THE OPTIMALITY OF LEPT IN PARALLEL MACHINE SCHEDULING

Cheng-Shang Chang
IBM Research Division, T.J. Watson Research Center
P.O. Box 704, Yorktown Heights, NY 10598

Rhonda Righter
Department of Decision and Information Sciences
Santa Clara University, Santa Clara, CA 95053

August 8, 1992, revised May 5, 1993

Abstract

We consider preemptive scheduling on parallel machines where the number of available machines may be an arbitrary, possibly random, function of time. Processing times of jobs are from a family of DLR (decreasing likelihood ratio) distributions, and jobs may arrive at random agreeable times. We give a constructive coupling proof to show that LEPT stochastically minimizes the makespan, and that it minimizes the expected cost when the cost function satisfies certain agreeability conditions.

PARALLEL PROCESSORS, MAKESPAN, SCHEDULING, LEPT RULE, COUPLING

AMS 1991 SUBJECT CLASSIFICATION: PRIMARY 90B35, SECONDARY 68M20

1 Introduction

There are n jobs to be scheduled preemptively on a set of identical parallel machines, and idling is permitted. The number of available machines may be a random function of time, so, for example, the machines may be subject to breakdown and repair. Processing times come from a DLR family of distributions. Jobs may have random release times, but these times must be agreeable with the processing times in the sense that "longer" jobs will be released earlier, with probability one. The cost at time t is $g(U_t)$ where U_t is the set of jobs present at t (those that have been released but are unfinished). We show that if certain conditions on processing times and cost rates are satisfied, then LEPT (longest expected processing time first), that is, always processing the jobs with the smallest indices among those present, minimizes the expected cost at any time t.

Chang et al. [4] shows that LEPT is optimal under our conditions for exponential processing time and arbitrary arrivals, and Weber [16] shows that for DLR processing times, LEPT stochastically minimizes makespan. We give an alternative, constructive, proof that generalizes Weber's results to more general cost functions. Kämpke [8,10] also considers an exponential model with weaker conditions on the cost function than ours. He obtains the weaker conclusion that LEPT minimizes the expected *total* cost for costs satisfying his conditions. Kämpke does not consider nonzero release dates and machine breakdowns.

As in [4], many results in the literature are a consequence of our result. For example, LEPT stochastically minimizes the makespan (the time at which all jobs are completed) and the total amount of work at any time t. When job i has a weight c_i and exponential processing time with rate μ_i, then our conditions reduce to LEPT being consistent with the "$c\mu$ rule". In this case, LEPT minimizes the expected weighted number of jobs in system at any time t, the expected weighted sum of job completion times, the expected weighted number of late jobs when jobs have a common random due date or independent due dates with the same distribution, and the expected weighted sum of job tardiness when jobs have a common random due date or agreeable, stochastically ordered due dates. For related work on scheduling exponential jobs on parallel machines see [1-18].

In section 3, using a coupled construction, we show that LEPT stochastically minimizes the makespan. Considering the makespan gives us a simple context in which to develop the ideas of the proof. We then extend our proof in section 4 to show that LEPT minimizes the expected cost at any time t when the cost function satisfies certain conditions. In section 2 we review various stochastic ordering relations and definitions.

We use smaller, positive, increasing, etc., in the nonstrict sense.

2 Preliminaries

Let us first recall some definitions for various ordering relations of random variables, (see, e.g., Ross [6]). Let X and Y be two nonnegative random variables with distributions F and G and with densities or probability mass functions f and g. Let $\bar{F}(x) = 1 - F(x)$ and $\bar{G}(x) = 1 - G(x)$. Then we have the following.

$X \geq_{st} Y$, X is stochastically larger than Y, if $E[h(X)] \geq E[h(Y)]$ for all increasing functions h, or equivalently, if $\bar{F}(t) \geq \bar{G}(t)$ for all $t \geq 0$. Also, $X \geq_{st} Y$ if and only if it is possible to construct two coupled random variables, $\hat{X}$ and $\hat{Y}$, so that $\hat{X} =_{st} X$ and $\hat{Y} =_{st} Y$ and $\hat{X} \geq \hat{Y}$ with probability 1. It is this last definition that we use in our proof for stochastic minimization of makespan.

