The PBS Policy: Some Properties and Their Proofs

Hanhua Feng Vishal Misra
Department of Computer Science
Columbia University

Dan Rubenstein
Department of Electric Engineering
Columbia University

March 20, 2007

Abstract

In this report we analyze a configurable blind scheduler containing a continuous, tunable parameter [2]. After the definition of this policy, we prove the property of no surprising interruption, the property of no permanent starvation, and two theorems about monotonicity of this policy.

1 The PBS Policy: an Introduction

We consider a system with a single work-conserving server where tasks are sequentially numbered by the order of their arrival times. In order words, $i < j$ always implies that task i arrives no later than task j. At every time t, for each task i in the system, we compute its priority value by:

$$P_i(t) = \frac{t_i(t)}{[x_i(t)]^\alpha}, \tag{1}$$

where $t_i(t)$ is the sojourn time of task i (elapsed time since its arrival), $x_(t)$ is the attained service time thus far, and the exponent α is a tunable parameter between 0 and $+\infty$. The priority-based blind scheduling (PBS) policy schedules a random task among the tasks with the maximal priority value.

We present two important properties [2] without proof.

Theorem 1.1 (basic fairness). *Under PBS, if $i > k$, $x_i(t) \leq x_k(t)$ for any time t such that both tasks are in the system.*

Theorem 1.2 (hospitality). *Under* PBS *with $\alpha > 0$, if $i > k$, then $P_i(t) \geq P_k(t)$ for any time t such that both tasks are in the system.*

2 Properties of the PBS policy

In this section, we prove two properties of the PBS policy. We say that a task is *scheduled* at time t if it receives service during interval $[t, t + \epsilon]$ for any $\epsilon > 0$. The processor share that the task receives at time t is defined to be the limit of the ratio between its received service and the elapsed time, namely ϵ, during this interval, as the elapsed time $\epsilon \to 0$. Mathematically, the processor share of task i at time t is defined to be the right-derivative of $x_i(t)$ at t, denoted by $x_i'(t)$. Clearly, if $x_i'(t) > 0$, task i must be scheduled at time t.

Note that two scheduled tasks, i and j, must have the same priority value, i.e.,

$$\frac{t_i(t)}{[x_i(t)]^\alpha} = \frac{t_k(t)}{[x_k(t)]^\alpha} \tag{2}$$

Equivalently, we have

$$\log t_i(t) - \alpha \log x_i(t) = \log t_k(t) - \alpha \log x_k(t). \tag{3}$$

2.1 No surprising interruption

The first property states that all scheduled tasks remain scheduled unless they complete, during any period in which no new tasks arrive at the system.

Property 2.1 (No surprising interruption)**.** *If $x_i'(t) > 0$ and no tasks arrive in the interval $[t, t_0]$, then either $x_i(t_0) = X_i$ or $x_i'(t_0) > 0$.*

Proof. Taking the derivative on (3), we get Taking the derivative we get

$$\frac{x_i'(t)}{x_i(t)} - \frac{x_k'(t)}{x_k(t)} = \frac{1}{\alpha}\left(\frac{1}{t_i(t)} - \frac{1}{t_k(t)}\right), \tag{4}$$

noting that $t_i(t)$ and $t_k(t)$ are linear functions of t. The right-hand side of (4) is positive for $\alpha > 0$ if task i is younger than task k. Then we get

$$\frac{x_i'(t)}{x_i(t)} > \frac{x_k'(t)}{x_k(t)}. \tag{5}$$

Taking the derivative on (4) again, we obtain (omitting variable t for short)

$$\frac{x_i''}{x_i} - \frac{x_k''}{x_k} = \frac{1}{\alpha}\left(\frac{1}{t_i} - \frac{1}{t_k}\right)\left[\left(\frac{x_i'}{x_i} + \frac{x_k'}{x_k}\right) - \left(\frac{1}{t_i} + \frac{1}{t_k}\right)\right]. \tag{6}$$

