Achievable sojourn times by non-size based policies in a GI/GI/1 queue

Nikhil Bansal *

July 23, 2004

Abstract

We investigate the best possible average sojourn time achievable under policies that do not make use of job sizes in their scheduling decisions (blind scheduling policies). Our main result is that for a single server GI/GI/1 queueing system, the average sojourn time under the best blind policy is at most $\log e/(1-\rho)$ time worse (upto constant factors) than the best average sojourn time possible under any arbitrary policy. Here ρ is the system load. Thus in a sense, the lack of knowledge of actual job sizes while making the scheduling decisions does not pose a serious problem if the blind policy is chosen carefully.

Our result makes use of previous results in the area of competitive analysis of online scheduling algorithms. Our main contribution is to show how these results can be used together with a game theoretic result known as Yao's Minimax Theorem to obtain almost tight results about queueing systems in fairly general setting such as a GI/GI/1 system.

1 Introduction

Sojourn time[1], defined as the time since a request arrives until it finishes, is one of the most useful measures of user satisfaction and performance in a scheduling system. Keeping the average sojourn time of jobs low is often an important criteria in the choice of a scheduling policy for a system.

For a single server system, optimizing for average sojourn time is well understood. The policy Shortest Remaining Processing Time (SRPT), that at any time works on the job with the smallest remaining service requirement, achieves the optimum possible average (or equivalently total) sojourn time for all possible problem instances [12],[15]. However, SRPT is rarely used in practice. One main drawback of SRPT, among many others, is that it requires exact knowledge of the service requirement for each job. Often, such information may be hard to obtain in practice. For example, in typical Operating Systems, the scheduler is only aware of the existence of a job and how much service it has received thus far. In particular, the scheduler only learns the size of this job when it completes its service requirement and leaves the system.

Scheduling policies that do *not* make use of job size in their scheduling decisions are widely studied. We shall call such policies *blind*. Blind policies are fairly common and usually have other attractive properties such as they are stateless and are easier to implement in a real system. Some common examples of blind policies are First In First Out (FIFO), where jobs are served in the order

*IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598. E-mail: `nikhil@us.ibm.com`.

[1]Also known as response time or flow time.

of their arrival, or Processor Sharing (PS), where the scheduler at any time shares the processor equally among all unfinished jobs present at that time.

A natural question that arises is that how much worse can the average sojourn time under a blind policy be as compared with SRPT? Let us first restrict our attention to the simpler M/M/1 queueing system. It is well known that in an M/M/1 system all blind policies are identical as far as average sojourn time is concerned. In particular, any blind policy[2] has average sojourn time equal to $1/\mu(1-\rho)$, where $1/\mu$ is the average job size and ρ is the system load [6]. As all blind policies are identical, the question reduces to determining the average sojourn time under SRPT. Schrage and Miller obtained the analytic expression for the average sojourn time under SRPT in a general M/G/1 system [13]. By evaluating this expression for the special case of the exponential job size distribution, Bansal [2] recently showed that the average sojourn time under SRPT in an M/M/1 system is $1/\log(e/(1-\rho)) \cdot 1/(\mu(1-\rho))$ upto lower order terms, that disappear as ρ approaches 1. Thus, the average sojourn time under SRPT in an M/M/1 system is about $\log(e/(1-\rho))$ times better than that under a blind policy. In particular, as $\log(e/(1-\rho))$ degrades exponentially slower than $1/(1-\rho)$, this suggests that SRPT does not significantly outperform a blind policy in an M/M/1 system.

In this paper, we study an analogous question for the more general GI/GI/1 queueing system. However, the situation is more involved in this case. First, unlike in the case of an M/M/1 system, all blind policies are no more identical. Second, no single blind policy is optimum. For example, in an M/G/1 system, FCFS is optimum for job size distributions with increasing failure rates, where as the blind policy Foreground Background (FB), that at any time works on the job with least attained service, is optimum for job size distributions with decreasing failure rates [7]. Third and an intriguing aspect is that the same blind policy can have very different behavior for different job size distributions. For example, while FB has average sojourn time equal to $1/\mu(1-\rho)$ when job sizes are exponentially distributed, the average sojourn time under under FB varies as $\log(e/(1-\rho))/\mu$ (upto constant factors) when the job sizes have the Pareto distribution with $1 < \alpha < 2$, [3]. In particular, note the substantially different dependence on ρ for the two distributions.

To keep the problem meaningful in a GI/GI/1 setting, we do the following: Clearly, any GI/GI/1 system is completely specified by the interarrival distribution G_a and the job-size distribution G_s. For a fixed G_a and G_s, let $Opt(G_a, G_s)$ denote the optimum possible average sojourn time achievable by a blind policy. We ask "How much worse can $Opt(G_a, G_s)$ be as compared with the average sojourn time under SRPT for this GI/GI/1 system?" Equivalently, for any GI/GI/1 system we compare SRPT only with the best possible blind policy for that system.

