A Tale of Two Traffics: Optimizing Tail Latency in the Light-Tailed $M/G/k$

GEORGE YU, Cornell University, USA
AMIT HARLEV, Cornell University, USA
REEVU ADAKROY, Cornell University, USA
ZIV SCULLY, Cornell University, USA

We consider the problem of scheduling to minimize asymptotic tail latency in the M/G/k queue with light-tailed job size distribution. This problem combines the challenges of scheduling for tail latency and scheduling in multiserver queues, but there is hope. In the simpler setting of the single-server M/G/1, the recently proposed γ-*Boost* policy is tail constant optimal, and it has excellent empirical tail latency. And for the simpler objective of mean latency, it is known that the optimal policy in the M/G/1, namely SRPT (shortest remaining processing time), is also excellent in the M/G/k: it is provably optimal in heavy traffic and has state-of-the-art empirical performance in lighter traffic.

One might therefore hope that γ-Boost is similarly effective in the M/G/k, but our results paint a more complicated picture. In heavy traffic, we prove that γ-Boost is indeed tail constant optimal. We also prove an analogous result for scheduling with unknown sizes, where γ-Boost is replaced by its unknown-size counterpart. But in lighter traffic, we find empirically that γ-Boost can be even worse than FCFS (first-come, first-served). This is a significant shortcoming, as the boundary between "lighter" and "heavy" traffic occurs at higher load when the number of servers k is larger. To overcome this, we design a new variant of γ-Boost that outperforms the original by, counterintuitively, *giving more priority to larger jobs*. The new variant, which we prove is also heavy-traffic optimal, has state-of-the-art empirical tail latency at lighter loads, outperforming even a much more computationally intensive mixed-integer-programming heuristic.

CCS Concepts: • **General and reference** → **Performance**; • **Mathematics of computing** → **Queueing theory**; • **Networks** → **Network performance modeling**; • **Computing methodologies** → *Model development and analysis*; • **Software and its engineering** → *Scheduling*.

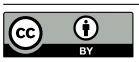
Additional Key Words and Phrases: scheduling; multiserver systems; response time; sojourn time; tail latency; service level objective (SLO); M/G/1 queue; M/G/k queue; light-tailed distribution; Gittins index; Boost scheduling; multi-armed bandit

ACM Reference Format:
George Yu, Amit Harlev, Reevu Adakroy, and Ziv Scully. 2025. A Tale of Two Traffics: Optimizing Tail Latency in the Light-Tailed M/G/k. *Proc. ACM Meas. Anal. Comput. Syst.* 9, 3, Article 46 (December 2025), 40 pages. https://doi.org/10.1145/3771561

1 Introduction

In today's large-scale computer systems, operators often care about service level objectives (SLOs) that relate to the tail of a system's response time distribution T, where a job's *response time* (a.k.a.

Authors' Contact Information: George Yu, School of Operations Research and Information Engineering, Cornell University, Ithaca, NY, USA; Amit Harlev, Center for Applied Mathematics, Cornell University, Ithaca, NY, USA; Reevu Adakroy, School of Operations Research and Information Engineering, College of Computing and Information Science, Cornell University, Ithaca, NY, USA; Ziv Scully, School of Operations Research and Information Engineering, Cornell University, Ithaca, NY, USA.



ACM 2476-1249/2025/12-ART46
https://doi.org/10.1145/3771561

sojourn time) is the total amount of time it spends in the system. In particular, SLOs often require that high percentiles of T are small, or, dually, that the tail of response time $\mathbf{P}[T > t]$ is small for large thresholds t. This work studies how to accomplish this via scheduling in the M/G/k multiserver queueing model, resulting in a new state-of-the-art scheduling policy, called *γ-CombinedBoost*, with good theoretical and empirical performance.

1.1 Recent Progress: Single-server Tail Scheduling and Multiserver Mean Scheduling

Recent work in the M/G/1 queueing model [7, 21, 25, 47, 49] has shown that when the job size distribution is light-tailed, aiming to asymptotically minimize $\mathbf{P}[T > t]$ in the $t \to \infty$ limit leads to policies with state-of-the-art $\mathbf{P}[T > t]$ performance at practical thresholds t. In particular, the *γ-Boost* policy introduced by Yu and Scully [49] achieves *tail constant optimality* [6, 48], meaning it achieves the best possible tail constant:

$$\inf_{\text{policies } \pi} \limsup_{t\to\infty} e^{\gamma t}\mathbf{P}[T_\pi > t] = \limsup_{t\to\infty} e^{\gamma t}\mathbf{P}[T_{\gamma\text{-Boost}} > t],$$

where $e^{\gamma t}$ is an appropriate scaling factor (details in Section 2.3). Empirically, optimizing this constant can lead to significantly less deadline violations for large thresholds t: Yu and Scully [49, Section 6] show that compared to optimizing for the decay rate alone,[1] one can reduce "large-t" violations by 30% or more by also optimizing the tail constant. This can be especially important in settings where SLOs are tight and any reduction in large response times is impactful.

The basic idea of γ-Boost is that it roughly mimics FCFS (first-come first-served), but it gives short jobs partial priority by "boosting" their arrival times backwards, as if they had actually arrived earlier. Short jobs get boosted more than long jobs: a job of size s has its arrival time boosted by

$$b_{\gamma\text{-Boost}}(s) = \frac{1}{\gamma}\log\frac{1}{1-e^{-\gamma s}}, \tag{1.1}$$

where $\gamma > 0$ is a parameter depending on the load and size distribution. See Section 2.4 for a full definition of γ-Boost.

However, the single-server M/G/1 model is a poor match for large-scale computer systems, which inevitably have multiple servers. And in the M/G/k, the M/G/1's k-server analogue, scheduling to optimize the tail $\mathbf{P}[T > t]$ remains an open problem. Instead, prior work on M/G/k scheduling theory primarily covers only the simpler problem of optimizing *mean* response time $\mathbf{E}[T]$ and weighted variants thereof [5, 9, 13–15, 17–19, 42–45]. Moreover, while these results prove bounds on (weighted) mean response time that hold at all loads (a.k.a. utilizations) $\rho \in (0, 1)$, they are only tight in *heavy traffic* as $\rho \to 1$, and thus optimality results are similarly limited to heavy traffic.

Despite the limitation to heavy traffic, existing M/G/k scheduling theory seems to lead to the right design decisions across a wide range of loads. For instance, while SRPT (shortest remaining processing time), which minimizes $\mathbf{E}[T]$ in the M/G/1, is only proven to minimize $\mathbf{E}[T]$ in the M/G/k in heavy traffic [18, 44], SRPT enjoys excellent empirical $\mathbf{E}[T]$ across a range of loads [16, 20], with the best known alternative improving upon SRPT by less than 1% [20].

1.2 Our Work: Multiserver Tail Scheduling

Taken together, the two lines of work discussed in Section 1.1 clearly suggest two questions:

Q1: *Is γ-Boost tail constant optimal in the M/G/k in heavy traffic?*

Q2: *Does γ-Boost have good empirical tail performance in the M/G/k in lighter traffic?*

Our work answers these and other questions about tail scheduling in the M/G/k. We were genuinely surprised by some of the answers, and we believe many readers will be, too. In an effort to mitigate

[1] Policies that achieve the optimal decay rate are said to be weakly optimal, and it is well-known that FCFS does so [6].

hindsight bias, this section gives a roughly chronological account of our findings. See Section 1.3 for a more straightforward statement of our contributions.

Throughout, we consider the M/G/k to have k servers of speed $1/k$ each, so that a job of size s takes time ks to complete. This means that the same arrival process induces the same load, and also the same value of γ for use in (1.1), with any number of servers k.

1.2.1 Findings for γ-Boost. Our first finding is the least surprising: we prove that Q1's answer is "yes" (Theorem 3.1). Namely, we show that γ-Boost is heavy-traffic tail constant optimal by showing that it achieves the same tail constant as γ-Boost in the M/G/1 as $\rho \to 1$:

$$\lim_{\rho \to 1} \frac{\limsup_{t\to\infty} e^{\gamma t}\mathbf{P}[T_{\text{M/G/}k\ \gamma\text{-Boost}} > t]}{\limsup_{t\to\infty} e^{\gamma t}\mathbf{P}[T_{\text{M/G/1}\ \gamma\text{-Boost}} > t]} = 1.$$

To do this, we follow the same overall strategy as the prior work on $\mathbf{E}[T]$ in the M/G/k, but handling the tail $\mathbf{P}[T > t]$ requires us to overcome new technical challenges. We discuss this in more detail in Section 3.2, but in brief: in a step that compares the amount of work in an M/G/k to that of an M/G/1 with the same arrival process, while simple worst-case bounds suffice for SRPT, we require a stochastic bound, and we provide a new stochastic bound that holds under *any* non-idling policy.

We prove an analogous result for scheduling with *unknown job sizes*, showing that a variant of γ-*Gittins*, which is tail constant optimal in the unknown-size M/G/1 [25], is also tail constant optimal in the unknown-size M/G/k in heavy traffic (Theorem 4.1). This latter result actually covers not just unknown job sizes, but a general partial-information model with *Markov-process jobs* [19, 25, 43, 44]. Simulations confirm that γ-Boost's "pre-asymptotic" tail performance is good at very high loads, e.g. $\rho = 0.99$ (Section 5).

