{j,d}$ if $r_j < d$ and 1 if $r_j \geqq d$. Applying Corollary 3.1 completes the proof.

Remark 3.3. Corollary 3.2 is an extension of Pinedo and Rammouz (1988), Theorem 2, to parallel machines. Pinedo (1983), Theorem 8, also showed that if $\mu_1 \geqq \mu_2 \geqq \cdots \geqq \mu_n$, $c_1 \geqq c_2 \geqq \cdots \geqq c_n$, and the common due date D has a concave distribution function, then the SEPT rule minimizes $\boldsymbol{E}[\sum_{j=1}^{n} c_j U_j]$. Corollary 3.2, states a different set of conditions under which the LEPT rule is optimal.

The following definition is useful in obtaining a result for more general weighted functions of completion times. The subsequent corollary extends Pinedo and Rammouz (1988), Theorem 1, to parallel machines.

Definition 3.4 (*Pinedo and Rammouz* (1988)). *Let* $F_1(t)$ *and* $F_2(t)$ *be two increasing functions.* F_1 *is said to be steeper than* F_2 *if* $F_1(t_2) - F_1(t_1) \geqq F_2(t_2) - F_2(t_1)$ *for all* $t_1 < t_2$. *We denote this by* $F_1 \geqq_s F_2$.

Corollary 3.5 (*minimization of the expected weighted sum of job completion times*). *Suppose* $(A1)$ *and* $(A4)$ *hold, and* $F_j(t), j = 1, \cdots, n$ *are increasing functions, such that* $F_1 \geqq_s F_2 \geqq_s \cdots \geqq_s F_n$. *Then* π *minimizes* $\boldsymbol{E}[\sum_{j=1}^{n} c_j F_j(Z_j)]$.

Proof. It is noted in Pinedo and Rammouz (1988) that Corollary 3.2 is equivalent to minimizing $\boldsymbol{E}[\sum_{j=1}^{n} c_j F(Z_j)]$, where $F(t)$ is the distribution function of the common due date D. Thus, π minimizes $\boldsymbol{E}[\sum_{j=1}^{n} c_j F(Z_j)]$ for all F increasing. Note that $\boldsymbol{E}[\sum_{j=1}^{n} c_j F_j(Z_j)] = \sum_{i=1}^{n} B_i$ with $B_i = \sum_{j=1}^{n+1-i} c_j \boldsymbol{E}[F_{n+1-i}(Z_j) - F_{n+2-i}(Z_j)]$, and taking $F_{n+1}(t) = 0$. Now $F_j \geqq_s F_{j+1}$ implies that $F_j - F_{j+1}$ is increasing. Combined with the fact that F_n is increasing, this implies that π minimizes B_i, $i = 1, \cdots, n$, and thus that it minimizes the sum of the B_i's.

Corollary 3.6 (*minimization of the expected weighted sum of job tardinesses*). *The tardiness of job* j, T_j, *is defined as* $\max(Z_j - D_j, 0)$. *Suppose* $(A1)$ *and* $(A4)$ *hold, and* $D_1 \leqq D_2 \leqq \cdots \leqq D_n$ a.s. *Then* π *minimizes* $\boldsymbol{E}[\sum_{j=1}^{n} c_j T_j]$.

Proof. Consider the event $\{D_j = d_j, j = 1, \cdots, n\}$. The objective function is equivalent to $\boldsymbol{E}[\sum_{j=1}^{n} c_j F_j(Z_j)]$, with $F_j(t) = \max(t - d_j, 0)$ (Pinedo and Rammouz (1988)). Clearly, the $F_j(t)$'s are increasing and $F_j \geqq_s F_{j+1}$. An application of Corollary 3.5 completes the proof.

Remark 3.7. Pinedo (1983), Theorem 3, showed that if $D_1 \leqq D_2 \leqq \cdots \leqq D_n$ a.s. and (A4) is satisfied, then π minimizes $\boldsymbol{E}[\sum_{j=1}^{n} c_j T_j]$ for a single-machine problem with all the jobs present at time 0. Corollary 3.6 is an extension of that theorem to parallel machines with release dates.

