Competitive Analysis of M/GI/1 Queueing Policies

Nikhil Bansal [1] Adam Wierman [2]

November 2002
CMU-CS-02-201

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We propose a framework for comparing the performance of two queueing policies. Our framework is motivated by the notion of competitive analysis, widely used by the computer science community to analyze the performance of online algorithms. We apply our framework to compare M/GI/1/FB and M/GI/1/SJF with M/GI/1/SRPT, and obtain new results about the performance of M/GI/1/FB and M/GI/1/SJF.

[1] Carnegie Mellon University, Computer Science Department. Email: nikhil@cs.cmu.edu
[2] Carnegie Mellon University, Computer Science Department. Email: acw@cs.cmu.edu
This work was supported by NSF Career Grant CCR-0133077 and by Pittsburgh Digital Greenhouse Grant 01-1.

Keywords: queueing; competitive analysis; scheduling; M/G/1; FB; LAS; SET; feedback; least attained service; shortest elapsed time; SRPT; shortest remaining processing time; regular variation

1 Introduction

An *online algorithm* for a problem is one in which decisions must be made based on past events without information about the future. Such algorithms are most natural in job scheduling, routing in communication networks, investment planning, and other scenarios where decisions must be made without knowledge of future events. In contrast, an *offline algorithm* for a problem is one where it is assumed that the algorithm is given the entire input instance in the beginning, and the goal is to compute the optimal or near optimal solution. For instance, problems within the domains of linear programming and graph optimization are most naturally studied as offline problems.

We will briefly introduce the notion of competitive analysis, which is used to analyze the performance of online algorithms. For details, an excellent introduction to competitive analysis and online algorithms in general can be found in [1]. We begin with the definition of an optimization problem. An optimization problem $\mathcal{P}$, consists of a set $\mathcal{I}$ of inputs and a cost function $\mathcal{C}$. With every input I is a set of feasible outputs $F(I)$, and associated with each feasible solution O in $F(I)$ is a positive cost $C(I, O)$, representing the cost of the output O on the input I. An optimal algorithm OPT is such that for all legal inputs I,

$$OPT(I) = \min_{O \in F(I)} C(I, O)$$

That is, the optimal algorithm computes the optimal solution on each instance. An online algorithm ALG is *c-competitive* if there is a constant α such that for all input sequences I,

$$ALG(I) \leq c \cdot OPT(I) + \alpha.$$

Since the online algorithm is at a disadvantage when compared with the optimal algorithm, c may depend on the size of the input, and typically $c > 1$. The goal is usually to give a constant competitive online algorithm for a problem.

The important point to note about the definition of competitive analysis is that if an algorithm is c-competitive, it guarantees that the output produced by the algorithm is no more than c times worse than optimal on all possible inputs. In particular, there are no statistical or probabilistic assumptions on the input. For our purposes, competitive analysis can be thought of as a framework for comparing the performance of two algorithms within which a performance guarantee holds for all possible inputs.

Let us now consider the comparison techniques used in queueing theory. A thorough survey of these techniques can be found in the books by Stoyan [3, 10]. Typically these techniques can be classified in two kinds:

1. Those that show a given random variable is stochastically smaller than another random variable, particularly when the job size distribution satisfies a condition such as having an *increasing/decreasing failure rate* (IFR/DFR), or being *new better/worse than used in expectation* (NBUE/NWUE). See [3, 10, 11] for a discussion of such results.

2. Those that bound the performance as a function of load, other parameters of the job size distribution, or a combination of both. For example, results such as $E[T]_{PS} \leq 1/(1-\rho)E[T]_{SRPT}$ or $E[T]_{FCFS} \leq (C^2+1)E[T]_{PS}$, where $E[T]_P$ denotes the average response time (i.e. sojourn time) under some policy P and C^2 denotes the coefficient of variation of the job size distribution.

The first approach, while interesting in its own right, is usually somewhat specific, and does not allow comparisons of policies in full generality. A drawback of the second approach is that the guarantee varies as a function of load or other parameters of the input instance such as the job size distribution. That is, it does not provide a uniform guarantee on the performance of the algorithm.

