Scheduling a Sequence of Parallel Programs Containing Loops within a Centralized Parallel Processing System*

Zhen LIU
INRIA Centre Sophia Antipolis
2004 Route des Lucioles
06560 Valbonne
France

Don TOWSLEY
Dept. of Computer Science
University of Massachusetts
Amherst, MA 01003
U.S.A.

July 12, 1993

Abstract

We investigate the problem of scheduling a sequence of jobs running in a centralized parallel processing system with identical processors. The jobs represent parallel programs that contain probabilistic loops of tasks that can be simultaneously executed. We show that the Smallest Phase first policy is optimal within the class of nonpreemptive policies when the task processing times are identical and independently distributed random variables with an increasing likelihood ratio distribution. The optimality extends to the class of preemptive policies when the task processing times have an exponential distribution. The optimality is understood to be the stochastic minimization of the process of the numbers of jobs in the system and the minimization of mean response times of the jobs. Stronger optimality results on the minimization of job response time are obtained for a simpler job model.

Keywords: stochastic scheduling, probabilistic loops, parallel processing, stochastic ordering.

*This work was supported in part by the National Science Foundation under grant ASC 88-8802764.

1 Introduction

We consider the problem of scheduling a sequence of jobs in a centralized parallel processing system with identical processors. The jobs represent parallel programs that contain probabilistic loops of tasks that can be simultaneously executed. When a job arrives in the system, it generates several tasks which form the first phase of the job. These tasks can be executed by any of the processors. When all of the tasks in a phase have completed execution, a new phase of the job is generated with some probability. This new phase contains again several tasks. A job leaves the system when all of its tasks finish execution. The processing times of the tasks are assumed to be random variables (r.v.'s) and to be identical and independently distributed (i.i.d.). The numbers of tasks in the phases also form sequences of i.i.d. r.v.'s.

In the literature (see e.g. [?, ?, ?, ?, ?, ?, ?, ?, ?]), the problem of stochastic scheduling of a sequence of jobs on two or more processors was traditionally studied for sequential jobs, i.e., jobs with single tasks. Stochastic scheduling of a single job consisting of tree structured tasks has been investigated in [?, ?, ?, ?, ?]. When there is a sequence of jobs which contain tasks constrained by precedence relations, extremal scheduling policies were established in [?], under the assumption that the tasks have dedicated processors. There also exist a number of papers that have analytically studied the performance of different policies for scheduling parallel programs (e.g. first come first serve [?, ?], processor sharing [?], priority scheduling [?]).

In this paper, we analyze the dynamic scheduling of a sequence of parallel programs containing loops of parallel tasks. The problem is to dynamically assign the tasks to the processors in such a way that the number of jobs in the system and the response time of the jobs are minimized. We define a Smallest Phase first (SP) policy to be one that assigns the highest priority to the tasks belonging to the phases containing the smallest number of tasks remaining in the system. We prove that when the number of tasks in each phase is constant and the task processing times have an increasing likelihood ratio (ILR) distribution, the SP policy is optimal within the class of policies that does not allow task preemptions. The optimality extends to the class of preemptive policies when the phase sizes form an i.i.d. sequence of r.v.'s having arbitrary distribution and the task processing times have an exponential distribution. This optimality is understood to be the stochastic minimization of the process of the numbers of jobs in the system and the minimization of mean response times of the jobs. Stronger optimality results on the minimization of job response time are obtained for a simpler job model—single phase jobs.

These properties support comparison results of [?], where a performance analysis was carried out for single phase jobs. Approximate techniques were used for the evaluation and comparison of four scheduling algorithms under the assumption of exponential task processing times. It was shown that the central splitting policy, which corresponds to our SP policy, provides the minimal mean job response time within the four scheduling policies.

The paper is organized as follows. In Section ??, we define the scheduling problem in more detail, and we present the basic notion on stochastic orderings. In Sections ?? and ??, we analyze the optimal policies for the minimization of the number of jobs and the response time of jobs, respectively.

2 Preliminaries

2.1 Basic Model

There are $K \geq 1$ identical processors and a central waiting queue. Jobs arrive at arbitrary times $0 \leq a_1 \leq a_2 \leq \cdots \leq a_n \leq \cdots$. At arrival, job J_n (which arrives at a_n) generates a phase (the first phase of J_n) of $v_n \geq 1$ tasks. All of these tasks are placed in the queue waiting to be executed by one of the K processors. When all of the tasks in a phase have completed execution, a new phase of the job is generated with probability $0 \leq p < 1$.

Let $\{u_n\}_{n=1}^{\infty}$ be a sequence of i.i.d. nonnegative integer r.v.'s such that for all $n \geq 1$, $P[u_n > 0] = p$. If $u_n > 0$, then the n-th completed phase initiates a new phase consisting of u_n tasks belonging to the same job as the n-th completed phase. Otherwise, it generates a phase of u_n tasks belonging to the same job. These u_n tasks are thus put into the waiting queue for execution. Otherwise it has no successor phase and the job that it belongs to is finished. Note that the number of tasks contained in the first phase of execution of a job need not have the same distribution as the number of tasks in subsequent phases.

The phases are denoted by ϕ_n, $n = 1, 2, \cdots$, where phase ϕ_n is the n-th generated phase (either by a job arrival or by a phase completion). The i-th task of phase ϕ_n is denoted by $T_{i,n}$.

A job is considered completed and leaves the system when, upon completion of a phase, it does not generate a successor phase. The processing times of all the tasks are i.i.d. r.v.'s with a common distribution.

The problem is to dynamically assign the tasks to the processors in such a way that the number of jobs in the system and the response time of the jobs are minimized. The scheduler has no information on the exact value of the processing times. We assume throughout this paper that the policies under consideration are *work conserving* or *nonidling*, i.e., a processor should never be idle when there is a task waiting for execution.

It is easy to see that optimal policies should be work conserving either when the policies are allowed to be preemptive, when the task processing times have an exponential distribution, or when the number of tasks in each phase is a constant.

Denote by Ψ the class of such scheduling policies. A policy is called *task nonpreemptive* if the executions of the tasks are never preempted. (Note that a task nonpreemptive policy need not contiguously schedule all tasks of the same job). A policy is a Smallest Phase (SP) policy if, when it schedules a task, it selects a task from a job whose current phase has the smallest number of tasks in the system. Denote by $\Psi^{\circ} \subset \Psi$ the class of task nonpreemptive policies, by $\Psi_{SP} \subset \Psi$ the class of SP policies and by $\Psi^{\circ}_{SP} \subset \Psi^{\circ}$ the class of task nonpreemptive SP policies. Note that the SP policies differ only in the way ties are broken.

