STOCHASTIC SCHEDULING ON PARALLEL PROCESSORS AND MINIMIZATION OF CONCAVE FUNCTIONS OF COMPLETION TIMES

RICHARD R. WEBER
QUEENS' COLLEGE, UNIVERSITY OF CAMBRIDGE
CAMBRIDGE CB3 9ET, U. K.

Abstract. We consider a stochastic scheduling problem in which n jobs are to be scheduled on m identical processors which operate in parallel. The processing times of the jobs are not known in advance but they have known distributions with hazard rates $\rho_1(t), \dots, \rho_n(t)$. It is desired to minimize the expected value of $\kappa(C)$, where C_i is the time at which job i is completed $C = (C_1, \dots, C_n)$, and $\kappa(C)$ is increasing and concave in C. Suppose processor i first becomes available at time τ_i. We prove that if there is a single static list priority policy which is optimal for every $\tau = (\tau_1, \dots, \tau_m)$, then the minimal expected cost must be increasing and concave in τ. Moreover, if $\kappa(C)$ is supermodular in C then this cost is also supermodular in τ. This result is used to prove that processing jobs according to the static list priority order $(1, 2, \dots, n)$ minimizes the expected value of $\Sigma w_i h(C_i)$, when $h(\cdot)$ is a nondecreasing, concave function, $w_1 \geq \dots \geq w_n$, and $\rho_1(t_1)w_1 \geq \dots \geq \rho_n(t_n)w_n$ for all $t_1, \dots, t_n$.

Key words. dynamic programming, flowtime, stochastic scheduling, supermodularity

1. Stochastic scheduling on parallel processors.

Problems of stochastic scheduling on parallel processors are those in which some number of jobs with unknown processing times are to be scheduled on identical processors which operate in parallel. The processing times of the jobs are not known in advance, but they are assumed to be independent and to have known distributions with hazard rates $\rho_1(t), \dots, \rho_n(t)$, $t \geq 0$. Each job can be processed by any of the available processors. We let C_i be the time at which job i is completed. Obviously $C = (C_1, \dots, C_n)$ is a random variable which depends upon the processing times and scheduling policy employed. The problem is to schedule the processing of the jobs to minimize the expected value of a given function of the completion times, the performance measure, $\kappa(C)$. While in general problem is clearly intractible, a number of results have been obtained for the case of *exponential jobs*, in which the processing times are exponentially distributed with parameters $\lambda_1, \dots, \lambda_n$. In this case, there are a number of important performance measures which can be optimized by using a static list priority policy. Glazebrook (1979) has shown that the expected value of the flowtime, $\kappa(C) = \Sigma C_i$, is minimized by the static list priority policy SEPT. This is the policy which at every instant processes those uncompleted jobs of smallest expected remaining processing times: meaning that if $\lambda_1 \geq \dots \geq \lambda_n$ then SEPT begins the processing of jobs according to the static list priority order $1, 2, \dots, n$. The policy which processes according to the reversed priority order $n, \dots, 1$ is known as LEPT. This policy minimizes the expected value of the *makespan*, $\kappa(C) = \max(C_i)$ (Bruno, Downey and Frederickson (1981)).

These results have been strengthed in a number of ways. It has been shown that SEPT and LEPT are optimal in the class of preemptive policies, for flowtime and makespan respectively, when jobs have processing times which are identically distributed according to a distribution with monotime hazard rate, but jobs have already received different amounts of processing prior to the start (see Weber (1982)). Moreover, in some cases (including that of exponential jobs) these policies minimize flowtime and makespan in distribution, as well as in expectation. Weiss and Pinedo (1980) showed that SEPT minimizes the expected value of the weighted flowtime $\kappa(C) = \Sigma w_i C_i$, in the case where exponential jobs have hazard rates $\lambda_1 \geq \dots \geq \lambda_n$ and *agreeable weights*: $w_1 \geq \dots \geq w_n$. They also showed that SEPT minimizes expected weighted flowtime when the processors have different speeds $s_1 \geq \cdots \geq s_m$. This is known as the case of *uniform processors* and in this case SEPT is defined as the policy which assigns the uncompleted job with k'th shortest remaining expected processing time to the k'th fastest processor, $k = 1, \ldots, m$. These results have been greatly strengthed by Kampke (1985) who has shown that, irrespective of how the λ_i's are ordered, the static list priority policy, which always processes the uncompleted job of k'th smallest index on the k'th fastest processor, $k = 1, \ldots, m$, minimizes expected flowtime if $w_1 \geq \cdots \geq w_n$ and $\lambda_1 w_1 \geq \cdots \geq \lambda_1 w_n$. Let SI be this *smallest index first* static list priority policy. Clearly the definition of SI is without loss of generality since SI can denote any static list priority policy by appropriate renumbering of the jobs. The main contribution of the present paper is to generalize Kampke's result to models in which the jobs have processing times which are not exponentially distributed. In section 3 we show that SI minimizes the expected value of $\Sigma w_i h C_i$, provided $h(c)$ is nondecreasing and concave in c, $w_1 \geq \cdots \geq w_n$ and $\rho_1(t_1)w_1 \geq \cdots \geq \rho_n(t_n)w_n$ for all $t_1, \ldots, t_n$.