Our framework for comparing queueing policies will partially address some problems stated above. Before we state the framework, we first state our main motivating problem:

> How well can we schedule jobs in the absence of knowledge of jobs sizes, so as to minimize the total response (sojourn) time?

It is well known that the *optimal algorithm* for minimizing the total response time is Shortest-Remaining-Processing-Time-First (`SRPT`), which works on the job with the shortest remaining processing requirement at all times [8]. However, one widespread criticism of `SRPT` is that it requires exact knowledge of job sizes. This may not be available in many practical settings, for example in operating systems, where the server has no idea about the size of the job on which it is working.

Many natural scheduling policies do not use knowledge of the job sizes. Some of the most commonly studied among these are First-Come-First-Served (`FCFS`), Processor-Sharing (`PS`) and Feedback (`FB`) [1]. Of particular interest is `FB`, which is designed to perform like `SRPT` in the absence of knowledge of job sizes. Indeed, `FB` has found widespread use in operating systems like Unix and other time sharing systems like CTSS. Under `FB`, at any moment,

[1] `FB` is also referred to as Shortest-Elapsed-Time (`SET`) and Least-Attained-Service (`LAS`).

the server works on the job that has received the smallest processing at that instant. If more than one job has received the least amount of processing, then the server time-shares among these jobs (i.e. gives each such job an equal share of processing). **Our goal will be to bound the performance of `FB` as compared with `SRPT`.**

2 Preliminaries

Throughout this paper we assume that the system is a single M/GI/1 queue with arrival rate λ. We will assume that the job size distribution is continuous with probability density function $f(t)$. The cumulative job size distribution will be denoted by $F(t)$. We will denote $1 - F(t)$ by $\bar{F}(t)$, and X will refer to the service time of a job. The load (utilization), ρ, of the server is $\rho \stackrel{\text{def}}{=} \lambda E[X]$. The load made up by the jobs of size less than or equal to x is $\rho(x) \stackrel{\text{def}}{=} \lambda \int_0^x t f(t) dt$.

The focus of our work will be comparing the performance of policies to the optimal policy, `SRPT`. Schrage and Miller first derived the expressions for the response times in an M/GI/1/SRPT system [8]. This was further analyzed and generalized by [4, 6]. The optimality of `SRPT` for mean response time was shown by [7, 9].

$$E[T(x)]_{SRPT} = \frac{\lambda \int_0^x t\bar{F}(t)dt}{(1-\rho(x))^2} + \int_0^x \frac{dt}{1-\rho(t)}$$

For simplicity, we will split $E[T(x)]_{SRPT}$ into two parts, the waiting time (denoted by $E[W(x)]_{SRPT}$) and the residence time (denoted by $E[R(x)]_{SRPT}$) where $E[W(x)]_{SRPT} = \frac{\lambda \int_0^x t\bar{F}(t)dt}{(1-\rho(x))^2}$ and $E[R(x)]_{SRPT} = \int_0^x \frac{dt}{1-\rho(t)}$.

2.1 The Framework

We define the following two notions for comparing response times under a scheduling policy:

1. *Competitiveness:* We say that a policy P is *competitive* with respect to (wrt) a class of distributions $\mathcal{D}$, if there exists a c such that the average response time under P is no more than c times than that under `SRPT` for any job size distribution in $\mathcal{D}$.

   If $\mathcal{D}$ consists of the class of all possible distributions, we state that P is *competitive.*

   Formally, let $E[T](\rho, G)_P$ denote the average response time under policy P with load ρ and job size distribution G . Then, P is *competitive* wrt $\mathcal{D}$ if $\exists$ a constant c such that

$$E[T](\rho, G)_P \leq cE[T](\rho, G)_{SRPT} \quad \forall \rho < 1, \forall G \in \mathcal{D} \tag{1}$$

2. *Strict Competitiveness:* This is a stronger notion than competitiveness. Here we require that for each value of the job size x, the expected response time under P be no more than a constant times that under SRPT.

   Formally, if $E[T(x)](\rho, G)_P$ denote the expected response time for job of size x under policy P, then P is *strictly competitive* if

$$E[T(x)](\rho, G)_P \leq cE[T(x)](\rho, G)_{SRPT}, \quad \forall x, \forall \rho < 1, \forall G \tag{2}$$

   Strict competitiveness for a class of distributions $\mathcal{D}$ is similarly defined.