$X \geq_{lr} Y$, X is larger than Y in the likelihood ratio ordering, if $\frac{f(x)}{f(y)} \geq \frac{g(x)}{g(y)}$ for all $x \geq y$. If $X \geq_{lr} Y$ then $X \geq_{st} Y$. If X and Y are exponentially distributed with rates μ and λ respectively, then $X \geq_{lr} Y$ if $\mu \leq \lambda$.

We will need the following lemma, which is easily verified, for likelihood ratio ordered random variables.

Lemma 2.1 *If $X \geq_{lr} Y$, then $(X | \min(X,Y) = m, \max(X,Y) = M) \geq_{st} (Y | \min(X,Y) = m, \max(X,Y) = M)$. That is, $P\{X = m | \min(X,Y) = m, \max(X,Y) = M\} \leq P\{Y = m | \min(X,Y) = m, \max(X,Y) = M\}$, for any $m \leq M$.*

Let $X_t = (X - t | X > t)$. We say that X is decreasing in likelihood ratio (DLR) if $X_s \leq_{lr} X_t$ for all $s \leq t$. Exponentially distributed random variables are, of course, DLR (and ILR). A random variable that is DLR is also DHR (decreasing in hazard rate), i.e., $h(t)$ is decreasing, where h is the hazard rate for X. We will also need lemma 2.2, which again is easily verified.

Lemma 2.2 *If $X \geq_{lr} Y$ then $X_t \geq_{lr} Y_t$ for $t \geq 0$. Therefore, if Y is also DLR, then $X_t \geq_{lr} Y_s$, for $s \leq t$.*

We define a DLR family of distributions as a set of DLR distributions with nonoverlapping hazard rates. That is, $h_1(t) \geq h_2(s)$ for all s and t, or vice versa, where h_1 and h_2 are the hazard rates of any two distributions of the set. We may also assume jobs start with different ages, i.e., they may have already received some processing before time 0. Exponential or geometric distributions with different means are a special case. If processing times come from a DLR family of distributions, then all processing times are likelihood ratio ordered at all times, and if the processing time of job i is longer than that of job j in the likelihood ratio sense, then from lemma 2.2, this will remain the case as jobs are worked on as long as job i is worked on more than job j. @@@Also note that if X_i and X_j come from a discrete DLR family of distributions and if $X_i \geq_{lr} X_j$ and $X_i \neq_{st} X_j$, then $X_i \geq (X_j - 1 | X_j > 1)$. That is, if $X_i \neq_{st} X_j$ and X_i is longer than X_j in the likelihood ratio sense, then X_i will still be longer even after job j receives one unit of processing.@@@

Let us define our model as follows. Let X_i be the processing time and R_i the release time of job i. All processing times come from a DLR family of distributions, and release dates are random and are *agreeable* with processing times in the sense that longer jobs are released earlier. That is, if the jobs are ordered so that $X_1 \geq_{lr} X_2 \geq_{lr} \cdots \geq_{lr} X_n$, then jobs are released in the order $1, \ldots, n$, i.e., $R_1 \leq_{no} R_2 \leq_{no} \ldots R_n$, where $R_i \leq_{no} R_j$ means the distributions are nonoverlapping, $P(R_i \leq R_j) = 1$. Note that LEPT will be nonpreemptive under our assumptions, and will correspond to FCFS for jobs released after time 0.

3 LEPT Stochastically Minimizes Makespan

In this section our objective is to stochastically minimize the makespan on parallel machines. We will show that for a system that uses any policy that does not agree with LEPT, we can construct a coupled system that uses a different policy such that the makespan under the new policy is smaller with probability 1. We will assume that we have a discrete time system; a limiting argument extends the results to the continuous case. We also assume, for simplicity, that all jobs are initially present. It is easy to extend the proof to agreeable release dates.

Lemma 3.1 *For any policy π which idles a machine when jobs are available, there exists a policy π' that does not idle the machine such that π' has stochastically smaller makespan.*

Proof. The proof follows by a straightforward coupling argument. □

Let a_j be the age of job j, that is the amount of processing received prior to the start of the problem. Let $X_{j,t}$ be the processing time for job j assuming it has already been worked on for t time units, i.e., $X_{j,t} = (X_j - t | X_j > t)$. That is, $X_{j,t}$ is the remaining processing time of job j when the job's age is $a_j + t$. Let $X_j = X_{j,0}$.