Let π be an arbitrary policy. The following notation will be used:

- $N_t(\pi)$: the number of jobs in the system at time $t \in I\!R^+$ under π;
- $N(\pi)$: the number of jobs in the stationary regime of the system under π, provided it exists;
- $Q_t^n(\pi)$: the number of tasks of phase ϕ_n, $n = 1, 2, \cdots$, in the system at time t under π;
- $h_t(\pi)$: the number of phases generated (due to either a job arrival or a phase completion) under π by time t;
- $c_{i,n}(\pi)$: the completion time of task $T_{i,n}$ under π;
- $C_n(\pi)$: the completion time of job J_n under π; when $p = 0$ (so that there is a single phase in each job), $C_n(\pi) = \max_{1 \le i \le v_n} c_{i,n}(\pi)$;

- $R_n(\pi) = C_n(\pi) - a_n$: the response time of job J_n under π;
- $R(\pi)$: the stationary response time of jobs under π, provided it exists.

Let $H_t(\pi)$ be the number of zeros in the set $\{u_1, u_2, \cdots, u_{h_t(\pi)}\}$. Then, the number of jobs in the system by time t under policy π can be expressed as

$$N_t(\pi) = n(t) - H_t(\pi), \tag{1}$$

where $n(t)$ denotes the number of jobs that arrive by time t. Note that $H_t(\pi)$ is monotonically increasing in $h_t(\pi)$.

2.2 Stochastic Ordering and Majorization

In the proof of our results, we will use the notion of majorization. Let $x, y \in I\!R^n$ be two real vectors. Vector x is said to be majorized by vector y (written $x \prec y$) iff

$$\begin{aligned} \sum_{i=1}^{k} x_{[i]} &\leq \sum_{i=1}^{k} y_{[i]}, \qquad k = 1, \cdots, n-1; \\ \sum_{i=1}^{n} x_{[i]} &= \sum_{i=1}^{n} y_{[i]}, \end{aligned}$$

where the notation $x_{[i]}$ is taken to be the i-th largest element of x. Last, a function $f : I\!R^n \to I\!R$ is *Schur convex* if $f(x) \leq f(y)$ for every pair of vectors such that $x \prec y$. Note that all convex symmetric functions are Schur convex. For example, the functions $\max_{1 \leq i \leq n} x_i$ and $\sum_{i=1}^{n} g(x_i)$, where $g : I\!R \to I\!R$ is convex, are Schur convex.

Let $X, Y \in I\!R^n$ be two random vectors. Vector X is stochastically smaller than vector Y in the sense of strong stochastic ordering ($X \leq_{st} Y$), convex ordering ($X \leq_{cx} Y$), increasing convex ordering ($X \leq_{icx} Y$), increasing Schur convex ordering ($X \leq_{E_1^\uparrow} Y$), and marginal convex ordering ($X \leq_{E_3^\uparrow} Y$) respectively, if

$$\begin{aligned} E[f(X)] \leq E[f(Y)], &\qquad \forall \text{ increasing } f : I\!R^n \to I\!R, \\ E[f(X)] \leq E[f(Y)], &\qquad \forall \text{ convex } f : I\!R^n \to I\!R, \end{aligned}$$

$$E[f(X)] \leq E[f(Y)], \qquad \forall \text{ increasing and convex } f : I\!R^n \to I\!R,$$
$$E[f(X)] \leq E[f(Y)], \qquad \forall \text{ increasing and Schur convex } f : I\!R^n \to I\!R,$$
$$E\left[\sum_{i=1}^{n} f(X_i)\right] \leq E\left[\sum_{i=1}^{n} f(Y_i)\right], \qquad \forall \text{ increasing and convex } f : I\!R \to I\!R,$$

respectively, provided the expectations exist. The notation "$\leq_{E_1^\uparrow}$" and "$\leq_{E_3^\uparrow}$" is taken from [?].

A real-valued process $\{X_t\}_t$ is said to be stochastically smaller than process $\{Y_t\}_t$, denoted $\{X_t\}_t \leq_{st} \{Y_t\}_t$, if for all n and all $t_1 < t_2 < \cdots < t_n$,

$$(X_{t_1}, X_{t_2}, \cdots, X_{t_n}) \leq_{st} (Y_{t_1}, Y_{t_2}, \cdots, Y_{t_n}).$$

The reader is referred to [?] for more properties concerning the $\leq_{st}$, $\leq_{cx}$, and $\leq_{icx}$ orderings and to [?] for more properties concerning the $\leq_{E_1^\uparrow}$ and $\leq_{E_3^\uparrow}$ orderings. In what follows, $=_{st}$ denotes equality in distribution. The following lemma is due to Strassen [?]:

Lemma 2.1 (Strassen) *Two random vectors X and Y satisfy $X \leq_{st} Y$ if and only if there exist two random vectors $\widehat{X}$ and $\widehat{Y}$ defined on a common probability space such that $X =_{st} \widehat{X}$, $Y =_{st} \widehat{Y}$, and $\widehat{X} \leq \widehat{Y}$ componentwise almost surely (a.s.).*

Some of our results will require that processing times have increasing likelihood ratio (ILR) distributions. In order to define such distributions, we first define the likelihood ratio ordering. Let $X, Y \in I\!R^+$ be two continuous nonnegative random variables with density functions f_X and f_Y respectively. The random variable X is smaller than the random variable Y in the sense of likelihood ratio ($X \leq_{lr} Y$) if

$$f_Y(x)/f_X(x) \leq f_Y(y)/f_X(y), \qquad 0 \leq x \leq y.$$

One of the properties of the likelihood ratio ordering is that it implies the strong stochastic ordering, i.e.,

$$X \leq_{lr} Y \;\Rightarrow\; X \leq_{st} Y.$$

The random variable $X \in I\!R^+$ is said to be increasing in likelihood ratio (ILR) (or has an ILR distribution) if

$$X_s \geq_{lr} X_t, \qquad 0 \leq s \leq t,$$

where X_t is the remaining life from t, having lifetime X which has reached the age of t.

A random variable is ILR iff its density function is log-concave (or, Polya frequency of order 2). A random variable is ILR if it has a gamma distribution with shape parameter greater or equal to 1.

The likelihood ratio ordering can also be defined for discrete random variables that are defined over the same set of values. We say that $X \leq_{lr} Y$ if $P(Y = x)/P(X = x)$ increases in x.

A r.v. $X \in I\!R^+$ has increasing failure rate (IFR) (or has an IFR distribution) iff

$$X_s \geq_{st} X_t, \qquad 0 \leq s \leq t,$$

where X_t is the remaining life of X which has reached the age of t. An ILR random variable has an IFR distribution.

3 Minimization of Number of Jobs

In this section, we consider the optimal policies for the minimization of the number of jobs in the system. Recall that a Smallest Phase first (SP) policy is one that always assigns a task of an unfinished phase having the mimimum number of unfinished tasks to a processor.