The important point in the definitions above is that the constant c does not depend upon the load ρ. Indeed, the question is interesting only when the load ρ is arbitrary. In particular, if the load is small it is easily seen that any reasonable policy will not be much worse than SRPT. Secondly, if c were allowed to depend on ρ, then the question becomes less interesting since it does not provide a strict enough criteria for distinguishing among different policies.

For example, let us consider the policy PS. It is well known that $E[T(x)](\rho, G)_{PS} = x/(1-\rho)$ and clearly, as $E[T(x)](\rho, G)_{SRPT} \geq x$, it is easy to see that $E[T(x)](\rho, F)_{PS} \leq 1/(1-\rho)E[T(x)](\rho, G)_{SRPT}$. However, the above definition implies that PS cannot be strictly competitive. To see this, consider any continuous job distribution with support on $[a, b]$. Then the expected response time of the job with size a will be $a/(1-\rho)$ under PS, where as it is easy to see that it will never exceed $2a$ under SRPT. Thus, choosing the load ρ arbitrarily close to 1, we can make the ratio under PS and SRPT as large as required. So, if some scheduling policy is competitive or strictly competitive, then it is close to SRPT in a very strong sense.

It is interesting to notice that, although in this paper we consider competitiveness with respect to SRPT, in general, we can consider competitiveness with respect to and arbitrary policy Q.

2.2 Our Results

We first consider policies that do not make use of job sizes. For ease of analysis, we restrict our attention to job size distributions that are continuous and have a finite mean. We show that the policy FB is strictly competitive if and only if (iff) the service distribution has a specific form. Further, if $f(x)$ is non-increasing, FB is strictly competitive iff the service distribution has a regularly varying tail (see Definition 3.1). In particular we show that

Theorem 3 *FB is strictly competitive with respect to a class of distributions $\mathcal{D}$ iff every $D \in \mathcal{D}$ has $\bar{F}(x)$ of the following form:*

$$\bar{F}(x) = ce^{-\int_z^x \alpha(t)/t dt} \text{ for some } z \text{ and all } x > z \text{ where } \lim_{t\to\infty} \alpha(t) = \alpha. \tag{3}$$

This gives a tight characterization of the class of distributions for which FB is strictly competitive and hence behaves like SRPT. The result implies that the performance of FB is likely good in practice, where distributions of job sizes are often Pareto. Surprisingly however FB will not be strictly competitive for bounded Pareto distributions or, in fact, for any bounded distributions.

Our second result deals with the policy Preemptive-Shortest-Job-First (PSJF). PSJF is often proposed as an approximation for SRPT, since it is easier to implement in a practical system. Note that under SRPT the priority of a job needs to be updated constantly as it is remaining processing time decreases. On the other hand, an incoming job is assigned a priority based on its job size, and this priority is never changed. We show that PSJF is strictly competitive for all job size distributions. In particular we show that

Theorem 4 *For all continuous job size distributions, PSJF is strictly 3-competitive.*

It is perhaps surprising that the guarantee holds for all job sizes, for all job size distributions, and for all values of load.

3 FB

We now move to an analysis of FB scheduling. Recall that under FB, the job with the least attained service gets the processor to itself. If several jobs all have the least attained service, they time-share the processor via PS. This is a very practical policy, since a job's *age* is always known, although it's *size* may not be known.

We need some preliminary notation.

$$\rho_x = \lambda \int_0^x \bar{F}(t)dt = \lambda \left(\int_0^x tf(t)dt + x\bar{F}(x) \right)$$

Then we have [2]

$$E[T(x)]_{FB} = \frac{\lambda \int_0^x t\bar{F}(t)dt}{(1-\rho_x)^2} + \int_0^x \frac{dt}{1-\rho_x}$$

Although FB does not technically have a waiting time, it will be useful when comparing to SRPT to define $E[W(x)]_{FB} = \frac{\lambda \int_0^x t\bar{F}(t)dt}{(1-\rho_x)^2}$ and $E[R(x)]_{FB} = \int_0^x \frac{dt}{1-\rho_x}$.