In the following two corollaries, we consider the special case that all jobs are present at time 0. A static list policy is one that assigns jobs to machines in the order of a fixed permutation of their indices. Clearly, the set of static list policies is a subset of the dynamic policies we have considered thus far, and π, which under (A1) assigns jobs in the order $1, \cdots, n$, is optimal amongst all static list policies when (A1) and (A4) are satisfied. In the following two corollaries the conditions needed in Corollaries 3.2 and 3.6 are relaxed from the strong sense (a.s.) to the weak sense (distribution). These corollaries generalize Pinedo (1983), Theorems 4 and 2, to parallel machines. Recall that a random variable X is stochastically smaller $\leqq_{\mathrm{st}}$ than Y if $\boldsymbol{P}(X > t) \leqq \boldsymbol{P}(Y > t)$ for all t.

Corollary 3.8 (*minimization of the expected weighted number of late jobs*). *Assume* $(A1)$ *and* $(A4)$ *hold, and due dates* $D_j, j = 1, \cdots, n$, *have a common distribution function* $F(t)$. *Let* U_j *be the indicator function for the event that job* j *is late. Then* π *minimizes* $\boldsymbol{E}[\sum_{j=1}^{n} c_j U_j]$ *amongst all the static list policies.*

Proof. Let $\sigma = \{\sigma_1, \sigma_2, \cdots, \sigma_n\}$ be a permutation of $\{1, 2, \cdots, n\}$ and suppose the jobs are scheduled according to the static list policy that assigns jobs to machines in priority order $\sigma_1, \sigma_2, \cdots, \sigma_n$. Under these circumstances, let $Q_{j,t}^{\sigma}$ be the indicator function for the event that job j is in the system at time t and let U_j^{σ} be the indicator function that job j is late. From Corollary 3.1, it follows that under (A1) and (A4)

$$\sum_{j=1}^{n} \boldsymbol{E} c_j Q_{j,t}^{\pi} \leqq \sum_{j=1}^{n} \boldsymbol{E} c_j Q_{j,t}^{\sigma}. \tag{9}$$

All the jobs are present at time 0, so $U_j^{\sigma} = Q_{j,D_j}^{\sigma}$. Since the order in which jobs will be assigned to machines is determined at time 0, it follows that $Q_{j,t}^{\sigma}$ is independent of the due date D_j. (This is not true if the policy is dynamic.) We have

$$\boldsymbol{E} Q_{j,D_j}^{\sigma} = \int_0^{\infty} \boldsymbol{E} Q_{j,t}^{\sigma}\, dF(t). \tag{10}$$

Combining (9) and (10) completes the proof.

Corollary 3.9 (*minimization of the expected weighted sum of job tardinesses*). *Assume* $(A1)$ *and* $(A4)$ *hold and due dates satisfy* $D_1 \leqq_{\mathrm{st}} D_2 \leqq_{\mathrm{st}} \cdots \leqq_{\mathrm{st}} D_n$. *Let* T_j *be the tardiness of job* j. *Then* π *minimizes* $\boldsymbol{E}[\sum_{j=1}^{n} c_j T_j]$ *among all static list policies.*

Proof. Again, let σ be a permutation of $\{1, 2, \cdots, n\}$ and define $Q_{j,t}^{\sigma}$ as above. Let T_j^{σ} be the tardiness of job j when jobs are processed under the static list policy defined by σ. All the jobs are present at time 0, so $T_j^{\sigma} = \int_{D_j}^{\infty} Q_{j,t}^{\sigma}\, dt$. Again, $Q_{j,t}^{\sigma}$ is independent of D_j and we have

$$\begin{aligned} \boldsymbol{E} T_j^{\sigma} &= \int_0^{\infty} \int_s^{\infty} \boldsymbol{E} Q_{j,t}^{\sigma}\, dt\, dF_j(s) \\ &= \int_0^{\infty} F_j(t) \boldsymbol{E} Q_{j,t}^{\sigma}\, dt, \end{aligned}$$

where $F_j(t)$ is the distribution function of the due date of job j, D_j. Analogous to the proof of Corollary 3.5, $\boldsymbol{E}[\sum_{j=1}^n c_j T_j^\sigma] = \sum_{i=1}^n B_i$, where

$$B_i = \int_0^\infty (F_{n+1-i}(t) - F_{n+2-i}(t)) \sum_{j=1}^{n+1-i} c_j EQ_{j,t}^\sigma dt$$

and we define $F_{n+1}(t) = 0$. The assumption $D_1 \leqq_{\text{st}} D_2 \leqq_{\text{st}} \cdots \leqq_{\text{st}} D_n$ implies $F_1(t) \geqq F_2(t) \geqq \cdots \geqq F_n(t) \geqq 0$. It follows from Corollary 3.1 that π minimizes B_i, $i = 1, \cdots, n$, and thus minimizes the sum of the B_i's.