Theorem 3.2 *(i) Suppose that @@@$X_1 \geq_{lr} \cdots \geq_{lr} X_i \geq_{lr} X_{i+1} \geq_{lr} X_{i+2} \geq_{lr} \cdots \geq_{lr} X_n$@@@, and X_j is from a DLR family of distributions for $j = 1, \ldots, i-1, i+1, \ldots, n$. Suppose policy π schedules job $i+1$ and not job i at time 0, and @@@schedules jobs in the order $1, 2, ..., n$@@@ from time 1 on. Then there exists a policy π' that schedules job i and not job $i+1$ at time 0, such that π' has a stochastically smaller makespan than π.*

(ii) Suppose that @@@$X_1 \geq_{lr} X_2 \geq_{lr} X_3 \geq_{lr} \cdots \geq_{lr} X_n$*@@@, and* X_j *is from a DLR family of distributions for* $j = 2, \ldots, n$*. Suppose policy* π *schedules job 2 and not job 1 at time 0, and @@@schedules jobs in the order* $1, 2, ..., n$*@@@ from time 1 on. Define policy* $\tilde{\pi}$ *as the policy that schedules job 1 and not job 2 at time 0, and otherwise agrees with* π *at time 0, and then @@@schedules jobs in the order* $1, 2, ..., n$*@@@ from time 1 on. Then* $\tilde{\pi}$ *will yield a stochastically smaller makespan than* π*.*
(iii) LEPT stochastically minimizes the makespan among all preemptive policies when processing times come from a DLR family of distributions.

Proof. Our proof is by (multiple) induction on a finite time horizon, T. We consider the truncated makespan, $\min(makespan, T)$. (We will continue to use the term makespan for truncated makespan.) These results hold trivially for $T = 0$. Assume they hold for $T - 1$ and consider the case for T. Let the current time be 0.

(i) Let S be the system that uses π. Let s be the first time π schedules job i. By assumption, after time 1 π @@@schedules jobs in the order $1, 2, ..., n$@@@ so under π job $i + 1$ will not be processed again before time s. We will construct π', with corresponding system S' and processing times X'_j, $j = 1, \ldots, n$, as follows. Let S' be in the same starting state as S, and let us directly couple the processing times of all jobs besides i and $i + 1$ in the two systems. That is, let $X'_j = X_j$ for $j \neq i, i + 1$. Let us also couple machine availabilities directly. Define π' as follows. At time 0, π' agrees with π except that π' schedules job i instead of job $i + 1$. From time 1 on π' @@@schedules jobs in the order $1, 2, ..., n$@@@, except that it does not schedule job i again before time s, and it schedules job $i + 1$ instead of i at time s. @@@(It may schedule both jobs i and $i + 1$ at time s.)@@@ then π' also schedules both at time s.) Note that from time 1 to time $s - 1$, π' will agree with π since in this interval both schedule only jobs with index less than i, and these jobs have the same processing times in both systems.

We consider three events, which we couple directly in the two systems:

(A) $X_i = X_{i+1} = 1$ (and $X'_i = X'_{i+1} = 1$).

(B) $X_i > 1$ and $X_{i+1} > 1$ (and $X'_i > 1$ and $X'_{i+1} > 1$).

(C) Exactly one of X_i and X_{i+1} equals 1 (and exactly one of X'_i and X'_{i+1} equals 1).

Let $\mu_k = h_k(0)$ be the hazard rate of job k at time 0. If event A occurs (with probability $\mu_i \mu_{i+1}$), both systems will be in the same state at time $s + 1$, and their makespans will be identical.

If event B occurs (with probability $(1 - \mu_i)(1 - \mu_{i+1})$), and if π does not also schedule job $i + 1$ at time s (so π' does not schedule job i at time s), then at time $s + 1$ both systems will again be in the same state and their makespans will be identical. Therefore, suppose that event B occurs and that π and π' schedule both jobs i and $i+1$ at time s. Since the processing times of jobs with indices lower than i are coupled, and @@@these jobs have priority over jobs i and $i+1$ under both policies,@@@ we can assume the machines on which lower indexed jobs are scheduled from time s on are simply unavailable and that those jobs have already finished. The machines will "randomly" become available when the jobs finish, at coupled times in the two systems. Let us call the machine that π schedules i on at time s machine a, and let π' schedule $i+1$ on a at time s. Then from time s on, system S is equivalent to a system in which the machines with lower indexed jobs are unavailable and the lower indexed jobs are finished, machine a is unavailable at time s, the current ages of jobs i and $i+1$ are a_i+1 and $a_{i+1}+1$ respectively, the states of the other jobs are as in S, and @@@jobs will be scheduled in the order $1, 2, ..., n$@@@ from time s on, except that at time s job i is not scheduled. The same is true for system S', except that it is job $i+1$ that is not scheduled at time s. By lemma 2.2 and the induction hypothesis (ii), the latter system will have stochastically smaller makespan than the former.