Our main result is that for any GI/GI/1 system (under some technical conditions that will be made precise later), the average sojourn time of the best blind policy for this system, $Opt(G_a, G_s)$, is only about a factor $\log(e/(1-\rho))$ times worse than that under SRPT (ignoring constant factors and lower order terms that disappear when ρ approaches 1). Thus, our result suggests that if we choose a blind policy for a system carefully, we can achieve an average sojourn time close to the best achievable by any arbitrary scheduling policy. Note that as the analytic expression for average sojourn time under an arbitrary GI/GI/1 system is not known and moreover no analytical expression for $Opt(G_a, G_s)$ for general G_a and G_s is known, we cannot adopt the approach (mentioned above) that [2] use for the case of M/M/1. To show our result, we apply a game-theoretic result known as Yao's Minimax Theorem [17] to existing results on competitive analysis of online

[2]Strictly speaking, to avoid considering meaningless policies, we also require the policy to be work-conserving, i.e. it always works on some job if there is work to do. Throughout this paper, we will only consider work-conserving policies.

scheduling algorithms and show how they can be interpreted in queueing theoretic context.

The paper is organized as follows: We define the relevant queueing notation and terminology in Section 1.1 and then state our results formally in Section 1.2. In Section 2 we define the relevant notions from competitive analysis of algorithms and then state Yao's result in Section 2. Then, in Section 2.2 we describe the results from online scheduling algorithms that we will require. Finally, in Section 3 we prove our main result and conclude with some discussions in Section 4.

1.1 Queueing Theoretic Preliminaries

A GI/GI/1 queueing system consists of an infinite sequence of jobs where the jobs sizes are independent and identically distributed (i.i.d.) random variables, S, chosen from a distribution G_s and the interarrival times are i.i.d. random variables, A, chosen from a distribution G_a. The load of the system, denoted by ρ, is defined as $E[S]/E[A]$, where $E[X]$ denotes the expected value of the random variable X. As usual, we also use λ to denote $1/E[A]$ and μ to denote $1/E[S]$. Finally, we will always assume the standard condition for stability that $\rho < 1$.

The sojourn time of a job is the time since a job arrives until the time it completes its service requirement. Given a GI/GI/1 queueing system and a scheduling policy $\mathcal{A}$, we denote the average sojourn time of a job by $E[T]_{\mathcal{A}}$. For a given GI/GI/1 system, consider optimum blind policy $\mathcal{A}$ which minimizes the average sojourn time (over the space of all blind policies) for that system. In general, $\mathcal{A}$ would depend on the choice of G_a and G_s. Also, $\mathcal{A}$ depends only on G_a and G_s, as they completely specify a GI/GI/1 system. For a given system, we will use $Opt(G_a, G_s)$ to denote the average sojourn time under $\mathcal{A}$. As SRPT is the optimum policy for minimizing average sojourn time, this paper will most deal with comparing $E[T]_{SRPT}$ and $Opt(G_a, G_s)$.

Finally, we say that a function $g(1/(1-\rho))$ is $O(f(1/(1-\rho)))$ (resp. $\Theta(O(f(1/(1-\rho)))$, if there is a universal constant c such that for any $\rho < 1$, $g(1/(1-\rho)) \leq cf(1/(1-\rho))$ (resp. $g(1/(1-\rho)) \geq cf(1/(1-\rho))$).

1.2 Results

Our main result is that for any GI/GI/1 system, the average sojourn time under SRPT is not significantly better than that under the optimum blind policy for that system. In particular we show that,

Theorem 1 *For a GI/GI/1 system with interarrival distribution G_a and service time distribution G_s, there is universal constant d (independent of ρ, G_a and G_s) such that*

$$Opt(G_a, G_s) \leq d \cdot \log\left(\frac{2C}{1-\rho}\right) \cdot E[T]_{SRPT} \tag{1}$$

where C is a bound on the sixth coefficient of variation of G_a and G_s, that is,

$$C = \left(\frac{E[A^6]}{E[A]^6} + \frac{E[S^6]}{E[S]^6}\right)^{1/6}$$

Observe that C only depends on the nature of the distribution of A and S, and not on the magnitude of $E[A]$ and $E[S]$. In particular, C is a constant independent of ρ. Thus, as ρ approaches 1, $\log(C/(1-\rho)) = \log 2C + \log(1/(1-\rho))$ dominated by $\log(1/(1-\rho))$. Thus, as a simple corollary of Theorem 1 we get that