Theorem 3.1 *Assume that the task processing times have a common exponential distribution. Then any preemptive Smallest Phase first policy stochastically minimizes the process of the number of jobs in the system:*

$$\forall \pi \in \Psi \ : \qquad \{N_t(SP)\}_t \leq_{st} \{N_t(\pi)\}_t.$$

Proof. It is easily seen that the processes of the numbers of jobs in system under SP policies are all stochastically identical.

Due to the memoryless property of the exponential distribution of the task processing times, the remaining processing times of the tasks at any stopping time are still exponentially distributed with the same parameter. Therefore, we can consider a system where the processors are continuously executing tasks. Whenever an execution completion occurs and there is no task assigned to that processor, it corresponds to the completion of a fictitious task. When a task is assigned to a processor, it is assigned a execution time equal to the remainder of the processing time already underway at that processor. We fix these task processing times as well as the job arrival times.

Let $\pi \in \Psi$ be an arbitrary non-SP policy. Let $0 = e_1 < e_2 < \cdots e_n < \cdots$ be the decision times of π. Assume that e_m is the first time when π makes a non-SP decision. At e_m, π assigns task T_{k_1,n_1} to a processor instead of task T_{k_2,n_2} if the SP rule were followed. In other words, task T_{k_2,n_2} is enabled and phase ϕ_{n_2} has fewer unfinished tasks than phase ϕ_{n_1} : $Q_{e_m}^{n_2}(\pi) < Q_{e_m}^{n_1}(\pi)$.

Construct a new policy π' as follows. The assignment decisions of π' are identical to π until time e_m. At time e_m, π' assigns task T_{k_2,n_2} to a processor instead of task T_{k_1,n_1}. If at time e_{m+1}, task T_{k_1,n_1} does not finish execution under π, then from time e_{m+1}, the assignment decisions of π' are still identical to π. If at time e_{m+1}, task T_{k_1,n_1} finishes execution under π, then the execution of task T_{k_2,n_2} is finished under π', and from time e_{m+1}, the assignment decisions of π' are still identical to π but for tasks T_{k_1,n_1} and T_{k_2,n_2}, where the decisions are switched, i.e., at any time, task T_{k_1,n_1} (resp. task T_{k_2,n_2}) is assigned under π iff task T_{k_2,n_2} (resp. task T_{k_1,n_1}) is assigned under π'.

It then follows that this construction either switches the completion times of tasks T_{k_1,n_1} and T_{k_2,n_2} or not. Moreover, task T_{k_2,n_2} finishes after T_{k_1,n_1} under π' only if T_{k_2,n_2} finishes after T_{k_1,n_1} under π. If $c_{k_2,n_2}(\pi') < c_{k_1,n_1}(\pi')$, then a phase may complete earlier under π' than under π. Therefore, it is readily checked by induction on the event epochs $e_1, e_2, \cdots, e_n, \cdots$ that for all time t,

$$\left(Q_t^1(\pi), Q_t^2(\pi), \cdots, Q_t^{h_t(\pi')}(\pi)\right) \prec \left(Q_t^1(\pi'), Q_t^2(\pi'), \cdots, Q_t^{h_t(\pi')}(\pi')\right).$$

and that (owing to [?, p. 117])

$$h_t(\pi) \leq h_t(\pi').$$

More generally, for all $t \geq 0$, all $j = 1, 2, \cdots$, and all $0 \leq t_1 < t_2 < \cdots < t_j \leq t$, we have that:

$$\forall i \in \{1, \cdots, j\} : \qquad h_{t_i}(\pi) \leq h_{t_i}(\pi'),$$

so that (cf. (??))

$$\left(N_{t_1}(\pi), \cdots, N_{t_j}(\pi)\right) \geq \left(N_{t_1}(\pi'), \cdots, N_{t_j}(\pi')\right).$$

Note that π' makes one less non-SP decision than π. This interchange process is repeated until we get a policy ρ such that, at each decision epoch, a task of the smallest phase is scheduled. We obtain that, for all $t \geq 0$, all $j = 1, 2, \cdots$, and all $0 \leq t_1 < t_2 < \cdots < t_j \leq t$,

$$\left(N_{t_1}(\pi), \cdots, N_{t_j}(\pi)\right) \geq \left(N_{t_1}(\rho), \cdots, N_{t_j}(\rho)\right).$$

Note that the new policy ρ may be an idling policy while π is a nonidling one due to the fact that more phases are generated under ρ than under π. Let SP be the policy obtained from ρ by removing the idling in ρ and assigning tasks as early as possible. Then, for all $t \geq 0$, all $j = 1, 2, \cdots$, and all $0 \leq t_1 < t_2 < \cdots < t_j \leq t$,

$$\left(N_{t_1}(\pi), \cdots, N_{t_j}(\pi)\right) \geq \left(N_{t_1}(\rho), \cdots, N_{t_j}(\rho)\right) \geq \left(N_{t_1}(SP), \cdots, N_{t_j}(SP)\right).$$

Unconditioning the arrival times and the task processing times entails that

$$\{N_t(SP)\}_t \leq_{st} \{N_t(\pi)\}_t.$$

■

Theorem 3.2 *Assume that the task processing times have a common ILR distribution. Assume further that the number of tasks in each phase is a constant V. Then, any task nonpreemptive Smallest Phase first policy (SP^o) stochastically minimizes the process of the number of jobs in the system within the class of policies Ψ^o:*

$$\forall \pi \in \Psi^o : \qquad \{N_t(SP^o)\}_t \leq_{st} \{N_t(\pi)\}_t.$$

Proof. The scheme of the proof is similar to that of Theorem ??. Fix the arrival times. For the given policy π, fix the task processing times and determine the task assignment times and task completion times. Denote by $s_{i,n}(\pi)$ and $\sigma_{i,n}(\pi)$ the execution starting time and the duration of the execution, respectively, of task $T_{i,n}$ under policy π.

Let σ be the generic r.v. of the task processing times. In the proof we will assume that σ is a continuous r.v. and has a density function f_σ. In case σ is a discrete r.v., we can replace f_σ by its distribution and the proof can be carried out in an analogous manner.

Assume that at time $b_1 = s_{k_1,n_1}(\pi)$, policy π assigns task T_{k_1,n_1}, whereas there is an unassigned task T_{k_2,n_2} such that $Q_{b_1}^{n_2}(\pi) < Q_{b_1}^{n_1}(\pi)$. Suppose task T_{k_2,n_2} is assigned at time $b_2 = s_{k_2,n_2}(\pi) > b_1$ under π. Let $e_1 = c_{k_1,n_1}(\pi) = b_1 + \sigma_{k_1,n_1}(\pi)$ and $e_2 = c_{k_2,n_2}(\pi) = b_2 + \sigma_{k_2,n_2}(\pi)$. Let also $\Delta = b_2 - b_1$.

