Uniform Bounds for Scheduling with Job Size Estimates

ZIV SCULLY, Carnegie Mellon University, USA
ISAAC GROSOF, Carnegie Mellon University, USA
MICHAEL MITZENMACHER, Harvard University, USA

We consider the problem of scheduling to minimize mean response time in M/G/1 queues where only *estimated* job sizes (processing times) are known to the scheduler, where a job of true size s has estimated size in the interval $[\beta s, \alpha s]$ for some $\alpha \geq \beta > 0$. We evaluate each scheduling policy by its *approximation ratio*, which we define to be the ratio between its mean response time and that of Shortest Remaining Processing Time (SRPT), the optimal policy when true sizes are known. Our question: is there a scheduling policy that (a) has approximation ratio near 1 when α and β are near 1, (b) has approximation ratio bounded by some function of α and β even when they are far from 1, and (c) can be implemented without knowledge of α and β?

We first show that naively running SRPT using estimated sizes in place of true sizes is *not* such a policy: its approximation ratio can be arbitrarily large for any fixed $\beta < 1$. We then provide a simple variant of SRPT for estimated sizes that satisfies criteria (a), (b), and (c). In particular, we prove its approximation ratio approaches 1 uniformly as α and β approach 1. This is the first result showing this type of convergence for M/G/1 scheduling.

We also study the Preemptive Shortest Job First (PSJF) policy, a cousin of SRPT. We show that, unlike SRPT, naively running PSJF using estimated sizes in place of true sizes satisfies criteria (b) and (c), as well as a weaker version of (a).

1 INTRODUCTION

Minimizing mean response time of jobs in a preemptive single-server queue is a fundamental scheduling problem. If the scheduler knows each job's size,[1] then the optimal policy is *Shortest Remaining Processing Time* (SRPT), which always serves the job of least remaining size: least size minus time served so far. However, in practical queueing systems, it is rare that the scheduler knows the exact job size, which is required for SRPT. Instead, it may be that the scheduler has only an *estimated size* for each job.

In settings where the scheduler knows only job size estimates, rather than true sizes, how should one schedule to minimize mean response time? We study this question in a stochastic online setting, namely an M/G/1 queue,[2] in which jobs arrive randomly over time. We use T to denote the distribution of response time, and we seek policies which achieve strong guarantees on $\mathbf{E}[T]$, the mean response time. We seek simple policies that do not depend on detailed knowledge of the distributions of true or estimated job sizes. While some simple heuristics have been proposed in prior work, no performance guarantees have been proven for these policies. We evaluate each scheduling policy by its *approximation ratio*, which we define to be the ratio between its mean response time and that of Shortest Remaining Processing Time (SRPT), the optimal policy when true sizes are known.

In the context of online algorithms with predictions [7, 11], there are two key goals for a policy: "consistency", which requires near-optimal performance under low error, and "robustness", which requires bounded approximation ratio under arbitrary error. Unfortunately, as we explain in Appendix B, robustness is provably unachievable in this context. Instead, we focus on a more

[1] A job's size is its processing time.
[2] The M/G/1 is a queueing model with Poisson arrivals and i.i.d. job sizes. We define the M/G/1 formally in Section 2.

Authors' addresses: Ziv Scully, Carnegie Mellon University, Computer Science Department, 5000 Forbes Ave, Pittsburgh, PA, 15213, USA, zscully@cs.cmu.edu; Isaac Grosof, Carnegie Mellon University, Computer Science Department, 5000 Forbes Ave, Pittsburgh, PA, 15213, USA, igrosof@cs.cmu.edu; Michael Mitzenmacher, Harvard University, School of Engineering and Applied Sciences, 33 Oxford Street, Cambridge, MA, 02138, USA, michaelm@eecs.harvard.edu.

appropriate guarantee, which we call "graceful degredation", which requires that the performance degrades smoothly and slowly as error increases.

We focus on the setting of multiplicative errors: We assume that a job of true size s has estimated size in the interval $[\beta s, \alpha s]$ for some $\alpha \geq \beta > 0$. We refer to this assumption by saying the jobs have (β, α)-bounded estimates. (Here β stands for "below" and α is "above.") In this context, a policy π is consistent if in the limit as $\alpha, \beta \to 1$, $\mathbf{E}[T_\pi] \to \mathbf{E}[T_{\text{SRPT}}]$. Graceful degradation requires that for any α, β, $\mathbf{E}[T_\pi] \leq C\frac{\alpha}{\beta}\mathbf{E}[T_{\text{SRPT}}]$, for some constant C.

In trying to achieve consistency and graceful degradation, a first policy one might consider is the *SRPT with Estimates* (SRPT-E) policy, which always serves the job of least estimated size minus time served so far. Unfortunately, in Theorem 6.1, we prove that SRPT-E can have infinite approximation ratio for any $\beta < 1$, so it has neither consistency nor graceful degradation.

In this work, we present the first policy which provably has both consistency and graceful degradation: the *SRPT with Bounce* (SRPT-B) policy, defined in Section 2. We also study the *Preemptive Shortest Job First with Estimates* (PSJF-E), for which we prove even better graceful degradation guarantees, and consistency guarantees relative to the perfect-information PSJF policy. That is, we also show $\mathbf{E}[T_{\text{PSJF-E}}] \to \mathbf{E}[T_{\text{PSJF}}]$ as $\alpha, \beta \to 1$. Specifically, SRPT-B's approximation ratio is at most $3.5\alpha/\beta$ and approaches 1 uniformly as α and β approach 1, PSFJ-E's approximation ratio is at most $1.5\alpha/\beta$.

1.1 Related Work

Policies for ordering jobs according to their service time have been studied extensively in single queues. For example, the text [5] provides an excellent introduction to the analysis of standard approaches such as shortest job first (SJF), PSJF, and SRPT in the single queue setting.

Settings where estimates or predictions of service times, such as one might obtain from machine learning algorithms, have been much less studied. The work closest to ours is that of Wierman and Nuyens [21]. They study policies that they dub ε-SMART policies. Such policies include variations of SRPT and PSJF with inexact job sizes, and they bound the performance of such policies based on how inexact the estimates can be [21]. However, their results only apply to two simpler types of error: addditive error, where the estimate of a job of size s is within $[s - \sigma, s + \sigma]$; and speedup error, where the estimate is updated as the job runs, and the estimate of a job of remaining size r is within $[(1 - \sigma)r, (1 + \sigma)r]$. In contrast, we primarily focus on the more realistic setting of multiplicative error. Their results on mean response time demonstrate only graceful degradation, are restricted to the setting of speedup error, and require additional assumptions on the job size distribution. While it is not our primary focus, we also discuss using the techniques in this paper to achieve consistency and graceful degradation results in the speedup-error setting in Section 7 and Appendix D. Such results would not require additional assumptions on the job size distribution.

Dell'Amico, Carra, and Michiardi empirically study scheduling policies for queueing systems with estimated sizes [3]; Dell'Amico and Mitzenmacher also perform an empirical study of scheduling policies with estimated sizes, but in the context of multiple queues using the power of two choices [10]. Mitzenmacher provides formulas for the mean response time for M/G/1 queues under scheduling policies where service times are predicted rather than known exactly according to a stochastic model, including for our SRPT-E and PSJF-E policies [9].[3] In later work Mitzenmacher studies similar models where only a single bit of prediction-based advice is given, and also studies single-bit advice in the mean-field setting under the power of two choices [8].

[3]In [9] the schemes using predictions are referred to as SPRPT (shortest predicted remaining processing time), and PSPFJ; there does not seem to be a consistent nomenclature for policies with predictions/estimates, and we hope our labeling is more readily understood.

In the setting of scheduling with predictions for finite collections of jobs, combinations of shortest predicted job first and round robin that yield good performance in terms of the competitive ratio were studied by [12]. For the online scheduling problem of weighted mean response time on a single machine with a finite arrival sequence, [1] consider what we would refer to as $(1, \mu)$-bounded jobs where μ is known, and give algorithms that are competitive up to a logarithmic factor in the maximum ratios of the processing times, densities, and weights. For unweighted mean response time, they prove a variant of graceful degradation: if job size estimates have a multiplicative error of at most μ, they prove a $O(\mu^2)$ competitive ratio bound. In contrast, our graceful degradation bounds are linear in $\mu = \frac{\alpha}{\beta}$, and do not depend on knowledge of α or β.

2 MODEL AND PRELIMINARIES

We now define our model and provide basic notation and definitions. Further definitions and background results appear in Section 5.

We consider a stochastic scheduling setting called the M/G/1 queue with job size estimates. The "M" in "M/G/1" refers to the assumption that jobs arrive according to a Poisson process (that is, with exponentially distributed interarrival times) with rate λ. The "G" in "M/G/1" refers to the assumption that job sizes are sampled i.i.d. from a general distribution. We additionally assume that each job j has a size s_j and an estimated size z_j, where the pair (s_j, z_j) is sampled i.i.d. from some joint distribution (S, Z). We assume (S, Z) is a continuous distribution with joint density function $f_{S,Z}(s, z)$. We write $f_S(s)$ and $f_Z(z)$ for the marginal densities of S and Z, respectively. Regardless of scheduling policy, the fraction of time the server is busy, also known as the load, is fixed. We denote load by ρ, and note that $\rho = \lambda \mathbf{E}[S]$. We assume that $\rho < 1$, to ensure that the server completes jobs faster than they arrive in the long run. The "1" in "M/G/1" refers to there being a single server.

We focus on the setting of multiplicatively-bounded size estimates:

Definition 2.1. Size estimates are (β, α)-bounded for some constants $0 < \beta \leq \alpha$ if for all jobs j,

$$z_j \in [\beta s_j, \alpha s_j].$$

Mnemonically, β is the bound below, and α is the bound above.

Definition 2.2. The *state* of job j is the triple $x_j = (s_j, z_j, a_j)$ consisting of

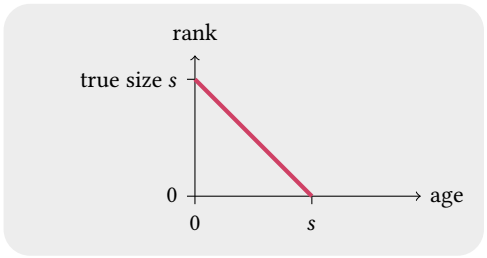
- its *(true) size* s_j, which is the amount of time it must be served to complete;
- its *estimated size* z_j, which is revealed when the job arrives and is guaranteed to be in the interval $[\beta s_j, \alpha s_j]$; and
- its *age* a_j, the amount of service the it has received so far.

Job j completes once $a_j = s_j$.

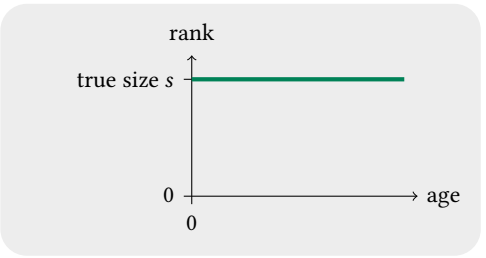
We now formally define the scheduling policies we consider.

Definition 2.3. We consider six scheduling policies in this work. We define each policy π by a *rank function*, denoted $\mathsf{rank}_\pi(x)$ or $\mathsf{rank}_\pi(s, z, a)$ assigning a *rank*, or priority, to a job based on its state $x = (s, z, a)$. The scheduler always serves whichever job has the least rank.[4] The policies,

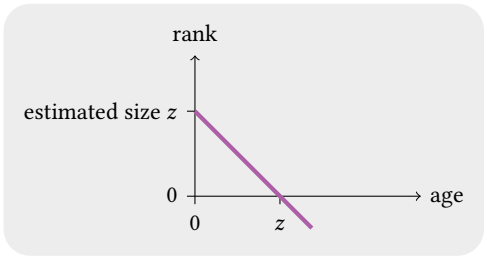
[4]We tiebreak arbitrarily. Given our continuous job-size assumption and our specific policies, ties happen with probability zero. See Appendix A for details.