However, despite Q1's answer being "yes", Q2's answer is "no"! We observe in simulations shown in Fig. 1.1 and Section 5 that with $k = 10$ servers, γ-Boost performs worse than FCFS even at loads as high as $\rho = 0.95$. We initially found this surprising, but in hindsight, we can at least partly explain it. The underlying reason why γ-Boost is tail constant optimal in the M/G/1 is that it solves a particular single-server batch scheduling problem with exponentially inflated response time costs [49] (see also Appendix E). But the multiserver version of the batch problem is computationally intractable to solve exactly,[2] and it seems that γ-Boost, which amounts to a greedy heuristic in the multiserver setting, fails to find high-quality solutions. In contrast, in the corresponding story for optimizing $\mathbf{E}[T]$, SRPT solves the corresponding multiserver batch problem [40, Theorem 5.3.1].

1.2.2 Improving Tail Performance in Lighter Traffic. Prompted by the failure of γ-Boost outside of heavy traffic, we ask:

Q3: *Is improving upon FCFS's tail possible in the M/G/k in lighter traffic?*

To answer this, we simulate a policy that uses the Gurobi mixed-integer program solver [22] to solve the batch scheduling problem discussed above *after every arrival*. As shown in Fig. 1.1, this policy, which we call simply *Gurobi*, does indeed improve upon FCFS, so Q3's answer is at least in principle "yes". We conjecture that an idealized version of Gurobi that exactly solves each mixed-integer program is tail constant optimal, though this is far from certain. Either way, solving a mixed-integer program with every arrival is infeasible in most practical applications, so the question of whether we can *practically* improve upon FCFS remains.

One intuition about why multiserver scheduling is hard is that it requires thinking not just about how to prioritize jobs relative to each other, but also about how to keep all of the servers busy, which amounts to *balancing work as equally as possible* across servers [34]. In batch scheduling

[2]While the specific batch problem in question (Appendix E) has not been studied in the literature, many related problems are known to be NP-hard [33, 40], and we believe a reduction from our problem to one of these is possible.

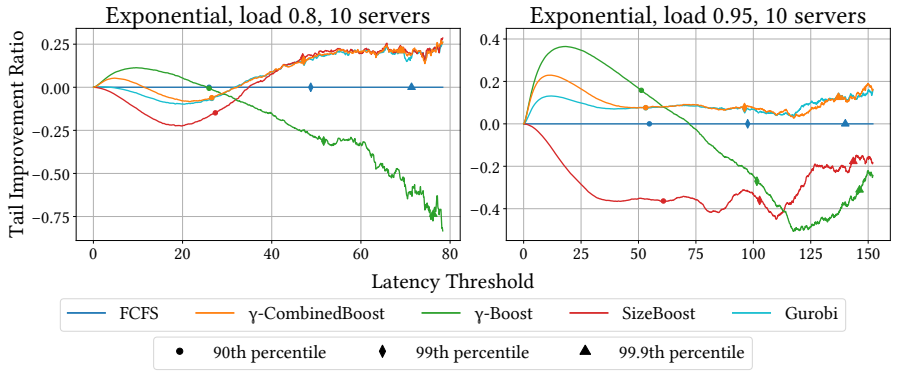


Fig. 1.1. (Higher is better.) Plot of tail performance of policies for $k = 10$ servers for both $\rho = 0.8$ and $\rho = 0.95$ load with job size distribution Exp(1). Tail improvement ratio of policy π at threshold t is $1 - \mathbf{P}[T_\pi > t]/\mathbf{P}[T_{\text{FCFS}} > t]$. Due to Gurobi's heavy computational requirements, we were only able to run for 100,000 samples for load 0.8 and 700,000 samples for load 0.95. Plotted results therefore may not have reached convergence; however, the results from both loads suggest that γ-CombinedBoost attains the same performance as Gurobi. See Section 5 for simulations of the non-Gurobi policies with hundreds of millions or billions of jobs.

problems, the key to doing this is to serve *long jobs first* [40, Theorems 5.1.1 and 5.2.7], which is in direct tension with SRPT's and γ-Boost's prioritizing short jobs.

Strictly prioritizing long jobs gives predictably poor tail performance (Appendix H), but the general idea of γ-Boost suggests a compromise: instead of boosting short jobs, we could *boost long jobs*. Following this reasoning, we propose the *SizeBoost* policy, which boosts a job of size s by a fraction, $\frac{k-1}{k}$, of its processing time. Under our $1/k$ server speed convention (Section 2), where jobs of size s have ks processing time, this amounts to a boost of

$$b_{\text{SizeBoost}}(s) = (k-1)s.$$

We have not carefully tuned the specific factor of $k - 1$, but rather chose it to make sense when specialized to $k = 1$, in which case boosting long jobs is a poor plan.[3] We see in Fig. 1.1 that SizeBoost is partly successful.

- At the lower load of $\rho = 0.8$, SizeBoost's tail $\mathbf{P}[T > t]$ *matches Gurobi's at large thresholds* t, which is essentially the best large-t behavior we can hope for.
- But at the higher load of $\rho = 0.95$, SizeBoost is significantly worse than FCFS.

It makes sense that SizeBoost should perform poorly in heavy traffic, because we know that γ-Boost, which boosts short jobs rather than long jobs, is the right choice in heavy traffic. But it seems that as load increases, SizeBoost's performance degrades before γ-Boost's improves, leaving a "medium-high" load of $\rho = 0.95$ where neither matches even FCFS, let alone Gurobi.

1.2.3 Bridging the Gap Between Lighter and Heavy Traffic. Seeing as γ-Boost and SizeBoost alone are not enough to achieve good tail performance at all loads, we ask:

Q4: *Can a practical heuristic match Gurobi at "medium-high" loads?*

One might expect that if this were possible at all, it would require new ideas beyond those underlying γ-Boost and SizeBoost, given that both policies perform poorly at $\rho = 0.95$ in Fig. 1.1.

[3]Empirically, SizeBoost works well; whether boost functions of the form $b(s) = cs$ for some constant c are optimal is a question we leave to future work.

Surprisingly, we find that not only is Q4's answer "yes", but one can match Gurobi with a naive combination of γ-Boost and SizeBoost: *just sum their boosts*. We propose the *γ-CombinedBoost* policy, which boosts jobs of size s by

$$b_{\gamma\text{-CombinedBoost}}(s) = b_{\gamma\text{-Boost}}(s) + b_{\text{SizeBoost}}(s) = \frac{1}{\gamma}\log\frac{1}{1-e^{-\gamma s}} + (k-1)s.$$

This policy *improves upon Gurobi* in Fig. 1.1 at both $\rho = 0.8$ and $\rho = 0.95$, with strictly better tail at small thresholds and matching tail at large thresholds. γ-CombinedBoost is also effective in heavy traffic: using essentially the same strategy as for γ-Boost, we prove that γ-CombinedBoost is tail constant optimal in heavy traffic (Theorem F.4). Additional simulations of γ-CombinedBoost in Section 5 show it to be a clear "overall best" among the policies we simulate, sometimes getting close to theoretical limits on what can be achieved (e.g. M/G/∞ bounds).

A priori, we predicted that γ-CombinedBoost might match γ-Boost in heavy traffic and SizeBoost in lighter traffic, because γ is roughly proportional to $1-\rho$ (Lemma A.1).

- In heavy traffic, we have $\gamma \to 0$. This means $b_{\gamma\text{-Boost}}(s)$ dominates $b_{\text{SizeBoost}}(s)$, and thus γ-CombinedBoost behaves like γ-Boost.
- In lighter traffic, γ is nonnegligible. This means $b_{\gamma\text{-Boost}}(s)$ is small except for very small sizes s, and thus γ-CombinedBoost behaves like a version of SizeBoost that gives extra priority to very small jobs, which seems either harmless or actively helpful.

What we find surprising is that at loads between these extremes, γ-CombinedBoost significantly outperforms γ-Boost and SizeBoost, despite being a naive interpolation between them.

1.3 Contributions and Outline

In this paper, we give the *first theoretical optimality guarantees for multiserver tail scheduling*, specifically strong tail optimality in the M/G/k in heavy traffic.

- For known sizes, our guarantees apply to the γ-Boost policy proposed by Yu and Scully [49] (Section 3) and our newly proposed γ-CombinedBoost policy (Appendix F).
- For unknown sizes, our guarantees apply to a variant of the γ-Gittins policy, specifically *γ-Surrogate*, proposed by Harlev et al. [25] (Section 4).

We also perform a broad suite of simulation experiments to evaluate tail performance at practical thresholds and outside of heavy traffic, finding that *our newly proposed γ-CombinedBoost policy sets a new state-of-the-art* (Section 5).

2 System Model

We consider an M/G/k queue with arrival rate λ, job size distribution S, and load $\rho = \lambda\mathbf{E}[S]$. For convenience, we assume that each of our k servers can complete work at rate $1/k$. Therefore, the M/G/k system has total service capacity 1. This will allow us to compare to a resource-pooled M/G/1, i.e. a single-server system with speed 1, as a lower bound.

2.1 Load and Stability

Under our server speed assumption, the load is $\rho = \lambda\mathbf{E}[S]$, and, because the system has total service capacity 1, we assume that $\rho < 1$. One would hope that this would be sufficient for stability in the M/G/k for non-idling scheduling policies, which we define formally in Definition 2.2. Similar to other prior work in this area [18, 26, 44], we will assume stability indeed holds. While we expect that $\rho < 1$ is indeed sufficient for stability, proving it is outside the scope of this paper. Hong and Scully [26, Appendix D] provide a proof sketch for stability in the G/G/k, which should also handle our M/G/k setting.