We also discuss more general performance measures. Our discussion is motivated by other recent work of Kampke (1987) who has studied cost functions of the following form. Suppose the jobs are completed in the order $i_1, \ldots, i_n$ at times $C_{i_1} \leq \cdots \leq C_{i_n}$ and $I = \{1, \ldots, n\}$ denotes the set of indices of all n jobs. Let

$$\begin{aligned}\kappa(C) = g(I)C_{i_1} + g(I - \{i_1\})(C_{i_2} - C_{i_1}) + g(I - \{i_1, i_2\})(C_{i_3} - C_{i_2}) \\ + \cdots + g(I - \{i_1, \ldots, i_{n-1}\})(C_{i_n} - C_{i_{n-1}}).\end{aligned}$$

Then $\kappa(C)$ may be interpreted as the total holding cost incurred, when a holding cost $g(U)$ is charged for each unit of time that the set of uncompleted jobs is U. $\kappa(C)$ is said to be *Markovian* because the contribution to the cost arising from events after time t depends only on the set of jobs uncompleted at time t. The weighted flowtime is a simple example of a Markovian cost function. It is the total holding cost incurred when a holding cost w_i is charged for each unit of time that job i is uncompleted.

Kampke considered cost functions that were of the above form and also nondecreasing and concave in C. The conditions that $\kappa(C)$ be nondecreasing and concave in C are easily seen to be equivalent to

(1) $$g(U) - g(U - \{i\}) \geq 0, \text{ for all } U \text{ and } i \in U, \text{ and}$$

(2) $$g(U) - g(U - \{i\}) - g(U - \{j\}) + g(U - \{i, j\} \geq 0, \text{ for all } U \text{ and } i, j \in U,$$

Statement (2) is the condition that $g(\cdot)$ be a supermodular set function (see Lovasz (1983)). It is also equivalent to the statement that $\kappa(c)$ be supermodular in c, in the sense that

$$\kappa(cVd) - \kappa(c) - \kappa(d) + \kappa(c\Lambda d) \geq 0, \text{ for all } c, d \geq 0,$$

where the maximums and minimums of the vectors c and d are taken componentwise and denoted cVd and $c\Lambda d$ respectively.

Kampke has demonstrated that SI minimizes the expected value of a nondecreasing, concave Markovian cost function, for exponential jobs on uniform processors, provided that for all $i < j$, all sets of uncompleted jobs U which contain i and j, and all t_1, t_2, one has $g(U - \{j\}) \geq g(U - \{i\})$ and $\lambda_i(g(U) - g(U - \{i\})) \geq \lambda_j(g(U) - g(U - \{j\}))$. It is an unresolved question whether this result can be generalized to models in which jobs have processsing times which are not exponentially distributed. Of course we conjecture that sufficient conditions are simply that for all U, $i,j \in U$, $i < j$, and t_1, t_2 one should have $g(U - \{j\}) \geq g(U - \{i\})$ and $\rho_i(t_1)(g(U) - g(U - \{i\})) \geq \rho_j(t_2)(g(U) - g(U - \{j\}))$. But we have only been able to prove this conjecture for the special case mentioned above.

However, we shall prove one new result about general cost functions. Suppose processors $1, \ldots, m$ first become available at time $\tau_1, \ldots, \tau_m$ respectively, and that for every τ the policy SI minimizes the expected value of $\kappa(C)$. Then the minimal expected cost inherits any properties of being nondecreasing, concave and supermodular in τ which are properties of $\kappa(C)$ as a function of C. This result is proved in section 2. It is used in section 3 to prove that SI minimizes the expected value of $\Sigma w_i h(C_i)$ under the conditions mentioned above. Section 4 contains some discussion of these results.