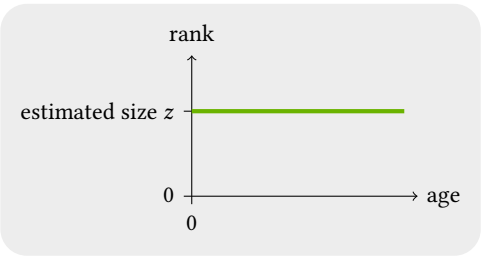
(a) Shortest Remaining Processing Time (SRPT)



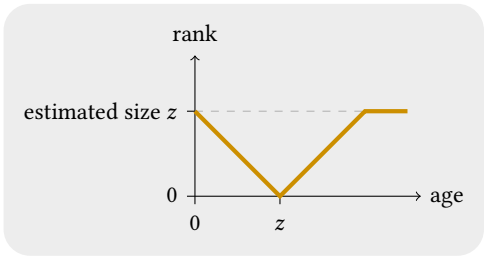
(b) Preemptive Shortest Job First (PSJF)



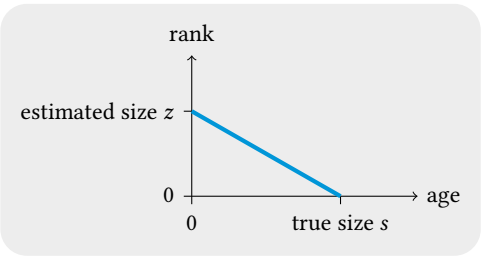
(c) SRPT with Estimates (SRPT-E)



(d) PSJF with Estimates (PSJF-E)



(e) SRPT with Bounce (SRPT-B)



(f) SRPT with Scaling Estimates (SRPT-SE)

Fig. 2.1. Rank Functions of Size-Estimate-Based Policies

which we illustrate in Fig. 2.1, are the following:

Shortest Remaining Processing Time (SRPT)	$\text{rank}_{\text{SRPT}}(s, z, a) = s - a,$
Preemptive Shortest Job First (PSJF)	$\text{rank}_{\text{PSJF}}(s, z, a) = s,$
SRPT with Estimates (SRPT-E)	$\text{rank}_{\text{SRPT-E}}(s, z, a) = z - a,$
PSJF with Estimates (PSJF-E)	$\text{rank}_{\text{PSJF-E}}(s, z, a) = z,$
SRPT with Bounce (SRPT-B)	$\text{rank}_{\text{SRPT-B}}(s, z, a) = \min\{\lvert z - a\rvert, z\},$
SRPT with Scaling Estimates (SRPT-SE)	$\text{rank}_{\text{SRPT-SE}}(s, z, a) = z/s \cdot (s - a).$

For SRPT, shown in Fig. 2.1(a) the rank of a job is the *remaining size* $s - a$, while for SRPT-E, shown in Fig. 2.1(c), the rank is the *estimated remaining size* $z - a$, using the estimate in place of the true size. (Note the rank for SRPT-E can be negative.) Similarly, PSJF-E's rank function z uses the estimate where PSJF uses the true size s for its rank, as shown in Figs. 2.1(b) and 2.1(d).

For SRPT-B, shown in Fig. 2.1(e), the rank is given by $\min\{|z - a|, z\}$. SRPT-B's rank is identical to SRPT-E's rank for ages $a \in [0, z]$, but rises back to z for larger a, yielding the bounce and thus the name SRPT with Bounce.

As a theoretical tool, we will also consider SRPT with Scaling Estimates, or SRPT-SE, shown in Fig. 2.1(f), for which the rank function is $z/s \cdot (s - a)$, a horizontally stretched version of SRPT-E. Note that SRPT-SE is not implementable in our model, as the scheduler does not have access to the true size s.

3 DESCRIPTION OF MAIN RESULTS

As discussed in the introduction, our goal is to derive the first provably consistent and gracefully-degrading policies in the size-estimate setting. In the setting of (β, α)-bounded size estimates, consistency requires that in the $\beta, \alpha \to 1$ limit, the policy achieves optimal mean response time, matching that of SRPT, the optimal known-size policy. "Graceful degradation" requires that a policy's mean response is bounded relative to that of SRPT and the α and β values:

$$\mathbf{E}[T_\pi] \leq C\frac{\alpha}{\beta}\mathbf{E}[T_{\text{SRPT}}],$$

for some constant C. Robustness, in the sense of achieving constant approximation ratio for arbitrary errors, is not possible in this setting, as we discuss in Appendix B.

First, we show that consistency and graceful degradation are not straightforward to achieve. Our first result shows that simply using SRPT-E (SRPT with Estimates), a natural method studied previously [9], yields mean response times that cannot be bounded within a constant factor of SRPT in the worst case, even with (β, α)-bounded size estimates.

Theorem 6.1 (Performance of SRPT-E). *Consider the M/G/1 with (β, α)-bounded size estimates.*

(a) For any size distribution S, there exists a joint distribution (S, Z) of true and estimated sizes such that the mean response time of SRPT-E is bounded below by

$$\mathbf{E}[T_{\text{SRPT-E}}] \geq \frac{\lambda(1-\beta)^2}{2}\mathbf{E}[S^2],$$

(b) The approximation ratio of SRPT-E may be arbitrarily large or infinite whenever $\beta < 1$.

We consider a novel variation of SRPT, SRPT with Bounce (SRPT-B), and prove it is consistent and gracefully-degrading, without knowledge of α and β. This is the first proof of a policy satisfying these criteria. Here, as α and β approach 1, SRPT-B approaches the performance of SRPT, and it achieves a suitable finite approximation ratio for all α and β. Formally, we prove the following:

Theorem 8.1 (Performance of SRPT-B). *Consider the M/G/1 with (β, α)-bounded size estimates.*

(a) The mean response time of SRPT-B is bounded above by

$$\mathbf{E}[T_{\text{SRPT-B}}] \leq \frac{\alpha}{\beta}\mathbf{E}[T_{\text{SRPT}}] + \left(\frac{3}{2}\alpha\mathbb{1}(\beta < 1) + 1\right)\min\left\{1, \max\left\{1 - \frac{1}{\alpha}, \frac{1}{\beta} - 1\right\}\right\}\left(\frac{1}{\rho}\ln\frac{1}{1-\rho} - 1\right)\mathbf{E}[S].$$

(b) The approximation ratio of SRPT-B is at most $3.5\alpha/\beta$.

(c) As α and β converge to 1, the approximation ratio of SRPT-B converges to 1. This convergence is uniform in the arrival rate and the joint distribution of true and estimated sizes.

The bound in Theorem 8.1(a) is a compromise between simplicity and tightness. It turns out that capping the rank function of SRPT-B at z is critical (see Remark 4.3). Otherwise, as explained in Appendix C, the approximation ratio could be arbitrarily poor when $\beta < 1/2$.

Finally, we consider PSJF-E, which we bound relative to PSJF. We prove PSJF-E is consistent relative to PSJF, achieving a mean response time ratio of α/β. While this is a weaker consistency result than that of SRPT-B, PSJF often performs within a few percent of SRPT in practice, making the result nearly as strong. In the worst case, PSJF's mean response time is within a factor of 1.5 of

SRPT's [20], so PSJF-E is within a factor of $1.5\alpha/\beta$ of optimal. This is a stronger graceful-degradation bound than we obtained for SRPT-B.

Theorem 9.1 (Performance of PSJF-E). *Consider the M/G/1 with (β, α)-bounded size estimates.*

(a) The mean response time of PSJF-E is bounded above by

$$\mathbf{E}[T_{\text{PSJF-E}}] \le \frac{\alpha}{\beta}\mathbf{E}[T_{\text{PSJF}}].$$

(b) The approximation ratio of PSJF-E is at most $1.5\alpha/\beta$.

One can view our PSJF-E result as bounding what Mitzenmacher [9] dubs the "price of misprediction" of PSJF-E. An algorithm's price of misprediction is its performance ratio relative not to the optimal algorithm, in this case SRPT, but relative to a version of the algorithm that has perfect information, in this case PSJF.

3.1 Discussion of Our Results

In the wider context of online algorithms with predictions, the three goals discussed in this paper can be stated more generally:

Consistency: In the limit as the prediction quality becomes perfect, the performance should approach that of the optimal algorithm with perfect information. For instance, one might try to achieve A-consistency [7], which requires that the competitive ratio is bounded by A as the error in the predictions goes to 0.

Robustness: In the limit as the prediction quality becomes arbitrarily poor, the performance should be comparable to that of the optimal algorithm in with perfect information. For instance, one might try to achieve B-robustness [7], which requires that the competitive ratio is bounded by B under arbitrarily poor predictions.

Graceful Degradation: As the prediction quality worsens from perfect to worthless, the performance should degrade smoothly and slowly. For instance, one might try to achieve C-graceful degradation, which requires that the competitive ratio is bounded by C times some measure of the estimate quality.

Looked at in this framework, we prove that SRPT-B is 1-consistent and has 3.5-graceful degradation, where α/β is our measure of estimate quality for (β, α)-bounded size estimates. We also prove that PSJF-E is 1.5 consistent and has 1.5-graceful degradation, where the factor of 1.5 comes from the maximum gap between PSJF and SRPT.

While we do not have robustness results for these algorithms, that is because in the context of scheduling in the M/G/1, the robustness property is provably unachievable. In particular, no online policy without prediction information can achieve a constant approximation ratio against SRPT, as discussed in Appendix B.

We feel our emphasis on graceful degradation is a key contribution of our work that may apply to many other algorithms-with-predictions problems. While consistency and robustness are well-known goals in the literature, the graceful degradation goal has received less focus. However, we argue that graceful degradation is *extremely important*. Real applications often have high-quality but imperfect predictions, which is the regime where performance is bounded by a graceful degradation result. The extreme cases of perfect or worthless estimates may come up less in practice.

To adapt the notion of robustness to the setting of M/G/1 scheduling, one possible method would be to compare an algorithm against the optimal blind policy, which knows the job size distribution, but not the job sizes. This policy is known, and is called the Gittins index policy [4]. We discuss this policy in Appendix B. This method of comparing against the optimal blind policy might be applicable to other algorithms-with-predictions problems.

4 PROOF OVERVIEW

We now explain the main ideas we use to prove our main results. We focus on our analysis of SRPT-B (the most complex result), but we briefly comment on how the same ideas apply to analyzing SRPT-E (see Remark 4.2) and PSJF-E (see Remark 4.1).

Our overall approach to comparing SRPT-B to SRPT is to compare each to a third policy, namely SRPT-SE. This approach proved useful because SRPT-SE's rank function has similarities with both SRPT's and SRPT-B's.

- Under both SRPT-SE and SRPT, a job's rank at every age is within a constant factor of its remaining size.
- Under both SRPT-SE and SRPT-B, a job's initial rank is its estimated size, and a job's rank never exceeds its initial rank.

In the remainder of this section, we explain how the above properties help us compare SRPT-SE to each of SRPT and SRPT-B. Interestingly, the two comparisons make use of two very different methods of analyzing mean response time.

4.1 Comparing SRPT-SE to SRPT

SRPT minimizes mean response by prioritizing jobs by remaining size [13]. SRPT-SE almost prioritizes jobs by remaining size, but it can make an error whenever two jobs' remaining sizes are within a constant factor of each other. The specific factor is α/β: if job 1 has remaining size r_1 and job 2 has greater remaining size $r_2 > r_1$, then SRPT will always serve job 1, but SRPT-SE might serve job 2 if $\beta r_2 \leq \alpha r_1$.

Intuitively, one might hope that because SRPT-SE only makes constant-factor errors when prioritizing jobs, its mean response time should suffer by only a constant factor. We show that this indeed is the case, and that the constant factor is the same. Specifically, Theorem 7.2 states

$$\mathbf{E}[T_{\text{SRPT-SE}}] \leq \frac{\alpha}{\beta}\mathbf{E}[T_{\text{SRPT}}]. \tag{4.1}$$

In order to show (4.1), we use a very recently developed formula for mean response time [15]. At a high level, the formula expresses a policy's mean response time in terms of an integral of various types of work. We first describe the formula (see Section 4.1.1) and then describe how we apply it to proving (4.1) (see Section 4.1.2)

4.1.1 Mean Response Time as a Work Integral. Define the (remsize $\leq r$)-*work* of a system to be the total remaining size of the jobs in the system that have remaining size r or less, as illustrated in Fig. 4.1. Scully et al. [15, Theorem 6.3] show that we can write the mean response time of any policy π in terms of the mean amount of (remsize $\leq r$)-work in the system:

$$\mathbf{E}[T_\pi] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[(\text{remsize} \leq r)\text{-work under } \pi]}{r^2}\,\mathrm{d}r. \tag{4.2}$$

See Definitions 5.1 and 5.2 for a formal definition of (remsize $\leq r$)-work and Proposition 5.3 for a formal statement of (4.2).

4.1.2 Comparing SRPT-SE's and SRPT's Work Integrals. With (4.2) in hand, to compare the mean response times of SRPT-SE and SRPT, it suffices to compare their amounts of (remsize $\leq r$)-work. In the proof of Theorem 7.2, we show

$$\mathbf{E}[(\text{remsize} \leq r)\text{-work under SRPT-SE}] \leq \mathbf{E}\left[\left(\text{remsize} \leq \frac{\alpha}{\beta}r\right)\text{-work under SRPT}\right]. \tag{4.3}$$

Combining (4.2) and (4.3) and using a change of variables implies (4.1).

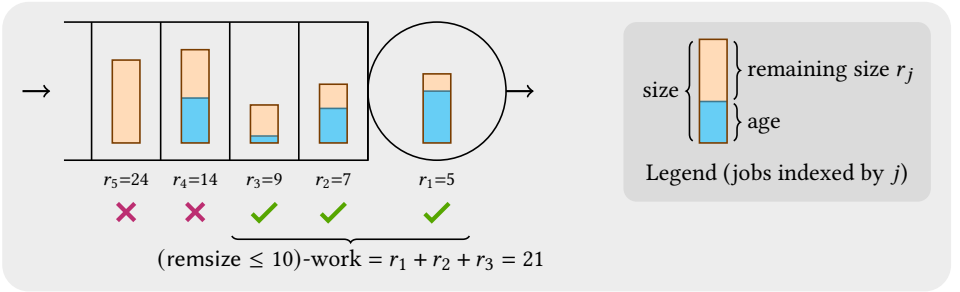


Fig. 4.1. Example of (remsize $\le r$)-Work

The intuition behind (4.3) is as follows. Because SRPT always serves the job of least remaining size, it satisfies the following guarantee:

> Whenever the system has nonzero (remsize $\le r$)-work, SRPT serves a job of remaining size r or less, thus decreasing the amount of (remsize $\le r$)-work.