Finally consider event C (which occurs with probability $\mu_i + \mu_{i+1} - 2\mu_i\mu_{i+1}$). That is, exactly one of the two jobs requires only one unit of processing in *both* systems, but we permit the *identity* of the shorter job to be different in the two systems. We first generate and couple

$$M = (\max(X_i, X_{i+1}) |\min(X_i, X_{i+1}) = 1, \max(X_i, X_{i+1}) > 1)$$

for the two systems. Let

$$\begin{aligned} q_M &= P\{X_i = 1 | \min(X_i, X_{i+1}) = 1, \max(X_i, X_{i+1}) = M\}, \\ p_M &= 1 - q_M = P\{X_{i+1} = 1 | \min(X_i, X_{i+1}) = 1, \max(X_i, X_{i+1}) = M\}. \end{aligned}$$

Since $X_i \geq_{lr} X_{i+1}$, we have $q_M \leq p_M$, using lemma 2.1 with $m = 1$. Now we couple the processing times of jobs i and $i+1$ as follows.

(C1) $X_i = X'_{i+1} = 1$ and $X_{i+1} = X'_i = M$ (with probability q_M),

(C2) $X_i = X'_{i+1} = M$ and $X_{i+1} = X'_i = 1$, (with probability q_M),

(C3) $X_i = X'_i = M$ and $X_{i+1} = X'_{i+1} = 1$ (with probability $p_M - q_M$).

Note that from (C1-C3), the marginals have the correct distribution, that is, conditioned on $m = 1$ and M, $P\{X_i = 1\} = P\{X'_i = 1\} = q_M$ and $P\{X_{i+1} = 1\} = P\{X'_{i+1} = 1\} = p_M$.

@@@Since jobs i and $i+1$ are "next to each other" under both policies, if only one of these two jobs is present it will be treated the same under both policies@@@ regardless of which one it is. Therefore, in cases C1 and C2 the two systems will have identical departure times and makespans.

In case C3, if π' does not schedule job i as well as job $i+1$ at time s, then at time $s+1$ both systems will be in the same states, and again the two systems will have identical makespans. Therefore, let us suppose that π' schedules job i as well as job $i+1$ at time s. Then, since job $i+1$ is gone at time s in S, it must be that π schedules the same jobs as π' at time s except that it schedules jobs i and j, instead of jobs i and $i+1$, where $j > i+1$ (or it idles a machine, in which case the result follows from lemma 3.1). Suppose that at time s π schedules i and π' schedules $i+1$ on machine a. From time s on system S is equivalent to a system in which the machines with lower indexed jobs are unavailable and the lower indexed jobs are finished, machine a is unavailable at time s, the current age of job i is a_i+1, job $i+1$ has finished, the states of the other jobs are as in S, and @@@jobs will be scheduled in the order $1, 2, ..., n$@@@ from time s on except that at time s job i is not scheduled and j is scheduled instead. The same is true for system S', except that job i is scheduled instead of job j at time s. Now we will show, through a sequence of intermediate policies, that the result follows from the induction hypothesis (ii).