Assumption 2.1. If $\rho < 1$, the M/G/k is stable under all non-idling scheduling policies π.

Our theoretical results are for the *heavy-traffic limit*. This limit, which we will denote by $\rho \to 1$, refers to the limit as $\lambda \to 1/\mathbf{E}[S]$, with the job size distribution S fixed.

2.2 Work in Multiserver Systems

Key to our analysis is quantifying the "work" in system, which is the amount of processing time a speed 1 server needs to complete all remaining jobs in the system. In particular, under our server speed convention, a job of size S contributes S work, because it takes S time to complete under a speed 1 server. However, the processing time of this job in the M/G/k will be kS, because the M/G/k servers only work at speed $1/k$.

We denote the steady-state work distribution in an M/G/k under policy π by W_π^k. For the M/G/1, we write simply $W_{\mathrm{M/G/1}}$, as the work is unaffected by the policy provided it is non-idling. However, in the M/G/k for $k > 1$, the work W_π^k *does* depend on the scheduling policy π. This is true even for non-idling policies, which do not leave servers unnecessarily idle, and which we will focus on in this paper.

Definition 2.2. A policy π is a *non-idling policy* if, under π, whenever there are fewer than k jobs in the system, all jobs are in service, and whenever there are at least k jobs in the system, all servers are busy.

Definition 2.3. The *idleness* is the fraction of servers that are idle. We denote by $I_\pi^k(t)$ the idleness under policy π at time t and by I_π^k the idleness under policy π in steady-state. If π is non-idling, then we can write the idleness as

$$I_\pi^k(t) = \frac{(k - N_\pi^k(t))^+}{k}, \qquad I_\pi^k = \frac{(k - N_\pi^k)^+}{k},$$

where $N_\pi^k(t)$ and N_π^k are the number of jobs in the system under policy π at time t and in steady-state, respectively.

The amount of work in the M/G/k under a policy π obeys a decomposition law that says, roughly, that the amount of work in the M/G/k is the amount of work in the M/G/1, which is policy-invariant, plus an amount related to how much work is in the system while servers are idle. The intuition is that work present when servers are idle "persists" once all of the servers are busy again. The statement below is from Scully [43, Theorem 8.3(b)], which in turn is a slight generalization of classical decomposition laws for M/G/1-like systems [10, 37].[4]

Theorem 2.4 (Work Decomposition Law). *For any non-idling scheduling policy π and any θ such that $\mathbf{E}[e^{\theta S}] < \infty$ and $\theta > \lambda(\mathbf{E}[e^{\theta S}] - 1)$,*

$$\mathbf{E}[e^{\theta W_\pi^k}] = \mathbf{E}[e^{\theta W_{\mathrm{M/G/1}}}] \frac{\mathbf{E}[I_\pi^k e^{\theta W_\pi^k}]}{1 - \rho}.$$

Handling the *wasted work* term, $\mathbf{E}[I_\pi^k e^{\theta W_\pi^k}]/(1 - \rho)$, is a key challenge in our analysis. We cover the techniques required to do so in Section 3.2.

2.3 Tail Asymptotics

Following Yu and Scully [49], we assume that the job size distribution S is light-tailed, specifically, that it is class I [1, 2]:

[4]Strictly speaking, the proof given by Scully [43, Theorem 8.3(b)] covers the $\theta \le 0$ case, i.e. the Laplace transform case, but the result for $\theta > 0$ follows from the probabilistic interpretation of the Laplace transform, which is also given by Scully [43, Theorem 8.3(a)].

Assumption 2.5. The size distribution S is *class I*, meaning its moment generating function's leftmost singularity

$$\theta^* = \sup\{\theta \in \mathbb{R} \mid \mathbf{E}[e^{\theta S}] < \infty\},$$

which may be ∞, satisfies $\theta^* > 0$ and $\lim_{\theta \to \theta^*} \mathbf{E}[e^{\theta S}] = \infty$.

Many "typical examples" of light-tailed distributions are class I, including bounded distributions, and distributions whose tails are asymptotically exponential, Gaussian, and Weibull with shape parameter at least 1 (i.e. the "lighter than exponential case"). The distributions that are light-tailed but *not* class I are exponentially damped heavy-tailed distributions [1, 2], such as those the form $\mathbf{P}[S > t] \sim ct^{-\alpha}e^{-\beta t}$ for constants $\alpha, \beta, c > 0$.

Our metric of focus is the *response time* of jobs, which is the total amount of time a job spends in the system. Let T_π^k denote the steady-state response time distribution under scheduling policy π in the M/G/k. Boxma and Zwart [6] show that in the M/G/1 with class I size distribution, policies π have

$$\limsup_{t \to \infty} e^{\gamma t}\mathbf{P}[T_\pi^1 > t] = C, \tag{2.1}$$

for some policy-dependent constant C, which may be infinite, where γ is the least positive real solution [36] to

$$\gamma = \lambda(\mathbf{E}[e^{\gamma S}] - 1). \tag{2.2}$$

Our goal is to find a policy that is *tail constant optimal* in the heavy-traffic limit. Roughly, we want to find a policy π that attains the smallest possible C. Formally, we define the tail constant of a policy π as follows:

Definition 2.6. Let $X \geq 0$ be a nonnegative random variable. Then the *tail constant* of X is

$$\mathbf{C}^+[X] = \limsup_{t \to \infty} e^{\gamma t}\mathbf{P}[X > t].$$

We also define the *lower tail constant* of X to be

$$\mathbf{C}^-[X] = \liminf_{t \to \infty} e^{\gamma t}\mathbf{P}[X > t].$$

The tail constant and lower tail constant of a policy π in the M/G/k are given by $\mathbf{C}^+[T_\pi^k]$ and $\mathbf{C}^-[T_\pi^k]$, respectively.

Definition 2.7. A scheduling policy π is *tail constant optimal*[5] if

$$\sup_{\pi'} \frac{\mathbf{C}^+[T_\pi^k]}{\mathbf{C}^+[T_{\pi'}^k]} = 1.$$

Heavy-traffic tail constant optimality therefore requires

$$\lim_{\rho \to 1} \sup_{\pi'} \frac{\mathbf{C}^+[T_\pi^k]}{\mathbf{C}^+[T_{\pi'}^k]} = 1.$$

One subtlety here is that we allow the policy π in the numerator to be a parametrized family that depends on the load ρ. For example, γ-Boost, defined below, depends on the value of γ from (2.2), which varies with the load ρ.

The tail constants of policies can be difficult to analyze directly, so to compare policies, we analyze their *tail transform constants*:

[5]We define the notion of tail constant optimality, instead of using the more widespread notion of tail optimality [6], as the policies we analyze from prior work [25, 49] turn out only to satisfy tail optimality under a certain class of policies, whereas they satisfy tail constant optimality across all policies. That said, we believe that tail constant optimality still captures a useful notion of optimal tail performance. Full details are provided in Appendix G.

Definition 2.8. Let $X \geq 0$ be a nonnegative random variable. Then the *(upper) tail transform constant* of X is

$$\tilde{\mathbf{C}}^+[X] = \limsup_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma} \mathbf{E}[e^{\theta X}],$$

and the analogous *lower tail transform constant* is given by

$$\tilde{\mathbf{C}}^-[X] = \liminf_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma} \mathbf{E}[e^{\theta X}].$$

The tail transform constant and lower tail transform constant of a policy π are denoted by $\tilde{\mathbf{C}}^+[T_\pi^k]$ and $\tilde{\mathbf{C}}^-[T_\pi^k]$, respectively.

In particular, a final value theorem [8] tells us that whenever the poles of $\mathbf{E}[e^{\theta T_\pi^k}]$ are in the open right half plane or at the origin, with at most one pole at the origin,

$$\mathbf{C}^-[T_\pi^k] = \tilde{\mathbf{C}}^-[T_\pi^k] = \tilde{\mathbf{C}}^+[T_\pi^k] = \mathbf{C}^+[T_\pi^k]. \tag{2.3}$$

For example, when the job size distribution is class I, the work in an M/G/1 system $W_{\text{M/G/1}}$, has asymptotically exponential tail [21, Equation (2)], and

$$\mathbf{C}^-[W_{\text{M/G/1}}] = \tilde{\mathbf{C}}^-[W_{\text{M/G/1}}] = \tilde{\mathbf{C}}^+[W_{\text{M/G/1}}] = \mathbf{C}^+[W_{\text{M/G/1}}],$$

with a simple pole at γ [1, 2]. Quantities we explicitly analyze in this paper (in, for example, Section 3), have the same poles as the work transform, and so satisfy this property. For policies for which this is not the case, we have, by Lemma A.3,

$$\mathbf{C}^-[T_\pi^k] \leq \tilde{\mathbf{C}}^-[T_\pi^k] \leq \tilde{\mathbf{C}}^+[T_\pi^k] \leq \mathbf{C}^+[T_\pi^k]. \tag{2.4}$$

2.4 Boost Policies and γ-Boost

A main theoretical result of our work is to prove heavy-traffic optimality of the scheduling policy γ-Boost in the M/G/k. γ-Boost is tail constant optimal in the M/G/1, and so is a natural candidate as a policy in the M/G/k. It belongs to a family of policies known as *boost policies*, introduced by Yu and Scully [49].