Define a new task nonpreemptive policy π' which differs from π only in that the scheduling of tasks T_{k_1,n_1} and T_{k_2,n_2} are switched. The processing times of all the other tasks are kept the same under π'. The processing times of T_{k_1,n_1} and T_{k_2,n_2} are interchanged in the following way:

$$\begin{aligned}
(\sigma_{k_2,n_2}(\pi'), \sigma_{k_1,n_1}(\pi')) \;&=\; \mathbf{1}(e_1 \le b_2)(\sigma_{k_1,n_1}(\pi), \sigma_{k_2,n_2}(\pi)) \\
&\quad +\mathbf{1}(b_2 < e_1 \le e_2)\,[U(\sigma_{k_1,n_1}(\pi), \sigma_{k_2,n_2}(\pi), \Delta)(\sigma_{k_1,n_1}(\pi), \sigma_{k_2,n_2}(\pi)) \\
&\qquad + (1 - U(\sigma_{k_1,n_1}(\pi), \sigma_{k_2,n_2}(\pi), \Delta))(\Delta + \sigma_{k_2,n_2}(\pi), \sigma_{k_1,n_1}(\pi) - \Delta)] \\
&\quad +\mathbf{1}(e_2 < e_1)(\Delta + \sigma_{k_2,n_2}(\pi), \sigma_{k_1,n_1}(\pi) - \Delta) \qquad (2)
\end{aligned}$$

where $U(a, b, \Delta)$ is a Bernoulli r.v. with probability distribution $\Pr[U(a, b, \Delta) = 0] = p(a, b, \Delta)$, $\Pr[U(a, b, \Delta) = 1] = 1 - p(a, b, \Delta)$ where

$$\begin{aligned}
p(a, b, \Delta) \;&=\; \frac{f_{\sigma|\sigma>\Delta}(b + \Delta) f_\sigma(a - \Delta)}{f_{\sigma|\sigma>\Delta}(a) f_\sigma(b)}, \\
&=\; \frac{f_\sigma(b + \Delta) f_\sigma(a - \Delta)}{f_\sigma(a) f_\sigma(b)}.
\end{aligned}$$

The construction of the processing times of tasks T_{k_1,n_1} and T_{k_2,n_2} under policy π' is illustrated in Figure **??**.

It is easy to see that this construction either switches the completion times of tasks T_{k_1,n_1} and T_{k_2,n_2} or not. Therefore, the (increasingly re-ordered) sequences of the assignment times and of the completion times of π' are identical to those of π. Moreover, task T_{k_2,n_2} finishes after T_{k_1,n_1} under π' only if T_{k_2,n_2} finishes after T_{k_1,n_1} under π. Hence, if the order of the completions

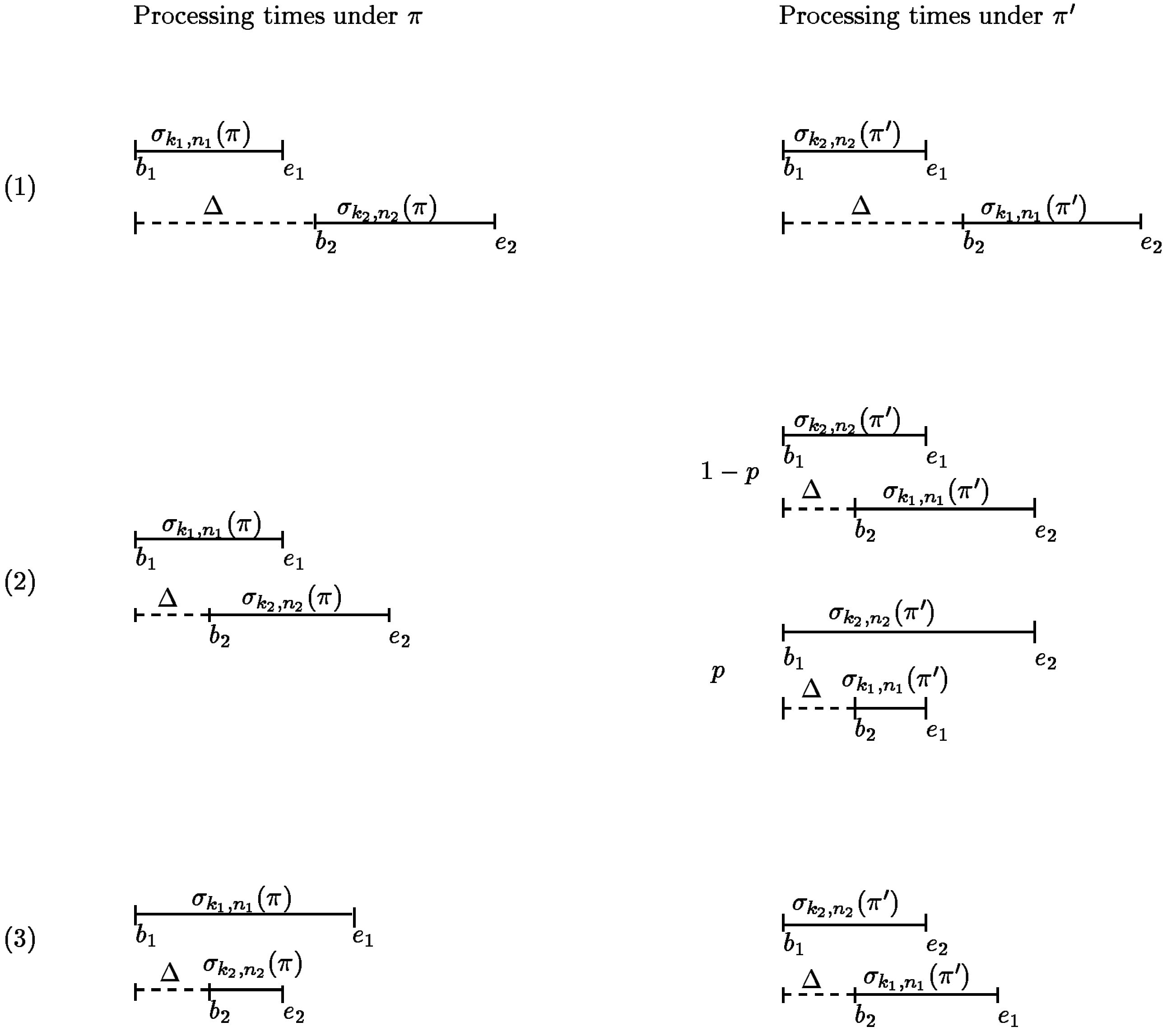

Figure 1: Construction of Processing Times of Tasks T_{k_1,n_1} and T_{k_2,n_2} under Policy π'

of tasks T_{k_1,n_1} and T_{k_2,n_2} under π' is the same as under π, then for all t: $h_t(\pi) = h_t(\pi')$ and

$$\left(Q_t^1(\pi), Q_t^2(\pi), \cdots, Q_t^{h_t(\pi')}(\pi)\right) = \left(Q_t^1(\pi'), Q_t^2(\pi'), \cdots, Q_t^{h_t(\pi')}(\pi')\right).$$