@@@Now at time s, it must be that, excluding machine a at time s, and the other machines that are unavailable because of lower indexed jobs, π schedules jobs $i+2, \ldots, i+k-1, i+k$, where $j = i+k$, and π' schedules jobs $i, i+2, \ldots, i+k-1$. Let $\pi_1 = \pi$, $\pi_k = \pi'$, and let π_2 schedule jobs $i, i+3, \ldots, i+k$ at time s and then schedule jobs in the order $1, 2, ..., n$, and let π_l agree with π_{l-1} at time s except that π_l schedules job $i+l-1$ instead of job $i+l$. Define S_l, $l = 1, \ldots, k$ as the corresponding systems. Since job $i+2$ has received at most one unit of processing by time s (possibly at time 0), the remaining processing time for job $i+2$ at time s is either X_{i+2} or $X_{i+2,1}$. Let $\hat{X}_i := (M-1|\text{ case C3})$ be the remaining processing time for job i at time s in case C3. Then, from lemma 3.3 (below) and the induction hypothesis (ii), S_2 has stochastically smaller makespan than $S_1 = S$. For each π_l, $l = 2, ..., k$, job i will have precedence over jobs $i+2, \ldots, n$ for $l = 2, \ldots, k$, from time s on.@@@ Let us couple $\hat{X}_i$ under policies π_l and π_{l+1} directly, for $l = 2, \ldots, k-1$. Then for $l = 2, \ldots, k-1$, system S_l from time s on is equivalent to a system in which only jobs $i+l-1, \ldots, n$ are uncompleted, and the machines that are processing lower indexed jobs are unavailable. Then, from the induction hypothesis (ii), S_{l+1} has stochastically smaller makespan than S_l.

(ii) From the induction hypothesis (i), there exists a policy, π', that agrees with π at time 0, except for scheduling job 1 instead of job 2 at time 0, such that π' has stochastically smaller makespan than π. By its construction, π' will schedule job 2 instead of 1 at time 1 (or the first time a machine is available), and otherwise @@@schedule jobs in the order $1, 2, ..., n$@@@. It may also schedule job 1 at that time, in which case π' is the same policy as $\tilde{\pi}$. If it does not, by the induction hypothesis (ii) @@@and lemma 2.2@@@, $\tilde{\pi}$ will have stochastically smaller makespan than π'.

(iii) Assume jobs are ordered in LEPT order. Let π be an arbitrary policy, with system S. @@@We will show that LEPT has stochastically smaller makespan than π.@@@ If π does not agree with LEPT from time 1 on, construct a new policy, π', that agrees with π at time 0 and follows LEPT thereafter. Then, by the induction hypothesis (iii), the makespan will be stochastically smaller under π' than under π. Therefore, without loss of generality, we suppose that from time 1 on π agrees with LEPT. If π also agrees with LEPT at time 0, then we are done. Therefore, suppose that at time 0 π does not agree with LEPT. @@@Also suppose, without loss of generality, that for all j and k, if $X_j =_{st} X_k$ and only one of jobs j and k is scheduled under π or LEPT, the lowest indexed job of j and k is scheduled. Then, @@@if either π or LEPT was followed at time 0, jobs will still be in LEPT order at time 1, since processing times come from a discrete DLR family of distributions. Therefore,@@@ from time 1, under both LEPT and π, jobs will be scheduled in the order $1, ..., n$. Let job i be a job such that i is not scheduled under π at time 0 but job $i+1$ is scheduled by π at time 0. (By assumption $X_i \neq_{st} X_{i+1}$.) From the induction hypothesis (i), there exists a policy π' that schedules job i and not job $i+1$ at time 0, such that π' has stochastically smaller makespan than π. From the induction hypothesis @@@(iii)@@@, the policy that agrees with π' at time 0 and follows LEPT from time 1 on will have stochastically smaller madespan than π'. By successively repeating the argument we obtain the result.@@@ □

Lemma 3.3 $\hat{X}_i := (M-1|\text{case C3}) \geq_{lr} X_{i+1,1} \geq_{lr} X_{i+2,1}$@@@ $\geq_{lr} X_{i+2}$ @@@.

Proof. The second inequality follows from lemma 2.2, @@@and the third from the fact that $X_i + 2$ is DLR. Let f_l be the probability mass function for X_l, $l = i, i+1$, (so $f_l(1) = \mu_l$) and let g be the probability mass function for $\hat{X}_i$, i.e.,

$$g(n) = P\{\hat{X}_i = n\} = P\{M = n+1| \text{ case C3 }\}.$$

We must show that $g(N)/g(n) \geq f_{i+1}(N+1)/f_{i+1}(n+1)$ for all $N \geq n$, using the fact that $X_i \geq_{lr} X_{i+1}$ and X_{i+1} is DLR. From the definition of case C3 we have the following.