Boost policies operate according to a simple rule: *serve the job of smallest boosted arrival time.* A job's boosted arrival time is defined to be

$$\text{boosted arrival time} = \text{arrival time} - \text{boost} = \text{arrival time} - b(\text{size}),$$

where a *boost function* $b : \mathbb{R}_+ \to [0, \infty)$ maps a job's size to its boost. Different boost policies differ in their choice of boost function b. We write b_π for the boost function of boost policy π, or simply b if the policy is generic or clear from context. The boost function of γ-Boost is

$$b_{\gamma\text{-Boost}}(s) = \frac{1}{\gamma} \log\left(\frac{1}{1 - e^{-\gamma s}}\right), \tag{2.5}$$

where γ is set according to (2.2).

One can consider either preemptive or nonpreemptive versions of any given boost policy [49]. We use nonpreemptive versions in our simulations, and one can check that our proofs in Section 3 and Appendix F apply to both versions.

3 Proving Heavy-Traffic Optimality of Boost

To prove heavy-traffic optimality of γ-Boost in the M/G/k, we will compare its tail constant in an M/G/k system with server speeds $1/k$ to the tail constant it achieves in a resource-pooled M/G/1, with server speed 1. Under our server speed convention, γ-Boost's M/G/1 tail constant provides a lower bound on the optimal tail constant in the M/G/k, because γ-Boost is tail constant optimal for the M/G/1 [49]. Specifically, recalling (2.4), we have

$$\mathbf{C}^+[T^1_{\gamma\text{-Boost}}] = \tilde{\mathbf{C}}^+[T^1_{\gamma\text{-Boost}}] \leq \inf_{\pi} \tilde{\mathbf{C}}^+[T^k_\pi] \leq \inf_{\pi} \mathbf{C}^+[T^k_\pi],$$

where the infimum is taken over all policies that can be implemented in an M/G/k. Our main result is that the gap in tail constants between γ-Boost in the M/G/k and γ-Boost in the resource-pooled M/G/1 approaches 0 in the heavy-traffic limit.

Theorem 3.1. *For an M/G/k with class I job size distribution S, γ-Boost is optimal in the heavy traffic limit:*

$$\lim_{\rho\to 1} \frac{\mathbf{C}^+[T^k_{\gamma\text{-Boost}}]}{\mathbf{C}^+[T^1_{\gamma\text{-Boost}}]} \leq \lim_{\rho\to 1} \frac{\mathbf{C}^+[W_{\text{M/G/1}}] \frac{\mathbf{E}[I^k_{\gamma\text{-Boost}} e^{\gamma W^k_{\gamma\text{-Boost}}}]}{1-\rho} \mathbf{E}[e^{\gamma(kS - b_{\gamma\text{-Boost}}(S))}] \mathbf{E}[e^{\gamma V_{\gamma\text{-Boost}}(\infty)}]}{\mathbf{C}^+[W_{\text{M/G/1}}] \mathbf{E}[e^{\gamma(S - b_{\gamma\text{-Boost}}(S))}] \mathbf{E}[e^{\gamma V_{\gamma\text{-Boost}}(\infty)}]} = 1,$$

where in the second limit, the numerator is an upper bound on the tail constant of γ-Boost in the M/G/k, and the denominator is the tail constant of γ-Boost in the M/G/1.

Why is γ-Boost's performance in the M/G/k different than in the M/G/1? The key difference is that the M/G/k system is not work-conserving. In particular, the system can have a variable amount of idleness, which is policy-dependent. Proving Theorem 3.1 therefore requires characterizing the effect of idleness on performance. In particular, for γ-Boost, we need to bound:

(1) The effect of idleness on the amount of relevant work[6] served between a job's boosted arrival time and actual arrival time.
(2) The effect of idleness on the steady-state work in system.

Because proving Theorem 3.1 relies on bounding these two effects, we will next focus on bounding them, and defer the proof of Theorem 3.1 to Section 3.3. For item (1), we prove an upper bound on the tail constant for general boost policies (Theorem 3.3), which shows that for our purposes, the effect of idleness on the amount of relevant work served is negligible. For item (2), we prove a bound on the effect of idleness on steady-state work for all non-idling policies (including all general boost policies) in Theorem 3.6, and show that in the heavy-traffic limit, the impact of idleness is negligible. We handle item (1) in Section 3.1 and item (2) in Section 3.2.

3.1 Characterizing Boost's Tail Constant in the M/G/k

To compare the tail constant of Boost in the M/G/k to that of Boost in the M/G/1, we will compare the tail transform constants in the two settings. In particular, as we know that $\tilde{\mathbf{C}}^+[T^1_{\gamma\text{-Boost}}]$ is a lower bound on the tail constant of any policy π in the M/G/k, it suffices to bound the gap of $\mathbf{C}^+[T^k_{\gamma\text{-Boost}}]$ to $\tilde{\mathbf{C}}^+[T^1_{\gamma\text{-Boost}}]$.

We do this by proving a more general bound, which holds for any Boost policy π with boost function b_π. To bound $\mathbf{C}^+[T^k_\pi]$, we will employ a *tagged job* analysis similar to that in Yu and Scully [49]. One key difference is that, because we do not characterize $\mathbf{C}^+[T^k_\pi]$ directly, we must instead first bound $\mathbf{C}^+[T^k_\pi]$ by $\mathbf{C}^+[X_\pi]$ for some appropriate quantity X_π s/t $\mathbf{C}^+[X_\pi] = \tilde{\mathbf{C}}^+[X_\pi]$. We can then show that when π is γ-Boost, the gap between $\tilde{\mathbf{C}}^+[X_\pi]$ and $\tilde{\mathbf{C}}^+[T^1_{\gamma\text{-Boost}}]$ (equivalently, $\mathbf{C}^+[X_\pi]$ and $\mathbf{C}^+[T^1_{\gamma\text{-Boost}}]$) vanishes. We update the notation here to handle multiple servers.

[6]Relevant work to a job J can roughly be thought of as work that has better priority than J.

Notation 3.2.

(a) We will analyze the response time of a tagged job with boosted arrival time 0.
  - We will write S for the tagged job's size and $B_\pi = b_\pi(S)$ for the tagged job's boost under boost policy π, suppressing π when the policy is obvious from context.

(b) We denote the amount of work in the M/G/k at time 0 under policy π by W_π^k, which is distributed according to the stationary work distribution.
  - In contrast to the M/G/1, the distribution is policy-dependent for $k > 1$, because the idleness (Definition 2.3) is policy-dependent.

(c) We denote crossing work and non-crossing work during $(0, u)$ by $V_\pi(u)$ and $\overline{V}_\pi(u)$, respectively, again suppressing π when the policy is obvious from context.
  - $V(u)$ is the amount of work from jobs arriving in $(0, u)$ with boosted arrival time in $(-\infty, 0]$.
  - $\overline{V}(u)$ is the complementary quantity, the amount of work from jobs arriving in $(0, u)$ with boosted arrival time in $(0, \infty)$.

We now adapt the analysis from Boost to handle the effect of idle servers in the M/G/k. Parts of our analysis will apply specifically to Boost policies, while other parts apply, generally, to all non-idling policies (which include boost policies). The end result is the following.

Theorem 3.3. *Let π be a boost policy and suppose that $\mathbf{E}[e^{\gamma k V(\infty)}] < \infty$, $\mathbf{E}[e^{\gamma k S}] < \infty$, and $\mathbf{E}[I_\pi^k e^{\gamma W_\pi^k}] < \infty$. The tail constant of π can be bounded above as follows:*

$$\mathbf{C}^+[T_\pi^k] \le \mathbf{C}^+[W_{\mathrm{M/G/1}}] \frac{\mathbf{E}[I_\pi^k e^{\gamma W_\pi^k}]}{1-\rho} \mathbf{E}[e^{\gamma(kS-B)}] \mathbf{E}[e^{\gamma V(\infty)}].$$

When $k = 1$, this bound reduces to the exact M/G/1 tail constant found by Yu and Scully [49, Theorem 3.1]. When $k > 1$, there are two differences, but, importantly for Theorem 3.1, both vanish in heavy traffic:

- The $\mathbf{E}[e^{\gamma(S-B)}]$ becomes $\mathbf{E}[e^{\gamma(kS-B)}]$, but this change is negligible in heavy traffic because $\gamma \to 0$.
- There is an additional factor of $\mathbf{E}[I_\pi^k e^{\gamma W_\pi^k}]/(1-\rho)$, which is related to the amount of work present when there are $k-1$ or fewer jobs in the system. We give an upper bound on this quantity under *any* non-idling scheduling policy using a "last-job lemma" (Lemmas 3.7 and 3.8), and the bound approaches 1 in heavy traffic.