Otherwise, it is checked by induction on the scheduling decision epochs that for all t:

$$h_t(\pi) \leq h_t(\pi'), \quad \text{and} \quad \left(Q_t^1(\pi), Q_t^2(\pi), \cdots, Q_t^{h_t(\pi')}(\pi)\right) \prec \left(Q_t^1(\pi'), Q_t^2(\pi'), \cdots, Q_t^{h_t(\pi')}(\pi')\right).$$

Therefore, for all $t \geq 0$, all $j = 1, 2, \cdots$, and all $0 \leq t_1 < t_2 < \cdots < t_j \leq t$, we have that:

$$\forall i \in \{1, \cdots, j\}: \qquad h_{t_i}(\pi) \leq h_{t_i}(\pi'),$$

so that

$$\left(N_{t_1}(\pi), \cdots, N_{t_j}(\pi)\right) \geq \left(N_{t_1}(\pi'), \cdots, N_{t_j}(\pi')\right). \tag{3}$$

We now need to show that the random variables of the task processing times under π' are i.i.d. This is done by evaluating the joint density function for the processing times $(\sigma_{k_1,n_1}(\pi'), \sigma_{k_2,n_2}(\pi'))$. According to (??), if $x \leq \Delta$, then

$$f_{\sigma_{k_1,n_1}(\pi'),\sigma_{k_2,n_2}(\pi')}(x, y) = f_\sigma(x) f_\sigma(y).$$

In the case that $\Delta < x \leq y + \Delta$ we have

$$\begin{aligned} f_{\sigma_{k_1,n_1}(\pi'),\sigma_{k_2,n_2}(\pi')}(x, y) &= f_\sigma(x) f_\sigma(y)(1 - p(y, x, \Delta)) + f_\sigma(\Delta + y) f_\sigma(x - \Delta), \\ &= f_\sigma(x) f_\sigma(y) \left[1 - \frac{f_\sigma(\Delta + y) f_\sigma(x - \Delta)}{f_\sigma(x) f_\sigma(y)}\right] + f_\sigma(\Delta + y) f_\sigma(x - \Delta), \\ &= f_\sigma(x) f_\sigma(y). \end{aligned}$$

Finally, in the case of $y + \Delta < x$,

$$\begin{aligned} f_{\sigma_{k_1,n_1}(\pi'),\sigma_{k_2,n_2}(\pi')}(x, y) &= f_\sigma(\Delta + y) f_\sigma(x - \Delta) \frac{f_\sigma(\Delta + y) f_\sigma(x - \Delta)}{f_\sigma(x) f_\sigma(y)}, \\ &= f_\sigma(x) f_\sigma(y). \end{aligned}$$

The necessity that the task processing times have a distribution with ILR should be obvious from this construction.

Therefore we conclude that for all $(x, y) \in I\!R^{+2}$,

$$f_{\sigma_{k_1,n_1}(\pi'),\sigma_{k_2,n_2}(\pi')}(x, y) = f_\sigma(x) f_\sigma(y) = f_{\sigma_{k_1,n_1}(\pi),\sigma_{k_2,n_2}(\pi)}(x, y).$$

It then follows that the task processing times under π' are i.i.d. r.v.'s with ILR distribution.

Hence, according to inequality (??), we obtain that

$$\{N_t(\pi')\}_t \leq_{st} \{N_t(\pi)\}_t.$$

This interchange argument can be repeated until a (possibly idling) task nonpreemptive policy ρ is obtained such that at each decision epoch, a task of the smallest phase is scheduled. Thus, we obtain that

$$\{N_t(\rho)\}_t \leq_{st} \{N_t(\pi)\}_t.$$

Note that policy ρ may be idling while π is nonidling due to the fact that more phases are generated under ρ than under π. Let SP° be the policy obtained from ρ by removing the idling in ρ and assigning tasks as early as possible. Since all the phases have V tasks, there is no gain in idling processors. Therefore,

$$\{N_t(SP^\circ)\}_t \leq_{st} \{N_t(\rho)\}_t \leq_{st} \{N_t(\pi)\}_t.$$

■

When a policy π admits a stationary regime for the number of jobs in system, then the above theorems imply:

Corollary 3.3 *If the task processing times have a common exponential distribution, then*

$$\forall \pi \in \Psi \ : \qquad N(SP) \leq_{st} N(\pi).$$

If the task processing times have a common ILR distribution, and if all of the phases have a constant number of tasks, then

$$\forall \pi \in \Psi^\circ : \qquad N(SP^\circ) \leq_{st} N(\pi).$$

4 Minimization of Response Time

Consider now the minimization of the response times of the jobs. First, by Little's formula (see e.g. [?]), we have that for all nonidling policies $\pi \in \Psi$ such that the mean stationary number of jobs and the mean stationary job response time exist,

$$E[N(\pi)] = \lambda E[R(\pi)],$$

where λ is the arrival rate of the jobs. Applying Corollary **??** immediately implies the following results.

Corollary 4.1 *Assume that the task processing times have a common exponential distribution. Then any SP policy minimizes the average job response time :*

$$\forall \pi \in \Psi \quad : \qquad E[R(SP)] \leq E[R(\pi)].$$

Corollary 4.2 *Assume that the task processing times have a common ILR distribution. Assume further that the number of tasks in each phase is a constant V. Then any task nonpreemptive Smallest Phase first policy minimizes the average job response time within the class of task nonpreemptive policies:*

$$\forall \pi \in \Psi^{\circ} \quad : \qquad E[R(SP^{\circ})] \leq E[R(\pi)].$$

We now focus on the class of task nonpreemptive SP policies Ψ°_{SP}. We will restrict ourselves to the case of single phase jobs with V tasks, where V is a constant. Define the First Come First Serve (FCFS) policy as the one that assigns higher priority to tasks of job J_m than job J_n if $m < n$. Clearly, FCFS policy is a task nonpreemptive SP policy.