From (5), we get

$$\frac{x_k'}{x_k} + \frac{x_i'}{x_i} = \frac{2x_i'}{x_i} - \frac{1}{\alpha}\left(\frac{1}{t_i} - \frac{1}{t_k}\right) \geq -\frac{1}{\alpha}\left(\frac{1}{t_i} - \frac{1}{t_k}\right),$$

and then (6) becomes

$$\frac{x_i''}{x_i} - \frac{x_k''}{x_k} \geq \frac{1}{\alpha}\left(\frac{1}{t_i} - \frac{1}{t_k}\right)\left[\left(\frac{1}{\alpha} - 1\right)\frac{1}{t_k} - \left(\frac{1}{\alpha} + 1\right)\frac{1}{t_i}\right].$$

For $t_i(t) < t_k(t)$ so that $i > k$, the right-hand-side of the preceding inequality is negative if $\alpha \geq 1$. In other words,

$$\frac{x_i''(t)}{x_i(t)} < \frac{x_k''(t)}{x_k(t)} \quad \text{for } \alpha \geq 1.$$

Therefore, if $x_k''(t) \leq 0$, we must have $x_i''(t) < 0$ for every younger task i. Suppose task $k = \min \mathcal{K}(t)$ is the oldest scheduled task where $\mathcal{K}(t)$ is the set of all scheduled tasks. Then $x_k''(t) \leq 0$ implies $\sum_{i \in \mathcal{K}(t)} x_i''(t) < 0$ if $|\mathcal{K}(t)| \geq 2$. This implication, however, contradicts the work-conserving principle (i.e., $\sum_{i \in \mathcal{K}(t)} x_i'(t) = 1$ and therefore $\sum_{i \in \mathcal{K}(t)} x_i''(t) = 0$). Hence we must have $x_k''(t) > 0$ for oldest scheduled task i, meaning that, for $\alpha \geq 1$, at any point of time the oldest scheduled task gains more and more processor share and therefore remains to be scheduled, and consequently all younger tasks also keep being scheduled by Theorem 1.2 (hospitality). Hence, for $\alpha \geq 1$ at any time, a scheduled task can only be interrupted by the future arrivals, not those already in the system. □

2.2 No permanent starvation

The second property states that no task will experience permanent starvation under the PBS policy with $0 < \alpha < \infty$ if the total number of tasks at any time is bounded.

Property 2.2 (No permanent starvation)**.** *For a system where the number of tasks is bounded, every task finishs in finite time under the PBS policy if either*

(i) $0 < \alpha \le 1$*, or*

(ii) $1 < \alpha < \infty$ *and there is a lower bound for every task.*

Proof. Suppose task j is the earliest starved task; it will remain in the system for ever. Since $x_j(t)$ is bounded, $x_j(t)/t_j(t)$, the fraction of processor time obtained by task j, must go towards to zero as $t \to \infty$.

For case (i), consider a later-arrived task k. By Theorem 1.1, $x_j(t) \ge x_k(t)$ and therefore $[x_j(t)]^{1-\alpha} \ge [x_k(t)]^{1-\alpha}$. On the other hand, by Theorem 1.2, $[x_j(t)]^\alpha/t_j(t) \ge [x_k(t)]^\alpha/t_k(t)$. Multiplying these two inequalities we get $x_j(t)/t_j(t) \ge x_k(t)/t_k(t)$ for every later-arrived task k. Hence, if task j is starved and takes less and less processor share, so are all other tasks. This scenario is not possible for a system with a finite number of tasks.

For case (ii), let y be the lower bound of task sizes. Since the priority value of task j goes to infinity, the priority value of every later-arrived task k on its departure, i.e., $t_k(\tau_k^d)/[x_k(\tau_k^d)]^\alpha$ where τ_k^d is the departure time of task k, also goes to infinity on its departure time. Since $x_k(\tau_k^d) \ge y$, we get $x_k(\tau_k^d)/t_k(\tau_k^d) \le [x_k(\tau_k^d)]^\alpha/[t_k(\tau_k^d)y^{\alpha-1}]$ goes to zero for every task k. This scenario is again not possible for a system with a finite number of tasks.