Corollary 1 *For any GI/GI/1 system, there is a universal constant d such that*

$$\lim_{\rho \to 1} \frac{Opt(G_a, G_s)}{\left(\log \frac{1}{1-\rho}\right) \cdot E[T]_{SRPT}} \leq d$$

As mentioned previously, Bansal [2] proved that for an M/M/1 system

$$\lim_{\rho \to 1} \frac{Opt(G_a, G_s)}{\left(\log \frac{1}{1-\rho}\right) \cdot E[T]_{SRPT}} = 1$$

Thus, our result is tight upto constant factors and possibly the dependence on C. In particular, the dependence on ρ cannot be improved.

2 Competitive Analysis

Competitive analysis is an extensive area of research in theoretical computer science and excellent surveys and books have been written on the topic [5, 8]. We restrict our presentation here only to the context of scheduling algorithms, and the relevant ideas that will be useful later for our purposes. A detailed exposition of competitive analysis applied to scheduling algorithms can be found in [14, 11].

A scheduling problem instance $\mathcal{I}$ consists of a collection of jobs specified by their sizes and their arrival times. We say that an instance has size n, denoted by $|\mathcal{I}| = n$ if it consists of n jobs. For an algorithm $\mathcal{A}$ and an instance $\mathcal{I}$, let $\mathcal{A}(\mathcal{I})$ denote the total sojourn time when the instance $\mathcal{I}$ is executed according to the algorithm $\mathcal{A}$. We say that a (deterministic) algorithm $\mathcal{A}$ has competitive ratio $c(n)$ if

$$\max_{\mathcal{I}:|\mathcal{I}| \leq n} \frac{\mathcal{A}(\mathcal{I})}{\mathcal{SRPT}(\mathcal{I})} \leq c(n)$$

Thus, the competitive ratio of an algorithm (possibly a function of n), is the worst case ratio over all input instances of size almost n of the cost achieved by $\mathcal{A}$ and the optimum cost on that instance.

Observe that the definition of competitive ratio is rather strict. Even if an algorithm is close to optimum on all but one input instance, its competitive ratio is determined by the bad performance on that single input instance. Motivated by this scenario, a useful notion is that of a *randomized* algorithm. A randomized algorithm makes random choices during its execution, these random choices can depend upon the state of the algorithm or the input seen thus far. In general, a randomized algorithm can be viewed as a probability distribution on a collection of deterministic algorithms [5]. The competitive ratio of a randomized algorithm is defined as

$$\max_{\mathcal{I}:|\mathcal{I}| \leq n} \frac{E[\mathcal{A}(\mathcal{I})]}{Opt(\mathcal{I})}$$

The expectation in the definition above is over the probability distribution (as defined by the randomized algorithm) on the collection of deterministic algorithms. The crucial thing to observe is that there is no probabilistic assumption on the input instance. The input instance is still chosen adversarially to maximize the ratio of the *expected* cost of the algorithm to that of the optimum. As one might expect, randomized algorithms can have significantly better competitive ratios than deterministic algorithms (a more elaborate discussion can be found in any standard text on online algorithms such [5]).

Observe that the notion of performance of a randomized algorithm is dual to the notion of average case performance (for instance, a queueing system). While the former deals with performance of a distribution over algorithms on a fixed input instance, the latter deals with the performance of a fixed algorithm on a distribution over input instances. However, these two notions are closely related in a way that we now make precise.

2.1 Yao's Minimax Theorem

Yao gave an ingenious way of viewing the worst case performance of an algorithm as a zero-sum game between two players [17]. He then applied Von Neumann's classic Minimax theorem to this game to obtain the following result that relates the performance of a randomized algorithm to a statement above the average case performance of another algorithm. We mention this elegant result without proof (Theorem 2 below). For details, we refer the reader to the classic paper of Yao [17].

Theorem 2 **[Yao's principle: Minimization problems [5]:]** *Consider a collection of algorithms $A_1, \ldots, A_m$ for a problem $\mathcal{P}$ and a collection of input sequences $I_1, \ldots, I_p$. Let $\mathcal{R}$ be any randomized algorithm for $\mathcal{P}$ and let Opt denote the optimum algorithm for $\mathcal{P}$. Let $y(j)$ be any probability distribution over request sequences. Then for any choice of the distribution $y(j)$*

$$c(\mathcal{R}) \geq \max \left\{ \min_i \frac{E_{y(j)}[A_i(\sigma_j)]}{E_{y(j)}[Opt(\sigma_j)]}, \min_i E_{y(j)} \left[\frac{A_i(\sigma_j)}{Opt(\sigma_j)} \right] \right\} \tag{2}$$

Let us consider the contrapositive of Theorem 2.

