Bounds on M/G/k Scheduling Under Moderate Load

Improving on SRPT-k and Tightening Lower Bounds

Isaac Grosof
University of Illinois, Urbana-Champaign
Urbana, IL, USA
igrosof@illinois.edu
isaac.grosof@northwestern.edu

Ziyuan Wang
Northwestern University
Evanston, IL, USA
ziyuanwang2027@u.northwestern.edu

ABSTRACT

A well-designed scheduling policy can significantly improve the performance of a queueing system, without requiring any additional resources. While scheduling is well-understood in the single-server setting, much less is known in the multiserver setting. Results are particularly sparse in moderate load settings, outside of the asymptotic regimes of heavy traffic and light traffic. Multiserver SRPT is known to achieve asymptotically optimal mean response time as in the limit as load approaches capacity, and no better policy is known outside of that asymptotic regime.

We give the first family of multiserver scheduling policies to achieve lower mean response time than SRPT, by delaying the service of small jobs to improve overall server utilization.

In light of this improvement, we seek to prove tighter lower bounds on mean response time in the M/G/k. We introduce the WINE lower-bounding framework, allowing multiple lower bounds to be combined into a single, stronger lower bound. Moreover, we introduce and analyze the Increasing Speed Queue, which captures the variable-service-speed nature of the M/G/k system, and use it to further strengthen our lower bound on M/G/k scheduling.

1 Introduction

Scheduling is an important tool for improving the performance of queueing systems, both in theory and in practice. A well-chosen scheduling policy can improve system performance without requiring increased resources.

Scheduling is well-understood in the single-server setting, with the Shortest Remaining Processing Time (SRPT) policy known to minimize mean response time when job sizes (durations) are known in advance [8].

Modern queueing systems are more often multiserver, having the capacity to serve several jobs at a time. Much less is known about scheduling in multiserver systems, such as the M/G/k, even for just $k = 2$ servers. This especially true under *moderate* load, where the system is neither inundated with jobs, nor working well below capacity. Asymptotic results are known in the heavy traffic limit ($\rho \to 1$) where ρ, the load, is the long-run average fraction of servers which are busy. SRPT-k, the multiserver equivalent of SRPT, achieves asymptotically optimal mean response time when job sizes are known [4]. Scheduling becomes irrelevant in the light-traffic limit ($\rho \to 0$).

But under *moderate* load, little is known about optimal scheduling in the M/G/k [2]. This is unfortunate, because moderate load is the regime of most importance in practice. Operating a queueing system is always a tradeoff between using resources efficiently and keeping queueing times low. Balancing this tradeoff places the system in moderate load. Scheduling can be a win-win for this tradeoff, but to understand how much improvement is possible we need to understand optimal scheduling under moderate load.

Observing SRPT-k's asymptotically optimal performance in heavy traffic provokes a natural question:

> Do there exist M/G/k scheduling policies with lower mean response time than SRPT-k under moderate load?

This question is open: No policies which outperform SRPT-k have preciously been presented, at any load, for any job size distribution. We present the SRPT-Except-$k+1$ (SEK) policy, which we empirically demonstrate in is the first policy to achieve lower mean response time than SRPT-k under moderate load (see Section 2).

Our result, showing that there is room for improvement beyond SRPT-k, provokes another natural question:

> How much improvement beyond SRPT-k is possible under moderate load?

Previously, only naive lower bounds on mean response time for M/G/k scheduling have appeared in the literature. A straightforward bound is the mean service time. This bound is tight at low load, but does not increase with increasing load, making it loose at moderate load. Another standard bound is the resource-pooled bound, where the M/G/k is compared against an M/G/1 system with a single giant server, as fast as all k servers put together, scheduling according to the single-server-optimal SRPT policy [4]. This bound is tight at high load, but does not increase with the number of servers k, making it loose at moderate load.

In Section 3, we introduce the WINE framework for lower bounding mean response time in the M/G/k. WINE is an existing technique for relating mean response time to mean relevant work, but it has only ever been used to upper-bound mean response time in the past [6,7,10]. We initiate its use in lower-bounding mean response time. This framework allows us to combine bounds such as the service-time bound and resource-pooled-SRPT bound to prove new, stronger lower bounds on M/G/k mean response time under an arbitrary scheduling policy.