How does idleness affect the analysis from Yu and Scully [49]? In the M/G/1, whenever the system has work, $I_\pi^1(t) = 0$ for all policies π. For boost policies, this means that we only need to consider two cases in the analysis: $W_\pi^1 > B$, in which case the idleness is 0 throughout the tagged job's time in the system, and $W_\pi^1 \le B$, in which case $W_\pi^1 = 0$ throughout the tagged job's time in the system. This is how Yu and Scully [49, Lemma 3.3] proceed in deriving the tail constant for general boost policies. But in the M/G/k, the system might have $W_\pi^k > B$, but also nonzero idleness after the tagged job's boosted arrival time. Our goal is to show that even in this case, we can bound a boost policy's tail constant by assuming the idleness has no impact on the boost. In particular, we want to handle cases when there was insufficient relevant work between the tagged job's boosted arrival time and actual arrival time. We can do so by capturing cases where we are guaranteed to have only served relevant work, using the following event:

Lemma 3.4. *Let Q be the event that at the tagged job's arrival time (i.e., time B), there are at least k jobs in the system with arrival time earlier than time 0. Let $u \ge 0$. The tagged job's response time*

under boost policy π can be upper bounded by

$$T_\pi^k \leq \begin{cases} W_\pi^k - \min\{B,u\} + kS + V(\infty) & \text{if } Q \text{ holds} \\ k(S + V(\infty) + \overline{V}(\min\{B,u\})) & \text{if } Q \text{ does not hold.} \end{cases}$$

Proof. We first observe that decreasing the tagged job's boost from B to $\hat{B} = \min\{B,u\}$ can only increase its response time, so it suffices to analyze the response time with this reduced boost. In both cases, we will bound the amount of time the tagged job can spend in the system.

If Q holds: At the tagged job's true arrival time, there are at least k jobs in the system with priority over it. In particular, these jobs have been in the system since the boosted arrival time of the tagged job, by our definition of Q. This means that between the tagged job's boosted arrival time and its actual arrival time, there were always at least k jobs in the system with better boosted arrival time than the tagged job's boosted arrival time. This implies that all servers have only worked on jobs with boosted arrival time better than the tagged job's boosted arrival time.

An upper bound on the amount of work with boosted arrival time better than the tagged job's boosted arrival time is $W_\pi^k + V(\infty)$. Between the tagged job's boosted arrival time and actual arrival time, all servers only worked on jobs belonging to this $W_\pi^k + V(\infty)$ amount of work. Therefore, the remaining amount of such work at the tagged job's actual arrival time is $W_\pi^k + V(\infty) - \hat{B}$. Once the tagged job arrives, one of the following must be true until it departs the system:

(1) All servers are occupied with work that has boosted arrival time better than the tagged job's boosted arrival time, i.e., this work leaves the system at rate 1.
(2) If not all servers are occupied with such work, the tagged job must be in service, because it has the best boosted arrival time among all remaining jobs. The tagged job is served at rate $1/k$.

The maximum amount of time that (1) can hold is $W_\pi^k + V(\infty) - \hat{B}$, and the maximum amount of time that (2) can hold is kS, so the maximum amount of time that either can hold is $W_\pi^k + V(\infty) - \hat{B} + kS$, which yields an upper bound on the amount of time the tagged job can spend in the system.

If Q does not hold: At the tagged job's true arrival time, there are at most $k-1$ jobs in the system with arrival time before the tagged job's boosted arrival time. In particular, even if all such jobs are in service, they only occupy $k-1$ of the k available servers. Then, once the tagged job arrives, one of the following must be true until it departs the system:

(1) For nonpreemptive policies, there could be a server working on a job from $\overline{V}(\min\{B,u\})$, with nothing from $V(\infty)$ in service, and with the tagged job not in service.
(2) There is at least one server working on work from $V(\infty)$, i.e. such work leaves the system at rate at least $1/k$.
(3) The tagged job is served at rate $1/k$.

(1) can only hold at the tagged job's arrival time. For each server that is serving a job from $\overline{V}(\min\{B,u\})$, after that job's completion, the server will take either the tagged job, or work from $V(\infty)$ into service. Namely, after a job from $\overline{V}(\min\{B,u\})$ completes, either event (2) or (3) will hold until the tagged job departs the system. A simple upper bound on how long (1) can hold is to imagine that all the work in $\overline{V}(\min\{B,u\})$ is served on a single server, leaving at the rate $1/k$, before the other two events hold. (2) can hold for at most $kV(\infty)$, and (3) can only hold for at most kS. Therefore, the tagged job will spend at most $k(\overline{V}(\min\{B,u\}) + V(\infty) + S)$ time in the system, as desired. □

Lemma 3.5. *For any boost policy π and event Q as defined in Lemma 3.4,*

$$\mathbf{P}[T_\pi^k > t] \leq \mathbf{P}[W_\pi^k - \min\{B,u\} + kS + V_\pi(\infty) > t] + \mathbf{P}[k(V(\infty) + \overline{V}(\min\{B,u\}) + S) > t].$$

Proof. From Lemma 3.4, we know that

$$\begin{aligned}\mathbf{P}[T_\pi > t] &= \mathbf{P}[\{T_\pi > t\} \cap Q] + \mathbf{P}[\{T_\pi > t\} \cap Q^c] \\ &\le \mathbf{P}[\{W_\pi^k - \min\{B, u\} + kS + V(\infty) > t\} \cap Q] + \mathbf{P}[\{k(V(\infty) + S) > t\} \cap Q^c] \\ &\le \mathbf{P}[W_\pi^k - \min\{B, u\} + kS + V(\infty) > t] + \mathbf{P}[k(V(\infty) + \overline{V}(\min\{B, u\}) + S) > t]. \quad \square\end{aligned}$$

With Lemmas 3.4 and 3.5 in hand, we can prove Theorem 3.3:

Proof Sketch of Theorem 3.3. The proof requires applying the upper bound from Lemma 3.5 with the scaling factor $e^{\gamma t}$, then applying final value theorem to both terms on the RHS and computing the requisite limits. The computations are similar to that of the Boost paper [49, Theorem 3.1], so we defer a complete proof to Appendix C. While the idleness does not impact the boost a job receives, it does impact the transform; the $\frac{\mathbf{E}[I_\pi^k e^{\gamma W_\pi^k}]}{1-\rho}$ term comes from expanding the transform of work in system, namely, by applying the work decomposition law (Theorem 2.4). $\square$

3.2 Characterizing Work Under System Idleness

In Section 3.1, we derived an upper bound on $\tilde{\mathbf{C}}^+[T^k_{\gamma\text{-Boost}}]$ for general boost policies, showing that it suffices to assume that between the tagged job's boosted arrival time and actual arrival time, the system worked solely on jobs with better priority than the tagged job. This bound also makes clear the dependence of a boost policy's tail constant on system idleness, namely, through the *wasted work* factor, $\frac{\mathbf{E}[I_\pi^k e^{\gamma W_\pi^k}]}{1-\rho}$. Heavy-traffic optimality of γ-Boost requires that this term approaches 1 in the $\rho \to 1$ limit.

Grosof et al. [18] bounded the effects of wasted work via a worst-case bound on the amount of extra additional "relevant work" that can be present in the M/G/k system. However, this technique cannot be used to bound the relevant work for boost policies, for the same reason it cannot be used to provide bounds for FCFS. Namely, under FCFS, all jobs in the system at the arrival time of the tagged job are permanently relevant to the tagged job, so with unbounded job size distributions S, the relevant work contributed by each relevant job is unbounded. Similarly, for any boost policy, all jobs in the system at the boosted arrival time of the tagged job are permanently relevant to the tagged job, and so all boost policies run into the same issue as FCFS. Therefore, we approach bounding the wasted work term via stochastic methods. In particular, we will prove the following, general bound on the wasted work for all non-idling policies.

Theorem 3.6. *For any non-idling policy π,*

$$\lim_{\rho \to 1} \tilde{\mathbf{C}}^+[W_\pi^k] \le \mathbf{C}^+[W_{\text{M/G/1}}].$$

Proving Theorem 3.6 requires a bound on the work in system when the idleness $I > 0$. For these non-idling policies, the amount of wasted processing time can be characterized as follows: we will show that to bound the work, it suffices to provide a bound on the size of the largest job in the system when the idleness is nonzero. Lemma 3.7 below provides such a bound in terms of the *excess* S_e of S, namely the distribution such that

$$\mathbf{P}[S_e > x] = \frac{1}{\mathbf{E}[S]} \int_x^\infty \mathbf{P}[S > y] \, dy.$$

Lemma 3.7 (Last Job Lemma). *Define L to be $k \times$ (size of largest job in the system). Then for any $x > 0$,*

$$\mathbf{E}[I_\pi^k \mathbb{1}(L > x)] \le k\mathbf{P}[kS_e > x].$$

Proof. By Assumption 2.1, we can assume stationarity of the involved processes, namely, of L and I_π^k. The key idea is then to apply Miyazawa's rate conservation law [38] to show that the size of the largest job in the system cannot be too big whenever the idleness $I_\pi^k > 0$. Formally, we apply the rate conservation law to the function $f(L) = (L - x)^+$, for arbitrary x. This value can change in the following ways:

- Work is done continuously whenever available. We denote the average continuous change in $f(L)$ by $\mathbf{E}[D_t f(L)]$.
- When a job arrives, it will increase L whenever its size is larger than L and x. By PASTA, the average change from this is given by $\lambda \mathbf{E}[(\max\{kS, L\} - x)^+ - (L - x)^+]$.