Corollary 2 *Suppose a cost minimization problem $\mathcal{P}$ has a $c(\mathcal{R}, n)$ competitive randomized online algorithm $\mathcal{R}$ for request sequences of length at most n. Let distribution $y(j)$ be any distribution over request sequences of length at most n. Then,*

$$\min_i E_{y(j)}[A_i(\sigma_j)] \leq c(n) E_{y(j)}[Opt(\sigma_j)] \tag{3}$$

Proof: This follows directly by Theorem 2, since if $\min_i E_{y(j)}[A_i(\sigma_j)] > c(\mathcal{R}, n) E_{y(j)}[Opt(\sigma_j)]$, then equation 2 would imply that no $c(n, \mathcal{R})$ competitive randomized online algorithm can exist for $\mathcal{P}$. □

In particular, for any distribution over the input instances, if we consider the algorithm A_i that has the best average performance on this distribution, then this performance is no worse than $c(n, \mathcal{R})$ times the performance of the optimum algorithm.

Having defined the notion of competitive ratio, we now describe the known results for competitive ratio of blind scheduling algorithms.

2.2 Competitive analysis of Blind Scheduling Algorithms

The study of competitive analysis of blind scheduling policies was first initiated by [10][3]. They showed that for the problem of minimizing the average sojourn time, no blind deterministic algorithm can have a competitive ratio better than $\Omega(n^{1/3})$. They also showed that any randomized algorithm has a competitive ratio at least $\Omega(\log n)$. In a significant breakthrough, Kalayanasundaram

[3]They used them *non-clairvoyant* scheduling policy for what we call a blind policy.

and Pruhs [9] gave a very elegant and non-trivial randomized algorithm that they called Randomized Multi-Level Feedback (RMLF) and proved that it has competitive ratio of $O(\log n \log\log n)$. Later, [4] showed that RMLF is infact an $O(\log n)$ competitive randomized algorithm and hence (by the results of [10]) the best possible up to constant factors. In other words,

Theorem 3 ([4]) *There is universal constant r, such that for any scheduling instance with at most n jobs, the expected total sojourn time under RMLF is at most $r \log n$ times that under SRPT.*[4]

The rest of the paper deals with applying Theorem 3 in the context of queueing.

3 Analysis

Applying Corollary 2 to Theorem 3 it follows that,

Lemma 1 *Given any probability distribution on the input instances with at most n jobs, there is some deterministic algorithm the expected total sojourn time of which is at most $r(\log n)$ times that under SRPT.*

If we consider a GI/GI/1 system and view each busy period as a separate instance, we can think of a GI/GI/1 system as defining a probability distribution on input instances. However, we cannot immediately apply Lemma 1 to this distribution on input instances because we cannot place a bound on the number of jobs a busy period might contain. More precisely, for every n, there is a non-zero fraction of busy periods that contain more than n jobs. Intuitively however, one would expect that as a busy period has about $1/(1-\rho)$ jobs on the average, the $\log n$ factor in Lemma 1 should essentially be $\log 1/(1-\rho)$. Our goal in the rest of this section will be to make this intuition precise. In particular, we will show that most of the action happens in busy periods with at most about $C^6/(1-\rho)^6$ jobs and hence the n in Lemma 1 can essentially be replaced by $C^6/(1-\rho)^6$.

Consider a GI/GI/1 system. If we restrict our attention only to whether the system is idle or working on a job, then all work-conserving policies look identical. In particular, the busy periods are independent of the actual scheduling policy involved. Let $\mathcal{B}$ denote the set of all possible busy periods. Let $\mathcal{M}$ denote the measure that the GI/GI/1 system induces on $\mathcal{B}$. For an algorithm A and a busy period B, let $T_A(B)$ denote the total sojourn time incurred when A is executed during the busy period B, and let $n(B)$ denote the number of jobs in B. As a GI/GI/1 system is renewal process, we can express the average sojourn time of a job under an algorithm A as

$$E[T]_A = \frac{E_{\mathcal{M}}[T_A(B)]}{E_{\mathcal{M}}[n(B)]} \tag{4}$$

Observe that for any GI/GI/1 system, $E[n(B)]$ is identical for every work-conserving scheduling policy. Thus, by equation 4 it suffices to compare the expected total sojourn time of a busy period, $E_{\mathcal{M}}[T_A(B)]$ for two scheduling policies, in order to compare the average sojourn time under them.

Call a busy period *bad* if it contains more than $N_0 = 24C^6/(1-\rho)^6$ jobs. We define the process P derived from the GI/GI/1 process as follows: Whenever there is a bad busy period of length B, we replaced it with another busy period with $n(B)$ "dummy" jobs of length 0.