Lemma 4.3 *Assume that all jobs have a single phase with a deterministic number V of tasks. Assume further that the task processing times have a common IFR distribution. Then the FCFS policy minimizes the vector containing the response times of the first $n \geq 1$ jobs in the sense of marginal convex ordering, within the class of task nonpreemptive SP policies:*

$$\forall \pi \in \Psi^{\circ}_{SP} \quad : \qquad (R_1(FCFS), \cdots, R_n(FCFS)) \leq_{E_3^{\uparrow}} (R_1(\pi), \cdots, R_n(\pi)).$$

Proof. Observe that under a task nonpreemptive SP policy $\pi \in \Psi_{SP}^{o}$, the tasks of the same job are scheduled contiguously, i.e., in between the first and the last assignments of the tasks of a job, no task of another job is assigned. Denote by $s_{k,m}(\pi)$ the time when task $T_{k,m}$ is assigned for execution under π. Then $\pi \in \Psi_{SP}^{o}$ implies that for all $m \geq 1$, there is no $l \neq m$ such that

$$\min_{1\leq i\leq V} s_{i,m}(\pi) < s_{j,l}(\pi) < \max_{1\leq i\leq V} s_{i,m}(\pi).$$

Consider the first n jobs and fix the arrival times. Let $f : I\!R \rightarrow I\!R$ be an arbitrary increasing and convex function. Define $F(\pi)$ as follows:

$$F(\pi) = \sum_{m=1}^{n} E_A \left[f\left(R_m(\pi) \right) \right] = \sum_{m=1}^{n} E_A \left[f\left(c_m(\pi) - a_m(\pi) \right) \right],$$

where E_A denotes the conditional expectation given the arrival times. We will show by induction that there exist policies $\pi_1, \pi_2, \cdots, \pi_n \in \Psi_{SP}^{o}$ such that in policy π_m, $1 \leq m \leq n$, the first m jobs are scheduled according to the FCFS rule, and that

$$F(\pi) \geq F(\pi_1) \geq F(\pi_2) \geq \cdots \geq F(\pi_n). \tag{4}$$

Note that π_n is identical to FCFS policy in scheduling the first n-jobs. Therefore, unconditioning with respect to the arrival times in the above relation readily implies the assertion of the theorem.

Since π is nonidling and is a task nonpreemptive SP policy, job J_1 is scheduled by π in accordance with FCFS rule. Thus, we can define π_1 to be identical to π. Assume that for some $2 \leq m \leq n$, there exist policies $\pi_1, \pi_2, \cdots, \pi_{m-1} \in \Psi_{SP}^{o}$ such that in policy π_l, $1 \leq l \leq m-1$, the first l jobs are scheduled according to the FCFS rule, and that

$$F(\pi) \geq F(\pi_1) \geq F(\pi_2) \geq \cdots \geq F(\pi_{m-1}).$$

We will construct a policy π_m based on π_{m-1} in such a way that in π_m, the first m jobs are scheduled according to the FCFS rule, and that $F(\pi_{m-1}) \geq F(\pi_m)$.

Denote by $j \geq m$ the random variable representing the index of the job that is the m-th scheduled under π_{m-1}. Define π_m as the one that differs from π_{m-1} only in that the scheduling of tasks $T_{i,j}$ and $T_{i,m}$ are switched for all $1 \leq i \leq V$.

Denote by $\sigma_{i,l}(\pi_{m-1})$ the processing time of task $T_{i,l}$ under policy π_{m-1} for $l \geq 1, 1 \leq i \leq V$. Define the processing times of policy π_m in the same probability space P in such a way that the processing times of tasks $T_{i,j}$ and $T_{i,m}$ are switched for all $1 \leq i \leq V$:

$$\sigma_{i,l}(\pi_m) = \sigma_{i,l}(\pi_{m-1}), \quad \sigma_{i,m}(\pi_m) = \sigma_{i,j}(\pi_{m-1}), \quad \sigma_{i,j}(\pi_m) = \sigma_{i,m}(\pi_{m-1}), \qquad l \neq m, \ l \neq j, \quad 1 \leq i \leq V.$$

With such a coupling, we have that

$$\begin{array}{llll} s_{i,l}(\pi_m) = s_{i,l}(\pi_{m-1}), & s_{i,m}(\pi') = s_{i,j}(\pi), & s_{i,j}(\pi') = s_{i,m}(\pi), & l \neq m, \ l \neq j, \ 1 \leq i \leq V; \\ c_{i,l}(\pi_m) = c_{i,l}(\pi_{m-1}), & c_{i,m}(\pi') = c_{i,j}(\pi), & c_{i,j}(\pi') = c_{i,m}(\pi), & l \neq m, \ l \neq j, \ 1 \leq i \leq V. \end{array}$$

Fix the processing times of the tasks belonging to the first $m-1$ jobs. Denote these processing times by $\mathcal{S} = \{\sigma_{i,l}(\pi_m), \ 1 \leq i \leq V, \ 1 \leq l \leq m-1\}$. Define the functions:

$$\begin{aligned} F_{\mathcal{S}}(\pi_{m-1}) &= \sum_{l=1}^{n} E_{A,\mathcal{S}}\left[f\left(R_l(\pi_{m-1})\right)\right] \\ F_{\mathcal{S}}(\pi_m) &= \sum_{l=1}^{n} E_{A,\mathcal{S}}\left[f\left(R_l(\pi_m)\right)\right] \end{aligned}$$

where $E_{A,\mathcal{S}}$ denotes the conditional expectation given the arrival times and the processing times of the first $m-1$ jobs $\mathcal{S}$.

We will study the difference $F_{\mathcal{S}}(\pi_{m-1}) - F_{\mathcal{S}}(\pi_m)$. Denote by $P_{\mathcal{S}}(j=k)$, where $k \geq m$, the conditional probability of the event $\{j=k\}$ given the arrival times and $\mathcal{S}$. Note that when $\mathcal{S}$ is fixed, for any $k \geq m$, the event $\{j=k\}$ is independent of the processing times of tasks belonging to jobs $J_m, J_{m+1}, \cdots$. Therefore,

$$\begin{aligned} & F_{\mathcal{S}}(\pi_{m-1}) - F_{\mathcal{S}}(\pi_m) \\ &= \sum_{k=m+1}^{n} P_{\mathcal{S}}(j=k) \Big\{ E_{A,\mathcal{S}}[f(c_m(\pi_{m-1}) - a_m)] + E_{A,\mathcal{S}}[f(c_k(\pi_{m-1}) - a_k)] \\ &\qquad - E_{A,\mathcal{S}}[f(c_m(\pi_m) - a_m)] - E_{A,\mathcal{S}}[f(c_k(\pi_m) - a_k)] \Big\} \\ &\quad + \sum_{k=n+1}^{\infty} P_{\mathcal{S}}(j=k) \left\{ E_{A,\mathcal{S}}[f(c_m(\pi_{m-1}) - a_m)] - E_{A,\mathcal{S}}[f(c_m(\pi_m) - a_m)] \right\} \end{aligned}$$

$$= \sum_{k=m+1}^{n} P_{\mathcal{S}}(j=k)\,\Big\{E_{A,\mathcal{S}}[f(c_m(\pi_{m-1})-a_m)] + E_{A,\mathcal{S}}[f(c_k(\pi_{m-1})-a_k)]$$

$$- E_{A,\mathcal{S}}[f(c_k(\pi_{m-1})-a_m)] - E_{A,\mathcal{S}}[f(c_m(\pi_{m-1})-a_k)]\Big\}$$

$$+ \sum_{k=n+1}^{\infty} P_{\mathcal{S}}(j=k)\,\{E_{A,\mathcal{S}}[f(c_m(\pi_{m-1})-a_m)] - E_{A,\mathcal{S}}[f(c_k(\pi_{m-1})-a_m)]\} \qquad (5)$$