Miyazawa's rate conservation law tells us that

$$\mathbf{E}[D_t f(L)] + \lambda \mathbf{E}[(\max\{kS, L\} - x)^+ - (L - x)^+] = 0.$$

We now observe that $\mathbf{E}[D_t f(L)]$ is upper bounded by $\mathbf{E}[I_\pi^k \mathbb{1}(L > x)]$. This is because we can bound the rate of decrease of $f(L)$:

- If $0 < I_\pi^k < 1$ and $L > x$, then the largest job must be in service, since there are servers idle, and work is done at rate $1/k$ at each server. Therefore, $f(L)$ decreases at rate 1.
- If the above conditions do not hold, then $f(L)$ decreases at least at rate 0.

Therefore, $\mathbf{E}[D_t f(L)] \le -\mathbf{E}[\mathbb{1}(0 < I_\pi^k < 1)\mathbb{1}(L > x)]$. Since I_π^k is at most 1, $1(0 < I_\pi^k < 1) \ge I_\pi^k$, which implies that $-\mathbf{E}[\mathbb{1}(0 < I_\pi^k < 1)\mathbb{1}(L > x)] \le -\mathbf{E}[I_\pi^k \mathbb{1}(L > x)]$. Applying this yields

$$-\mathbf{E}[I_\pi^k \mathbb{1}(L > x)] + \lambda \mathbf{E}[(\max\{kS, L\} - x)^+ - (L - x)^+] \ge 0.$$

For the second term, we observe that:

- If $L < kS$, then it equals $(kS - x)^+ - (L - x)^+$.
- If $L \ge kS$, then it equals 0.

We can upper bound both cases by $(kS - x)^+$. We now have

$$-\mathbf{E}[I_\pi^k \mathbb{1}(L > x)] + \lambda \mathbf{E}[(kS - x)^+] \ge 0.$$

Some algebra and properties of S_e [3, 24] now yields:

$$\mathbf{E}[I_\pi^k \mathbb{1}(L > x)] \le \lambda \mathbf{E}[(kS - x)^+] = \lambda \frac{k\mathbf{E}[S]}{k\mathbf{E}[S]} \mathbf{E}[(kS - x)^+] = k\rho \mathbf{P}[kS_e > x] \le k\mathbf{P}[kS_e > x]. \quad \square$$

This immediately yields the following bound on the idleness-weighted work transform:

Lemma 3.8. *Let π be a non-idling policy. For a fixed $\varepsilon > 0$ and assuming $\mathbf{E}[e^{(\gamma+\varepsilon)kS_e}] < \infty$,*

$$\frac{\mathbf{E}[I_\pi^k e^{\gamma W_\pi^k}]}{1 - \rho} \le \frac{\gamma + \varepsilon}{\varepsilon} \left(\frac{k\mathbf{E}[e^{(\gamma+\varepsilon)S_e}]}{1 - \rho} \right)^{\gamma/(\gamma+\varepsilon)}.$$

Proof Sketch. The complete proof is in Appendix B. The proof essentially involves some key observations, and a direct application of the Last Job Lemma (Lemma 3.7). The first observation is that if $L = k \times$ (size of the largest job in the system), then by definition, $\mathbf{E}[I_\pi^k e^{\theta W_\pi^k}] \le \mathbf{E}[I_\pi^k e^{\theta L}]$, so it suffices to find a bound on $\frac{\mathbf{E}[I_\pi^k e^{\theta L}]}{1-\rho}$. The second observation is that because $\mathbf{E}[I_\pi^k] = 1 - \rho$ (Lemma A.2), we have $\frac{\mathbf{E}[I_\pi^k \mathbb{1}(L>x)]}{1-\rho} = \mathbf{P}_I[L > x]$, where $\mathbf{P}_I[\cdot]$ represents the probability measure given by $\mathbf{P}_I[A] = \frac{\mathbf{E}[I_\pi^k A]}{1-\rho}$, for an event A. Then a Chernoff bound argument applied to the Last Job Lemma implies

$$e^{\theta x} \mathbf{P}_I[L > x] \le \min\{ \frac{k}{1 - \rho} \mathbf{E}[e^{(\gamma+\varepsilon)kS_e}] e^{-(\gamma+\varepsilon)x} e^{\theta x}, e^{\theta x} \},$$

after which the conclusion follows from applying the tail integral formula (Lemma A.4) and taking limits. $\square$

Now we are ready to prove Theorem 3.6.

Proof of Theorem 3.6. First, we use the fact that $1/\rho = \mathbf{E}[e^{\gamma S_e}]$ (Lemma A.1, proof in Appendix A). Therefore, as $\rho \to 1$, we also have $\gamma \to 0$, and $\frac{1}{1-\rho} = \frac{\mathbf{E}[e^{\gamma S_e}]}{\mathbf{E}[e^{\gamma S_e}]-1}$. In particular, since we will be taking the heavy traffic limit, we can assume that we start at sufficiently high load ρ' (respectively, at $\gamma \in (0, \gamma']$) such that for some $\varepsilon > 0$, $\mathbf{E}[e^{(\gamma+\varepsilon)kS_e}] < \infty$ for all $\gamma \in (0, \gamma']$. Using Theorem 2.4, we know that

$$\begin{aligned}
\limsup_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta W_\pi}] &= \limsup_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta W_{\mathrm{M/G/1}}}]\frac{\mathbf{E}[I_\pi^k e^{\theta W_\pi^k}]}{1-\rho} \\
&= \left(\limsup_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta W_{\mathrm{M/G/1}}}]\right)\left(\limsup_{\theta\to\gamma} \frac{\mathbf{E}[I_\pi^k e^{\theta W_\pi^k}]}{1-\rho}\right) \\
&\le \left(\limsup_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta W_{\mathrm{M/G/1}}}]\right)\left(\frac{k\mathbf{E}[e^{(\gamma+\varepsilon)S_e}]}{1-\rho}\right)^{\gamma/(\gamma+\varepsilon)}\left(\frac{\gamma+\varepsilon}{\gamma}\right) \quad \text{(Lemma 3.8)} \\
&= \mathbf{C}^+[W_{\mathrm{M/G/1}}]\left(\frac{k\mathbf{E}[e^{(\gamma+\varepsilon)S_e}]}{1-\rho}\right)^{\gamma/(\gamma+\varepsilon)}\left(\frac{\gamma+\varepsilon}{\varepsilon}\right).
\end{aligned}$$

It therefore suffices to show that

$$\lim_{\rho\to 1}\left(\frac{k\mathbf{E}[e^{(\gamma+\varepsilon)S_e}]}{1-\rho}\right)^{\gamma/(\gamma+\varepsilon)}\left(\frac{\gamma+\varepsilon}{\gamma}\right) = 1.$$

Rewriting our desired limit in terms of γ—using again that $1/\rho = \mathbf{E}[e^{\gamma S_e}]$ (Lemma A.1):

$$\lim_{\gamma\to 0}\left(\frac{k\mathbf{E}[e^{(\gamma+\varepsilon)S_e}]\mathbf{E}[e^{\gamma S_e}]}{\mathbf{E}[e^{\gamma S_e}]-1}\right)^{\gamma/(\gamma+\varepsilon)}\left(\frac{\gamma+\varepsilon}{\gamma}\right).$$

We begin by analyzing the first term. Because we are taking $\gamma \to 0$ and all terms are positive, we can bound terms as follows:

$$\left(\frac{k}{\mathbf{E}[e^{\gamma S_e}]}\right)^{\gamma/(\gamma+\varepsilon)} \le \left(\frac{k\mathbf{E}[e^{(\gamma+\varepsilon)S_e}]\mathbf{E}[e^{\gamma S_e}]}{\mathbf{E}[e^{\gamma S_e}]-1}\right)^{\gamma/(\gamma+\varepsilon)} \le \left(\frac{k\mathbf{E}[e^{(\gamma'+\varepsilon)S_e}]\mathbf{E}[e^{\gamma' S_e}]}{\gamma\mathbf{E}[S_e]}\right)^{\gamma/(\gamma+\varepsilon)},$$

where

- For the lower bound, we use the fact that $1 \le \mathbf{E}[e^{\gamma S_e}] \le \mathbf{E}[e^{(\gamma+\varepsilon)S_e}]$. For the denominator, it is clear that we have increased its value.
- For the upper bound, the numerator uses the fact that $\mathbf{E}[e^{\gamma S_e}] \le \mathbf{E}[e^{\gamma' S_e}]$ and $\mathbf{E}[e^{(\gamma+\varepsilon)S_e}] \le \mathbf{E}[e^{(\gamma'+\varepsilon)S_e}]$. The denominator uses the fact that $e^x - 1 \ge x$ for all $x \ge 0$.

The numerator in both upper and lower bounds are finite and independent of γ, namely $k^{\gamma/(\gamma+\varepsilon)} \to 1$ as $\gamma \to 0$, and similarly $(k\mathbf{E}[e^{(\gamma'+\varepsilon)S_e}]\mathbf{E}[e^{\gamma' S_e}])^{\gamma/(\gamma+\varepsilon)} \to 1$ as $\gamma \to 0$.