This guarantee implies that SRPT minimizes mean (remsize $\le r$)-work among all scheduling policies. In contrast, SRPT-SE satisfies a weaker guarantee:

> Whenever the system has nonzero (remsize $\le r$)-work, SRPT-SE serves a job of remaining size $\alpha/\beta \cdot r$ or less, thus decreasing the amount of (remsize $\le \alpha/\beta \cdot r$)-work.

Roughly speaking, this means that whenever the system's (remsize $\le r$)-work is nonzero, SRPT-SE reduces (remsize $\le \alpha/\beta \cdot r$)-work just as efficiently as SRPT does, suggesting a relationship like (4.3) might hold.

The main technical challenge in proving (4.3) is formalizing the above intuition. The key ingredient turns out to be introducing a new variant of (remsize $\le r$)-work. The new variant, called (remsize-e $\le r$)-*work* (see Definitions 5.1 and 5.2), uses scaled estimated remaining size instead of true remaining size. This new variant is important because SRPT-SE always serves the job of least scaled estimated remaining size, so it satisfies the following guarantee:

> Whenever the system has nonzero (remsize-e $\le r$)-work, SRPT-SE serves a job of scaled estimated remaining size r or less, thus decreasing the amount of (remsize-e $\le r$)-work.

This guarantee implies that, analogously to SRPT minimizing mean (remsize $\le r$)-work, SRPT-SE minimizes mean (remsize-e $\le r$)-work (see Proposition 7.1).

Remark 4.1. The proof of Theorem 9.1, which compares PSJF-E to PSJF, follows a similar strategy to the comparison of SRPT-SE to SRPT outlined above. However, the details are significantly more complicated. The main obstacle is that while (4.2) uses remaining size, PSJF-E and PSJF prioritize jobs by *original* estimated and true size, respectively. We overcome this obstacle by introducing several more new variants of (remsize $\le r$)-work.

4.2 Comparing SRPT-B to SRPT-SE

Our approach to comparing SRPT-B to SRPT-SE looks very different from our approach to comparing SRPT-SE to SRPT. In particular, we use a different method of characterizing each policy's mean response time. The method, often called the "tagged-job method", has been used since the early days of M/G/1 scheduling theory to analyze a variety of policies, including SRPT [14]. Recently, Scully et al. [19] generalized the tagged-job method to *all* policies in which a job's rank varies as a function of its age, including all of the policies we study (see Definition 2.3). Below, we outline how the tagged-job method of Scully et al. [19] applies to SRPT-B and SRPT-SE.

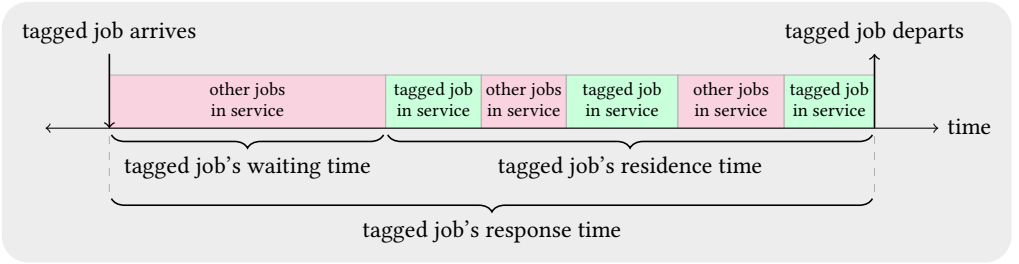


Fig. 4.2. Response Time = Waiting Time + Residence Time

At a high level, the tagged-job method works by following a single "tagged" job on its journey through the system. The tagged job's response time is a random variable with several sources of randomness:

- the random true size S and estimated size Z of the tagged job,
- the random state of the system at the moment the tagged job arrives, and
- the random arrivals that occur after the tagged job.

One can show that the expected response time of the tagged job, where the expectation is taken over all of the above sources of randomness, is indeed the system's mean response time [5].

To compute the tagged job's expected response time, we first condition on its true and estimated sizes. Specifically, let the random variable $T_\pi(s,z)$ denote the response time of the tagged job under policy π given that it has true size $S = s$ and estimated size $Z = z$. We will find $\mathbf{E}[T_\pi(s,z)]$, from which mean response time follows by integrating over s and z:

$$\mathbf{E}[T_\pi] = \int_0^\infty \int_0^\infty \mathbf{E}[T_\pi(s,z)] f_{S,Z}(s,z)\, \mathrm{d}s\, \mathrm{d}z. \tag{4.4}$$

To analyze the tagged job's response time $T_\pi(s,z)$, we split it into two parts:

Waiting time: the amount of time between the tagged job's arrival and the moment the tagged job first receives service, denoted $T_\pi^{\mathsf{wait}}(s,z)$.

Residence time: the amount of time between the tagged job first receives service and the tagged job's completion, denoted $T_\pi^{\mathsf{res}}(s,z)$.

We illustrate waiting time and residence time in Fig. 4.2. We define mean waiting and residence times $\mathbf{E}[T_\pi^{\mathsf{wait}}]$ and $\mathbf{E}[T_\pi^{\mathsf{res}}]$ analogously to (4.4).

To compare SRPT-B to SRPT-SE, we separately compare the two policies' waiting times (see Section 4.2.1 and Proposition 8.7) and residence times (see Section 4.2.2 and Proposition 8.4). At a high level, because SRPT-B and SRPT-SE have similar enough rank functions, we are able to show that SRPT-B's waiting and residence times are not too much larger than SRPT-SE's.

4.2.1 Comparing Waiting Times. Consider a tagged job of estimated size z. Under both SRPT-B and SRPT-SE, the tagged job's initial rank is z. The tagged job's waiting time therefore lasts until any other jobs that remain in the system have rank greater than z, at which point the tagged job, having better rank than all other jobs in the system, is served for the first time. Therefore, the tagged job's waiting time depends on how long each *other* job spends with rank z or less. In particular, the tagged job's waiting time does not depend on its own size, so we denote its waiting time by simply $T_\pi^{\mathsf{wait}}(z)$. Specifically, letting

$$u_\pi(z) = \mathbf{E}\left[\begin{pmatrix}\text{amount of service time during which}\\ \text{a job has rank } z \text{ or less under policy } \pi\end{pmatrix}^2\right],$$

it turns out that comparing $\mathbf{E}[T^{\text{wait}}_{\text{SRPT-B}}(z)]$ to $\mathbf{E}[T^{\text{wait}}_{\text{SRPT-SE}}(z)]$ boils down to comparing $u_{\text{SRPT-B}}(z)$ to $u_{\text{SRPT-SE}}(z)$ (see Theorem 5.4(a)).

One can use simple geometry to show that under SRPT-B, the amount of service time a job spends with rank z or less is at most than twice the amount it would be under SRPT-SE, implying $u_{\text{SRPT-B}}(z) \le 4u_{\text{SRPT-SE}}(z)$. This is strong enough to show graceful degradation of SRPT-B, but it does not imply consistency. But the rank functions of SRPT-B and SRPT-SE do become closer and closer together as α and β approach 1, so one would expect consistency of SRPT-B to hold, too.

The main technical challenge in comparing waiting times is obtaining a bound tight enough to show consistency. In particular, it does not suffice to simply bound $u_{\text{SRPT-B}}(z) - u_{\text{SRPT-SE}}(z)$ with a quantity that vanishes as α and β approach 1. While this is a necessary first step, it shows only that as α and β approach 1, the difference $\mathbf{E}[T^{\text{wait}}_{\text{SRPT-B}}(z)] - \mathbf{E}[T^{\text{wait}}_{\text{SRPT-SE}}(z)]$ vanishes for all z. We seek bounds on mean response time, so we need to integrate over z to show that $\mathbf{E}[T^{\text{wait}}_{\text{SRPT-B}}] - \mathbf{E}[T^{\text{wait}}_{\text{SRPT-SE}}]$ also vanishes. This second step is purely computational but requires some care: there are several choices one must make when bounding the integral, and many choices lead to either intractable expressions or bounds that are too weak to show consistency.

Remark 4.2. The reason for SRPT-E's poor performance is that under SRPT-E, a job can spend up to a $1-\beta$ fraction of its service time below rank 0. This means one can have $u_{\text{SRPT-E}}(z) \ge (1-\beta)\mathbf{E}[S^2]$, from which Theorem 6.1 easily follows. SRPT-B avoids this problem thanks to the bounce in its rank function.

4.2.2 Comparing Residence Times. Consider a tagged job of true size s and estimated size z. When the tagged job starts its residence time, its rank z is less than the rank of every other job in the system. Moreover, under both SRPT-B and SRPT-SE, a job's rank never exceeds this initial rank of z (see Figs. 2.1(e) and 2.1(f)). Therefore, the only reason that the tagged job might be preempted is if new jobs arrive.

Suppose a new job of estimated size z' arrives while the tagged job has age a. What determines whether the new arrival delays the tagged job?

- If the new job's initial rank z' is less than the tagged job's rank at age a, then the new job has priority over the tagged job.
- If z' is at least the tagged job's rank at age a, then the tagged job has priority over the new job, initially. But if later the tagged job will have rank greater than z' at some *future* age $a' > a$, then when the tagged job reaches age a', the new job will have priority over the tagged job.

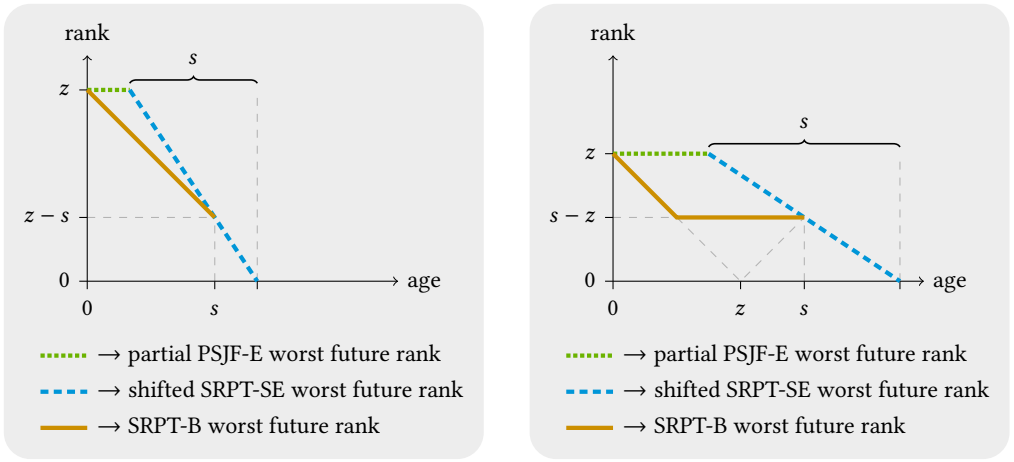
The conclusion of this discussion is what Scully et al. [19] call the "Pessimism Principle", the upshot of which is the following:

> When determining whether a new arrival will delay the tagged job, what matters is not the tagged job's current rank but rather its *worst future rank*.

Therefore, bounding the tagged job's residence time of SRPT-B boils down to bounding the tagged job's worst future rank under SRPT-B.

One simple bound on the tagged job's worst future rank is its initial rank z, because under SRPT-B, a job's rank never exceeds its initial rank (see Fig. 2.1(e)). As it happens, under PSJF-E, the tagged job's worst future rank would always be z. This means that SRPT-B's mean residence time is at most that of PSJF-E, which turns out to be simple to bound (see Lemma 5.7 and Proposition 5.8). This is strong enough to show graceful degradation of SRPT-B, but it does not imply consistency.

The main technical challenge in comparing residence times is obtaining a bound tight enough to show consistency. To show consistency of SRPT-B, we would like to bound SRPT-B's residence time in terms of that of SRPT-SE, not PSJF-E. However, the tagged job's worst future rank at a given age can be greater under SRPT-B than under SRPT-SE. Our solution is, roughly speaking, to bound



(a) Overestimated Size (b) Underestimated Size

Fig. 4.3. Bounding Residence Time of SRPT-B

the worst future rank under SRPT-B to a "shifted" version of worst future rank under SRPT-SE, as illustrated in Fig. 4.3. The result is a bound of the form

$$\mathbf{E}[T^{\text{res}}_{\text{SRPT-B}}(s,z)] \leq \mathbf{E}[T^{\text{res}}_{\text{SRPT-SE}}(s,z)] + c\mathbf{E}[T^{\text{res}}_{\text{PSJF-E}}(s,z)],$$

where c approaches 0 as α and β approach 1. This immediately implies an analogous bound on mean residence times.

Remark 4.3. The Pessimism Principle, namely the fact that a job's residence time is governed by its worst future rank instead of its current rank, is the reason we limit the bounce in SRPT-B's rank function (see Fig. 2.1(e)) to no more than the job's initial rank.