2. Concavity and supermodularity properties of the minimal cost. In this section we consider a stochastic scheduling problem on m identical parallel processors in which it is desired to minimize the expected value of a function of the completion times, $\kappa(C_1, \ldots, C_n)$. In fact, we suppose there are a number of cost functions, $\kappa(C, \alpha)$, parameterized by a real-valued vector α. Suppose that for every fixed α the same static list priority policy is optimal. For the moment we do not specify whether this policy is optimal within the class of preepmptive policies or within the class of nonpreemptive policies. In either case the theorem of this section will hold. Suppose that not all processors are necessarily available from the start. Processor i does not become available until time $\tau_i \geq 0$. Given a particular value of τ, let $\kappa(\tau, \alpha)$ be the minimal expected cost. Here, τ is just another indexing parameter which it is convenient to denote separately from α.

THEOREM 1. *Suppose that the same static list priority policy is optimal for every (τ, α), $\tau \geq 0$. Then the minimal expected cost $K(\tau, \alpha)$ inherits any properties of being nondecreasing, concave, or supermodular in (τ, α) which are properties of $\kappa(C, \alpha)$ as a function of (C, α).*

Proof. Without loss of generality we can suppose the optimal policy is SI. This is the policy which prioritises the jobs in increasing order of their indices. Whenever an opportunity arises to reassign jobs to processors SI always assigns processing to those uncompleted jobs of smallest indices. We shall prove the inheritance of concavity first. The proof is by induction on n. It is trivial for n=0. Assuming the theorem is true when less than n jobs are uncompleted, the inductive step is established using the following identity.

$$\begin{aligned} (3) \qquad K(\tau, \alpha) = \min[&E\{K^1((\tau_1 + Y, \tau_2, \ldots, \tau_m), \tau_1 + Y, \alpha)\}, \\ &E\{K^1((\tau_1, \tau_2 + Y, \ldots, \tau_m), \tau_2 + Y, \alpha)\}, \ldots, E\{K^1((\tau_1, \tau_2, \ldots, \tau_m + Y), \tau_m + Y, \alpha)\}]. \end{aligned}$$

Here Y is the processing time required by job 1. $K^1(\tau, c_1, \alpha)$ denotes the minimal expected cost given that processors become available at times $\tau_1, \ldots, \tau_m$, that only jobs $2, \ldots, n$ need be completed, and that the cost function which remains to be minimized is $\kappa((c_1, C_2, \ldots, C_n)\alpha)$. This cost function is concave in $((c_1, C_2, \ldots, C_n), \alpha)$ and by hypothesis is minimized by L for every value of the parameters (c_1, α). Therefore, employing the inductive hypothesis, each of the m terms within expectations in (3) is concave in (τ, α) for every fixed Y.

Now consider the justification of identity (3). By hypothesis, it is optimal to assign job 1 to the first available processor. So, for example, if processor 1 is the first available, then the first term within the minimum in (3) is equal to $K(\tau, \alpha)$. Moreover, no other term within the minimum is less than $K(\tau, \alpha)$, for the value of the j'th term could be achieved by a realizable policy which holds back the processing of job 1 until the j'th processor becomes free, but otherwise processes the jobs according to the priority order $2, \dots, n$. It follows that $K(\tau, \alpha)$ is concave in (τ, α) for the result of taking expectations and then a minimum of concave functions is always a concave function.

That $K(\tau, \alpha)$ will be nondecreasing in (τ, α) if $\kappa(C, \alpha)$ is nondecreasing in (C, α) follows trivially from (3). To establish that $K(\tau, \alpha)$ inherits supermodularity, if this is a property of $\kappa(C, \alpha)$, it is sufficient to show that $K((\tau_1', \tau_2, \dots, \tau_m), \alpha) - K((\tau_1, \tau_2, \dots, \tau_m), \alpha)$ is increasing in τ_2 for $\tau_1' > \tau_1$. The proof is by induction on n, with the case $n = 0$ holding by the supermodularity of $\kappa(C, \alpha)$. To establish a step of the induction, assuming the result is true when less than n jobs remain to be completed, we must check a number of cases. The explanation is simplified if we suppose τ and all processing times take only integer values. The assumption is without loss of generality, so let us assume $\tau_1' = \tau_1 + 1$ and $\tau_2' = \tau_2 + 1$. We need to check that

$$\begin{aligned} (4) \qquad & \{ K((\tau_1 + 1, \tau_2 + 1, \dots, \tau_m), \alpha) - K((\tau_1, \tau_2 + 1, \dots, \tau_m), \alpha) \} \\ & \quad - \{ K((\tau_1 + 1, \tau_2, \dots, \tau_m), \alpha) - K((\tau_1, \tau_2, \dots, \tau_m), \alpha) \} \geq 0. \end{aligned}$$