For the lower bound, it suffices to show that $\lim_{\gamma\to 0} \frac{\gamma}{\gamma+\varepsilon}\log(1/\mathbf{E}[e^{\gamma S_e}]) = 0$. Clearly, the first term goes to 0, and by monotone convergence, the second term goes to $\log 1 = 0$, as desired. So the lower bound converges to 1. It now remains for the upper bound to show that $\lim_{\gamma\to 0} \frac{\gamma}{\gamma+\varepsilon}\log(\gamma\mathbf{E}[S_e]) = 0$. This can be rewritten as $\frac{\gamma}{\gamma+\varepsilon}\log\gamma + \frac{\gamma}{\gamma+\varepsilon}\log\mathbf{E}[S_e]$. Clearly, the second term converges to 0. The first term, by L'Hôpital's rule, also converges to 0. Therefore, both lower and upper bounds converge to 1 in the $\gamma \to 0$ limit. Finally, $(\gamma+\varepsilon)/\varepsilon \to 1$ as $\gamma \to 0$, so the entire limit converges to 1. □

3.3 Proof of Heavy-Traffic Optimality

Having bounded the effects of idleness in Sections 3.1 and 3.2, we are now ready to prove our main result, that γ-Boost is heavy-traffic optimal in the M/G/k. To proceed, we first need to confirm

a technical assumption for γ-Boost, namely that $\mathbf{E}[e^{\gamma k V(\infty)}]$ is finite. Specifically, we show the following:

Lemma 3.9. *Let $u \in \mathbb{R}_+ \cup \{\infty\}$. Then for γ-Boost, if $\mathbf{E}[e^{\gamma k S}] < \infty$, then*

$$\mathbf{E}[e^{\gamma k V(u)}] = \exp(\lambda \mathbf{E}[(e^{\gamma k S} - 1)\min\{B, u\}]) < \infty.$$

Proof. We consider each new arrival as a triple (b, s, t), as in [49, Lemma 3.5], and let X correspond to the set of random triples arriving after time 0, so that

$$V(u) = \sum_{(b,s,t)\in X} s\mathbb{1}(t \le \min\{b, u\}).$$

Then, $kV(u) = \sum_{(b,s,t)\in X} ks\mathbb{1}(t \le \min\{b, u\})$. To compute $\mathbf{E}[e^{\gamma k V(u)}]$, we apply Campbell's theorem [31, Section 3.2], which implies that

$$\mathbf{E}[e^{\gamma k V(u)}] = \exp(\lambda \mathbf{E}[(e^{\gamma k S} - 1)\min\{B, u\}]),$$

so long as the RHS is finite. For γ-Boost, we will show this is indeed the case. It suffices to show that $\lambda \mathbf{E}[(e^{\gamma k S} - 1)\min\{B, u\}]$ is finite. We have:

$$\begin{aligned}\lambda \mathbf{E}[(e^{\gamma k S} - 1)\min\{B, u\}] &\le \lambda \mathbf{E}[(e^{\gamma k S} - 1)\frac{1}{\gamma}\log\Big(\frac{e^{\gamma S}}{e^{\gamma S} - 1}\Big)] \qquad \text{(Equation (2.5))}\\ &= \frac{\lambda}{\gamma}\mathbf{E}[(e^{\gamma k S} - 1)\log\Big(\frac{e^{\gamma S}}{e^{\gamma S} - 1}\Big)].\end{aligned}$$

Recalling that $x \log \frac{x+1}{x} \le 1$, which implies that $\log \frac{x+1}{x} \le 1/x$, for all $x > 0$, we have:

$$\frac{\lambda}{\gamma}\mathbf{E}[\log\Big(\frac{e^{\gamma S}}{e^{\gamma S} - 1}\Big)(e^{\gamma k S} - 1)] \le \frac{\lambda}{\gamma}\mathbf{E}[(e^{\gamma k S} - 1)/(e^{\gamma S} - 1)].$$

Finally, we only need observe that $\mathbf{E}[(e^{\gamma k S} - 1)/(e^{\gamma S} - 1)] = \sum_{i=0}^{k-1} \mathbf{E}[e^{\gamma i S}]$. Since $\mathbf{E}[e^{\gamma k S}] < \infty$, each term in the sum is finite, so our expression is finite, as desired. □

Lemma 3.10. *Under γ-Boost,*

$$\lim_{\rho\to 1} \frac{\mathbf{E}[e^{\gamma(kS-B)}]}{\mathbf{E}[e^{\gamma(S-B)}]} = 1.$$

Proof. Since $1/\rho = \mathbf{E}[e^{\gamma S_e}]$ (Lemma A.1), as $\rho \to 1$, we have $\gamma \to 0$. Then it suffices to show

$$\begin{aligned}
\lim_{\gamma\to 0} \frac{\mathbf{E}[e^{\gamma(kS-B)}]}{\mathbf{E}[e^{\gamma(S-B)}]} &= \lim_{\gamma\to 0} \frac{\mathbf{E}[e^{\gamma kS}\frac{e^{\gamma S}-1}{e^{\gamma S}}]}{\mathbf{E}[e^{\gamma S}-1]} && \text{(Equation (2.5))}\\
&= \lim_{\gamma\to 0} \frac{\mathbf{E}[e^{\gamma(k-1)S}(e^{\gamma S}-1)]}{\mathbf{E}[e^{\gamma S}-1]}\\
&= \lim_{\gamma\to 0} \frac{\mathbf{E}[e^{\gamma(k-1)S}(e^{\gamma S}-1)]/\gamma}{\mathbf{E}[e^{\gamma S}-1]/\gamma}\\
&= \frac{\mathbf{E}[\lim_{\gamma\to 0} e^{\gamma(k-1)S}(e^{\gamma S}-1)/\gamma]}{\mathbf{E}[\lim_{\gamma\to 0}(e^{\gamma S}-1)/\gamma]} && \text{(Monotone convergence)}\\
&= \frac{\mathbf{E}[\lim_{\gamma\to 0} e^{\gamma kS}/\gamma - e^{\gamma(k-1)S}/\gamma]}{\mathbf{E}[\lim_{\gamma\to 0}(e^{\gamma S}-1)/\gamma]}\\
&= \frac{\mathbf{E}[\lim_{\gamma\to 0} kS + o(\gamma)/\gamma - (k-1)S - o(\gamma)/\gamma]}{\mathbf{E}[\lim_{\gamma\to 0} S + o(\gamma)/\gamma]}\\
&= \mathbf{E}[S]/\mathbf{E}[S] = 1.
\end{aligned}$$

The use of monotone convergence theorem is valid because $\frac{e^{\gamma S}-1}{\gamma}$ and $e^{\gamma(k-1)S}$ are both positive and increasing in γ. □

Proof of Theorem 3.1. One can take any policy in the M/G/k and run it in the M/G/1. Because γ-Boost is optimal in the M/G/1 across all policies [49], this implies that any policy in the M/G/k has tail constant at best equal to that of $\mathbf{C}^+[T^1_{\gamma\text{-Boost}}]$. In particular, we know that for all ρ,

$$\frac{\mathbf{C}^+[T^k_{\gamma\text{-Boost}}]}{\mathbf{C}^+[W_{\text{M/G/1}}]\mathbf{E}[e^{\gamma(S-B)}]\mathbf{E}[e^{\gamma V(\infty)}]} \geq 1,$$

where the denominator comes from [49, Theorem 3.1]. Therefore, it suffices to show that

$$\lim_{\rho\to 1} \frac{\mathbf{C}^+[T^k_{\gamma\text{-Boost}}]}{\mathbf{C}^+[W_{\text{M/G/1}}]\mathbf{E}[e^{\gamma(S-B)}]\mathbf{E}[e^{\gamma V(\infty)}]} \leq 1.$$

Since $1/\rho = \mathbf{E}[e^{\gamma S_e}]$ (Lemma A.1), as we take $\rho \to 1$, we have $\gamma \to 0$. Then, for sufficiently high load $\rho \in (\rho', 1)$, because the job size distribution is class I, we can assume that $\mathbf{E}[e^{\gamma kS}] < \infty$ and, since $\mathbf{E}[e^{\theta S_e}] = \frac{\mathbf{E}[e^{\theta S}]-1}{\gamma\mathbf{E}[S]}$ [24, Chapter 25], that there exists $\varepsilon > 0$ such that $\mathbf{E}[e^{(\gamma+\varepsilon)kS_e}] < \infty$. Under these assumptions, Lemma 3.8 implies that $\frac{\mathbf{E}[I^k_{\gamma\text{-Boost}} e^{\gamma W^k_{\gamma\text{-Boost}}}]}{1-\rho}$ is bounded, and Lemma 3.9 implies that $\mathbf{E}[e^{\gamma kV(\infty)}] < \infty$, so we can apply Theorem 3.3 to get:

$$\mathbf{C}^+[T^k_{\gamma\text{-Boost}}] \leq \mathbf{C}^+[W_{\text{M/G/1}}] \frac{\mathbf{E}[I^k_{\gamma\text{-Boost}} e^{\gamma W^k_{\gamma\text{-Boost}}}]}{1-\rho} \mathbf{E}[e^{\gamma(kS-B)}]\mathbf{E}[e^{\gamma V(\infty)}].$$

Therefore, it suffices to show that

$$\lim_{\rho\to 1} \frac{\mathbf{C}^+[W_{\text{M/G/1}}]\frac{\mathbf{E}[I^k_{\gamma\text{-Boost}} e^{\gamma W^k_{\gamma\text{-Boost}}}]}{1-\rho}}{\mathbf{C}^+[W_{\text{M/G/1}}]} \leq 1, \quad \lim_{\rho\to 1} \frac{\mathbf{E}[e^{\gamma(kS-B)}]}{\mathbf{E}[e^{\gamma(S-B)}]} \leq 1.$$

The first inequality follows immediately from the fact that γ-Boost is a non-idling policy, so Theorem 2.4 implies that the numerator is $\tilde{C}^+[W^k_{\gamma\text{-Boost}}]$ and Theorem 3.6 implies the desired inequality. The second inequality follows immediately from Lemma 3.10. □

4 Heavy-Traffic Optimality for Unknown Sizes

In this section we prove that γ-Surrogate is heavy-traffic optimal in the M/G/k where job sizes are unknown. Since we are considering the γ-Surrogate policy, we use the system model introduced by Harlev et al. [25], who introduced it, but extend it to the M/G/k setting. In particular, this means that the system model in this section is the same as the one used throughout this paper except for the following changes:

- Service is quantized, and we let the service quantum be length 1 without loss of generality.[7] (The arrival process remains continuous.)
- Jobs are modeled as independent *absorbing discrete-time Markov chains* with countable state space. When a job is served, its state advances once per unit of service, and the job completes when it enters the unique absorbing state. The intuition is that a job's state encodes all the information the scheduler has about the job.
- Scheduling policies must be *non-clairvoyant*. That is, they must choose which job's to serve using only the information available at the time, which is each job's state trajectory up to its current state.