5 BACKGROUND ON M/G/1 SCHEDULING THEORY

In this section, we review definitions and results from M/G/1 scheduling theory that we use in our proofs. Specifically, we review two recently developed methods for computing a policy's mean response time.

- Section 5.1 reviews the "work integral" method, which we use to compare SRPT-SE to SRPT.
- Section 5.2 reviews the "tagged job" method, which we use to compare SRPT-B to SRPT-SE.

Having given in Section 4 an intuitive overview of each method, the main purpose of this section is to present them more formally.

5.1 Mean Response Time via the Work Integral Method

We saw in Section 4.1 that one can compute a policy's mean response time by looking at the amount of different types of work in the system. Specifically, the key definition is $(\text{remsize} \leq r)$-work, the amount of work contributed by jobs which have remaining size r or less.

Below, we give a formal definition of a general kind of work, which includes $(\text{remsize} \leq r)$-work as a special case. Recall from Definition 2.3 that a job's state is a tuple $x = (s, z, a)$ consisting of its true size s, estimated size z, and age a.

Definition 5.1. Let $\varphi : \mathbb{R}^3_+ \to \{\text{false}, \text{true}\}$ be a predicate on job states.

(a) The *φ-work of a job in state x*, denoted $w(x, \varphi)$, is the amount of service a job in state x requires to either complete or reach a state that does not satisfy φ. That is, a job's φ-work, roughly speaking, its remaining processing time while satisfying φ. Formally,

$$w((s, z, a), \varphi) = \sup\{w \in [0, s - a) \mid \varphi(s, z, a + w)\}.$$

(b) The *(system) φ-work* is the total φ-work of all jobs in the system. We denote by $W_\pi(\varphi)$ the steady-state distribution of the system φ-work under policy π.

When discussing φ-work, it is helpful to have a shorthand notation for describing predicates. The following definition describes such a shorthand.

Definition 5.2.

(a) Let $\mathsf{func} : \mathbb{R}_+^3 \to \mathbb{R}$ be a real function on job states and $r \in \mathbb{R}$ be a constant. The predicate $(\mathsf{func} \le r)$ is true for those states x such that $\mathsf{func}(x) \le r$. We define other inequality predicates similarly, e.g. $\mathsf{func}_1 \le r < \mathsf{func}_2$, and we omit the parentheses when they would be redundant, e.g. $W_\pi(\mathsf{func} \le r)$. To disambiguate between functions and constants, we use sans-serif font for functions.

(b) We frequently use the following functions on job states in the above shorthand:

(true) size	$\mathsf{size}(s, z, a)$	$= s$,
(true) remaining size	$\mathsf{remsize}(s, z, a)$	$= s - a$,
estimated size	$\mathsf{size\text{-}e}(s, z, a)$	$= z$,
scaled estimated remaining size	$\mathsf{remsize\text{-}e}(s, z, a)$	$= z/s \cdot (s - a)$.

All of the functions in Definition 5.2(b) happen to also be rank functions of one of the policies we study (see Definition 2.3). For example, $\mathsf{remsize} = \mathsf{rank}_{\mathrm{SRPT}}$. We introduce the names in Definition 5.2(b) to emphasize that, for instance, we can consider $(\mathsf{remsize} \le r)$-work under policies other than SRPT.

This last example is especially important, because a policy's mean response time is connected to its steady-state $(\mathsf{remsize} \le r)$-work.

Proposition 5.3 (special case of [15, Theorem 6.3]). *In the M/G/1, the mean response time of any policy π is*

$$\mathbf{E}[T_\pi] = \frac{1}{\lambda} \int_0^\infty \frac{\mathbf{E}[W_\pi(\mathsf{remsize} \le r)]}{r^2} \, \mathrm{d}r.$$

The above result is a recently derived, powerful identity for the mean response time. We use it in our analyses of SRPT-SE (see Section 7) and PSJF-E (see Section 9).

5.2 Mean Response Time via the Tagged Job Method

We describe the main ideas behind the tagged job method in Section 4.2. As a reminder, the approach is to focus on a single "tagged" job; split its response time into two parts, *waiting time* and *residence time* (see Fig. 4.2); and analyze each part separately. The purpose of this section is to define the concepts and notation that we need in order to write down formulas for the tagged job's expected waiting and residence times.

Consider a tagged job of of true size s and estimated size z arriving to a steady-state system under some scheduling policy π. An important quantity when computing the tagged job's waiting and residence times is the rate at which jobs with rank less than the tagged job arrive to the system. We can interpret the system load $\rho = \lambda \mathbf{E}[S]$ as the overall rate at which work arrives. Analogously,

define[5]

$$\rho_S(s) = \lambda \mathbf{E}[S\mathbb{1}(S \le s)], \qquad \rho_Z(z) = \lambda \mathbf{E}[S\mathbb{1}(Z \le z)]. \tag{5.1}$$

These are the average rates at which work arrives when one only counts work from jobs whose true size or estimated size, respectively, is at most some threshold. Specifically, $\rho_S(s)$ is important for analyzing policies that use a job's true size (SRPT and PSJF), while $\rho_Z(z)$ is important for analyzing policies that use a job's estimated size (SRPT-E, PSJF-E, SRPT-B, and SRPT-SE).

Having defined (5.1), there are two more quantities we need to define before stating formulas for the tagged job's waiting and residence times. We give formal and informal expressions for each.

- In the waiting time formula, we use the quantity[6]

$$u_\pi(r) = \mathbf{E}\left[\left|\{a \in [0,S) \mid \mathsf{rank}_\pi(S,Z,a) \le r\}\right|^2\right] = \mathbf{E}\left[\begin{pmatrix}\text{amount of service time during which}\\ \text{a job has rank } r \text{ or less under policy } \pi\end{pmatrix}^2\right].$$

- In the residence time formula, we use the *worst future rank* of a job, which we define as

$$\mathsf{rank}^{\mathsf{worst}}_\pi(s,z,a) = \sup_{b\in[a,s)} \mathsf{rank}_\pi(s,z,b) = \begin{pmatrix}\text{maximum rank a job currently in state } (s,z,a) \text{ has}\\ \text{under policy } \pi \text{ between now and its completion}\end{pmatrix}.$$

We are now ready to state the waiting and residence time formulas for the policies we consider.

Proposition 5.4 (special case of [19, Theorem 5.5]). *Consider an M/G/1 under policy* $\pi \in$ {SRPT-E, PSJF-E, SRPT-B, SRPT-SE}.

(a) The expected waiting time of a (tagged) job of estimated size z *is*

$$\mathbf{E}[T^{\mathsf{wait}}_\pi(z)] = \frac{\lambda}{2}\frac{u_\pi(z)}{(1-\rho_Z(z))^2}.$$

(b) The expected residence time of a (tagged) job of true size s *and estimated size* z *is*

$$\mathbf{E}[T^{\mathsf{res}}_\pi(s,z)] = \int_0^s \frac{1}{1-\rho_Z(\mathsf{rank}^{\mathsf{worst}}_\pi(s,z,a))}\,\mathrm{d}a.$$

Some intuition for the formulas above is warranted. We explain one simple case below, referring the reader to Scully et al. [19, Section 4] for more discussion.

Remark 5.5. Consider how Theorem 5.4(b) applies to PSJF-E. We have $\mathsf{rank}^{\mathsf{worst}}_{\text{PSJF-E}}(s,z,a) = z$, so

$$\mathbf{E}[T^{\mathsf{res}}_{\text{PSJF-E}}(s,z)] = \frac{s}{1-\rho_Z(z)}. \tag{5.2}$$

The intuitive interpretation of (5.2) is as follows. During the tagged job's residence time, new jobs may arrive at any time, preempting the job in service if they have rank below z. On average, the server spends a $\rho_Z(z)$ fraction of its time serving these new jobs of rank below z, leaving a $1-\rho_Z(z)$ fraction for serving the tagged job. This means that the tagged job's age increases at average rate $1/(1-\rho_Z(z))$, so it takes $s/(1-\rho_Z(z))$ time to go from age 0 to age s.

Formulas very similar to those in Proposition 5.4 hold for SRPT and PSJF. In fact, such formulas are classic results [5, 14]. The differences is that expected waiting and residence times both depend only on a job's size s, so we replace $u_\pi(z)$ with $u_\pi(s)$, and we replace $\rho_Z(z)$ with $\rho_S(s)$. Alternatively, one may view the SRPT and PSJF formulas as special cases of the SRPT-SE and PSJF-E formulas in

[5]Recall that a random job's true size S and estimated size Z are *not* independent, which is important in the definition of $\rho_Z(z)$.

[6]In the formal expression, $|\cdot|$ denotes interval length. The definition we give is simplified by the fact that for the scheduling policies we consider, a job's rank is below a threshold r for at most one contiguous interval of ages. A more complicated definition is needed for policies with general rank functions [19].

a system where $S = Z$ for all jobs. We omit the exact statements because in our proofs, we end up analyzing SRPT and PSJF with the work integral method.

5.3 Useful Lemmas

The following simple lemmas will be useful in our later analyses.

Lemma 5.6.

$$\frac{\mathrm{d}}{\mathrm{d}s}\rho_S(s) = \lambda s f_S(s), \qquad \frac{\mathrm{d}}{\mathrm{d}z}\rho_Z(z) = \lambda \mathbf{E}[S \mid Z = z] f_Z(z).$$

Proof. These follow from (5.1) and the fact that we can write

$$\mathbf{E}[S\mathbb{1}(S \le s)] = \int_0^s s' f_S(s')\,\mathrm{d}s' \qquad \mathbf{E}[S\mathbb{1}(Z \le z)] = \int_0^z \mathbf{E}[S \mid Z = z'] f_Z(z')\,\mathrm{d}z'. \qquad \square$$

Lemma 5.7. *The mean residence time of PSJF-E is*

$$\mathbf{E}[T^{\text{res}}_{\text{PSJF-E}}] = \left(\frac{1}{\rho}\ln\frac{1}{1-\rho}\right)\mathbf{E}[S].$$

Proof. Combining (5.2) and Lemma 5.6 yields

$$\begin{aligned}
\mathbf{E}[T^{\text{res}}_{\text{PSJF-E}}] &= \int_0^\infty \int_0^\infty \mathbf{E}[T^{\text{res}}_{\text{PSJF-E}}(s,z)] f_{S,Z}(s,z)\,\mathrm{d}z && \text{[conditioning on } s \text{ and } z\text{]} \\
&= \int_0^\infty \int_0^\infty \frac{s}{1-\rho_Z(z)} f_{S,Z}(s,z)\,\mathrm{d}z && \text{[(5.2)]} \\
&= \int_0^\infty \frac{\mathbf{E}[S \mid Z = z]}{1-\rho_Z(z)} f_Z(z)\,\mathrm{d}z \\
&= \frac{1}{\lambda}\ln\frac{1}{1-\rho}. && \text{[Lemma 5.6]}
\end{aligned}$$

The lemma follows from $\rho = \lambda\mathbf{E}[S]$. $\square$

The main reason that Lemma 5.7 is useful is the following result of Wierman et al. [20], which shows that the mean residence time of PSJF-E is a *lower bound* on the mean response time of SRPT.

Proposition 5.8 ([20, Theorem 5.8]). *The mean response time of SRPT is bounded below by*

$$\mathbf{E}[T_{\text{SRPT}}] \ge \left(\frac{1}{\rho}\ln\frac{1}{1-\rho}\right)\mathbf{E}[S].$$

6 SRPT WITH ESTIMATES (SRPT-E)

Our first result shows that, for (β, α)-bounded size estimates with $\beta < 1$, the performance of SRPT-E can lead to arbitrarily large approximation ratios. This formalizes previous empirical results (see e.g. [9]), where it was noted that underestimates of large jobs, particularly when job sizes are highly variable, can lead to poor performance for SRPT-E, as a large underestimated job being served can obtain a negative estimated remaining time and block service for all other jobs, even when the actual remaining time is large. This formalization motivates our seeking a variation of SRPT that avoids this problem, and our examination of PSJF-E, which we show in contrast has bounded approximation ratio for (β, α)-bounded size estimates.

Theorem 6.1 (Performance of SRPT-E). *Consider the M/G/1 with (β, α)-bounded size estimates.*

(a) For any size distribution S, there exists a joint distribution (S, Z) of true and estimated sizes such that the mean response time of SRPT-E is bounded below by

$$\mathbf{E}[T_{\text{SRPT-E}}] \geq \frac{\lambda(1-\beta)^2}{2}\mathbf{E}[S^2],$$

(b) The approximation ratio of SRPT-E may be arbitrarily large or infinite whenever $\beta < 1$.

Proof. For a given job distribution S, let $Z = \beta S$.

From Theorem 5.4(a), we know that the expected waiting time of SRPT-E is

$$\mathbf{E}[T^{\text{wait}}_{\text{SRPT-E}}(z)] = \frac{\lambda}{2}\frac{\mathbf{E}\left[\left(\begin{smallmatrix}\text{amount of service time during which}\\ \text{a job has rank } z \text{ or less under SRPT-E}\end{smallmatrix}\right)^2\right]}{(1-\rho_Z(z))^2}$$
$$\geq \frac{\lambda}{2}\mathbf{E}\left[\left(\begin{smallmatrix}\text{amount of service time during which}\\ \text{a job has rank 0 or less under SRPT-E}\end{smallmatrix}\right)^2\right].$$

Note that this lower bound applies regardless of z.