We finally note that the lower bound of task sizes for case (ii) is indeed necessary; we can construct a series tasks with decreasing sizes so as to starve a longer task. ☐

3 The Monotonicity of the PBS policy

3.1 The deterministic case

We need to define a few quantities before showing theorems about monotonicity of the PBS policy. We denote by $\nu(t)$ the total number of tasks arriving before or on time t. Quantity $S(t,k)$ is the total attained service time by the first k tasks at time t:

$$S(t,k) := \sum_{i=1}^{k} x_i(t), \qquad 0 < k \le \nu(t),$$

which can be split into two portions by a threshold ξ, $0 \le \xi \le \infty$, for the attained service time of every task. The first portion is to count only attained service of first ξ seconds of the first k tasks, and the second one is to count only attained service beyond the first ξ seconds, i.e.,

$$S_\xi^-(t,k) := \sum_{i=1}^{k}[x_i(t) \wedge \xi], \quad S_\xi^+(t,k) := \sum_{i=1}^{k}[x_i(t) - \xi]^+,$$

where $z^+ := z \vee 0$ and we use $\wedge$ and $\vee$ to denote minimum and maximum, respectively. For the case that $k = \nu(t)$, we use shorthand notations

$$S_\xi^+(t) := S_\xi^+(t,\nu(t)) \quad \text{and} \quad S_\xi^-(t) := S_\xi^-(t,\nu(t))$$

as $S(t)$ is a shorthand notation of $S(t,\nu(t))$.

The following theorem states that the total attained service beyond ξ of the first k tasks is greater for a smaller α:

Theorem 3.1. *Consider two policies $\mathcal{P}_1 := \mathrm{PBS}(\alpha_1)$ and $\mathcal{P}_2 := \mathrm{PBS}(\alpha_2)$. If $0<\alpha_1\le\alpha_2<\infty$, then $S_\xi^+(t,k)^{\mathcal{P}_1} \ge S_\xi^+(t,k)^{\mathcal{P}_2}$, for any $t\ge 0$, $\xi\in[0,\infty]$ and $k=1,2,\ldots,\nu(t)$.*

Before proving this theorem, we need the following lemma, which states that, if some scheduler task receive more service with a larger α, all younger tasks must also receive more service with this larger α.

Lemma 3.2. *Consider two policies $\mathcal{P}_1 := \mathrm{PBS}(\alpha_1)$ and $\mathcal{P}_2 := \mathrm{PBS}(\alpha_2)$, with $0<\alpha_1\le\alpha_2<\infty$. Suppose at least one task in the first k tasks is scheduled at time t by $\mathcal{P}_2$, and suppose $x_k(t)^{\mathcal{P}_1} < x_k(t)^{\mathcal{P}_2}$. Then, $x_i(t)^{\mathcal{P}_1} \le x_i(t)^{\mathcal{P}_2}$ for all $i>k$.*

Proof of Lemma 3.2. For every task i, consider two cases at time t:

(i) The task i has finished its service under $\mathcal{P}_2$, i.e., $x_i(t)^{\mathcal{P}_2} = X_i$. Clearly we have $x_i(t)^{\mathcal{P}_1} \le x_i(t)^{\mathcal{P}_2}$ and we are done.

(ii) The task i has not finished its service under $\mathcal{P}_2$. Because there is an older scheduled task under $\mathcal{P}_2$, by Theorem 1.1 (basic fairness), the task i must also be scheduled under $\mathcal{P}_2$.

Now we consider two sub-cases of case (ii) at time t:

(iia) The task k has not finished the service under $\mathcal{P}_2$. Then, tasks k and i must both be scheduled, i.e., they have the same priority value:

$$\frac{t_i(t)}{[x_i(t)^{\mathcal{P}_2}]^{\alpha_2}} = \frac{t_k(t)}{[x_k(t)^{\mathcal{P}_2}]^{\alpha_2}}.$$