There are a number of cases to consider. Without loss of generality, let τ_m be one of the smallest of $\tau_3, \dots, \tau_m$. Let Y be the processing time of job 1. If $\tau_m \leq \tau_1, \tau_2$ then job 1 can optimally be assigned to processor m and the left hand side of (4) is

$$\begin{aligned} & E[\{ K((\tau_1 + 1, \tau_2 + 1, \dots, \tau_m + Y), \tau_m + Y, \alpha) \\ & - K((\tau_1, \tau_2 + 1, \dots, \tau_m + Y), \tau_m + Y, \alpha) \} \\ & - \{ K((\tau_1 + 1, \tau_2, \dots, \tau_m + Y), \tau_m + Y, \alpha) \\ & - K((\tau_1, \tau_2, \dots, \tau_m + Y), \tau_m + Y, \alpha) \}] \end{aligned}$$

which is nonnegative by the inductive hypothesis. If $\tau_1 < \tau_m$ and $\tau_1 + 1 \leq \tau_2$ then a similar expression holds with job 1 assigned to processor 1. If $\tau_2 < \tau_m$, $\tau_2 + 1 \leq \tau_1$ then a similar expression holds with job 1 assigned to processor 2. The remaining case is $\tau_1 = \tau_2 < \tau_m$. The left hand side of (4) is

$$\begin{aligned} & E[\{ K((\tau_1 + 1 + Y, \tau_1 + 1, \dots, \tau_m), \tau_1 + 1 + Y, \alpha) \\ & - K((\tau_1 + Y, \tau_1 + 1, \dots, \tau_m), \tau_1 + Y, \alpha) \} \\ & - \{ K((\tau_1 + 1, \tau_1 + Y, \dots, \tau_m), \tau_1 + Y, \alpha) \\ & - K((\tau_1 + Y, \tau_1, \dots, \tau_m), \tau_1 + Y, \alpha) \}]. \end{aligned}$$

However, the third term in the above, $E[K((\tau_1 + 1, \tau_1 + Y, \dots, \tau_m), \tau_1 + Y, \alpha)]$, is not more than $E[K((\tau_1 + 1 + Y, \tau_1, \dots, \tau_m), \tau_1 + 1 + Y, \alpha)]$ and making this substitution we deduce that the above in nonegative by using the inductive hypothesis. This completes the proof of the theorem. □

It is interesting to consider an intuitive explanation of theorem 1. Concavity of $K(\tau)$ in τ_1 is equivalent to the statement that $K(\tau_1 + \delta, \tau_2, \dots, \tau_m) - K(\tau_1, \tau_2, \dots, \tau_m)$ decreases in τ_1. Supermodularity is equivalent to the statement that this increases in τ_2. When the cost function is the flowtime, ΣC_i, our intuition suggests

$$\begin{aligned} & K(\tau_1 + \delta, \tau_2, \dots, \tau_m) - K(\tau_1, \tau_2, \dots, \tau_m) \\ & \qquad\qquad = \delta E[\text{ number of jobs processed on processor 1 }] + o(\delta). \end{aligned}$$

Moreover, it is intuitive that as τ_1 increases, or τ_2 decreases, the expected number of jobs that are processed on processor 1 decreases.

That this intuition is correct is part of the proof given by Weber, Varaiya and Walrand (1986) that nonpreemptive SEPT minimizes expected flowtime on parallel processors when the jobs have processing times which are stochastically ordered as $Y_1 \leq_{st} \cdots \leq_{st} Y_n$. The more general result that is stated here can be used to simplify that proof. The application of the theorem which we shall use in the next section is (b) of the following corollary.

COROLLARY 1. *Let $K(\tau)$ be the minimal cost achieved by the single static list priority policy which is optimal for all $\tau \geq 0$. Then*

(a) $K(\tau_1, \tau_2+1, \tau_3, \ldots, \tau_m) - K(\tau_2, \tau_1+1, \tau_3, \ldots, \tau_m)$ is nondecreasing in τ_1 and nonincresing in τ_2, and

(b) if two random variables Y_1, Y_2 are stochastically ordered $Y_1 \leq_{st} Y_2$, then $E\{\, K(\tau_1+Y_1, \tau_1+1+Y_2, \tau_3, \ldots, \tau_m) - K(\tau_1+Y_2, \tau_1+1+Y_1, \tau_3, \ldots, \tau_m)\} \leq 0$.