Because $Z = \beta S$, a job of size s starts at rank βs, and reaches rank 0 at age βs. Therefore, it receives $(1-\beta)s$ service with rank ≤ 0. Therefore, we can lower bound waiting time explicitly:

$$\mathbf{E}[T^{\text{wait}}_{\text{SRPT-E}}] \geq \frac{\lambda}{2}\mathbf{E}\left[\left((1-\beta)S\right)^2\right].$$

From this result, we see that SRPT-E's response time grows with $\mathbf{E}[S^2]$, which can be arbitrarily large or infinite. In contrast, the response time of SRPT is finite even for job size distributions with infinite $\mathbf{E}[S^2]$, such as a Pareto distribution with exponent 1.5. □

7 SRPT WITH SCALING ESTIMATES (SRPT-SE)

SRPT-SE is a policy that uses a job's true size and estimated size to assign its rank. SRPT-SE is thus not a practical policy, as one would prefer SRPT if true sizes were known. However, analyzing it is helpful for a few reasons. First, it is a useful warmup for the analysis of PSJF-E, which follows the same outline but is somewhat more complicated (see Section 9). Second, it is the first step of analyzing SRPT-B, whose performance we bound relative to SRPT-SE (see Section 8). Third, there are settings in which a policy similar to SRPT-SE, which enjoys similarly good performance, could be implemented in practice (see Appendix D).

Our main tool for analyzing SRPT-SE is Proposition 5.3, which expresses mean response time in terms of mean $(\text{remsize} \leq r)$-work, with less $(\text{remsize} \leq r)$-work corresponding to lower response time.

Before analyzing SRPT-SE, it is helpful to consider how one might use Proposition 5.3 to show that SRPT minimizes mean response time. The key is that for every value of r, SRPT minimizes mean $(\text{remsize} \leq r)$-work, or equivalently mean $(\text{rank}_{\text{SRPT}} \leq r)$-work. The intuition is that whenever the system has nonzero $(\text{rank}_{\text{SRPT}} \leq r)$-work, SRPT serves a job of rank r or less, thus reducing the amount of $(\text{rank}_{\text{SRPT}} \leq r)$-work.

It turns out that an analogous property holds for any policy that can be defined using a rank function [19], including those in Definition 2.3.

Proposition 7.1 (very similar to [18, Theorem VII.7][7]). *Consider a policy $\pi \in \{\text{SRPT}, \text{PSJF}, \text{SRPT-E}, \text{PSJF-E}, \text{SRPT-B}, \text{SRPT-SE}\}$ and any rank r. In the M/G/1, the policy that minimizes the mean amount of steady-state $(\text{rank}_\pi \leq r)$-work is π itself. That is, for any policy π',*

$$\mathbf{E}[W_\pi(\text{rank}_\pi \leq r)] \leq \mathbf{E}[W_{\pi'}(\text{rank}_\pi \leq r)].$$

Because remsize-e = $\mathsf{rank}_{\text{SRPT-SE}}$, we have from Proposition 7.1 that SRPT-SE minimizes mean (remsize-e $\le r$)-work. Of course, Proposition 5.3 uses (remsize $\le r$)-work, not (remsize-e $\le r$)-work. Fortunately, we can leverage the fact that we have (β, α)-bounded size estimates to relate (remsize $\le r$)-work to (remsize-e $\le r$)-work, yielding the following result.

Theorem 7.2 (Performance of SRPT-SE). *Consider the M/G/1 with (β, α)-bounded size estimates.*

(a) The mean response time of SRPT-SE is bounded above by

$$\mathbf{E}[T_{\text{SRPT-SE}}] \le \frac{\alpha}{\beta}\mathbf{E}[T_{\text{SRPT}}],$$

(b) The approximation ratio of SRPT-SE is at most α/β.

(c) As α and β converge to 1, the approximation ratio of SRPT-SE converges to 1. This convergence is uniform in the arrival rate and the joint distribution of true and estimated sizes.

Proof. It clearly suffices to prove (a). Recall the following facts about job states x:

- $\mathsf{rank}_{\text{SRPT}}(x) = \mathsf{remsize}(x)$,
- $\mathsf{rank}_{\text{SRPT-SE}}(x) = \mathsf{remsize\text{-}e}(x)$, and
- $\mathsf{remsize\text{-}e}(x)/\mathsf{remsize}(x) = \mathsf{size\text{-}e}(x)/\mathsf{size}(x) \in [\beta, \alpha]$.

Using the above facts together with Propositions 7.1 and 5.3, we compute

$$\begin{aligned}
\mathbf{E}[T_{\text{SRPT-SE}}] &= \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W_{\text{SRPT-SE}}(\mathsf{remsize} \le r)]}{r^2}\,\mathrm{d}r && \text{[Proposition 5.3]}\\
&\le \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W_{\text{SRPT-SE}}(\mathsf{remsize\text{-}e} \le \alpha r)]}{r^2}\,\mathrm{d}r && \text{[using } \mathsf{remsize\text{-}e}(x)/\mathsf{remsize}(x) \le \alpha\text{]}\\
&\le \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W_{\text{SRPT}}(\mathsf{remsize\text{-}e} \le \alpha r)]}{r^2}\,\mathrm{d}r && \text{[Proposition 7.1]}\\
&\le \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}\left[W_{\text{SRPT}}\left(\mathsf{remsize} \le \frac{\alpha}{\beta} r\right)\right]}{r^2}\,\mathrm{d}r && \text{[using } \mathsf{remsize\text{-}e}(x)/\mathsf{remsize}(x) \ge \beta\text{]}\\
&= \frac{\alpha}{\beta}\frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W_{\text{SRPT}}(\mathsf{remsize} \le r')]}{(r')^2}\,\mathrm{d}r' && \text{[setting } r' = \alpha/\beta \cdot r\text{]}\\
&= \frac{\alpha}{\beta}\mathbf{E}[T_{\text{SRPT}}]. && \text{[Proposition 5.3]}
\end{aligned}$$

□

Theorem 7.2 completes our analysis of SRPT-SE. We describe a related result for a similar scheme that does not use the job's true size in Appendix D.

8 SRPT WITH BOUNCE (SRPT-B)

As we have mentioned, previous works have noted that when using SRPT-E, large jobs that are underestimated will have estimated remaining sizes that become negative while the true remaining time is still relatively large, leading to long waiting times for jobs stuck behind them. An open question has been to modify SRPT with estimated sizes in a way that avoids this issue in a robust manner, without assumptions on job size distributions. Our suggested solution, SRPT with Bounce (SRPT-B), handles this by modifying the rank function from $z - a$ to $\min\{|z - a|, z\}$.[8] We prove the following results for SRPT-B.

[7] While Scully and Harchol-Balter [18, Theorem VII.7] consider a specific policy, namely a generalization of SRPT, the same proof applies virtually verbatim to any policy that can be defined by a rank function [19].

[8] We note that it is an interesting open question to consider the effects of other possible forms for the bounce, which we do not investigate here.

Theorem 8.1 (Performance of SRPT-B). *Consider the M/G/1 with (β, α)-bounded size estimates.*

(a) The mean response time of SRPT-B is bounded above by

$$\mathbf{E}[T_{\text{SRPT-B}}] \le \frac{\alpha}{\beta}\mathbf{E}[T_{\text{SRPT}}] + \left(\frac{3}{2}\alpha\mathbb{1}(\beta < 1) + 1\right)\min\left\{1, \max\left\{1 - \frac{1}{\alpha}, \frac{1}{\beta} - 1\right\}\right\}\left(\frac{1}{\rho}\ln\frac{1}{1-\rho} - 1\right)\mathbf{E}[S].$$

(b) The approximation ratio of SRPT-B is at most $3.5\alpha/\beta$.

(c) As α and β converge to 1, the approximation ratio of SRPT-B converges to 1. This convergence is uniform in the arrival rate and the joint distribution of true and estimated sizes.

Our overall approach to analyzing SRPT-B is to compare it to SRPT-SE. We separately compare the waiting times and residence times of the two policies (see Section 4.2). Both comparisons boil down to comparing the amount of time a job spends above or below a given rank under each policy. We begin with the residence time comparison (see Section 8.1) before moving on to the more complicated waiting time comparison (see Section 8.2). Combining the two comparisons and some additional computation (see Section 8.3) yields Theorem 8.1.

8.1 Residence Time Difference between SRPT-B and SRPT-SE

By Theorem 5.4(b), the expected residence time of the job under SRPT-B, SRPT-SE, or PSJF-E is an integral of $1/(1-\rho_Z(\cdot))$ terms over the job's ages, where the value plugged is the worst future rank of the job.

Consider a job of true size s and estimated size z. In order to bound $\mathbf{E}[T^{\text{res}}_{\text{SRPT}}(s,z)]$, we will find functions $g_{s,z}(\cdot), h_{s,z}(\cdot)$, and values $c_{s,z}, t_{s,z} > 0$ such that

$$\int_0^{t_{s,z}} \frac{1}{1-\rho_Z(g_{s,z}(a))}\,\mathrm{d}a = \mathbf{E}[T^{\text{res}}_{\text{SRPT-B}}(s,z)] + c_{s,z}s, \tag{8.1}$$

$$\int_0^{t_{s,z}} \frac{1}{1-\rho_Z(h_{s,z}(a))}\,\mathrm{d}a = \mathbf{E}[T^{\text{res}}_{\text{SRPT-SE}}(s,z)] + c_{s,z}\mathbf{E}[T^{\text{res}}_{\text{PSJF-E}}(s,z)], \tag{8.2}$$

$$g_{s,z}(a) \le h_{s,z}(a) \quad \text{for all } a \in (0, t_{s,z}). \tag{8.3}$$

Because $1/(1-\rho_Z(\cdot))$ is nondecreasing, this implies

$$\mathbf{E}[T^{\text{res}}_{\text{SRPT-B}}(s,z)] \le \mathbf{E}[T^{\text{res}}_{\text{SRPT-SE}}(s,z)] + c_{s,z}(\mathbf{E}[T^{\text{res}}_{\text{PSJF-E}}(s,z)] - s).$$

Finally, we will bound $c_{s,z}$ by a value c which is independent of s and z, obtaining

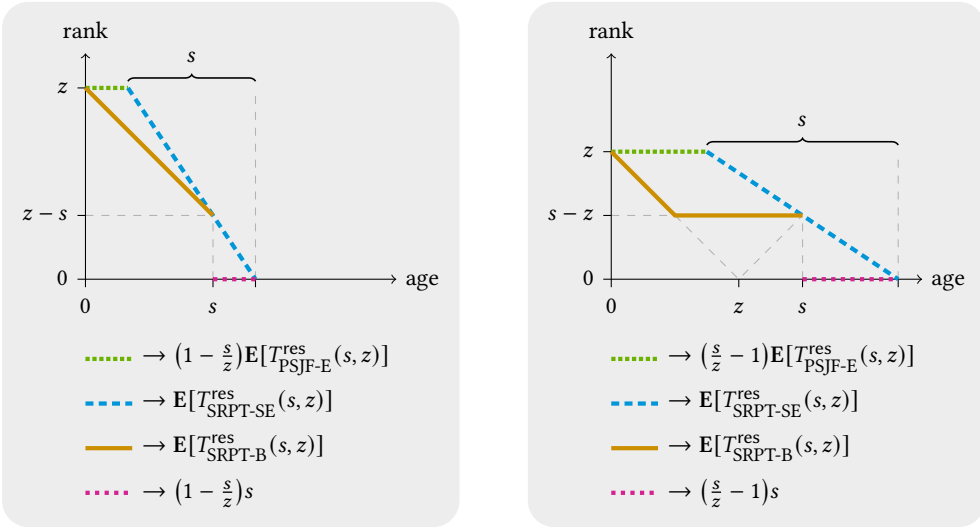
$$\mathbf{E}[T^{\text{res}}_{\text{SRPT-B}}] \le \mathbf{E}[T^{\text{res}}_{\text{SRPT-SE}}] + c(\mathbf{E}[T^{\text{res}}_{\text{PSJF-E}}] - \mathbf{E}[S]). \tag{8.4}$$

We begin by computing the worst future ranks of each policy.