MAMA 2024 Venice, Italy

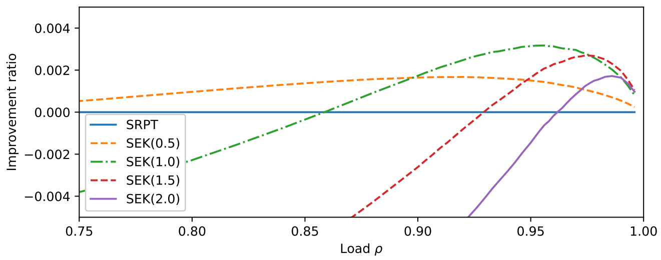

Figure 1: Improvement ratio $1 - \frac{E[T^{SEK}]}{E[T^{SRPT}]}$ between the SEK policy and the SRPT-k policy, in an M/M/2 with $S = Exp(1)$, for several cutoff parameters c and over loads $\rho \in [0.75, 0.996]$. Simulated 10^8 arrivals on coupled arrival processes.

However, to close the gap towards optimality, we need to go beyond the existing sources of lower bounds on mean relevant work. Neither resource-pooled bound nor the service-time bound capture the fundamental nature of work completion in the M/G/k system: Moving incrementally between no servers active, one server, two servers, up to k servers active. To capture this behavior, we define and analyze the Increasing Speed Queue (ISQ) in Section 4, a novel queueing system which lower bounds relevant work in the M/G/k. Our analysis employs a novel application of the drift method [1], which may be of independent interest. Using our ISQ analysis, we obtain the tightest-known lower bounds on mean response time under moderate load.

2 SRPT-Except-$k+1$

We introduce a new scheduling policy for the M/G/k system, the *SRPT-Except-*$k+1$ (SEK) policy [3, Section 8.3.1].

If there are k or fewer jobs in the system, all are served. If there are $k+2$ or more jobs in the system, SEK matches SRPT-k: Serve the k jobs of least remaining size.

SEK can diverge from SRPT-k when there are $k+1$ jobs in the system. SEK is parameterized by a switching parameter c. If there are k jobs in the system with remaining size $\leq c$, and the final job has remaining size $\geq c$, SEK serves the $k-1$ jobs with least remaining size, as well as the job with largest remaining size. Otherwise, SEK matches SRPT-k.

The intuition behind the SEK policy is that when SEK diverges from SRPT-k, SRPT-k is about to waste service capacity, if no job arrives in the next c time. By running the job with largest remaining size, the wasted capacity is diminished, or at least delayed.

As shown in Fig. 1, in the M/M/2 setting, the SEK policy outperforms the SRPT-k policy by a small but consistent margin, over a range of loads ρ and a range of parameter values c. The largest improvement observed in this dataset is a margin of 0.32%, achieved by $SEK(c=1)$ at load 0.956. For each parameter value c, SEK empirically improves upon SRPT for all loads ρ above some threshold depending on c.

In further empirical investigation, we have observed larger improvement margins under higher-variance workloads, up to about a 1% improvement. In subsequent work, we hope to characterize the combinations of load ρ, parameter c, and size distribution S for which SEK can be guaranteed to improve upon SRPT.

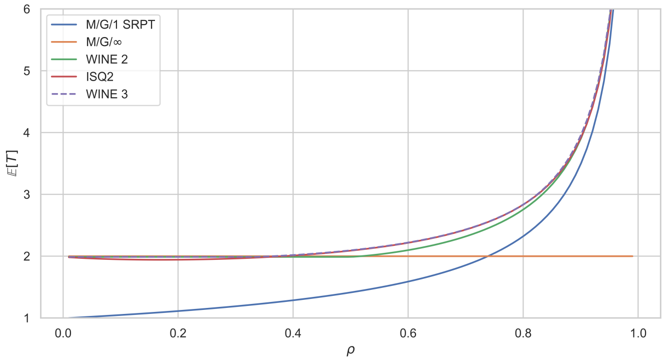

Figure 2: Comparison of lower bounds on optimal mean response time for scheduling policies in an M/M/2 with $S = Exp(1)$. Bounds: resource-pooled M/G/1 SRPT, service time (M/G/∞), the WINE combination of those two, the Increasing Speed Queue, and the WINE combination of all three.