Proof. The expression in (a) is

$$\{\, K(\tau_1, \tau_2+1, \tau_3, \ldots, \tau_m) - K(\tau_1, \tau_2, \tau_3, \ldots, \tau_m)\,\} \\ + \{\, K(\tau_2, \tau_1, \tau_3, \ldots, \tau_m) - K(\tau_2, \tau_1+1, \tau_3, \ldots, \tau_m)\,\}$$

The two bracketed terms are nondecreasing in τ_1, by supermodularity and concavity respectively. Similarly, they are nonincreasing in τ_2. Part (b) follows from (a), by replacing τ_1 and τ_2 in (a) by τ_1+Y_1 and τ_1+Y_2 respectively and taking the expected value.

3. Minimizing the expected value of a concave cost function. The following section is concerned with establishing conditions under which the expected value of the cost function $\Sigma w_i h(C_i)$ is minimized by a static list priority policy. We assume that $h(c)$ is nondecreasing and concave in c. Without loss of generality we also suppose $h(0) = 0$. Suppose that we can establish conditions under which a single static list priority policy minimizes the expected value of the *truncated* weighted flowtime, $\Sigma w_i \min(C_i, t)$, for all t. Then we can express $h(c)$ as $h(c) = E[\min(c, T)] = \int_0^c P(T \geq t)\, dt$, where T is a random variable. It follows that the given static list priority policy minimizes the expected value of $\Sigma w_i h(C_,)$.

We adopt a discrete time formulation. While the result can be proved in a continuous time setting (as in Weber (1982)), a discrete time exposition eliminates the need for a digression on continuous time dynamic programming. As in Weber and Nash (1979), we suppose that time proceeds in discrete steps $s = 0, 1, \ldots$. *Interval s* is $[s, s+1)$ If job i has received x_i intervals of processing it is said to be of *age* x_i, and it will be completed at then end of its next interval of processing with probability $p_i(x_i)$. The age of job i at time 0 may be nonzero if there has been some processing of job i prior to time 0. If job i has been completed we denote this by setting $x_i = \infty$. Holding cost w_i is charged at time s for each job i which is uncompleted at time s. The truncated weighted flowtime is equivalent to the total holding cost incurred by time t. There are $m(s)$ processors available at time s, where $m(s)$ is integer-valued and nondecreasing in s. When other parameters are fixed, the pair (x, s) is a state variable for the problem. SI is the static list priority policy which processes jobs of smallest index first.

THEOREM 2. *Suppose that $w_1 \geq \cdots \geq w_n$ and $\rho_1(t_1)w_1 \geq \cdots \geq \rho_n(t_n)w_n$, for all $t_1, \ldots, t_n$. Then SI minimizes the expected value of $\Sigma w_i h(C_i)$, for any $m(\cdot)$ and nondecreasing, concave function $h(\cdot)$.*

Proof. By the comments above, it is sufficient to prove the theorem for an arbitrary t and $h(c) = \min(c,t)$. We call this the problem *with truncation after t intervals.* The proof is by an induction on t and employs three hypotheses, $H_{1,s}$, $H_{2,s}$ and $H_{3,s}$. We shall introduce these hypotheses shortly and see that each is clearly true when $s = 1$. Let H_t be the hypothesis that $H_{1,s}$, $H_{2,s}$ and $H_{3,s}$ are true, for all $m(\cdot)$, n, $\rho_1(\cdot),\ldots,\rho_n(\cdot)$, and $1 \le s \le t$. We shall show that H_{t-1} implies H_t. The first hypothesis is,

$$H_{1,t}\colon E[\Sigma w_i \min(C_i,t)] \text{ is minimized by SI.}$$

$H_{1,1}$ is certainly true. The first step of the induction is to show that $H_{1,t}$ follows from H_{t-1}. Consider the processing of jobs during interval 0. Let $m = m(0)$. Consider some candidate optimal policy, Π, which processes a job j, $j > m$, during interval 0 but does not process a job i, where $i \le m$. During interval 0, Π processes job j and some $m-1$ other jobs. Let this set of other jobs be denoted U, and let Π' be the policy which processes i and the jobs of U during interval 0 and which is optimal thereafter. Clearly, since Π it is a candidate optimal policy and we have assumed $H_{1,t-1}$ is true, Π must be identical to SI from time 1 onward. This is because the scheduling problem that remains for the jobs uncompleted at time 1 has the same optimal schedule as a second problem with truncation after $t-1$ intervals. We will show that Π' is at least as good as Π.