Lemma 8.2. *The worst future ranks of a job of true size s, estimated size z, and age a under SRPT-B, SRPT-SE, and PSJF-E are*

$$\begin{aligned}
\text{rank}^{\text{worst}}_{\text{SRPT-B}}(s,z,a) &= \max\{z-a, \min\{s-z, z\}\},\\
\text{rank}^{\text{worst}}_{\text{SRPT-SE}}(s,z,a) &= \text{SRPT-SE}(s,z,a) = \frac{z}{s}(s-a),\\
\text{rank}^{\text{worst}}_{\text{PSJF-E}}(s,z,a) &= \text{PSJF-E}(s,z,a) = z.
\end{aligned}$$

Proof. SRPT-SE and PSJF-E have nondecreasing rank as a function of age a, so the job's worst future rank is its current rank. If $s \le z$, then the same is true for SRPT-B. If instead $z < s \le 2z$, then under SRPT-B, the job's worst future rank is its current rank $z-a$ until age $a = s - 2(s-z)$, after which the worst future rank is its final rank, namely $s-z$. Finally, if $s > 2z$, then the job's worst future rank is always its final rank, namely z. □



(a) Estimated Size $z \geq$ True Size s (b) Estimated Size $z <$ True Size $s \leq 2z$

Fig. 8.1. Relating Residence Times of SRPT-B, SRPT-SE, and PSJF-E

Lemma 8.3. *The following definitions satisfy* (8.1)–(8.3)*:*

$$c_{s,z} = \min\left\{\left|1 - \frac{s}{z}\right|, 1\right\},$$
$$t_{s,z} = (1 + c_{s,z})s,$$
$$g_{s,z}(a) = \begin{cases} \max\{z - a, \min\{s - z, z\}\} & \text{if } a \leq s \\ 0 & \text{if } a > s, \end{cases}$$
$$h_{s,z}(a) = \begin{cases} z & \text{if } a \leq c_{s,z}s \\ \frac{z}{s}(s - (a - c_{s,z}s)) & \text{if } a > c_{s,z}s. \end{cases}$$

Proof. Combining Theorem 5.4(b) and Lemma 8.2 yields (8.1) and (8.2). A simple case analysis, illustrated in Fig. 8.1, yields (8.3). The illustration shows the $z \geq s$ and $z < s \leq 2z$ cases, and the $s > 2z$ case is essentially the same as the $s = 2z$ case. □

Proposition 8.4. *The mean residence time of SRPT-B is bounded above by*

$$\mathbf{E}[T^{\text{res}}_{\text{SRPT-B}}] \leq \mathbf{E}[T^{\text{res}}_{\text{SRPT-SE}}] + \min\left\{1, \max\left\{1 - \frac{1}{\alpha}, \frac{1}{\beta} - 1\right\}\right\}\left(\frac{1}{\rho}\ln\frac{1}{1-\rho} - 1\right)\mathbf{E}[S].$$

Proof. Let $c_{s,z} = \min\{1, |1 - s/z|\}$, as in Lemma 8.3. Because we have (β, α)-bounded size estimates, $c_{s,z} \leq \min\{1, \max\{1 - 1/\alpha, 1/\beta - 1\}\}$ for all feasible pairs of true size s and estimated size z. This bound on $c_{s,z}$ is independent of s and z, so by Lemma 8.3 and the discussion at the start of this section, (8.4) holds with $c = \min\{1, \max\{1 - 1/\alpha, 1/\beta - 1\}\}$. The result then follows from Lemma 5.7, which gives the value of $\mathbf{E}[T^{\text{res}}_{\text{PSJF-E}}]$. □

8.2 Waiting Time Difference between SRPT-B and SRPT-SE

By Theorem 5.4(a), computing the waiting time of SRPT-B and SRPT-SE boils down to computing how much of a job's service happens below a given rank. That is, letting

$$u_\pi(z) = \mathbf{E}\left[\begin{pmatrix}\text{amount of service time during which}\\ \text{a job has rank } z \text{ or less under policy } \pi\end{pmatrix}^2\right], \tag{8.5}$$

our goal is to compare $u_{\text{SRPT-B}}(z)$ and $u_{\text{SRPT-SE}}(z)$. We begin by computing both quantities.

Lemma 8.5.

$$u_{\text{SRPT-B}}(z) = \mathbf{E}\Big[S^2\mathbb{1}(Z \le z) + \big(\max\{0, \min\{S-(Z-z), 2z\}\}\big)^2\mathbb{1}(Z > z)\Big],$$
$$u_{\text{SRPT-SE}}(z) = \mathbf{E}\left[S^2\mathbb{1}(Z \le z) + \left(\frac{S}{Z}z\right)^2\mathbb{1}(Z > z)\right].$$

Proof. Consider a job with true size S and estimated size Z drawn from the joint distribution of true and estimated sizes. Under both policies, if $Z \le z$, then the job's rank remains z or less for its entire service time, which explains the $\mathbb{1}(Z \le z)$ terms. If instead $Z > z$, then the following reasoning explains the $\mathbb{1}(Z > z)$ terms.

- Under SRPT-B, the job has rank z or less when its age is in the interval $[0, S) \cap [Z - z, Z + z]$.
- Under SRPT-SE, the job spends a z/Z fraction of its service time with rank z or less. □

Lemma 8.6.

$$u_{\text{SRPT-B}}(z) - u_{\text{SRPT-SE}}(z) \le 3z\max\{1-\beta, 0\}\mathbf{E}[S\mathbb{1}(Z > z)].$$

Proof. We begin by applying Lemma 8.5:

$$u_{\text{SRPT-B}}(z) - u_{\text{SRPT-SE}}(z) = \mathbf{E}\left[\left(\big(\max\{0, \min\{S-(Z-z), 2z\}\}\big)^2 - \left(\frac{S}{Z}z\right)^2\right)\mathbb{1}(Z > z)\right].$$

If $S \le Z$, then because $z/Z < 1$ whenever the indicator is nonzero, the following computation shows that the expression inside the expectation is nonpositive:

$$\frac{z}{Z} < 1 \quad\Rightarrow\quad \frac{z}{Z}(Z - S) \le Z - S \quad\Rightarrow\quad S - (Z - z) \le \frac{S}{Z}z.$$

This means adding an $S > Z$ to the indicator gives an upper bound, from which we compute

$$\begin{aligned}
&u_{\text{SRPT-B}}(z) - u_{\text{SRPT-SE}}(z)\\
&\quad\le \mathbf{E}\left[\left(\big(\max\{0, \min\{S-(Z-z), 2z\}\}\big)^2 - \left(\frac{S}{Z}z\right)^2\right)\mathbb{1}(S > Z > z)\right] && \text{[nonpositive when } S \le Z\text{]}\\
&\quad\le \mathbf{E}\big[\big((z + \min\{S - Z, z\})^2 - z^2\big)\mathbb{1}(S > Z > z)\big]\\
&\quad\le \mathbf{E}[3z(S - Z)\mathbb{1}(S > Z > z)]\\
&\quad\le 3z\max\{1-\beta, 0\}\mathbf{E}[S\mathbb{1}(Z > z)]. && \text{[using } Z \ge \beta S\text{]}
\end{aligned}$$

□

Proposition 8.7. *The mean waiting time of SRPT-B is bounded above by*

$$\mathbf{E}[T^{\text{wait}}_{\text{SRPT-B}}] \le \mathbf{E}[T^{\text{wait}}_{\text{SRPT-SE}}] + \frac{3}{2}\alpha\max\{1-\beta, 0\}\left(\frac{1}{\rho}\ln\frac{1}{1-\rho} - 1\right)\mathbf{E}[S].$$

Proof. The high-level steps of the proof are the following:

- We use Theorem 5.4(a) to express $\mathbf{E}[T^{\text{wait}}_{\text{SRPT-B}}] - \mathbf{E}[T^{\text{wait}}_{\text{SRPT-SE}}]$ in terms of $u_{\text{SRPT-B}}(z) - u_{\text{SRPT-SE}}(z)$.
- We bound $u_{\text{SRPT-B}}(z) - u_{\text{SRPT-SE}}(z)$ using Lemma 8.6.

- We use integration by parts, obtaining an expression that is similar to one that appears in the proof of Lemma 5.7.
- Mirroring the remainder of the proof of Lemma 5.7, which involves applying Lemma 5.6, yields the desired result.

We begin by computing

$$\begin{aligned}
&\mathbf{E}[T^{\text{wait}}_{\text{SRPT-B}}] - \mathbf{E}[T^{\text{wait}}_{\text{SRPT-SE}}] \\
&\quad= \int_0^\infty (\mathbf{E}[T^{\text{wait}}_{\text{SRPT-B}}(z)] - \mathbf{E}[T^{\text{wait}}_{\text{SRPT-SE}}(z)]) f_Z(z)\,\mathrm{d}z && \text{[conditioning on } Z\text{]} \\
&\quad= \frac{\lambda}{2}\int_0^\infty \frac{u_{\text{SRPT-B}}(z) - u_{\text{SRPT-SE}}(z)}{(1-\rho_Z(z))^2} f_Z(z)\,\mathrm{d}z && \text{[Theorem 5.4(a) and (8.5)]} \\
&\quad\le \frac{3}{2}\max\{1-\beta, 0\}\int_0^\infty \frac{\lambda z \mathbf{E}[S\mathbb{1}(Z>z)]}{(1-\rho_Z(z))^2} f_Z(z)\,\mathrm{d}z && \text{[Lemma 8.6]} \\
&\quad\le \frac{3}{2}\alpha\max\{1-\beta, 0\}\int_0^\infty \frac{\lambda \mathbf{E}[S \mid Z=z]\mathbf{E}[S\mathbb{1}(Z>z)]}{(1-\rho_Z(z))^2} f_Z(z)\,\mathrm{d}z. && \text{[using } Z \le \alpha S\text{]}
\end{aligned}$$

It remains only to bound the last integral. By Lemma 5.6, we have

$$\frac{\lambda \mathbf{E}[S \mid Z=z] f_Z(z)}{(1-\rho_Z(z))^2} = \frac{\mathrm{d}}{\mathrm{d}z}\frac{1}{1-\rho}, \tag{8.6}$$

and conditioning on Z yields

$$\mathbf{E}[S\mathbb{1}(Z>z)] = \int_z^\infty \mathbf{E}[S \mid Z=z']\,\mathrm{d}z'. \tag{8.7}$$

Combining these equations and integrating by parts, we obtain

$$\begin{aligned}
&\int_0^\infty \frac{\lambda \mathbf{E}[S \mid Z=z]\mathbf{E}[S\mathbb{1}(Z>z)]}{(1-\rho_Z(z))^2} f_Z(z)\,\mathrm{d}z \\
&\quad= \int_0^\infty \left(\frac{\mathrm{d}}{\mathrm{d}z}\frac{1}{1-\rho_Z(z)}\right)\left(\int_z^\infty \mathbf{E}[S \mid Z=z'] f_Z(z')\,\mathrm{d}z'\right)\mathrm{d}z && \text{[(8.6) and (8.7)]} \\
&\quad= 0 - \mathbf{E}[S] + \int_0^\infty \frac{\mathbf{E}[S \mid Z=z] f_Z(z)}{1-\rho_Z(z)}\,\mathrm{d}z && \text{[integrating by parts]} \\
&\quad= \left(\frac{1}{\rho}\ln\frac{1}{1-\rho} - 1\right)\mathbf{E}[S]. && \text{[Lemma 5.6]}
\end{aligned}$$

□

8.3 Combining Waiting Time and Residence Time Bounds

Proof of Theorem 8.1. By Proposition 5.8, it suffices to prove (a), which follows by combining Propositions 8.4 and 8.7, applying Theorem 7.2(a), and observing

$$\max\{1-\beta, 0\} \le \mathbb{1}(\beta < 1)\min\left\{1, \max\left\{1 - \frac{1}{\alpha}, \frac{1}{\beta} - 1\right\}\right\}.$$

□

9 PSJF WITH ESTIMATES (PSJF-E)

Theorem 9.1 (Performance of PSJF-E). *Consider the M/G/1 with (β, α)-bounded size estimates.*

(a) The mean response time of PSJF-E is bounded above by

$$\mathbf{E}[T_{\text{PSJF-E}}] \le \frac{\alpha}{\beta}\mathbf{E}[T_{\text{PSJF}}].$$

(b) The approximation ratio of PSJF-E is at most $1.5\alpha/\beta$.

We prove Theorem 9.1 using an argument similar to the proof of Theorem 7.2. However, because PSJF prioritizes jobs by original size instead of remaining size, combining Propositions 7.1 and 5.3 to compare PSJF with PSJF-E is not as straightforward as comparing SRPT with SRPT-SE. We begin by working out how Proposition 5.3 applies to PSJF and PSJF-E.

Lemma 9.2. *The mean response time of PSJF is*

$$\mathbf{E}[T_{\text{PSJF}}] = \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W_{\text{PSJF}}(\text{size} \le r)]}{r^2}\,\mathrm{d}r + \frac{1}{2\lambda}\ln\frac{1}{1-\rho}.$$

Proof. Applying Proposition 5.3 yields

$$\begin{aligned}\mathbf{E}[T_{\text{PSJF}}] &= \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W_{\text{PSJF}}(\text{remsize} \le r)]}{r^2}\,\mathrm{d}r \\ &= \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W_{\text{PSJF}}(\text{size} \le r)]}{r^2}\,\mathrm{d}r + \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W_{\text{PSJF}}(\text{remsize} \le r < \text{size})]}{r^2}\,\mathrm{d}r. \qquad (9.1)\end{aligned}$$

It remains only to compute $\mathbf{E}[W_{\text{PSJF}}(\text{remsize} \le r < \text{size})]$.