It is easy to see that for any distribution, if load $\rho < 1$, then $E[T(x)]_{FB}$ is no more than $1/(1-\rho)^2$ times $E[T(x)]_{SRPT}$. We now ask a stronger question as to whether FB is strictly competitive with respect to SRPT.

Moving towards our goal, we notice that the ratio of $E[R(x)]_{FB}$ and $E[R(x)]_{SRPT}$ is at least $(1-\rho(x))/(1-\rho_x)$ and that the ratio of $E[W(x)]_{FB}$ and $E[W(x)]_{SRPT}$ is exactly $(1-\rho(x))^2/(1-\rho_x)^2$. Thus, a necessary condition for $E[T(x)]_{FB}/E[T(x)]_{SRPT} < c$, is that $(1-\rho(x))/(1-\rho_x) \leq c$. We state this condition as

Observation 1 $E[T(x)]_{FB}/E[T(x)]_{SRPT} > (1-\rho(x))/(1-\rho_x)$ *Thus*

$$\frac{1-\rho(x)}{1-\rho_x} > c \Rightarrow \frac{E[T(x)]_{FB}}{E[T(x)]_{SRPT}} > c$$

Surprisingly, this condition is also sufficient in the sense of the following theorem.

Theorem 1 $(1-\rho(x))/(1-\rho_x) \leq c$ *implies that*

$$\frac{E[T(x)]_{FB}}{E[T(x)]_{SRPT}} \leq 3c^2$$

Before we prove this theorem, we need two lemmata.

Lemma 3.1

$$E[R(x)]_{FB} - E[R(x)]_{SRPT} \leq 2E[W(x)]_{FB} \tag{4}$$

Proof: Consider,

$$
\begin{aligned}
E[R(x)]_{FB} - E[R(x)]_{SRPT} &= \int_0^x \frac{(\rho_x - \rho(t))dt}{(1-\rho(t))(1-\rho_x)} \\
&\leq \int_0^x \frac{(\rho_x - \rho(t))dt}{(1-\rho_x)(1-\rho_x)} \\
&\leq \frac{\int_0^x (\rho_x - \rho(t))dt}{(1-\rho_x)(1-\rho_x)} \\
&= \frac{x\rho_x - \int_0^x \rho(t)dt}{(1-\rho_x)(1-\rho_x)} \\
&= \frac{x\rho_x - x\rho(x) + \lambda \int_0^x t^2 f(t)dt}{(1-\rho_x)(1-\rho_x)} \\
&= \frac{\lambda(\int_0^x t^2 f(t)dt + x^2 \bar{F}(x))}{(1-\rho_x)(1-\rho_x)} \\
&= 2E[W(x)]_{FB}
\end{aligned}
$$

The first step follows by simply substituting the expressions for $E[R(x)]_{FB}$ and $E[R(x)]_{SRPT}$. The second step follows from the first since $1 - \rho(t) \geq 1 - \rho(x) \geq 1 - \rho_x$. □

We can now extend this lemma as follows.

Lemma 3.2 *If $E[W(x)]_{FB} \leq kE[W(x)]_{SRPT}$, then $E[T(x)]_{FB} \leq 3kE[T(x)]_{SRPT}$.*

Proof: We know that $E[W(x)]_{FB} \leq kE[W(x)]_{SRPT}$ and hence

$$3E[W(x)]_{FB} \leq 3kE[W(x)]_{SRPT}$$

Adding $E[R(x)]_{SRPT}$ to the left hand side and $3kE[R(x)]_{SRPT}$ to the right hand side we obtain:

$$3E[W(x)]_{FB} + E[R(x)]_{SRPT} \leq 3kE[W(x)]_{SRPT} + 3kE[R(x)]_{SRPT} \tag{5}$$

But, by Lemma 3.1 we have that $E[R(x)]_{FB} \leq 2E[W(x)]_{FB} + E[R(x)]_{SRPT}$, and hence that

$$E[W(x)]_{FB} + E[R(x)]_{FB} \leq 3E[W(x)]_{FB} + E[R(x)]_{SRPT} \tag{6}$$

Combining Equation 5 and Equation 6 gives that

$$E[W(x)]_{FB} + E[R(x)]_{FB} \leq 3k(E[W(x)]_{SRPT} + E[R(x)]_{SRPT})$$

and hence the result follows. $\Box$

We can now easily prove Theorem 1.