[4]The analysis of [4] is extremely involved and as is standard in such analyses, no attempt is made to optimize the value of r, or to calculate it explicitly. However, looking at the analysis of [4], a crude upper bound on r seems to be about 100.

Let $\mathcal{M}'$ denote the measure on busy periods induced by the new process P. By construction, $E_{\mathcal{M}'}[n(B)] = E_{\mathcal{M}}[n(B)]$. The crucial point about the process P is that all busy periods with more than N_0 jobs have length 0. Thus, the only contribution to the expected total sojourn time under the measure $\mathcal{M}'$ is due to the busy periods that contain at most N_0 jobs.

Let Opt' denote the blind algorithm A that minimizes $E_{\mathcal{M}'}[T_A(B)]$. Since, the only contribution to $E_{\mathcal{M}'}[T_A(B)]$ is due to busy periods with at most N_0 jobs it follows from Lemma 1 that:

Lemma 2

$$E_{\mathcal{M}'}[T_{Opt'}(B)] \leq r \log N_0 E_{\mathcal{M}'}[T_{SRPT}(B)] \tag{5}$$

Next, we will show that for any work-conserving algorithm A, the quantities $E_{\mathcal{M}'}[T_A(B)]$ and $E_{\mathcal{M}}[T_A(B)]$ do not differ from each other significantly. In particular we show that

Lemma 3 *For any work conserving algorithm A,*

$$E_{\mathcal{M}'}[T_A(B)] \leq E_{\mathcal{M}}[T_A(B)] \leq 2E_{\mathcal{M}'}[T_A(B)]$$

In order to prove Lemma 3, we will use the following tail bounds on sum of independent random variables and some results about the number of jobs in busy period and its length.

Lemma 4 *Let $S_n = X_1 + X_2 + \ldots + X_n$, where X_i are independent and identically distributed according to the random variable X. Then, for any $\epsilon > 0$,*

$$Pr\left[|S_n - nE[X]| \geq \epsilon n E[X]\right] \leq \frac{E[(X - E[X])^6]}{\epsilon^6 E[X]^6 n^3} \tag{6}$$

Proof: Let $Y_i = X_i - E[X_i]$ and $T_n = \sum_{i=1}^n Y_i$. Clearly, $E[Y_i] = 0$ for all $1 \leq i \leq n$, and $E[T_n] = 0$. We want to bound the probability of the event that $|T_n| \geq \epsilon n E[X]$. Clearly,

$$Pr[|T_n| \geq \epsilon n E[X]] \leq Pr[T_n^6 \geq (\epsilon n E[X])^6] \leq \frac{E[T_n^6]}{(\epsilon n E[X])^6}$$

where the last step follows from Markov's inequality applied to the random variable $|T_n|^6$.

By definition, $E[T_n^6] = E[(\sum_{i=1}^n Y_i)^6]$. As $E[Y_i] = 0$ and the $Y_i's$ are i.i.d., the only non-zero terms in the binomial expansion of $E[[(\sum_{i=1}^n Y_i)^6]$ are of the type $E[Y_i^6]$, $E[Y_i^4 Y_j^2]$, $E[Y_i^3 Y_j^3]$ and $E[Y_i^2 Y_j^2 Y_k^2]$ for all i, j, k distinct. Moreover by the convexity of the function x^p for $p \geq 1$, we have that $E[Y_i^j] \leq E[Y_i^6]^{j/6}$ for all $1 \leq j \leq 6$. Thus, it follows that

$$E[T_n^6] = E[(\sum_{i=1}^n Y_i)^6] \leq (n + 2n(n-1) + n(n-1)(n-2))E[Y^6] \leq n^3 E[Y^6]$$

Finally, $E[Y^6] = E[(X - E[X])^6] \leq E[X^6]$, we get that

$$Pr[|T_n| \geq \epsilon n E[X]] \leq \frac{E[X^6]}{\epsilon^6 n^3 E[X]^6}$$

□

Lemma 5 *Given a GI/GI/1 system, let A and S be the random variables representing arrival time and service time respectively. Let $\delta = (1-\rho)$ and $C = (E[A^6]/E[A]^6 + E[S^6]/E[S]^6)^{1/6}$. Then,*

$$Pr[n(B) \geq n] \leq \frac{C^6}{3\delta^6 n^3} \tag{7}$$

$$Pr[l(B) \geq t] \leq \frac{4C^6}{\delta^6 (\lambda t)^3} \tag{8}$$

Proof: Let Z denote the random variable $A - S$ and let $A(n), S(n)$ and $Z(n)$ denote the sum of n i.i.d. copies of A, S and Z respectively.