Let $\varphi_r(s,z,a) = (s - a \le r < s)$. That is, φ_r is true for states with remaining size less than r but original size greater than r. Our goal is to compute $\mathbf{E}[W(\varphi)]$. Recall from Definition 5.1 that $W(\varphi_r)$ is the sum of each individual job's φ_r-work. We can therefore use a generalization of Little's law [2, 6] to express the average total amount of φ_r-work, namely $\mathbf{E}[W(\varphi_r)]$, as the arrival rate λ times the average cumulative amount of φ_r-work a job contributes over the course of its time in the system. Specifically, consider a random job with true size S, estimated size Z, response time T, and age $A(t)$ after being in the system for time $t \in [0,T)$.[9] Then by the generalization of Little's law, we can write the mean φ_r-work as

$$\mathbf{E}[W_{\text{PSJF}}(\varphi_r)] = \lambda\mathbf{E}\left[\int_0^T w(\varphi_r, (S,Z,A(t)))\,\mathrm{d}t\right], \qquad (9.2)$$

We now compute the expectation of the right-hand side of (9.2), which concerns a single job's time in the system. Because $\varphi_r(s,z,a)$ holds only if $a > 0$ and a job's age is 0 during its waiting time, we can restrict attention to residence time. Letting T^{wait} and T^{res} denote the random job's waiting and residence times, respectively, we have

$$\begin{aligned}\mathbf{E}\left[\int_0^T w(\varphi_r, (S,Z,A(t)))\,\mathrm{d}t\right] &= \mathbf{E}\left[\int_0^{T^{\text{wait}}} w(\varphi_r, (S,Z,0))\,\mathrm{d}t + \int_{T^{\text{wait}}}^{T^{\text{wait}}+T^{\text{res}}} w(\varphi_r, (S,Z,A(t)))\,\mathrm{d}t\right] \\ &= \mathbf{E}\left[\int_{T^{\text{wait}}}^{T^{\text{wait}}+T^{\text{res}}} w(\varphi_r, (S,Z,A(t)))\,\mathrm{d}t\right]. \qquad (9.3)\end{aligned}$$

Under PSJF, a job's rank is always its size, and the load of jobs with rank better than s is $\rho_S(s)$. This means that during the residence time of a job of size s, it takes $\Delta/(1-\rho_S(s))$ time for a job's age to increase by Δ.[10] This means that for any function g and any size s,

$$\mathbf{E}\left[\int_{T^{\text{wait}}}^{T^{\text{wait}}+T^{\text{res}}} g(A(t))\,\mathrm{d}t \;\middle|\; S = s\right] = \frac{1}{1-\rho_S(s)}\int_0^s g(a)\,\mathrm{d}a. \qquad (9.4)$$

[9]The random variables S, Z, T, and $A(t)$ refer to the same job, so they are not independent. In addition, the response time T and age $A(t)$ depend on PSJF scheduling. The same applies to T^{wait} and T^{res}, which we introduce shortly.

[10]This $1/(1-\rho_S(s))$ factor under PSJF is similar to the $1/(1-\rho_Z(z))$ factor that appears under PSJF-E, as discussed in Remark 5.5. One can use the concept of *busy periods* from M/G/1 scheduling theory to formalize this [5, 19].

From this, we compute

$$\begin{aligned}
&\mathbf{E}\left[\int_0^T w(\varphi_r, (S, Z, A(t)))\,\mathrm{d}t\right] \\
&\quad= \mathbf{E}\left[\int_{T^{\text{wait}}}^{T^{\text{wait}}+T^{\text{res}}} w(\varphi_r, (S, Z, A(t)))\,\mathrm{d}t\right] && \text{[(9.3)]} \\
&\quad= \int_0^\infty \mathbf{E}\left[\int_{T^{\text{wait}}}^{T^{\text{wait}}+T^{\text{res}}} w(\varphi_r, (s, Z, A(t)))\,\mathrm{d}t \;\middle|\; S = s\right] f_S(s)\,\mathrm{d}s && \text{[conditioning on } S\text{]} \\
&\quad= \int_0^\infty \mathbf{E}\left[\int_{T^{\text{wait}}}^{T^{\text{wait}}+T^{\text{res}}} (s - A(t))\mathbb{1}(s - A(t) \le r < s)\,\mathrm{d}t \;\middle|\; S = s\right] f_S(s)\,\mathrm{d}s && \text{[expanding } \varphi_r\text{]} \\
&\quad= \int_0^\infty \int_0^s (s - a)\mathbb{1}(s - a \le r < s)\,\mathrm{d}a\,\frac{f_S(s)}{1 - \rho_S(s)}\,\mathrm{d}s && \text{[(9.4)]} \\
&\quad= \frac{r^2}{2}\int_r^\infty \frac{f_S(s)}{1 - \rho_S(s)}\,\mathrm{d}s. && (9.5)
\end{aligned}$$

Finally, we use (9.2) to plug this back into (9.1), obtaining

$$\begin{aligned}
\frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W_{\text{PSJF}}(\text{remsize} \le r < \text{size})]}{r^2}\,\mathrm{d}r &= \frac{1}{2}\int_0^\infty \int_r^\infty \frac{f_S(s)}{1 - \rho_S(s)}\,\mathrm{d}s\,\mathrm{d}r && \text{[(9.2) and (9.5)]} \\
&= \frac{1}{2}\int_0^\infty \frac{s f_S(s)}{1 - \rho_S(s)}\,\mathrm{d}s && \text{[swapping integrals]} \\
&= \frac{1}{2\lambda}\ln\frac{1}{1 - \rho}. && \text{[Lemma 5.6]}
\end{aligned}$$

□

Lemma 9.3. *The mean response time of PSJF-E is bounded by*

$$\mathbf{E}[T_{\text{PSJF-E}}] \le \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W_{\text{PSJF-E}}(\text{size-e} \le \alpha r)]}{r^2}\,\mathrm{d}r + \frac{\alpha}{\beta}\frac{1}{2\lambda}\ln\frac{1}{1 - \rho}.$$

Proof. Applying Proposition 5.3 and using the fact that $\text{remsize-e}(x)/\text{remsize}(x) \le \alpha$ yields

$$\begin{aligned}
\mathbf{E}[T_{\text{PSJF-E}}] &= \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W_{\text{PSJF-E}}(\text{remsize} \le r)]}{r^2}\,\mathrm{d}r \\
&\le \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W_{\text{PSJF-E}}(\text{remsize-e} \le \alpha r)]}{r^2}\,\mathrm{d}r \\
&= \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W_{\text{PSJF-E}}(\text{size-e} \le \alpha r)]}{r^2}\,\mathrm{d}r + \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W_{\text{PSJF-E}}(\text{remsize-e} \le r < \text{size-e})]}{r^2}\,\mathrm{d}r. \qquad (9.6)
\end{aligned}$$

It remains only to bound $\mathbf{E}[W_{\text{PSJF-E}}(\text{remsize-e} \le \alpha r < \text{size-e})]$.

The first part of this computation is similar to part of the proof of Lemma 9.2. Specifically, under PSJF-E, a job's rank is always its estimated size, so analogues of (9.3)–(9.5) hold for PSJF-E. The main difference is that in (9.4), we condition on a job having both size s and estimated size z, and we use $\rho_Z(z)$ instead of $\rho_S(s)$. The end result is

$$\begin{aligned}
&\frac{1}{\lambda}\mathbf{E}[W_{\text{PSJF-E}}(\text{remsize-e} \le \alpha r < \text{size-e})] \\
&\qquad= \int_0^\infty \int_0^\infty \int_0^s (s - a)\mathbb{1}\left(\frac{z}{s}(s - a) \le \alpha r < z\right)\mathrm{d}a\,\frac{f_{S,Z}(s, z)}{1 - \rho_Z(z)}\,\mathrm{d}s\,\mathrm{d}z.
\end{aligned}$$

Simplifying the right-hand side yields

$$\begin{aligned}
&\frac{1}{\lambda}\mathbf{E}[W_{\text{PSJF-E}}(\text{remsize-e} \le \alpha r < \text{size-e})] \\
&\quad= \int_0^\infty \int_0^\infty \int_0^s q \mathbb{1}\left(\frac{z}{s}q \le \alpha r < z\right) \mathrm{d}q \, \frac{f_{S,Z}(s,z)}{1-\rho_Z(z)} \, \mathrm{d}s \, \mathrm{d}z && \text{[setting } q = s - a\text{]} \\
&\quad= \int_{\alpha r}^\infty \int_0^\infty \frac{\frac{1}{2}\left(\frac{\alpha r s}{z}\right)^2}{1-\rho_Z(z)} \mathrm{d}q \, f_{S,Z}(s,z) \, \mathrm{d}s \, \mathrm{d}z \\
&\quad= \int_{\alpha r}^\infty \frac{\frac{1}{2}\mathbf{E}\left[\left(\frac{\alpha r S}{z}\right)^2 \mid Z = z\right] f_Z(z)}{1-\rho_Z(z)} \mathrm{d}z \\
&\quad= \alpha \frac{r^2}{2} \int_{\alpha r}^\infty \frac{\frac{\alpha}{z}\mathbf{E}\left[\left(\frac{S}{z}\right) S \mid Z = z\right] f_Z(z)}{1-\rho_Z(z)} \mathrm{d}z. && (9.7)
\end{aligned}$$

Finally, we plug this back into (9.6), obtaining

$$\begin{aligned}
&\frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W_{\text{PSJF-E}}(\text{remsize-e} \le \alpha r < \text{size-e})]}{r^2} \mathrm{d}r \\
&\quad= \alpha \frac{1}{2} \int_0^\infty \int_{\alpha r}^\infty \frac{\frac{\alpha}{z}\mathbf{E}\left[\left(\frac{S}{z}\right) S \mid Z = z\right] f_Z(z)}{1-\rho_Z(z)} \mathrm{d}z \, \mathrm{d}r && \text{[(9.7)]} \\
&\quad= \alpha \frac{1}{2} \int_0^\infty \frac{\mathbf{E}\left[\left(\frac{S}{z}\right) S \mid Z = z\right] f_Z(z)}{1-\rho_Z(z)} \mathrm{d}z && \text{[swapping integrals]} \\
&\quad\le \frac{\alpha}{\beta}\frac{1}{2} \int_0^\infty \frac{\mathbf{E}[S \mid Z = z] f_Z(z)}{1-\rho_Z(z)} \mathrm{d}z && \text{[using } Z/S \ge \beta\text{]} \\
&\quad= \frac{\alpha}{\beta}\frac{1}{2\lambda} \ln \frac{1}{1-\rho}. && \text{[Lemma 5.6]} \qquad \square
\end{aligned}$$

Proof of Theorem 9.1. A result of Wierman et al. [20, Theorem 5.1] states

$$\mathbf{E}[T_{\text{PSJF}}] \le \frac{3}{2}\mathbf{E}[T_{\text{SRPT}}],$$

so it suffices to prove (a). Lemma 9.2 expresses $\mathbf{E}[T_{\text{PSJF}}]$ as a sum of two terms, and Lemma 9.3 bounds $\mathbf{E}[T_{\text{PSJF-E}}]$ by a sum of two similar terms. The ratio of the second terms is α/β. To bound the ratio of the first terms by α/β, we proceed similarly to the proof of Theorem 7.2:

$$\begin{aligned}
&\frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W_{\text{PSJF-E}}(\text{size-e} \le \alpha r)]}{r^2} \mathrm{d}r \\
&\quad\le \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W_{\text{PSJF}}(\text{size-e} \le \alpha r)]}{r^2} \mathrm{d}r && \text{[Proposition 7.1]} \\
&\quad\le \frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}\left[W_{\text{PSJF}}\left(\text{size} \le \frac{\alpha}{\beta} r\right)\right]}{r^2} \mathrm{d}r && \text{[using size-e}(x)/\text{size}(x) \ge \beta\text{]} \\
&\quad= \frac{\alpha}{\beta}\frac{1}{\lambda}\int_0^\infty \frac{\mathbf{E}[W_{\text{PSJF}}(\text{size} \le r')]}{(r')^2} \mathrm{d}r'. && \text{[setting } r' = \alpha/\beta \cdot r\text{]} \qquad \square
\end{aligned}$$

10 CONCLUSION

We have examined scheduling policies in the context of estimated sizes. Our main result is to resolve an issue previously seen empirically, namely that the performance of SRPT-E can degrade significantly due to long jobs being underestimated, by developing and analyzing a novel policy,

SRPT-B, which combines the best aspects of SRPT-E and PSJF-E. In analyzing SRPT-B, we have demonstrated that is has three key properties for (β, α)-bounded estimates: (a) an approximation ratio near 1 when α and β are near 1, (b) an approximation ratio bounded by some function of α and β, and (c) implementation without knowledge of α and β. We have also shown that PSJF-E also has properties (b) and (c), and has an approximation ratio near 1 relative to PSJF when α and β are near 1. We have also shown that the empirical observation of the poor performance of SRPT-E can be characterized through a lower bound.

For practical settings, our results provide theoretical backing for previous empirical findings that PSJF-E performs very well with estimated job sizes. While SRPT-B provides an additional promising alternative that will sometimes perform better in practice, we recommend PSJF-E as a simple, natural scheduling algorithm that seems to generally perform the best or near the best among standard alternatives.

Our work leaves several open directions, including considering other estimation models, optimizing the behavior of the bounce in SRPT-B, and improving the bounds. For instance, another important estimation model is a model where estimates are typically good, but not guaranteed to be good, such as Gaussian errors. One might try to adapt our results to that setting by bounding the worst expected error over a given interval of time. We also note it may be possible to tighten the bound on the ratio between PSJF and SRPT, which may correspondingly tighten the bound between PSJF-E and SRPT.