Proof: (of Theorem 1) Clearly if $(1-\rho(x))/(1-\rho_x) \leq c$ this implies that $E[W(x)]_{FB}/E[W(x)]_{SRPT} \leq c^2$, which by Lemma 3.2 implies that $E[T(x)]_{FB}/E[T(x)]_{SRPT} \leq 3c^2$. Thus we are done. $\Box$

By Observation 1 and Theorem 1 it follows that in order to prove strict competitiveness it is sufficient to consider the quantity $(1-\rho(x))/(1-\rho_x)$ and show that it is bounded by a constant for all x and ρ. Our goal will be characterize the service distributions for which this property is satisfied.

We first show that `FB` is not strictly competitive for any bounded distribution.

Theorem 2 *FB is not strictly competitive under any bounded service distribution.*

Proof: Since the service distribution is bounded, there is a finite p such that $\bar{F}(x) > 0$ for all $x < p$ and $\bar{F}(p) = 0$.

We will show that $(1-\rho(x))/(1-\rho_x)$ can be made arbitrarily large by choosing $\rho \to 1$ and x arbitrarily close to p. By observation 1 we know that $(1-\rho(x))/(1-\rho_x) \leq c$ is necessary for $E[T(x)]_{FB}/E[T(x)]_{SRPT} \leq c$, thus this will give us the desired result.

When $\rho \to 1$,

$$\begin{aligned}
\frac{1-\rho(x)}{1-\rho_x} &\to \frac{\rho-\rho(x)}{\rho-\rho_x} \\
&= \frac{\int_x^p tf(t)dt}{\int_x^p \bar{F}(t)dt} \\
&\geq \frac{x\int_x^p f(t)dt}{\int_x^p \bar{F}(t)dt} \\
&\geq \frac{x\int_x^p f(t)dt}{(p-x)\bar{F}(x)} \\
&= \frac{x\bar{F}(x)}{(p-x)\bar{F}(x)} \\
&= \frac{x}{p-x}
\end{aligned}$$

The fourth step follows from the third since $\bar{F}(x)$ is decreasing in x.

Now, choosing $x = p - \varepsilon$ and making ε arbitrarily small we can make $\frac{1-\rho(x)}{1-\rho_x}$ and hence $E[T(x)]_{FB}/E[T(x)]_{SRPT}$ as large as required. □

Thus, we only need to consider distributions which have an infinite support. For $(1 - \rho(x))/(1 - \rho_x) \leq c$ to hold for all x, clearly a necessary condition is that it holds when $x \to \infty$:

$$\lim_{x\to\infty} \frac{1-\rho(x)}{1-\rho_x} \leq c \tag{7}$$

We will now show that Condition 7 holds only for probability distributions satisfying Equation 3. And, finally, all distributions of this type will satisfy our condition.

Before stating the theorem, recall the following definition.

Definition 3.1 *A distribution function $F(x)$, $x \geq 0$ is said to have a regularly varying tail with index $\alpha < 0$ if, for arbitrary $t > 0$,*

$$\lim_{x\to\infty} \frac{\bar{F}(tx)}{\bar{F}(x)} = t^{\alpha}$$

It is important to point out that in the special case when $f(x)$ is non-increasing, we have additionally that Equation 3 holds iff $F(x)$ has a regularly varying tail [5, Pages 54-74].

Theorem 3 *FB is strictly competitive iff every the service distribution is of the form:*

$$\bar{F}(x) = ce^{-\int_z^x \alpha(t)/t dt} \text{ for some } z \text{ and all } x > z \text{ where } \lim_{t\to\infty} \alpha(t) = \alpha. \tag{8}$$

Proof: We begin by showing that Condition 7 holds for distributions of the form 8. Again, let $\rho \to 1$. (Notice that for all ρ bounded away from 1 Condition 7 holds trivially.)

$$\frac{1-\rho(x)}{1-\rho_x} \quad \to \quad \frac{\int_x^\infty tf(t)dt}{\int_x^\infty \bar{F}(t)dt}$$

As $x \to \infty$, both the fractions tend to 0, so we need to evaluate the limit by applying L'Hopital's Rule. Thus, the limit is $\frac{xf(x)}{\bar{F}(x)}$. Let $\mu(x) \stackrel{\text{def}}{=} f(x)/\bar{F}(x)$ (a.k.a. the hazard rate of X). Then Condition 7 becomes:

$$\lim_{x\to\infty} x\mu(x) \leq \alpha$$

This condition is met iff $\bar{F}(x)$ is of the form specified in Equation 8 [5, Pages 54-74].