References

BARAS, J. S., DORSEY, A. J. AND MAKOWSKI, A. M. (1985) Two competing queues with linear costs and geometric service requirements: the μc-rule is often optimal. *Adv. Appl. Prob.* **17**, 186–209.

BUYUKKOC, C., VARAIYA, P. AND WALRAND, J. (1985) The $c\mu$ rule revisited. *Adv. Appl. Prob.* **17**, 237–238.

CHANG, C. S. (1992) Ordering for stochastic majorization: theory and applications. *Adv. Appl. Prob.* **24**(3).

CHANG, C. S. AND YAO, D. D. (1990) Rearrangement, majorization and stochastic scheduling.

FREDERICKSON, G. N., BRUNO, J. AND DOWNEY, P. (1981) Sequencing tasks with exponential service times to minimize the expected flow time or makespan. *J. Assoc. Comput. Mach.* **28**, 100–113.

KÄMPKE, T. (1987) On the optimality of static priority policies in stochastic scheduling on parallel machines. *J. Appl. Prob.* **24**, 430–448.

KÄMPKE, T. (1989) Optimal scheduling of jobs with exponential service times on identical parallel processors. *Operat. Res.* **37**, 126–133.

KEILSON, T. (1979) *Markov Chain Models — Rarity and Exponentiality.* Springer-Verlag, New York.

MARSHALL, A. W. AND OLKIN, I. (1979) *Inequalities: Theory of Majorization and its Applications.* Academic Press, New York.

MUIRHEAD, R. F. (1903) Some methods applicable to identities and inequalities of symmetric algebraic functions with n letters. *Proc. Edinburgh Math. Soc.* **21**, 144–157.

PINEDO, M. (1983) Stochastic scheduling with release dates and due dates. *Operat. Res.* **31**, 559–572.

PINEDO, M. AND RAMMOUZ, E. (1988) A note on stochastic scheduling on a single machine subject to breakdown and repair. *Prob. Eng. Inf. Sci.* **2**, 41–49.

PINEDO, M. AND WEISS, G. (1979) Scheduling stochastic tasks on two parallel processors. *Naval Res. Log. Quart.* **26**, 527–535.

ROSS, S. M. (1983) *Introduction to Stochastic Dynamic Programming.* Academic Press, New York.

SHANTHIKUMAR, J. G. AND YAO, D. D. W. (1991) Multiclass queueing systems: polymatroidal structure and optimal scheduling control. *Operat. Res.*

VAN DER HEYDEN, L. (1981) Scheduling jobs with exponential processing and arrival times on identical processors so as to minimize expected makespan. *Math. Operat. Res.* **6**, 305–312.

WALRAND, J. (1988) *An Introduction to Queueing Networks.* Prentice Hall, Englewood Cliffs, NJ.

WEBER, R. R. (1982) Scheduling jobs with stochastic processing requirements on parallel machines to minimize makespan or flowtime. *J. Appl. Prob.* **19**, 167–182.

WEBER, R. R. (1983) Scheduling stochastic jobs on parallel machines to minimize makespan or flowtime. In *Applied Probability — Computer Science: The Interface*, ed. R. Disney and T. Ott, pp. 327–338. Birkhauser, Boston, MA.

WEBER, R. R. (1988) Stochastic scheduling on parallel processors and minimization of concave functions of completion times. In *Stochastic Differential Systems, Stochastic Control Theory and Applications*, **10**, ed. W. Fleming and P. L. Lions, pp. 601–609. Springer-Verlag, New York.

WEISS, G. (1982) Multiserver stochastic scheduling. In *Deterministic and Stochastic Scheduling*, ed. M. A. H. Dempster, J. K. Lenstra, and A. H. G. Rinnooy Kan, pp. 157–179. Reidel, Dordrecht.

WEISS, G. AND PINEDO, M. (1980) Scheduling tasks with exponential service times on non-identical processors to minimize various cost functions. *J. Appl. Prob.* **17**, 187–202.