REFERENCES

[1] Yossi Azar, Stefano Leonardi, and Noam Touitou. 2021. Flow Time Scheduling with Uncertain Processing Time. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC 2021)*. ACM, Rome, Italy, 1070–1080. https://doi.org/10.1145/3406325.3451023

[2] Shelby L. Brumelle. 1971. On the Relation between Customer and Time Averages in Queues. *Journal of Applied Probability* 8, 3 (1971), 508–520. https://doi.org/10.2307/3212174

[3] Matteo Dell'Amico, Damiano Carra, and Pietro Michiardi. 2016. PSBS: Practical Size-Based Scheduling. *IEEE Trans. Comput.* 65, 7 (July 2016), 2199–2212. https://doi.org/10.1109/TC.2015.2468225

[4] John C. Gittins. 1979. Bandit Processes and Dynamic Allocation Indices. *Journal of the Royal Statistical Society: Series B (Methodological)* 41, 2 (Jan. 1979), 148–164. https://doi.org/10.1111/j.2517-6161.1979.tb01068.x

[5] Mor Harchol-Balter. 2013. *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*. Cambridge University Press, Cambridge, UK.

[6] Daniel P. Heyman and Shaler Stidham. 1980. The Relation between Customer and Time Averages in Queues. *Operations Research* 28, 4 (Aug. 1980), 983–994. https://doi.org/10.1287/opre.28.4.983

[7] Thodoris Lykouris and Sergei Vassilvitskii. 2021. Competitive Caching with Machine Learned Advice. *J. ACM* 68, 4 (2021), 24:1–24:25. https://doi.org/10.1145/3447579

[8] Michael Mitzenmacher. [n.d.]. *Queues with Small Advice*. 1–12. https://doi.org/10.1137/1.9781611976830.1 arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611976830.1

[9] Michael Mitzenmacher. 2020. Scheduling with Predictions and the Price of Misprediction. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020) (Leibniz International Proceedings in Informatics (LIPIcs))*, Thomas Vidick (Ed.), Vol. 151. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, Article 14, 18 pages. https://doi.org/10.4230/LIPICS.ITCS.2020.14

[10] Michael Mitzenmacher and Matteo Dell'Amico. 2020. The Supermarket Model with Known and Predicted Service Times. *arXiv:1905.12155 [cs]* (Oct. 2020). http://arxiv.org/abs/1905.12155

[11] Michael Mitzenmacher and Sergei Vassilvitskii. 2020. Algorithms with Predictions. In *Beyond the Worst-Case Analysis of Algorithms*, Tim Roughgarden (Ed.). Cambridge University Press, 646–662. https://doi.org/10.1017/9781108637435.037

[12] Manish Purohit, Zoya Svitkina, and Ravi Kumar. 2018. Improving Online Algorithms via ML Predictions. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS 2018)*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc., Montréal, Canada, 9684–9693. https://proceedings.neurips.cc/paper/2018/file/73a427badebe0e32caa2e1fc7530b7f3-Paper.pdf

[13] Linus E. Schrage. 1968. A Proof of the Optimality of the Shortest Remaining Processing Time Discipline. *Operations Research* 16, 3 (June 1968), 687–690. https://doi.org/10.1287/opre.16.3.687

[14] Linus E. Schrage and Louis W. Miller. 1966. The Queue M/G/1 with the Shortest Remaining Processing Time Discipline. *Operations Research* 14, 4 (Aug. 1966), 670–684. https://doi.org/10.1287/opre.14.4.670

[15] Ziv Scully, Isaac Grosof, and Mor Harchol-Balter. 2020. The Gittins Policy Is Nearly Optimal in the M/G/k under Extremely General Conditions. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 4, 3, Article 43 (Nov. 2020), 29 pages. https://doi.org/10.1145/3428328

[16] Ziv Scully, Isaac Grosof, and Mor Harchol-Balter. 2020. Optimal Multiserver Scheduling with Unknown Job Sizes in Heavy Traffic. *ACM SIGMETRICS Performance Evaluation Review* 48, 2 (Nov. 2020), 33–35. https://doi.org/10.1145/3439602.3439615

[17] Ziv Scully and Mor Harchol-Balter. 2018. SOAP Bubbles: Robust Scheduling under Adversarial Noise. In *56th Annual Allerton Conference on Communication, Control, and Computing*. IEEE, Monticello, IL, 144–154. https://doi.org/10.1109/ALLERTON.2018.8635963

[18] Ziv Scully and Mor Harchol-Balter. 2021. The Gittins Policy in the M/G/1 Queue. In *18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt 2020)*. IEEE, Philadelphia, PA.

[19] Ziv Scully, Mor Harchol-Balter, and Alan Scheller-Wolf. 2018. SOAP: One Clean Analysis of All Age-Based Scheduling Policies. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2, 1, Article 16 (April 2018), 30 pages. https://doi.org/10.1145/3179419

[20] Adam Wierman, Mor Harchol-Balter, and Takayuki Osogami. 2005. Nearly Insensitive Bounds on SMART Scheduling. In *Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 2005)*. ACM, Banff, Alberta, Canada, 12. https://doi.org/10.1145/1064212.1064236

[21] Adam Wierman and Misja Nuyens. 2008. Scheduling despite Inexact Job-Size Information. In *Proceedings of the 2008 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 2008)*. ACM, Annapolis, MD, 25–36. https://doi.org/10.1145/1375457.1375461

A HANDLING RANK FUNCTION TIES

One might ask how to handle two jobs have the same rank under rank-based policies. For example, it is possible that two jobs of the same size arrive and await service; when service becomes available, should one be given priority, or a processor-sharing based policy be used? We avoid such questions by assuming continuous distributions for size and estimated size, so almost surely no two jobs arrive with the same values for these quantities.

If one considers a discrete job size distribution, then PSJF-E in the presence of arbitrarily small errors is equivalent to PSJF with random priority tiebreaking, which can have significantly worse mean response time than a standard tiebreaking rule such as First-Come-First-Served. This issue is sidestepped by our focus on continuous distributions.

Alternatively, ties could occur if a job being served while its rank was increasing reached the rank of another job which also would have increasing rank if it was served. Whichever job is served immediately loses minimum rank status, resulting in a processor-sharing effect. The tiebreaking policy is therefore irrelevant. The only scheme analyzed in this paper with increasing rank, SRPT-B, avoids this scenario by capping the rank at the initial size estimate. Under SRPT-B, a job being served with increasing rank can only be preempted by a newly arriving job, which will finish before the preempted job can continue.

B UNACHIEVABILITY OF TRADITIONAL ROBUSTNESS

In the context of online algorithms with predictions [7, 11], one typically tries to achieve constant-factor robustness, which requires that the approximation ratio is bounded by a constant under arbitrarily poor predictions.

In the context of M/G/1 scheduling with predictions, constant-factor robustness is provably unachievable. This follows from the fact that the mean response time of an algorithm with arbitrarily poor predictions is bounded below by the optimal blind policy, that has no predictions at all.

In the context of M/G/1 scheduling, the optimal blind policy is known: the Gittins Index policy [4, 18] is optimal in this context, though it requires knowledge of the job size distribution.

The approximation ratio of the Gittins index policy to the SRPT policy can be arbitrarily poor in the limit as $\rho \to 1$ [16]. Specifically, whenever the job size distribution $S \in \mathrm{MDA}(\Lambda)$, the Gumbel domain of attraction, the approximation ratio of Gittins to SRPT grows arbitrarily large as $\rho \to 1$. Loosely, we can think of the Gumbel domain of attraction as including all light-tailed, unbounded job size distributions.

In all such cases, constant-factor robustness is unachievable. A better goal might be to compare the performance of a policy against that of the Gittins policy. We leave that question for future work.

C POOR PERFORMANCE FOR SRPT-B WITHOUT THE RANK CAP

We have discussed that for SRPT-B capping the bounce at initial estimated size of a job is important in our analysis. We here explain why our results would not hold without such a cap.

Under SRPT-B, if a job begins service, it can only leave service by being preempted by newly arriving jobs, not by having its rank rise above the rank of other jobs in the queue. In contrast, under a policy *SRPT with Unlimited Bounce* (SRPT-UB) with rank function $|z - a|$, without a cap at z, the situation would be very different.

We consider a case where all jobs have nearly the same size (S is nearly constant) and $\beta = 1/2 - \varepsilon$. For specificity, let each job have size $s \in [1, 1 + \delta]$, for $\delta \ll \varepsilon$. Let each job's estimate $z = \beta s$.

Under SRPT-UB, consider a job j of size 1. It begins service with rank $\beta = 1/2 - \varepsilon$, descends to rank 0, and rises back to a rank of $1/2 - \varepsilon$ at age $1 - \varepsilon$. At this age, or in the next δ service, job j's rank will rise high enough that it will be preempted by any fresh job (that has yet to receive service), whether that job arrived before or after job j. As a result, job j will have to wait until there are no more fresh jobs to complete. Approximately, job j must wait until the system empties to complete. From standard results in queueing theory, the mean response time under SRPT-UB is therefore $\Theta\left(\frac{1}{(1-\rho)^2}\right)$. In contrast, a better policy such as SRPT or SRPT-B has mean response time that grows as $\Theta\left(\frac{1}{1-\rho}\right)$. The gap between these two response times grows arbitrarily wide as $\rho \to 1$.

D SRPT WITH CONTINUOUSLY UPDATING ESTIMATES (SRPT-CUE)

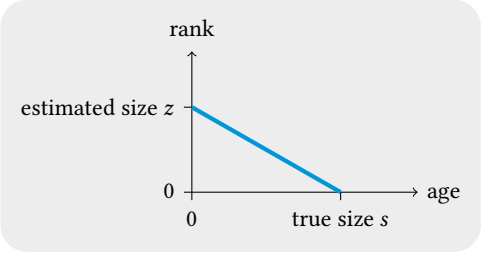
The SRPT-SE scheme requires knowledge of the job's true size, and hence while it is useful in our analysis, it is not a practical policy as we have defined it. However, there may be settings where a variation of SRPT-SE is implementable. We now describe such a setting, the variant of SRPT-SE that can be implemented in it, and how to extend our bounds in Theorem 7.2 to cover the new variant.

Suppose that jobs are given size estimates not just when they initially arrive but also continuously during service. More specifically, suppose that once a job of true size s reaches each age a, it is given an estimated remaining size in the interval $[\beta(s - a), \alpha(s - a)]$. This could model settings where there is some visible metric of job progress but the speed at which progress is made is uncertain, such as sending files of known size to clients with unknown packet loss rates.

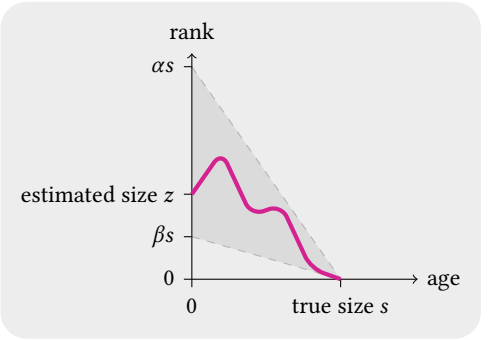
In this setting, a natural policy is one we call *SRPT with Continuously Updating Estimates* (SRPT-CUE), which at every moment in time serves the job with the smallest remaining size estimate. The difference between SRPT-SE and SRPT-CUE is that while each job's rank decreases linearly under SRPT-SE, a job's rank may follow a more complicated path under SRPT-CUE, though it will stay in the interval $[\beta(s - a), \alpha(s - a)]$. We illustrate the difference in Fig. D.1.

The proof of Theorem 7.2 can be modified to give the same result for SRPT-CUE as for SRPT-SE, namely

$$\mathbf{E}[T_{\text{SRPT-CUE}}] \le \frac{\alpha}{\beta}\mathbf{E}[T_{\text{SRPT}}]. \tag{D.1}$$



(a) SRPT with Scaling Estimates (SRPT-SE)



(b) SRPT with Continuously Updating Estimates (SRPT-CUE)

Fig. D.1. Comparison Between SRPT-SE and SRPT-CUE

The two main steps are (a) formalizing the definition of SRPT-CUE using a rank function and (b) showing that Proposition 7.1 holds for SRPT-CUE. The difficulty of these steps depends on the details of the estimation error model. For example, if the estimated remaining size functions for each job are sampled i.i.d. from some function distribution, then (a) can be done using methods of Scully et al. [19], and (b) follows for the same reasons as the policies we consider.

It may be possible to handle even adversarial estimation errors, provided they stay (β, α)-bounded, by using the methods of Scully and Harchol-Balter [17]. In this model, (a) is done by assigning each job state a rank *interval* instead of a single rank, from which the job's rank may be adversarially chosen. However, (b) may require placing some limit on the adversary's power, such as making them oblivious to the system state or the scheduling policy.

We note that for (D.1) to hold, it is important that under SRPT-CUE, a job's rank changes only while that job is in service. For example, if a job's rank can change while it is not in service, then small fluctuations in rank might cause the system to split its effort between two jobs of similar remaining size, which is worse than serving one of the jobs before the other. See Scully and Harchol-Balter [17, proof of Theorem 6] for further discussion.