(iib) The task k have already left the system under $\mathcal{P}_2$. Let τ be its departure time. Let task j be that older task scheduled at time t. Note that $j<k<i$. Then, at time τ, we have

$$\frac{t_j(\tau)}{[x_j(\tau)^{\mathcal{P}_2}]^{\alpha_2}} \le \frac{t_k(\tau)}{[x_k(\tau)^{\mathcal{P}_2}]^{\alpha_2}} = \frac{t_k(\tau)}{\left[X_k^{\mathcal{P}_2}\right]^{\alpha_2}}, \tag{7}$$

because task k is scheduled at time τ. Then, the following shows that a finished younger task, namely k, should have a higher priority value than the scheduled older task j, not only at the departure time of the former, but also at any later time:

$$\begin{aligned}
\frac{t_k(t)}{[x_k(t)^{\mathcal{P}_2}]^{\alpha_2}} &= \frac{t_k(\tau)}{[x_k(\tau)^{\mathcal{P}_2}]^{\alpha_2}} + \frac{t-\tau}{[x_k(\tau)^{\mathcal{P}_2}]^{\alpha_2}} \quad [x_k(t)=x_k(\tau)] \\
&\ge \frac{t_j(\tau)}{[x_j(\tau)^{\mathcal{P}_2}]^{\alpha_2}} + \frac{t-\tau}{[x_k(\tau)^{\mathcal{P}_2}]^{\alpha_2}} \quad [\text{by (7)}] \\
&\ge \frac{t_j(\tau)}{[x_j(t)^{\mathcal{P}_2}]^{\alpha_2}} + \frac{t-\tau}{[x_j(t)^{\mathcal{P}_2}]^{\alpha_2}} \quad [\text{Basic fairness}] \\
&= \frac{t_j(t)}{[x_j(t)^{\mathcal{P}_2}]^{\alpha_2}} = \frac{t_i(\tau)}{[x_i(t)^{\mathcal{P}_2}]^{\alpha_2}}. \quad [i,j \text{ both scheduled at } t]
\end{aligned}$$

In short, for both cases (iia) and (iib), we have

$$\frac{t_i(t)}{[x_i(t)^{\mathcal{P}_2}]^{\alpha_2}} \le \frac{t_k(t)}{[x_k(t)^{\mathcal{P}_2}]^{\alpha_2}}. \tag{8}$$

Now consider tasks under $\mathcal{P}_1$. The task k must not have finished its service at time t, by the premise of the lemma, although it may or may not be scheduled under $\mathcal{P}_1$. If the younger task i is finished under $\mathcal{P}_1$, it should have a priority value higher than scheduled task k (for the same reason as above); if it is not finished, by Theorem 1.2 (hospitality), it must also be scheduled, and has the same priority as task k. In either case,

$$\frac{t_i(t)}{[x_i(t)^{\mathcal{P}_1}]^{\alpha_1}} \geq \frac{t_k(t)}{[x_k(t)^{\mathcal{P}_1}]^{\alpha_1}}. \tag{9}$$

Note that the sojourn time $t_j(t)$ is policy invariant. Then we have (the variable t are omitted at some places

$$\begin{aligned} x_i(t)^{\mathcal{P}_1} &= \left(x_i^{\mathcal{P}_1}\right)^{\frac{\alpha_1}{\alpha_2}} \left(x_i^{\mathcal{P}_1}\right)^{1-\frac{\alpha_1}{\alpha_2}} \\ &\leq \left(\frac{t_i}{t_k}\right)^{\frac{1}{\alpha_2}} \left(x_k^{\mathcal{P}_1}\right)^{\frac{\alpha_1}{\alpha_2}} \left(x_i^{\mathcal{P}_1}\right)^{1-\frac{\alpha_1}{\alpha_2}} \quad \text{[by (9)]} \\ &\leq \left(\frac{x_i^{\mathcal{P}_2}}{x_k^{\mathcal{P}_2}}\right) \left(x_k^{\mathcal{P}_1}\right)^{\frac{\alpha_1}{\alpha_2}} \left(x_i^{\mathcal{P}_1}\right)^{1-\frac{\alpha_1}{\alpha_2}} \quad \text{[by (8)]} \\ &= x_i^{\mathcal{P}_2} \left(\frac{x_i^{\mathcal{P}_1}}{x_k^{\mathcal{P}_1}}\right)^{1-\frac{\alpha_1}{\alpha_2}} \left(\frac{x_k^{\mathcal{P}_1}}{x_k^{\mathcal{P}_2}}\right) < x_i(t)^{\mathcal{P}_2}. \end{aligned}$$