Now, for the other direction we must show that for distributions satisfying Equation 8 there exists some constant k such that $(1-\rho(x))/(1-\rho_x) \leq k$ holds for all x. Consider a finite x. We have already shown that there exists a constant c such that $\lim_{x\to\infty}(1-\rho(x))/(1-\rho_x) \leq \alpha$. Thus, because $t\mu(t)$ is continuous and $\bar{F}(t)$ has unbounded support, there exists a constant for each x, α_x, such that $t\mu(t) \leq \alpha_x$ for all $t \geq x$. Applying this bound as $\rho \to 1$:

$$\begin{aligned}
\frac{1-\rho(x)}{1-\rho_x} &\to \frac{\int_x^\infty t f(t)dt}{\int_x^\infty \bar{F}(t)dt} \\
&= \frac{\int_x^\infty t\mu(t)\bar{F}(t)dt}{\int_x^\infty \bar{F}(t)dt} \\
&\leq \frac{\alpha_x \int_x^\infty \bar{F}(t)dt}{\int_x^\infty \bar{F}(t)dt} = \alpha_x
\end{aligned}$$

Finally, we can again notice that $x\mu(x)$ is continuous and has a finite limit both as $x \to 0$ and as $x \to \infty$; thus there exists a $k < \infty$ such that $\sup \alpha_x = k$, which completes the proof. $\square$

4 PSJF

At any given point, the `PSJF` policy schedules the job in the system that arrived with the smallest size. This is similar to `SRPT` in that `PSJF` biases towards the short jobs, however it can be viewed as a harsher policy because it does not allow the large jobs to increase their priority when they become short. Further, `PSJF` is much simpler to implement than `SRPT` since we need only assign priorities to jobs upon arrival; we do not change the priority of a job as it is worked on. We can write the expected time in system for a job of size x under this policy as follows

$$E[T(x)]^{PSJF} = \frac{\lambda \int_0^x t^2 f(t)dt}{2(1-\rho(x))^2} + \frac{x}{1-\rho(x)}$$

Again, we will call the first term the waiting time, i.e. $E[W(x)]_{PSJF} = \frac{\lambda \int_0^x t^2 f(t)dt}{2(1-\rho(x))^2}$ and the second term will be the residence time i.e. $E[R(x)]_{PSJF} = \frac{x}{1-\rho(x)}$. Interestingly, although `PSJF` is much more unfair to large jobs than `SRPT`, it is still strictly competitive under all distributions.

Theorem 4 *For all continuous job size distributions, PSJF is strictly 3-competitive with respect to SRPT.*

Before we prove Theorem 4, we will need a lemma, which bounds the residence time under `PSJF`.

Lemma 1 $E[R(x)]_{PSJF} - E[R(x)]_{SRPT} \leq 2E[W(x)]_{PSJF}$.

Proof: We consider the quantity $E[R(x)]_{PSJF} - E[R(x)]_{SRPT}$.

$$\begin{aligned}
E[R(x)]_{PSJF} - E[R(x)]_{SRPT} &= \frac{x}{1-\rho(x)} - \int_0^x \frac{dt}{1-\rho(t)} \\
&= \int_0^x \frac{(\rho(x)-\rho(t))dt}{(1-\rho(t))(1-\rho(x))} \\
&\leq \int_0^x \frac{(\rho(x)-\rho(t))dt}{(1-\rho(x))^2} \\
&= \frac{x\rho(x) - \int_0^x \rho(t)dt}{(1-\rho(x))^2} \\
&= \frac{x\rho(x) - x\rho(x) + \lambda \int_0^x t^2 f(t)dt}{(1-\rho(x))^2} \\
&= \frac{\lambda \int_0^x t^2 f(t)dt + x^2 \bar{F}(x)}{(1-\rho(x))^2} \\
&= 2E[W(x)]_{PSJF}
\end{aligned}$$