Consider an event $\{j=k\}$, where $k>m$. Let b_1 and b_2 denote the times at which jobs J_k and J_m, respectively, begin execution under π_{m-1}. By defintion, $b_1 \le b_2$. Denote by $\tilde{\sigma}_{i,k}(\pi_{m-1}) = \max(s_{i,k}(\pi_{m-1}) + \sigma_{i,k}(\pi_{m-1}) - b_2, 0)$ the residual life of task $T_{i,l}$ at time b_2 under π. Since the processing times have an IFR distribution, we get that $\tilde{\sigma}_{i,k}(\pi_{m-1}) \le_{st} \sigma_{i,k}(\pi_{m-1})$. Therefore,

$$c_k(\pi_{m-1}) \le b_2 + \max_{1\le i\le V} \tilde{\sigma}_{i,k}(\pi_{m-1}) \le_{st} b_2 + \max_{1\le i\le V} \sigma_{i,k}(\pi_{m-1}) =_{st} b_2 + \max_{1\le i\le V} \sigma_{i,m}(\pi_{m-1}) \le c_m(\pi_{m-1}).$$

Applying Strassen's theorem to the random variables $c_k(\pi_{m-1}), c_m(\pi_{m-1})$ entails that there are two random variables $\hat{c}_k(\pi_{m-1}), \hat{c}_m(\pi_{m-1})$ on a common probability space P' such that

$$\hat{c}_k(\pi_{m-1}) =_{st} c_k(\pi_{m-1}), \quad \hat{c}_m(\pi_{m-1}) =_{st} c_m(\pi_{m-1}), \quad \text{and} \quad \hat{c}_k(\pi_{m-1}) \le \hat{c}_m(\pi_{m-1}), \quad P' - a.s..$$

Using the facts that $a_k \ge a_m$ and that the function f is increasing and convex implies that

$$f(\hat{c}_m(\pi_{m-1}) - a_m) + f(\hat{c}_k(\pi_{m-1}) - a_k) \ge f(\hat{c}_k(\pi_{m-1}) - a_m) + f(\hat{c}_m(\pi_{m-1}) - a_k),$$

$$f(\hat{c}_m(\pi_{m-1}) - a_m)] \ge f(\hat{c}_k(\pi_{m-1}) - a_m).$$

Therefore, it follows from relation (??) that

$$F_{\mathcal{S}}(\pi_{m-1}) \ge F_{\mathcal{S}}(\pi_m).$$

Unconditioning with respect to $\mathcal{S}$ implies that $F(\pi_{m-1}) \ge F(\pi_m)$, so that, by induction, relation (??) holds. This completes the proof. ■

A sequence of random variables $\{X_n\}_n$ is said to converge to a stationary random variable X in the sense of Cesaro with respect to the class of functions $\mathcal{C}$, if for all $f \in c\mathcal{C}$,

$$\lim_{n\to\infty} \frac{1}{n} \sum_{m=1}^{n} E[f(X_n)] = E[f(X)].$$

A sufficient condition for such a convergence to hold is that X_n couples in finite time with a stationary and ergodic sequence [?]. For example, the coupling exists when X_n is the response time of job J_n under FCFS policy, provided the arrival process is stationary (cf. [?, Theorem 5.5.7 and Lemma 5.5.8]). In the remainder of this paper, we will assume that the policies under consideration admit the convergence of the response times in the sense of Cesaro with respect to the class of increasing and convex functions.

Theorem 4.4 *Assume that all jobs have a single phase with a deterministic number V of tasks. Assume further that the task processing times have a common IFR distribution. Then the FCFS policy minimizes the stationary job response time in the sense of convex ordering, within the class of task nonpreemptive SP policies:*

$$\forall \pi \in \Psi_{SP}^{o} \quad : \qquad R(FCFS) \leq_{cx} R(\pi).$$

Proof. Let $f : I\!R \to I\!R$ be an arbitrary increasing and convex function. According to Lemma ??,

$$E\left[f\left(R(FCFS)\right)\right] = \lim_{n\to\infty} \frac{1}{n} \sum_{m=1}^{n} E\left[f\left(R_m(FCFS)\right)\right] \leq \lim_{n\to\infty} \frac{1}{n} \sum_{m=1}^{n} Ef\left[\left(R_m(\pi)\right)\right] = E\left[f\left(R(\pi)\right)\right].$$

Thus, $R(FCFS) \leq_{icx} R(\pi)$. Since π is a SP policy, $E[R(FCFS)] = E[R(\pi)]$. Therefore, $R(FCFS) \leq_{cx} R(\pi)$. ∎

In the case that the task processing times are constant, we can prove that FCFS minimizes the response times within the general class of non-preemptive policies.

Lemma 4.5 *Assume that all jobs have a single phase with a deterministic number V of tasks. Assume further that the task processing times are a constant. Then the FCFS policy minimizes* *the vector of the response times of the first $n \geq 1$ jobs, in the sense of increasing Schur convex ordering, within the class of task nonpreemptive policies:*

$$\forall \pi \in \Psi^{\circ} \;:\qquad (R_1(FCFS), \cdots, R_n(FCFS)) \leq_{E_1^{\uparrow}} (R_1(\pi), \cdots, R_n(\pi))\,.$$

Proof. Fix the arrival times. Let c_n denote the n-th task completion time (which is identical for all nonidling policies). For all $\pi \in \Psi^{\circ}$, let $I_n(\pi)$ be the index of the n-th completed job. Under the assumption of determistic task processing times, it is clear that

$$I_n(FCFS) = n, \quad \text{and} \quad C_n(FCFS) = c_{nV} \leq C_{I_n(\pi)}(\pi), \qquad l = 1, 2, \cdots$$

Thus, for all $n \geq 1$,

$$\begin{aligned}
&(R_1(FCFS), \cdots, R_n(FCFS)) \\
&\leq \; \Big((C_{I_1(\pi)}(\pi) - a_1), \cdots, (C_{I_n(\pi)}(\pi) - a_n)\Big) \\
&\prec \; ((C_1(\pi) - a_1), \cdots, (C_n(\pi) - a_n))
\end{aligned}$$