Note that for the last inequality we need (1) that $x_k(t)^{\mathcal{P}_1} < x_k(t)^{\mathcal{P}_2}$ by the lemma premise, (2) that $x_i(t)^{\mathcal{P}_1} \leq x_k(t)^{\mathcal{P}_1}$ by Theorem 1.1 (basic fairness) since task k has not finished under $\mathcal{P}_1$, and (3) that $1 - \alpha_1/\alpha_2 \geq 0$. □

Proof of Theorem 3.1. Again we prove it by contradiction. Let us hypothesize that Theorem 3.1 does not hold and let k be *the smallest integer* such that

$$S_\xi^+(t,k)^{\mathcal{P}_1} < S_\xi^+(t,k)^{\mathcal{P}_2}, \tag{10}$$

for some t. Since $S_\xi^+(t,k)$ is continuous with respect to t, we can assume $S_\xi^+(t,k)^{\mathcal{P}_2}$ is also strictly increasing at time t because $S_\xi^+(t,k)^{\mathcal{P}_1}$ is non-decreasing, using the same argument as in the proof of Theorem 1.1. Therefore, there must be at least one scheduled task in the first k ones, and all younger tasks still in the system must also be scheduled at time t, by Theorem 1.2 (hospitality).

By the contradictory hypothesis of this proof we have

$$S_\xi^+(t,k-1)^{\mathcal{P}_1} \geq S_\xi^+(t,k-1)^{\mathcal{P}_2}, \tag{11}$$

and the difference between (10) and (11) shows

$$\left(x_k(t)^{\mathcal{P}_1} - \xi\right)^+ < \left(x_k(t)^{\mathcal{P}_2} - \xi\right)^+,$$

or equivalently,

$$x_k(t)^{\mathcal{P}_2} > x_k(t)^{\mathcal{P}_1} \vee \xi. \tag{12}$$

Then by Theorem 1.1 (basic fairness), for all $j \leq k$, either (i) we have $x_j(t)^{\mathcal{P}_2} \geq x_k(t)^{\mathcal{P}_2} > \xi$ or (ii) the older task j has finished the service under $\mathcal{P}_2$, i.e., $x_j(t)^{\mathcal{P}_2} = X_j$. For case (i), $x_j(t)^{\mathcal{P}_1} \wedge \xi \leq$

$\xi = x_j(t)^{\mathcal{P}_2} \wedge \xi$, and for case (ii), $x_j(t)^{\mathcal{P}_1} \wedge \xi \leq X_j \wedge \xi = x_j(t)^{\mathcal{P}_2} \wedge \xi$. Thus, in either case, we have $x_j(t)^{\mathcal{P}_1} \wedge \xi \leq x_j(t)^{\mathcal{P}_2} \wedge \xi$ Summing up, we get

$$S_\xi^-(t,k)^{\mathcal{P}_1} \leq S_\xi^-(t,k)^{\mathcal{P}_2} \tag{13}$$

As stated earlier (above (11)), at time t, under policy $\mathcal{P}_2$, there must be at least one scheduled task in the first k tasks. By Lemma 3.2, we get from (12) that, for all $i > k$,

$$x_i(t)^{\mathcal{P}_1} \leq x_i(t)^{\mathcal{P}_2}.$$

With (10), (13) and the preceding inequality, we finally get

$$\begin{aligned} S(t)^{\mathcal{P}_1} &= \sum_{i=k+1}^{\nu(t)} x_i(t)^{\mathcal{P}_1} + S_\xi^-(t,k)^{\mathcal{P}_1} + S_\xi^+(t,k)^{\mathcal{P}_1} \\ &< \sum_{i=k+1}^{\nu(t)} x_i(t)^{\mathcal{P}_2} + S_\xi^-(t,k)^{\mathcal{P}_2} + S_\xi^+(t,k)^{\mathcal{P}_2} = S(t)^{\mathcal{P}_2}, \end{aligned}$$

which violates the work-conserving principle. Hence (10) is not possible, and the proof is complete. □

3.2 The stochastic case

We now consider a $G/GI/1$ queue with an arrival rate denoted by λ. Task sizes are independent, identically distributed random variable, whose cumulative distribution function and probability density function are denoted by $F(\cdot)$ and $f(\cdot)$, respectively. We denote by random variable T the response time of a task, and by T_x the response time conditioned on its size x.