3 Lower Bounds: WINE Framework

In Section 2, the SEK policy was empirically shown to achieve lower mean response time than SRPT-k, at certain loads. But how much more improvement is possible? We seek lower bounds on M/G/k mean response time at moderate load.

Two straightforward bounds are the *service-time* bound and the *resource-pooled M/G/1* bound. First, we can lower bound response time by service time. We adopt the convention that the M/G/k servers run at speed $1/k$, so a job of size s requires ks time in service. As a result, $E[T^\pi] \geq kE[S]$, for any M/G/k policy π, where S is the job size distribution. This bound is tight at low load ($\rho \to 0$). We refer to this bound as the M/G/∞ bound, because in an infinite-server system with servers that run at speed $1/k$, all job's response times would be $kE[S]$. In Fig. 2, we show this bound for the M/M/2 system as the orange line.

Throughout this section, we use the M/M/2 as an example, though our bounds are generic over all job size distributions. Note that job sizes are known to the scheduling policy in advance, so jobs are not interchangeable.

The resource-pooled bound compares the M/G/k against a resource-pooled M/G/1 where the single server has speed 1, the same as the total speed as all k servers in the M/G/k. The resource-pooled M/G/1 has a superset of the options available as are available in the M/G/k. The policy which minimizes mean response time in the M/G/1 is the single-server SRPT policy, so we know that $E[T^\pi] \geq E[T^{SRPT\text{-}1}]$ This bound is tight in heavy traffic ($\rho \to 1$), and is at the heart of the heavy-traffic optimality proof of SRPT-k and several other policies in more general settings [4–6]. In Fig. 2, we show this bound as the blue curve.

Unfortunately, both of these bounds are loose under moderate load. The service-time bound does not increase with load, while the resource-pooled M/G/1 bound does not increase with the number of servers k. To achieve a strong lower bound under moderate load, we need a framework that can incorporate both load and number of servers.

We introduce the WINE framework for lower-bounding mean response time in the M/G/k. The recently-developed WINE formula characterizes of mean response time under an arbitrary policy in terms of mean relevant work under that policy [9, 10]:

$$E[T^\pi] = \frac{1}{\lambda} \int_{x=0}^{\infty} \frac{E[W_x^\pi]}{x^2}.$$

In past work, WINE has been used to upper-bound mean response time by upper-bounding mean relevant work under a specific policy [6, 7, 10]. In contrast, we use WINE to lower-bound mean response time under an arbitrary policy by lower-bounding mean relevant work under an arbitrary policy. Essentially, WINE provides a way to combine different methods of lower-bounding mean relevant work into a combined lower bound on mean response time.

We give three ways of lower-bounding mean relevant work: in Section 3.1, using the service-time lower bound and using the resource-pooled M/G/1 lower bound, and in Section 4 using a novel approach, which we call the Increasing Speed Queue. By taking the maximum of these bounds at each relevancy cutoff x, we can achieve a significantly stronger lower bound on response time than with past approaches.

3.1 Relevant-work Lower Bounds

For the service-time lower bound, consider the size y at which a job first becomes relevant, either upon arrival or after receiving service to bring its size down to x. The job will spend ky time in service after that point, contributing a total of $ky^2/2$ relevant work over its time in the system:

$$E[W_x^\pi] \geq \frac{k\lambda}{2} E[\min(S, x)^2].$$

For the resource-pooled M/G/1 lower bound, we lower bound relevant work in the M/G/k by relevant work in the SRPT-1 system, which minimizes relevant work for each relevancy cutoff x over all single-server policies:

$$E[W_x^\pi] \geq \frac{\lambda}{2} \frac{E[\min(S, x)^2]}{1 - \rho_x}, \text{ where } \rho_x = \lambda E[S \mathbb{1}\{S \leq x\}].$$

Simply taking the maximum of these bounds for each x and applying WINE gives new, stronger bounds on response time under moderate load. In Fig. 2, we show this bound for the M/M/2 system as the green curve labeled "WINE 2". Note that the green curve significantly improves upon the prior blue and orange curves for loads $\rho > 0.5$.