(cf. [?] for the second inequality). Thus, upon removal of the conditioning on the arrival times, it follows that for all increasing and Schur convex function $f : I\!R^n \to I\!R$,

$$E\left[f\left(R_1(FCFS), \cdots, R_n(FCFS)\right)\right] \leq E\left[f\left(R_1(\pi), \cdots, R_n(\pi)\right)\right].$$

■

Theorem 4.6 *Assume that all jobs have a single phase with a deterministic number V of tasks and that the task processing times are constant. Then the FCFS policy minimizes the stationary response time in the sense of increasing convex ordering, within the class of task nonpreemptive policies:*

$$\forall \pi \in \Psi^{\circ} \;:\qquad R(FCFS) \leq_{icx} R(\pi).$$

Proof. It follows from the preceding lemma and the definition of the $\leq_{E_1^{\uparrow}}$ ordering that

$$\sum_{i=1}^{n} E\left[f\left(R_m(FCFS)\right)\right] \leq \sum_{i=1}^{n} E\left[f\left(R_m(\pi)\right)\right]$$

for $n = 1, 2, \ldots$ and all increasing and convex function $f : I\!R \rightarrow I\!R$. Dividing by n on both sides of the above inequality and letting n go to infinity yields

$$E\left[f\left(R(FCFS)\right)\right] = \lim_{n\rightarrow\infty} \frac{1}{n} \sum_{m=1}^{n} E\left[f\left(R_m(FCFS)\right)\right] \leq \lim_{n\rightarrow\infty} \frac{1}{n} \sum_{m=1}^{n} Ef\left[\left(R_m(\pi)\right)\right] = E\left[f\left(R(\pi)\right)\right].$$

Therefore, we obtain

$$R(FCFS) \leq_{icx} R(\pi).$$

∎

References

[1] A. K. Agrawala, E. G. Coffman, M. R. Garey, S. K. Tripathi, "A Stochastic Optimization Algorithm Minimizing Expected Flow Times on Uniform Processors", *IEEE Trans. on Computers*, Vol.C-33, No.4, pp.351-356, April 1984.

[2] F. Baccelli, Z. Liu, D. Towsley, "Extremal Scheduling of Parallel Processing with and without Real-Time Constraints", to appear in the *Journal of the ACM.*

[3] A. Brandt, P. Franken, B. Lisek, *Stationary Stochastic Models.* Akademic-Verlag, 1989.

[4] J. Bruno, "On Scheduling Tasks with Exponential Service Times and In-Tree Precedence Constraints", *Acta Informatica*, **22** (1985), pp. 139–148.

[5] E. G. Coffman, L. Flatto, M. R. Garey, R. R. Weber, "Minimizing Expected Makespans on Uniform Processor Systems", *Adv. Appl. Prob.*, Vol.19, pp.177-201, 1987.

[6] E. G. Coffman, Z. Liu, "On the Optimal Stochastic Scheduling of Out-Forests", Rapport de Recherche INRIA No. 1156, 1990, *Operations Research*, Vol. 40, Supp. No. 1, pp. S67–S75, January 1992.

[7] C. S. Chang, R. Nelson, M. Pinedo, "Scheduling Two Classes of Exponential Jobs on Parallel Processors: Structural Results and Worst-Case Analysis", *Adv. Appl. Prob.*, Vol. 23, pp. 925-944, 1991.

[8] E. Frostig, "A Stochastic Scheduling Problem with Intree Precedence Constraints", *Operations Research*, **36** (1988), pp. 937–943.

[9] A.W. Marshall, I. Olkin, *Inequalities: Theory of Majorization and Its Applications*, Academic Press, 1979.

[10] R. Nelson, D. Towsley, A. N. Tantawi, "Performance Analysis of Parallel Processing Systems", *IEEE Trans. on Software Engineering*, Vol. 14, No. 4, pp. 532-540, April 1988.

[11] R. Nelson, "A Performance Evaluation of a General Parallel Processing Model," *Performance Evaluation Review*, **18**, 1, 13-26, 1990.

[12] R. Nelson, D. Towsley, "A Performance Evaluation of Several Priority Policies for Parallel Processing Systems," to appear in *J. ACM.*

[13] L. M. Ni, K. Hwang, "Optimal Load Balancing in a Multiple Processor Systems with Many Job Classes", *IEEE Trans. on Software Engineering*, Vol.SE-11, No.5, pp.491-496, May 1985.

[14] C. H. Papadimitriou, J. N. Tsitsiklis, "On Stochastic Scheduling with In-Tree Precedence Constraints", *SIAM Jour. Comput.*, Vol.16, No.1, pp.1-6, Feb. 1987.

[15] M. Pinedo, G. Weiss, "Scheduling Jobs with Exponentially Distributed Processing Times and Intree Precedence Constraints on Two Parallel Machines", *Operations Research*, Vol. 33, pp. 1381–1388, 1985.

[16] M. Pinedo, G. Weiss, "The 'Largest Variance First' Policy in Some Stochastic Scheduling Problems", *Oper. Res.*, Vol. 35, No.6, pp.884-891, Nov.-Dec. 1987.

[17] R. Righter, S. H. Xu, "Scheduling Jobs on Non-Identical IFR Processors to Minimize General Cost Functions", *Adv. Appl. Prob.*, Vol. 23, pp. 909-924, 1991.

[18] Z. Rosberg, A. Makowski, "Optimal Routing to Parallel Heterogeneous Servers—Small Arrival Rates", *IEEE Trans. Auto. Control*, Vol. 35, pp. 789-796, 1990.

[19] S. Stidham, "A Last Word on $L = \lambda W$", *Oper. Res.*, Vol. 22, pp.417-421, 1974.

[20] D. Stoyan, *Comparison Methods for Queues and Other Stochastic Models.* English translation (D.J. Daley editor), J.Wiley and Sons, New York, 1983.

[21] V. Strassen, "The existence of Probability Measures with Given Marginals," *Ann. Math. Stat.*, Vol. 36, pp. 423-439, 1965.

[22] D. Towsley, G. Rommel, J.A. Stankovic "Analysis of Fork-Join Program Response Times on Multiprocessors," *IEEE Transactions on Parallel and Distributed Systems*, **1**, 3, 286-303, July 1990.

[23] R. Weber, P. Varaiya, J. Walrand, "Scheduling Jobs with Stochastically Ordered Processing Times on Parallel Machines to Minimize Expected Flowtime", *J. Appl. Prob.*, Vol. 23, 841-847, 1986.

[24] Y. T. Wang, R. J. T. Morris, "Load Sharing in Distributed Systems", *IEEE Trans. on Computers*, Vol.C-34, No.3, pp.204-217, March 1985.