The following theorem states that the mean response time of the PBS policy is monotonic with respect to α with IHR and DHR task-size distributions in a $G/GI/1$ queue. A random variable is IHR (DHR) distributed if its hazard rate $f(x)/[1-F(x)]$ is increasing (decreasing).

Theorem 3.3. *Consider two policies $\mathcal{P}_1 :=$ PBS(α_1) and $\mathcal{P}_2 :=$ PBS(α_2). If $0 < \alpha_1 \leq \alpha_2 < \infty$, then in a $G/GI/1$ queue, $\mathbb{E}T^{\mathcal{P}_1} \geq \mathbb{E}T^{\mathcal{P}_2}$ with DHR task-size distributions, and $\mathbb{E}T^{\mathcal{P}_1} \leq \mathbb{E}T^{\mathcal{P}_2}$ with IHR task-size distributions.*

We need the following Lemma to prove Theorem 3.3.

Lemma 3.4. [1] *Let $P(x)$ and $Q(x)$ be increasing (i.e., non-decreasing) functions, and $g(x)$ be a non-negative function. If*

$$\int_0^\xi g(x)dP(x) \leq \int_0^\xi g(x)dQ(x)$$

for every $\xi \in [0,\infty]$, then

$$\int_0^\infty g(x)h(x)dP(x) \leq \int_0^\infty g(x)h(x)dQ(x)$$

[1] This Lemma is more general than those in [1] because $P(x)$ and $Q(x)$ can be non-continuous (cf. Lemma 1 in Appendix of [1]). The integral is, however, well-defined in measure theory ($P(\cdot)$ and $Q(\cdot)$ can be considered as two different measures on the real line). And yet, the proof is much simpler.

for a decreasing function $h(x)$. If

$$\int_{\xi}^{\infty} g(x)dP(x) \leq \int_{\xi}^{\infty} g(x)dQ(x)$$

for every $\xi \in [0,\infty]$, then

$$\int_{\xi}^{\infty} g_1(x)h(x)dP(x) \leq \int_{\xi}^{\infty} g_2(x)h(x)dQ(x)$$

for an increasing function $h(x)$.

Proof. For the first claim in the lemma, with decreasing function $h(x)$, we let $H(x) = h(0) - h(x)$ and therefore $h(x) = [H(\infty) - H(x)] + h(\infty) = \int_x^\infty dH(y) + h(\infty)$. Then we get

$$\begin{aligned}
&\int_0^\infty g(x)h(x)dP(x) \\
&= h(\infty)\int_0^\infty g(x)dP(x) + \int_{x=0}^\infty g(x)\int_{y=x}^\infty dH(y)dP(x) \\
&= h(\infty)\int_0^\infty g(x)dP(x) + \int_{y=0}^\infty \int_{x=0}^{y} g(x)dP(x)dH(y)
\end{aligned}$$

Therefore, if we change $P(x)$ to $Q(x)$, the preceding quantity gets greater if $g(x) \geq 0$, since $dH(y)$ is non-negative.

For the second claim in the lemma, with increasing function $h(x)$, we get

$$\begin{aligned}
&\int_0^\infty g(x)h(x)dP(x) \\
&= h(0)\int_0^\infty g(x)dP(x) + \int_{x=0}^\infty g(x)\int_{y=0}^{x} dh(y)dP(x) \\
&= h(0)\int_0^\infty g(x)dP(x) + \int_{y=0}^\infty \int_{x=y}^{\infty} g(x)dP(x)dh(y)
\end{aligned}$$

Therefore, if we change $P(x)$ to $Q(x)$, it becomes greater if $g(x) \geq 0$, since $dh(y)$ is non-negative. $\square$

Proof of Theorem 3.3. It is important to know that the blind policies do not know the future service of a task (i.e., its remaining service), and therefore, the response time of a task of size x, namely T_x, is statistical equivalent to the sojourn time of a task of size greater than x at the time when its attained service time is x. For the same reason, $\mathbb{E}T_x$ is an increasing function of x.