Consider the states reached at time 1 after following Π or Π' and processing either $U + \{j\}$ or $U + \{i\}$ during the interval 0. Each job in U has received one unit of processing, and job j or job i has received one unit of processing, under Π or Π' respectively. Let the age of job h at time 1 be X_h for $h \ne i,j$ and let $X_i = x_i$ and $X_j = x_j$. The difference in the expected costs achieved by Π and Π' is clearly just the same as the difference which would have been achieved if we had been in state X at time 0, with $m(0)$ altered to 1, and Π and Π' were the policies which in that starting circumstance processed jobs j and i respectively during interval 0 and behaved optimally thereafter. So suppose this scenario is indeed the case. We will show that starting at time 0 in state x, with $m(0)$ altered to 1, it is better to process job i for a single unit of time than to process job j. Let $G_h(x,s)$ denote the expected cost attained when, starting in state x at time s, with $m(s)$ altered to 1, we follow the policy Π_h, where this is defined as the policy which processes job h during interval s and proceeds optimally thereafter. Let $D_{ij}(x,s) = G_i(x,s) - G_j(x,s)$. Showing $D_{ij}(x,0) \le 0$ will confirm that Π' (which is Π_i) is as good as Π (which is Π_j). Thus the inductive step for $H_{1,t}$ will be completed when we have shown H_2, t, where

$$H_{2,t} : D_{ij}(x,0) \le 0, \text{ when truncation is after } t \text{ intervals.}$$

We now derive a useful expression for $D_{ij}(x,0)$. Consider the processing which takes place during interval 0. Let $k = m(1) + 1$. The state reached at time 1 after employing Π_i to schedule processing during intervals 0 and 1 will be the same as if no job were processed during interval 0, the jobs $1,\ldots,m(1)$ were processed during interval 1, and then either job i or job k is were given one extra unit of processing as job i is or is not still uncompleted respectively. Let X denote the state reached after jobs $1,\ldots,m(1)$ have received one unit of processing. We introduce a notational convention that when a quantity is superscripted with a job index it indicates that the quantity is to be evaluated assuming the job with that index is completed. For example, $G^i(x,s)$ is the expected total holding cost starting at time s in a state where the job ages are given by x, but job

i is already complete. Let $p_i = 1 - q_i = p_i(x_i)$, by the above observations we have, conditional on X,

$$\begin{aligned}
&G_i(x,0) = -p_i w_i + G_i(X,1) + \Sigma w_h 1(X_h = \infty) && \text{for } i > m(1),\\
&G_i(x,0) = -p_i w_i + p_i G^i_k(X,1) + q_i G_i(X,1) && \\
&\qquad\qquad + \Sigma w_h 1(X_h = \infty) && \text{for } i \le m(1),\\
(5)\quad &D_{ij}(x,0) = p_j w_j - p_i w_i + D_{ij}(X,1) && \text{for } i,j > m(1),\\
(6)\quad &D_{ij}(x,0) = p_j w_j - p_i w_i + p_i D^i_{kj}(X,1) + q_i D_{ij}(X,1) && \text{for } i \le m(1) < j,\\
(7)\quad &D_{ij}(x,0) = p_j w_j - p_i w_i + p_i q_j D^i_{kj}(X,1) && \\
&\qquad\qquad + q_i p_j D^j_{ik}(X,1) + q_i q_j D_{ij}(X,1) && \text{for } i,j \le m(1).
\end{aligned}$$

We now establish $D_{ij}(x,0) \le 0$. Notice that repeated applications of $D_{ij} = D_{ik} + D_{kj}$ imply that it is sufficient to show $D_{ij} \le 0$ for the case in $j = i+1$. So assume this is so. In the cases described in (5) and (6) the inductive step is easy, since $k \le j$ and $p_j w_j - p_i w_i \le 0$ and we have assumed H_{t-1}. In case (7) the argument is much more intricate. The key idea is to think of decreasing w_i to w_j while simultaneously increasing p_i in such a way that $p_i w_i$ does not increase. Define $\overline{w}_i = w_j$, and for each x_i define $\overline{p}_i(x_i) = \max\{p_i(x_i), \phi_j\}$ where $\phi_j = \sup_{s \ge 0}\{p_j(s)\}$. Consider the new problem which is obtained when we decrease w_i to $\overline{w}_i$ and increase $p_i(x_i)$ to $\overline{p}_i(x_i)$ in this fashion. We shall express this change of problem by saying "*the parameters for i are altered towards j*."Note that the definitions imply $w_i \ge \overline{w}_i = w_j$, and $p_i(x_i)w_i \ge \overline{p}_i(x_i)\overline{w}_i \ge p_j(x_j)w_j$ for all x_i and x_j. Since we are supposing that $j = i+1$, this insures that the new problem still has parameters which satisfy the conditions in the statement of theorem 2. We now introduce an inductive hypothesis which states that the terms D^j_{ik} and D_{ij} in on the right hand side of (7) do not decrease as the parameters for i are altered towards j.