Let $\Gamma(z) = \sum_{n=1}^{\infty} Pr[(n(B) = n]z^n$, denote the generating function for the number of jobs in a busy period. It is a classic result (see for example, [16], page 422), that

$$\ln \frac{1}{1-\Gamma(z)} = \sum_{n=1}^{\infty} z^n P(Z(n) < 0)$$

An easy consequence of this fact is that,

$$Pr[n(B) = n] \leq \frac{Pr[Z(n) \leq 0]}{n} \tag{9}$$

By Lemma 4, we can upper bound $Pr[Z(n) \leq 0]$ as,

$$Pr[Z(n) \leq 0] \leq Pr[|Z(n) - E[Z(n)]| \geq E[Z(n)]] \leq \frac{E[Z^6]}{n^3 E[Z]^6} \tag{10}$$

Now, $E[Z] = E[A] - E[S] = (1-\rho)E[A]$, and $E[Z^6] = E[(A-S)^6] \leq E[\max(A,S)^6] \leq (E[A^6] + E[S^6])$. Thus, by equations 9 and 10,

$$Pr[n(B) = n] \leq \frac{1}{n} \frac{(E[A^6] + E[S^6])}{n^3 (1-\rho)^6 E[A]^6} \leq \frac{1}{n^4 \delta^6} \left(\frac{E[A^6]}{E[A]^6} + \frac{E[S^6]}{E[S]^6} \right) \leq \frac{C^6}{\delta^6 n^4}$$

Thus, (7) follows as

$$Pr[n(B) \geq n] = \sum_{x \geq n} Pr[n(B) = x] \leq \frac{C^6}{\delta^6 x^4} \leq \frac{C^6}{3\delta^6 n^3}$$

To prove (8), we use the following fact about busy periods that relates the number of jobs in a busy period to its length [1].

$$Pr[l(B) \geq A(n)] = Pr[n(B) \geq n+1] \tag{11}$$

Conditioning on the event $A(n) \geq t$, we have that,

$$\begin{aligned} Pr[l(B) \geq t] &= Pr\left[l(B) \geq t | A(n) \geq t\right] \cdot Pr[A(n) \geq t] \\ &\quad + Pr\left[l(B) \geq t | A(n) < t\right] \cdot Pr[A(n) < t] \\ &\leq Pr[A(n) \geq t] + Pr\left[l(B) \geq A(n) | A(n) < t\right] \\ &\leq Pr[A(n) \geq t] + Pr[l(B) \geq A(n)] \end{aligned} \tag{12}$$

Choosing $n = \lambda t/3$, we have that $E[A(n)] = t/3$. Thus, by Lemma 4 we get that,

$$Pr[A(n) \geq t] = Pr[A(n) \geq 3E[A(n)] \leq \frac{E[A^6]}{2^6 n^3 E[A]^6} \leq \frac{C^6}{2^6 n^3} \leq \frac{C^6}{\lambda^3 t^3} \quad (13)$$

Now, by equation 11 and 7,

$$Pr[l(B) \geq A(n)] = Pr[n(B) \geq n+1] \leq Pr[n(B) \geq n] \leq \frac{C^6}{3\delta^6 n^3} = \frac{8C^6}{3\delta^6 \lambda^3 t^3} \quad (14)$$

Equations 13 and 14 together with equation 12 finally imply that

$$Pr[l(B) \geq t] \leq \frac{4C^6}{\delta^6 \lambda^3 t^3}$$

□

We now prove Lemma 3.

Proof of Lemma 3: As M and M' only differ on bad busy periods and the bad busy periods contribute 0 to $E_{\mathcal{M}'}[T_A(B)]$, it easily follows that for any work-conserving algorithm A, $E_{\mathcal{M}'}[T_A(B)] \leq E_{\mathcal{M}}[T_A(B)]$.

To show the other side of the inequality, we need to upper bound the contribution due to bad busy periods in $E_{\mathcal{M}}[T_A(B)]$. Clearly, in a busy period of length l and consisting of n jobs, the total sojourn time of the jobs involved can be at most nl, irrespective of the choice of the algorithm.