4 Lower Bounds: Increasing Speed Queue

The increasing-speed queue (ISQ) is a single-server variable-speed queue. When the first job arrives to an empty queue, the server initially runs at speed $1/k$. If another job arrives before the system empties, the server now runs at speed $2/k$. With each subsequent arrival, the server's speed increases, until it reaches speed 1. The server will then stay at speed 1 until the system empties.

The total work in the ISQ system is a lower bound on work in the M/G/k, because the M/G/k cannot serve more jobs than have arrived since the system was last empty, nor can it have total service rate higher than 1.

A lower bound on relevant work in the M/G/k can be derived by considering an ISQ system whose job size distribution is truncated at size x. More sophisticated, tighter lower bounds can be derived by also incorporating the jobs with initial size larger than x into the analysis.

Thus, if we can analyze the ISQ system, we can prove tighter lower bounds on mean response time in the M/G/k.

To analyze the ISQ system, we use a novel variant of the *drift method* [1]. By carefully crafting a test function to have appropriate drift, we can characterize mean work in the ISQ system. In the $k = 2$ system, we use the following test function $f(w, i)$, where w is work and i is system speed:

$$f(w, 1) = w^2, \quad f(0, 0) = 0,$$
$$f(w, 1/2) = w^2 + \frac{w}{\lambda} + \frac{1 - e^{-2w\lambda}}{2\lambda^2}.$$

We thereby derive the following formula for mean work:

$$E[W^{ISQ\text{-}2}] = \frac{\lambda E[S^2]}{2(1 - \lambda E[S])} + \frac{E[S] - (1 - \widetilde{S}(2\lambda))/2\lambda}{3 - \widetilde{S}(2\lambda)},$$

where $\widetilde{S}$ is the Laplace-Stieljes transform of the job size distribution S. Similar results hold for all k.

From this analysis, we can lower bound mean relevant work in the M/G/k. By integrating this bound using the WINE framework, we can directly lower bound mean response time in the M/G/k. In Fig. 2, we show this bound for the M/M/2 system as the red curve labeled "ISQ-2". This bound is appreciably stronger than the green WINE-2 curve, with up to a 6% improvement.

By combining all three lower bounds using the WINE framework, we can improve further. In Fig. 2, we show this bound for the M/M/2 system as the dotted purple curve labeled "WINE 3". This bound slightly improves upon the red ISQ-2 curve, with up to a 2% improvement. The improvement is largest at low load, when the ISQ-2 bound falls below the service-time bound, and at moderately high load, where for some x the M/G/1/SRPT bound is stronger than the ISQ-2 bound. This magnitude of bound-strengthening is important, when compared to SEK's margin of improvement over SRPT-k.

5 References

[1] A. Eryilmaz and R. Srikant. Asymptotically tight steady-state queue length bounds implied by drift conditions. *Queueing Systems*, 72:311–359, 2012.

[2] I. Grosof. Open problem—M/G/k/SRPT under medium load. *Stochastic Systems*, 9(3):297–298, 2019.

[3] I. Grosof. *Optimal Scheduling in Multiserver Queues.* PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 2023.

[4] I. Grosof, Z. Scully, and M. Harchol-Balter. SRPT for multiserver systems. *Performance Evaluation*, 127-128:154–175, 2018.

[5] I. Grosof, Z. Scully, and M. Harchol-Balter. Load balancing guardrails: Keeping your heavy traffic on the road to low response times. *Proc. ACM Meas. Anal. Comput. Syst.*, 3(2), June 2019.

[6] I. Grosof, Z. Scully, M. Harchol-Balter, and A. Scheller-Wolf. Optimal scheduling in the multiserver-job model under heavy traffic. *Proc. ACM Meas. Anal. Comput. Syst.*, 6(3), Dec. 2022.

[7] Y. Hong and Z. Scully. Performance of the Gittins policy in the G/G/1 and G/G/k, with and without setup times. *Performance Evaluation*, 163:102377, 2024.

[8] L. Schrage. A proof of the optimality of the shortest remaining processing time discipline. *Operations Research*, 16(3):687–690, 1968.

[9] Z. Scully. *A New Toolbox for Scheduling Theory.* PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 2022.

[10] Z. Scully, I. Grosof, and M. Harchol-Balter. The Gittins policy is nearly optimal in the M/G/k under extremely general conditions. *Proc. ACM Meas. Anal. Comput. Syst.*, 4(3), Nov. 2020.