H_3, t : Suppose $j = i+1 < k \le m(1)+1$ and truncation is after t intervals.
Then $D^j_{ik}(x,0)$ and $D_{ij}(x,0)$ do not decrease as the parameters
for i are altered towards j.

Assuming by H_{t-1} that $H_{3,t-1}$ is true, we first alter the parameters for i towards j in the terms $D^j_{ik}(X,1)$ and $D_{ij}(X,1)$ in (7). The right hand side does not decrease and the signs of these two quantities are still nonpositive, by hypothesis $H_{2,t-1}$. Then as we alter the parameters for i towards j with respect to the quantities p_i, q_i and w_i, the right hand side does not decrease. At the end of these alterations the model has been changed to one in which jobs i and j have identical holding costs. Also, since $p_i \ge \phi_j$, we have $\overline{p}_i(x_i) \ge p_j(x_j)$ for all x_i and x_j. Following the alteration of parameters for i we write the left hand side in (7) as $\overline{D}_{ij}(x,0)$. Since we have shown that $D_{ij}(x,0) \le \overline{D}_{ij}(x,0)$, the proof will be concluded by showing $\overline{D}_{ij}(x,0) \le 0$, and verifying the inductive step for $H_{3,t}$.

Firstly, we check the inductive step for $H_{3,t}$. This is straightforward. The check for $D_{ij}(x,0)$ has already been carried out in our analysis of the fact that the right hand side of (7) does not decrease as the parameters for i are altered towards j. To check the statement for $D^j_{ik}(x,0)$ we denote $h = m(1)+2$ and use

$$D^j_{ik}(x,0) = p_k w_k - p_i w_i + q_i p_k D^{hk}_{ih}(X,1) + p_i q_k D^{ik}_{hk}(X,1) + q_i q_k D^j_{ik}(X,1),$$

and employ $H_{3,t-1}$ on the right hand side to show this is no more than $\overline{D}^{i}_{kj}(x,0)$. Moreover, $H_{2,t}$ and $H_{3,t}$ hold trivially at the base of induction $t = 1$. Note that throughout the above we have assumed a fixed realization for X. The inductive steps are to be completed by taking an expectation over X.

Finally, we use corollary 1 of the previous section to show $\overline{D}_{ij}(x,0) \leq 0$. Recall that $i,j \leq m(1)$. So we have

$$\begin{aligned}\overline{D}_{ij}(x,0) &= E[K(\overline{Y}_i, Y_j + 1, \tau_3, \ldots, \tau_m) - K(\overline{Y}_i + 1, Y_j, \tau_3, \ldots, \tau_m)],\\ &= E[K(\overline{Y}_i, Y_j + 1, \tau_3, \ldots, \tau_m) - K(Y_j, \overline{Y}_i + 1, \tau_3, \ldots, \tau_m)],\end{aligned}$$

Where $\overline{Y}_i$ and Y_j are the remaining processing times of jobs i and j, and $\tau_3, \ldots, \tau_m$ are such that $m(s) = 2 + \Sigma 1(\tau_i \leq s)$, $s \geq 1$. Clearly $\overline{Y}_i \leq_{st} Y_j$ since $\overline{p}_i(x_i) \geq p_j(x_j)$ for all x_i and x_j. Moreover since $\overline{Y}_i$ and Y_j are both at least 1, the $K(\cdot)$'s above are evaluated for a problem in which no processor is available for processing the remaining $n-2$ jobs until time 1. Therefore $H_{1,t-1}$ applies, and corollary 1(b) can be envoked to deduce that the final term above is nonpositive. This completes the proof of the theorem. □

4. Discussion. Our result is both stronger and weaker than that which has been obtained by Kampke for the minimization of expected weighted flowtime with exponential jobs. We have generalized that result to more general processing time distributions. We have also shown that the policy SI minimizes the expected holding cost incurred by time t, for every t.

Kampke's method of proof shows only that the total expected holding cost is minimized. However, it is interesting to note that his result for a general concave Markovian cost function does imply that SI minimizes total expected discounted holding cost for discounting at rate α and exponential jobs. One need only think of adding an extra processor and job of index 0, which has a processing time exponentially distributed with parameter α, and alter the cost funciton to g_α, where $g_\alpha(U) = \lambda_1 g(U)/\alpha$ for $0 \in U$, and $g_\alpha(U) = 0$ for $0 \notin U$. This will ensure that job 0 has the highest priority and will always be processed from time 0 until it completes. The total expected cost in the new problem will be just λ_1/α times the total expected discounted cost in the original problem.