Define

$$U_\xi^-(t) = \sum_{i=1}^{\nu(t)} [(\xi \wedge X_i) - x_i(t)]^+ = \left[\sum_{i=1}^{\nu(t)} (\xi \wedge X_i)\right] - S_\xi^-(t),$$

and

$$U_\xi^+(t) = \sum_{i=1}^{\nu(t)} [X_i - (\xi \vee x_i(t))]^+ = \left[\sum_{i=1}^{\nu(t)} (\xi \vee X_i)\right] - S_\xi^+(t),$$

and let U_ξ^- and U_ξ^+ be the corresponding random variables, respectively. Note that departed tasks contribute zero to either U_ξ^- or U_ξ^+. The physical meaning of U_ξ^- (U_ξ^+) is the total remaining service time of all tasks, counting only the portion in the first ξ seconds (after the first ξ seconds) of service. We also have

$$U := U_\infty^- = U_0^+ = U_\xi^- + U_\xi^+$$

for any ξ. Note that U stands for the total remaining time, which is also policy invariant.

If $0 < \alpha_1 \leq \alpha_2 < \infty$, $S_\xi^-(t)^{\mathcal{P}_1} \leq S_\xi^-(t)^{\mathcal{P}_2}$ holds for every sample path, we have then $U_\xi^-(t)^{\mathcal{P}_1} \geq U_\xi^-(t)^{\mathcal{P}_2}$, noting that $\left[\sum_{i=1}^{\nu(t)} (\xi \wedge X_i)\right]$ is policy invariant. Taking the expectation we get

$$\mathbb{E}\left(U_\xi^-\right)^{\mathcal{P}_1} \geq \mathbb{E}\left(U_\xi^-\right)^{\mathcal{P}_2}.$$

We have the following equation from Eq. (6) in [1]: (it in fact holds for $G/GI/1$ queues because it uses only Little's Law and the task-size distribution information.)

$$\mathbb{E}U_\xi^- = \lambda \int_{x=0}^{\xi} \mathbb{E}T_x[1-F(x)]dx = \lambda \int_{x=0}^{\xi} \frac{1-F(x)}{f(x)} \mathbb{E}T_x dF(x),$$

and since $U = U_\xi^- + U_\xi^+$, we obtain

$$\mathbb{E}U_\xi^+ = \lambda \int_{\xi}^{\infty} \mathbb{E}T_x[1-F(x)]dx = \lambda \int_{x=\xi}^{\infty} \frac{1-F(x)}{f(x)} \mathbb{E}T_x dF(x).$$

Letting $dP(x) = \mathbb{E}T_x^{\mathcal{P}_1} dF(x)$, $dQ(x) = \mathbb{E}T_x^{\mathcal{P}_2} dF(x)$, $g(x) = 1$, and $h(x) = [1-F(x)]/f(x)$ in Lemma 3.4, we obtain

$$\mathbb{E}T^{\mathcal{P}_1} = \int_0^\infty \mathbb{E}T_x^{\mathcal{P}_1} dF(x) \geq \int_0^\infty \mathbb{E}T_x^{\mathcal{P}_2} dF(x) = \mathbb{E}T^{\mathcal{P}_2}$$

for DHR distributions where $h(x)$ is increasing, and swap $dP(x)$ and $dQ(x)$ we obtain

$$\mathbb{E}T^{\mathcal{P}_1} = \int_0^\infty \mathbb{E}T_x^{\mathcal{P}_1} dF(x) \leq \int_0^\infty \mathbb{E}T_x^{\mathcal{P}_2} dF(x) = \mathbb{E}T^{\mathcal{P}_2}$$

for IHR distributions where $h(x)$ is decreasing. □

References

[1] S. Aalto, U. Ayesta, and E. Nyberg-Oksanen. Two-level processor-sharing scheduling disciplines: Mean delay analysis. In *Proceedings of ACM SIGMETRICS '04*, pages 97–105, 2004.

[2] H. Feng, V. Misra, and D. Rubenstein. PBS: A unified priority-based scheduler. In *Proceedings of ACM SIGMETRICS '07*, 2005.