Thus, it suffices to bound,

$$\begin{aligned}
&= \sum_{n \geq N_0} n \int_{t=0}^{\infty} t Pr[l(B) = t \text{ and } n(B) = n] dt \\
&= \sum_{n \geq N_0} n \int_{t=0}^{\infty} Pr[l(B) \geq t \text{ and } n(B) = n] dt \\
&= \int_{t=0}^{\infty} N_0 Pr[l(B) \geq t \text{ and } n(B) \geq N_0] dt + \sum_{n > N_0} \int_{t=0}^{\infty} Pr[l(B) \geq t \text{ and } n(B) \geq n] dt \quad (15)
\end{aligned}$$

We bound the two terms separately. The first term can be written as

$$\begin{aligned}
&\int_{t=0}^{N_0/\lambda} N_0 Pr[l(B) \geq t \text{ and } n(B) \geq N_0] dt + \int_{t > N_0/\lambda} N_0 Pr[l(B) \geq t \text{ and } n(B) \geq N_0] dt \\
\leq\ & \frac{N_0^2}{\lambda} \cdot Pr[n(B) \geq N_0] + \int_{t > N_0/\lambda} N_0 Pr[l(B) \geq t] dt \\
\leq\ & \frac{N_0^2}{\lambda} \cdot \frac{C^6}{\delta^6 N_0^3} + N_0 \cdot \int_{t > N_0/\lambda} \frac{2C^6}{\delta^6 \lambda^3 t^3} dt \\
\leq\ & \frac{N_0^2}{\lambda} \cdot \frac{C^6}{\delta^6 N_0^3} + N_0 \cdot \frac{2C^6}{\delta^6 \lambda N_0^2} = \frac{3C^6}{\delta^6 \lambda N_0}
\end{aligned}$$

We now focus on bounding the second term in equation 15.

$$\sum_{n > N_0} \int_{t=0}^{\infty} Pr[l(B) \geq t \text{ and } n(B) \geq n] dt$$

$$
\begin{aligned}
&= \sum_{n>N_0} \int_{t=0}^{n/\lambda} Pr[l(B) \geq t \text{ and } n(B) \geq n] dt + \sum_{n>N_0} \int_{t=n/\lambda}^{\infty} Pr[l(B) \geq t \text{ and } n(B) \geq n] dt \\
&\leq \sum_{n>N_0} \frac{n}{\lambda} Pr[n(B) \geq n] + \sum_{n>N_0} \int_{t=n/\lambda}^{\infty} Pr[l(B) \geq t] dt \\
&\leq \frac{1}{\lambda} \frac{C^6}{\delta^6 N_0} + \sum_{n>N_0} \frac{4C^6}{2\delta^6 \lambda n^2} \\
&\leq 3 \frac{C^6}{\delta^6 \lambda N_0}
\end{aligned}
$$

Thus the total contribution of busy periods with N_0 or more jobs under any work-conserving algorithm is at most $6C^6/\delta^6\lambda N_0$. Without loss of generality, we assume that $\rho > 1/2$. Note that if $\rho \leq 1/2$, then the blind policy PS has average sojourn time at most $2/\mu$. Since any arbitrary has average sojourn time at least equal to the average job size, $1/\mu$, it follows that Theorem 1 is trivially true for $\rho \leq 1/2$. Thus, if $\rho > 1/2$ we have that $6C^6/\delta^6\lambda N_0$ is at most $12C^6/\delta^6\mu N_0$. Plugging the value of $N_0 = 24C/(\delta^6)$, it follows that

$$
E_{\mathcal{M}}[T_A(B)] - E_{\mathcal{M}'}[T_A(B)] \leq \frac{1}{2\mu} \tag{16}
$$

Since the average job size is $1/\mu$, and each busy period has at least one job, it follows that for any algorithm $\mathcal{A}$. $E_{\mathcal{M}}[T_A(B)] \geq 1/\mu$. By equation 16 it follows that,

$$
E_{\mathcal{M}'}[T_A(B)] \geq E_{\mathcal{M}}[T_A(B)] - \frac{1}{2\mu} \geq \frac{1}{2} E_{\mathcal{M}}[T_A(B)]
$$

Thus the desired result follows. □

We now finish the proof of Theorem 1.

Proof of Theorem 1: Consider the blind policy Opt' that minimizes $E_{\mathcal{M}'}[T_{Opt'}(B)]$. Since Opt' is work-conserving, by Lemma 3 it follows that

$$
E_{\mathcal{M}}[T_{Opt'}(B)] \leq 2E_{\mathcal{M}'}[T_{Opt'}(B)] \tag{17}
$$

Clearly, by definition of $Opt(G_a, G_s)$ is follows that,

$$
Opt(G_a, G_s) \leq E_{\mathcal{M}}[T_{Opt'}(B)] \tag{18}
$$

By Lemma 2, we also have that

$$
E_{\mathcal{M}'}[T_{Opt'}(B)] \leq r \log N_0 E_{\mathcal{M}'}[T_{SRPT}(B)]
$$

which by Lemma 3 implies that

$$
E_{\mathcal{M}'}[T_{Opt'}(B)] \leq r \log N_0 E_{\mathcal{M}}[T_{SRPT}(B)] \tag{19}
$$