$$p_M - q_M = \frac{f_i(M)f_{i+1}(1) - f_i(1)f_{i+1}(M)}{f_i(M)f_{i+1}(1) + f_i(1)f_{i+1}(M)}$$

$$\begin{aligned} P\{M = n+1 \text{ and case C3}\} &= [f_i(n+1)f_{i+1}(1) + f_i(1)f_{i+1}(n+1)](p_{n+1} - q_{n+1}) \\ &= f_i(n+1)f_{i+1}(1) - f_i(1)f_{i+1}(n+1) \end{aligned}$$

$$
\begin{aligned}
P\{\mathrm{C3}\} &= \sum_{M=2}^{\infty} f_i(M) f_{i+1}(1) - f_i(1) f_{i+1}(M) \\
&= f_{i+1}(1)(1 - f_i(1)) - f_i(1)(1 - f_{i+1}(1)) \\
&= f_{i+1}(1) - f_i(1) = \mu_{i+1} - \mu_i.
\end{aligned}
$$

Therefore,

$$
g(n) = \frac{f_i(n+1) f_{i+1}(1) - f_i(1) f_{i+1}(n+1)}{f_{i+1}(1) - f_i(1)}
$$

and

$$
\begin{aligned}
\frac{g(N)}{g(n)} &= \frac{f_i(N+1) f_{i+1}(1) - f_i(1) f_{i+1}(N+1)}{f_i(n+1) f_{i+1}(1) - f_i(1) f_{i+1}(n+1)} \\
&= \frac{\frac{f_i(N+1)}{f_i(n+1)} \frac{f_i(n+1)}{f_i(1)} - \frac{f_{i+1}(N+1)}{f_{i+1}(n+1)} \frac{f_{i+1}(n+1)}{f_{i+1}(1)}}{\frac{f_i(n+1)}{f_i(1)} - \frac{f_{i+1}(n+1)}{f_{i+1}(1)}} \\
&\geq \frac{\frac{f_{i+1}(N+1)}{f_{i+1}(n+1)} \left[\frac{f_i(n+1)}{f_i(1)} - \frac{f_{i+1}(n+1)}{f_{i+1}(1)} \right]}{\frac{f_i(n+1)}{f_i(1)} - \frac{f_{i+1}(n+1)}{f_{i+1}(1)}} = \frac{f_{i+1}(N+1)}{f_{i+1}(n+1)}.
\end{aligned}
$$

□

Note that $\hat{X}_i$ is not necessarily DLR.

4 General Cost Functions

We now extend our proof of theorem 3.2 to permit more general cost functions of the jobs present at time t, $g(U_t)$. Let $U^i = U - \{i\}$ and $U^{ij} = U - \{i, j\}$, and let $h_i(t)$ be the hazard rate of job i at time t. Assume processing times come from a DLR family of distributions. Consider the following conditions on g and X_i, $i = 1, \ldots, n$.

(A1) $X_i \geq_{lr} X_j \quad \forall i < j$,

(A2) $g(U) \geq g(U^i) \quad \forall i$,

(A3) $h_i(s)[g(U) - g(U^i)] \geq h_j(t)[g(U) - g(U^j)] \quad \forall s, t \; \forall U, \; i, j \in U, i < j$.

Note that a special case is when $h_i(s) = \mu_i \; \forall s$, (processing times are exponential or geometric) and $g(U) - g(U^i) = c_i$, $i = 1, ..., n$. For the makespan case, $g(U) = 1$ if U is not empty, so condition (A3) holds trivially.

Theorem 4.1 *When processing times are from a DLR family, and processing times and the cost function satisfy conditions (A1-3), then LEPT minimizes the expected cost at any time t among all preemptive policies.*

Proof. Our proof follows as before, with minor changes to Theorem 3.2(i). (Lemma 3.1 follows from condition (A2)).We use the same definitions of π, π', S, S', i, and s, and the same coupling of events. Let Δg_t be the cost for system S minus the cost for S' at time t. We must show $E[\Delta g_t] \geq 0$ for all $t \geq 1$.

First consider the case $t > s$. If event A occurs, the systems are identical and $E[\Delta g_t|\text{A}] = 0$. If either event B or event C3 occurs, then, depending on whether both jobs i and $i+1$ are scheduled at time s or not, either the two systems are identical, or the result follows from the induction hypothesis as before. Events C1 and C2 have the same probability, and $E[\Delta g_t|\text{C1}] = -E[\Delta g_t|\text{C2}]$, so $E[\Delta g_t \text{ and (C1 or C2)}] = 0$. We therefore have the result for $t > s$.