The third step follows from the first since $1-\rho(x) \leq 1-\rho(t)$ for $t \leq x$. □

Proof: (of Theorem 4:) By Lemma 1 we know that $E[R(x)]_{PSJF} \leq E[R(x)]_{SRPT} + 2E[W(x)]_{PSJF}$. Now, adding $E[W(x)]_{PSJF}$ to both the sides we get that

$$E[W(x)]_{PSJF} + E[R(x)]_{PSJF} \leq E[R(x)]_{SRPT} + 3E[W(x)]_{PSJF}$$

Finally using the fact that $E[W(x)]_{PSJF} \leq E[W(x)]_{SRPT}$, $\forall x$, we get the following chain on inequalities

$$\begin{aligned}
E[T(x)]_{PSJF} &= E[W(x)]_{PSJF} + E[R(x)]_{PSJF} \\
&\leq 3E[W(x)]_{PSJF} + E[R(x)]_{SRPT} \\
&\leq 3E[W(x)]_{SRPT} + E[R(x)]_{SRPT} \\
&\leq 3E[W(x)]_{SRPT} + 3E[R(x)]_{SRPT} \\
&= 3E[T(x)]_{SRPT}
\end{aligned}$$

□

5 Conclusion

We consider a new model for comparing the performance of queueing policies. To summarize our approach, instead of asking whether one policy is strictly better than the other, we relax this condition to ask if one policy is no more than some constant times worse than the other policy. This allows us to compare policies a new way. In particular, it allows us to compare and bound the performance of a policy that might be inherently disadvantaged when compared with the optimal policy.

For example, we compared `FB` (which does not make use of job size while scheduling) to `SRPT`, and `PSJF` (which does not change the priority of a job as it executes) to `SRPT`. When comparing `FB` to `SRPT` we found that `FB` is only strictly competitive for a specific class distributions, which when the density function of the service distribution is non-increasing reduces to exactly those distributions with regularly varying tails. Thus, the performance of `FB` is likely to be good on many practical distributions, such as Pareto distributions. Surprisingly though, `FB` is not strictly competitive for any bounded distributions, including the bounded Pareto. `PSJF` however, is strictly competitive for all service distributions. Thus, by not allowing jobs to change priority while in the system, `PSJF` is only giving up a constant factor of performance.

References

[1] A. Borodin and R. El-Yaniv. *Online computation and competitive analysis.* Cambridge University Press, 1998.

[2] Leonard Kleinrock. *Queueing Systems*, volume II. Computer Applications. John Wiley & Sons, 1976.

[3] A. Muller and D. Stoyan. *Comparison Methods for Stochastic Models and Risks.* John Wiley & Sons, 2002.

[4] A.V. Pechinkin, A.D. Solovyev, and S.F. Yashkov. A system with servicing discipline whereby the order of remaining length is serviced first. *Tekhnicheskaya Kibernetika*, 17:51–59, 1979.

[5] Sidney I. Resnick. *Extreme Values, Regular Variation, and Point Processes.* Springer-Verlag, 1987.

[6] R. Schassberger. The steady-state appearance of the M/G/1 queue under the discipline of shortest remaining processing time. *Advances in Applied Probability*, 22:456–479, 1990.

[7] L.E. Schrage. A proof of the optimality of the shortest processing remaining time discipline. *Operations Research*, 16:678–690, 1968.

[8] L.E. Schrage and L.W. Miller. The queue M/G/1 with the shortest processing remaining time discipline. *Operations Research*, 14:670–684, 1966.

[9] D.R. Smith. A new proof of the optimality of the shortest remaining processing time discipline. *Operations Research*, 26:197–199, 1976.

[10] D. Stoyan. *Comparison Methods for Queues and Other Stochastic Models*. John Wiley & Sons, 1983.

[11] Ronald W. Wolff. *Stochastic Modeling and the Theory of Queues*. Prentice Hall, 1989.