Combining 17,18 and 19 we get that

$$
Opt(G_a, G_s) \leq 2r \log N_0 E_{\mathcal{M}}[T_{SRPT}(B)]
$$

and hence that

$$Opt(G_a, G_s) \leq 2r \log \frac{24C^6}{\rho(1-\rho)^6} E_{\mathcal{M}}[T_{SRPT}(B)]$$

or equivalently,

$$Opt(G_a, G_s) \leq 12r \log \frac{2C}{(1-\rho)} E_{\mathcal{M}}[T_{SRPT}(B)]$$

This proves Theorem 1. □

4 Conclusions

We proved that for any GI/GI/1 queueing system, the performance of the best blind policy for that system achieves average sojourn close to the best possible by any other (non-blind) policy. Unfortunately, our result does not give a construct way to produce the optimum (deterministic) blind scheduling policy for an arbitrary GI/GI/1 system[5]. In many cases however, the optimum blind policy may be known. For example, in many practical systems, the job sizes follow heavy tailed distribution or more generally have a decreasing failure rate a.k.a. hazard rate. These systems are often modeled as M/G/1 systems with G_s having a decreasing hazard rate, for which FB is known to be the optimum blind policy [7].

The dependence on the sixth coefficient of variations of inter-arrival times and sizes in our results seems to be an artifact of our analysis. The only reason why We need the sixth moments to be finite, is that this allows us to show that the probability a busy period has more than n jobs decays at least as $1/n^3$. This $1/n^3$ decay is required for the way we bound the sojourn time contribution of bad busy periods. One way of improving the result would be to bound the sojourn time contribution of bad busy periods more carefully. On the other hand, if we are only interested in the heavy-traffic limit as $\rho \to 1$, then we can improve the result so that we only require that the second moments of A and S be finite. This follows easily, as in the heavy-traffic regime we only need to consider the probability that a busy period has more than n jobs in the limit as n approaches infinity. In this case, the Central Limit Theorem directly implies that this probability decays exponentially with n. We believe that the finiteness of the second moments is actually the right term even in the non-heavy traffic case. It would be interesting to show that this is indeed true.

References

[1] A. Baltrunas, D. J. Daley, and C. Kluppelberg. Tail behaviour of the busy period of GI/G/1 queue with subexponential service times. *Stochastic Processes and Applications*, 111:237–258, 2004.

[2] N. Bansal. On the average sojourn time under M/M/1/SRPT. Operations Research Letters (to appear).

[3] N. Bansal and D. Gamarnik. Handling load with less stress. Research Report RC23277, IBM, submitted for publication.

[4] L. Becchetti and S. Leonardi. Non-clairvoyant scheduling to minimize the average flow time on single and parallel machines. In *ACM Symposium on Theory of Computing (STOC)*, pages 94–103, 2001.

[5] Note however that if one allows randomized blind algorithms then clearly $RMLF$ is at most a factor $O(\log(C/(1-\rho)))$ times worse than SRPT.

[5] A. Borodin and R. El-Yaniv. *On-Line Computation and Competitive Analysis.* Cambridge University Press, 1998.

[6] R. W. Conway, W. L. Maxwell, and L. W. Miller. *Theory of Scheduling.* Addison-Wesley Publishing Company, 1967.

[7] H. Feng and V. Misra. Mixed scheduling disciplines for network flows (the optimality of FBPS). In *Proceedings of The Fifth Workshop on MAthematical performance Modeling and Analysis (MAMA)*, 2003.

[8] A. Fiat and G. J. Woeginger. *Online Algorithms: The State of the Art, eds. A. Fiat and G. J. Woeginger.* Springer-Verlag, 1998.

[9] B. Kalyanasundaram and K. Pruhs. Minimizing flow time nonclairvoyantly. In *IEEE Symposium on Foundations of Computer Science*, pages 345–352, 1997.

[10] R. Motwani, S. Phillips, and E. Torng. Nonclairvoyant scheduling. *Theoretical Computer Science*, 130(1):17–47, 1994.

[11] K. Pruhs, J. Sgall, and E. Torng. Online scheduling. In Handbook on Scheduling: Algorithms, Models and Performance Analysis (to appear), CRC press.

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

[13] 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.

[14] J. Sgall. Online scheduling. In *Online Algorithms: The State of the Art, eds. A. Fiat and G. J. Woeginger*, pages 196–231. Springer-Verlag, 1998.

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

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

[17] A. C-C. Yao. Probabilistic computations: Toward a unified measure of complexity (extended abstract). In *IEEE Symposium on Foundations of Computer Science (FOCS)*, 1977.