For the case of exponential jobs, Kampke's proved that $w_1 \geq \cdots \geq w_n$ and $\lambda_1 w_1 \geq \cdots \geq \lambda_n w_n$ are sufficient conditions for SI to minimize expected weighted flowtime. There is a sense in which these conditions are also necessary ones. For consider $m = 2$, $n = 3$. Suppose w_1 is very much larger than w_2 and w_3, so it is definitely optimal to process job 1 from the start. Let the jobs have processing times Y_1, Y_2, Y_3. Then it is optimal to start by processing jobs 1 and 2 rather than jobs 1 and 3 provided $w_3 E(\min\{Y_1, Y_2\}) \leq w_2 E(\min\{Y_1, Y_3\})$. This is equivalent to $\lambda_1(w_2 - w_3) + (\lambda_2 w_2 - \lambda_3 w_3) \geq 0$, and if this is to hold for all λ_1 we will need $w_2 > w_3$ and $\lambda_2 w_2 \geq \lambda_3 w_3$. Similarly one can check that the conditions of theorem 2 are necessary if SI is to minimize the expected total holding cost incurred by time t for all τ and t.

Since $m(s)$ was an arbitrary nondecreasing function, theorem 2 holds even if $m(s)$ is a stochastic process nondecreasing in s. Notice that it was in order to check the inductive step for $H_{3,t}$ that we needed $m(s)$ to be nondecreasing. $H_{3,t}$ was a hypothesis concerning jobs $i, j \leq m(1)$ which receive processing during interval 1. To check an inductive step for $H_{3,t}$ these jobs must continue to be processed from time 1 until they are complete. The question remains open as to whether theorem 2 can be shown to hold for processors of differing speeds, $s_1 \geq \cdots \geq s_n$, as has been proved by Kampke for the case of exponential jobs. The generalization would follow from the work in the previous section if we had been able to prove theorem 2 for arbitrary $m(s)$. One would simply imagine that $m(s)$ is a stochastic process, taking values that are independent from interval to interval, and equal to i with probability $(s_i - s_{i+1})/s_1$. This would approximate a model with processors of different speeds as the discretization of time is made arbitrarily fine. However, it is not possible to relax the condition that $m(s)$ be nondecreasing and the generalization of theorem 2 to the case of two uniform processors will require some other analysis.

We expect that theorem 2 can be generalized not only to uniform processors of different speeds, but to any concave Markovian cost function. The conditions of the theorem would be replaced by $g(U-\{j\}) \geq g(U-\{i\})$ and $p_i(t_1)(g(U)-g(U-\{i\})) \geq p_j(t_2)(g(U)-g(U-\{j\}))$ for all $i < j$, all sets of uncompleted jobs U which contain both i and j, and all t_1, t_2. However, it appears difficult to prove this generalization by methods in this paper or those adopted by Kampke.

It should be interesting to investigate whether the results of section 2 have any parallels when one consider the minimization of the expected value of a convex function of the completion times, such as the first time at which all of m processors become free. A cost function which is convex and Markovian is also submodular in the completion times. However, the question is unresolved as to whether $K(\tau)$ might inherit convexity and submodularity in τ when these are properties of $\kappa(C)$.

Acknowledgement. I am grateful to Thomas Kampke for conjecture and discussion of the result proved in section 3 and to Kevin Glazebrook and Gideon Weiss for some comments which have improved the paper.

REFERENCES

[1] J. Bruno and P. Downey and G. N. Frederickson, *Sequencing tasks with exponential service times to minimize the expected flowtime or makespan*, J. Ass. Comp. Mach., 28 (1981), pp. 100–113.

[2] K. D. Glazebrook, *Scheduling tasks with exponential service times on parallel processors*, J. Appl. Prob., 16 (1979), pp. 658–689.

[3] T. Kampke, *Optimalitatsaussagen fur Spezeille Stochastische Schedulingprobleme*, Diploma-Mathematiker, Rheinisch-Westfalischen Technischen Hochschule Aachen, 1985.

[4] T. Kampke, *On the optimality of static priority policies in stochastic scheduling on parallel machines*, J. Appl. Prob., 24 (1987) (to appear).

[5] L. Lovasz, *Submodular functions and convexity, Mathematical Programming, the State of the Art*, eds A. Bachem et al, Springer, Berlin, 1983, pp. 235–257.

[6] R. R. Weber and P. Nash, *An optimal strategy in multi-server stochastic scheduling*, J. R. Statist. Soc., B 40 (1979), pp. 323–328.

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

[8] R. R. Weber and P. Varaiya and J. Walrand, *Scheduling jobs with stochastically ordered processing times on parallel machines to minimize expected flowtime*, J. Appl. Prob., 23 (1986), pp. 841–847.

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