Now consider $1 \leq t \leq s$. Let V_t be the set of jobs present at time t in either system, $t \leq s$, excluding jobs i and $i+1$. (Because of our coupling, V_t will be the same for both systems.) Again, $\mu_k = h_k(0)$ is the hazard rate of job k at time 0. Then

$$\begin{aligned} E[\Delta g_t] &= \mu_i\mu_{i+1}[g(V_t \cup \{i\}) - g(V_t \cup \{i+1\})] \\ &\quad +(1-\mu_i)(1-\mu_{i+1})0 \\ &\quad +\mu_i(1-\mu_{i+1})0 \\ &\quad +\mu_i(1-\mu_{i+1})[g(V_t \cup \{i\}) - g(V_t \cup \{i+1\})] \\ &\quad +(\mu_{i+1}-\mu_i)[g(V_t \cup \{i\}) - g(V_t \cup \{i, i+1\})], \end{aligned}$$

where the lines of the equation correspond to event A, B, C1, C2, and C3, respectively. After some algebra we have

$$E[\Delta g_t] = \mu_i[g(V_t \cup \{i, i+1\}) - g(V_t \cup \{i+1\})] - \mu_{i+1}[g(V_t \cup \{i, i+1\}) - g(V_t \cup \{i\})] \geq 0$$

where the inequality follows from A3. □

Acknowledgements

We would like to thank the referee for his or her careful reading, and for pointing out an error in our original approach.

References

[1] Bruno, J., and P. Downey (1977). Sequencing tasks with exponential service times on two machines. Technical Report. EECS, UC Santa Barbara.

[2] Bruno, J., P. Downey, and G. Frederickson (1981). Sequencing tasks with exponential service times to minimize the expected flow time or makespan. *J. Assoc. Comp. Mach.* 28: 100-113.

[3] Chang C.-S. (1992). A new ordering for stochastic majorization: Theory and applications. *Adv. Appl Prob.* 24: 604-634.

[4] Chang, C.-S., X. L. Chao, M. Pinedo, and R. R. Weber (1992). On the optimality of LEPT and $c\mu$ rules for machines in parallel. *J. Appl. Prob.* 29:667-681.

[5] Glazebrook, K. D. (1979). Scheduling tasks with exponential service times on parallel processors. *J. Appl. Prob.* 16: 685-689.

[6] Glazebrook, K. D., and P. Nash (1977). On multi-server stochastic scheduling. *J. Roy. Stat. Soc., B* 38: 67-72.

[7] Hordijk, A., and G. Koole (1991). Optimal policies in two stochastic scheduling problems with arrivals of customers. Preprint.

[8] Kämpke, T. (1987). On the optimality of static priority policies in stochastic scheduling on parallel machines. *J. Appl. Prob.* 24: 430-448.

[9] Kämpke, T. (1987). Necessary optimality conditions for priority policies in stochastic weighted flowtime scheduling problems. *Adv. Appl. Prob.* 19: 749-750.

[10] Kämpke, T. (1989). Optimal scheduling of jobs with exponential service times on identical parallel processors. *Opns. Res.* 37: 126-133.

[11] Nash, P., and R. R. Weber (1982). Sequential open-loop scheduling strategies. *Deterministic and Stochastic Scheduling.* M. A. H. Dempster, ed., D. Reidel. 385-397.

[12] Pinedo, M. (1983). Stochastic scheduling with release dates and due dates. *Opns. Res.* 31: 559-572.

[13] Ross, S. M. (1983) *Stochastic Processes.* New York: J. Wiley & Sons.

[14] Ross, S. M. (1983) *Introduction to Stochastic Dynamic Programming.* New York: Academic Press.

[15] Van der Heyden, L. (1981). Scheduling jobs with exponential processing and arrival times on identical processors so as to minimize the expected makespan, *Math. of Oper. Res.* 6: 305-312.

[16] Weber, R. R. (1982). Scheduling jobs with stochastic processing requirements on parallel machines to minimize makespan or flowtime. *J. Appl. Prob.* 19: 167-182.

[17] Weiss, G., and M. Pinedo (1980). Scheduling tasks with exponential service times on non-identical processors to minimize various cost functions. *J. Appl. Prob.* 17: 187-202.

[18] Weiss, G. (1982). Multiserver stochastic scheduling. *Deterministic and Stochastic Scheduling* M. A. H. Dempster, ed. D. Reidel. 157-179.