In this setting, boost policies assign each job's boost based on its trajectory rather than its size. This means that a job's boost can change with service, which does not happen in the known size model. If a job's boosted arrival time exceeds that of a job in the queue, the boost policy will preempt it and replace it with the job in the queue.

Harlev et al. [25] introduced three boost policies and proved that they are tail constant optimal in the M/G/1 among all non-clairvoyant policies. We extend this result by showing that one of these policies, γ-Surrogate, is also heavy-traffic optimal in the M/G/k, using the same approach as in the known-size setting. To define the three policies we introduce the following notation:

- We write X_u for the random state of the job after u units of service.
- We denote a job's trajectory during its first u units of service as $X_{0:u} = (X_0, X_1, \ldots, X_u)$.
- We let S represent a job's size, which is the hitting time of the unique absorbing state.

The three policies are the following:

- The γ-Gittins boost policy has boost function

$$b_{\gamma\text{-Gittins}}(X_{0:u}) = \frac{1}{\gamma}\log\Gamma_\gamma(X_u) + \frac{1}{\gamma}\log\frac{e^\gamma}{e^\gamma - 1}.$$

  Here γ is the same solution to (2.2) as in the known-size model, $\Gamma_\gamma(x)$ is a variant of the *Gittins index* [11, 12] and defined in the appendix (Definition D.1), and the $\frac{1}{\gamma}\log\frac{e^\gamma}{e^\gamma-1}$ term is added by convention to ensure boosts are nonnegative.
- The γ-Surrogate boost policy is a version of γ-Gittins with decreasing boost function:

$$b_{\gamma\text{-Surrogate}}(X_{0:u}) = \min_{t\in\{0,\ldots,u\}} b_{\gamma\text{-Gittins}}(X_{0:t}).$$

[7] One can rescale time to study arbitrarily small service quanta. We study quantized service because this is what Harlev et al. [25] study, who in turn make this choice for purely technical reasons: prior work on the Gittins index in continuous time [4, 29, 30, 35] treats the traditional case of *time-discounted* rewards, whereas the Gittins index developed by Harlev et al. [25] is for *time-inflated* costs.

- The γ-Insulated boost policy is a "minimally preemptive" version of γ-Gittins:

$$b_{\gamma\text{-Insulated}}(X_{0:u}) = \begin{cases} b_{\gamma\text{-Gittins}}(X_{0:u}) & \text{if } b_{\gamma\text{-Gittins}}(X_{0:u}) = b_{\gamma\text{-Surrogate}}(X_{0:u}) \\ \infty & \text{otherwise.} \end{cases}$$

4.1 Proving Heavy-Traffic Optimality of γ-Surrogate in the M/G/k

We will follow the approach of Section 3 to prove heavy-traffic optimality of γ-Surrogate in the M/G/k.[8] In particular, we prove an analog of Theorem 3.1 for the unknown-size setting.

Theorem 4.1. *For an M/G/k with class I job size distribution, γ-Surrogate is optimal in the heavy traffic limit:*

$$\lim_{\rho\to 1} \frac{\mathbf{C}^+[T^k_{\gamma\text{-Surrogate}}]}{\mathbf{C}^+[T^1_{\gamma\text{-Surrogate}}]} \le \lim_{\rho\to 1} \frac{\mathbf{C}^+[W_{\text{M/G/1}}] \frac{\mathbf{E}[I^k_{\gamma\text{-Surrogate}} e^{\gamma W^k_{\gamma\text{-Surrogate}}}]}{1-\rho} \mathbf{E}[e^{\gamma(kS-\underline{B})}]\mathbf{E}[e^{\gamma V(0,\infty)}]}{\mathbf{C}^+[W_{\text{M/G/1}}]\mathbf{E}[e^{\gamma(S-\underline{B})}]\mathbf{E}[e^{\gamma V(0,\infty)}]} = 1,$$

where the numerator is an upper bound on the tail constant of γ-Surrogate in the M/G/k, and the denominator is the tail constant of γ-Surrogate in the M/G/1.

While this looks *notationally* almost identical to the statement of Theorem 3.1, it is important to note that both the boost term, $\underline{B}$, and the crossing work term, $V(0,\infty)$, are defined differently than their known-size setting counterparts. They do, however, capture the same ideas, adjusted appropriately for the new setting:

- The boost term, $\underline{B}$, represents the worst-ever boost experienced by a job under a boost policy π, and is defined as $\underline{B} = \min_{t\in\{0,\dots,S\}} b_\pi(X_{0:t})$. The intuition for why this quantity appears is that it determines the worst priority a job gets prior to completion, and thus determines which other work will eventually be prioritized over a job.
- The crossing work term, $V(0,\infty)$, is similar to the crossing work in the known-size setting (Notation 3.2(c)), but it is now possible for only certain parts of each arriving job to be included in the crossing work.

To prove Theorem 4.1, we replicate the proof of Theorem 3.1, reusing results from the known-size setting when possible, and otherwise proving analogues for the unknown-size setting. Since the ideas are the same, we present only an outline with a sketch of the proofs, and then present a complete proof in Appendix D.

Note that Theorem 3.6 and Lemma 3.8 hold for all non-idling policies, including γ-Gittins, γ-Surrogate, and γ-Insulated. The external result used in the proof Theorem 3.1 is [49, Theorem 3.1] which has a direct analogue in the unknown-size setting, [25, Theorem 4.11]. The only other policy specific results used in the proof of Theorem 3.1 are Theorem 3.3 and Lemmas 3.9 and 3.10:

- An analogue of Theorem 3.3 follows from considering the worst boosted arrival time of a tagged job and then carefully checking that each step of the proof still holds for the unknown-size setting definition of crossing work.
- An analogue of Lemma 3.9 follows almost immediately from the fact that boost is uniformly bounded in the unknown-size setting due to the service quantization, and so it is impossible for a job that arrives far in the future to affect the crossing work. See Lemma D.6.
- We prove an analogue of Lemma 3.10 in Lemma D.8. Just as in the known-size setting we expand the boost term using its definition and then interchange the limit and the expectation to get the desired result. Justifying the interchange in the unknown-size setting requires

[8] Our results in the unknown-size setting extend specifically to γ-Surrogate and not the other two policies introduced in Harlev et al. [25], γ-Gittins and γ-Insulated. This is because, as discussed in [45, Appendix A], nonmonotonic rank functions are difficult to analyze in the multiserver setting.

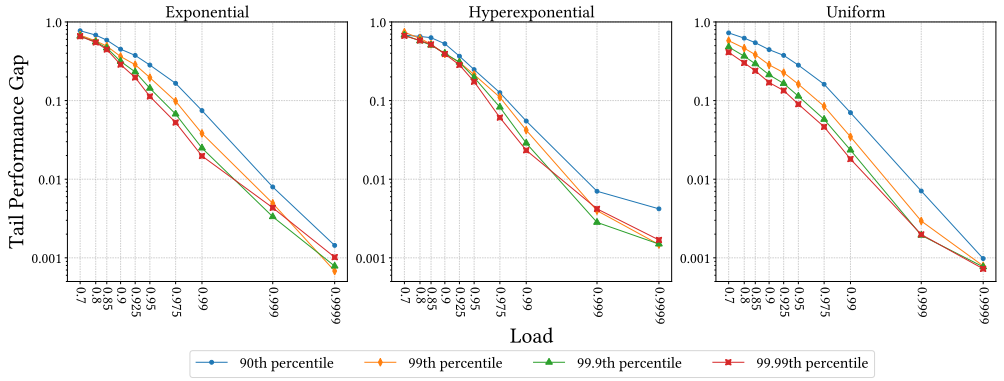


Fig. 5.1. (Lower is better.) γ-Boost's heavy-traffic performance for $k = 10$ servers and different job size distributions. We plot the Tail Performance Gap (TPG) against several loads and percentiles. Namely, let t_q^1 denote the qth percentile response time of γ-Boost in the M/G/1 and t_q^k denote the qth percentile response time of γ-Boost in the M/G/k. Then the TPG at the qth percentile for load ρ is given by $1 - t_q^1/t_q^k$. As load increases to 1, we would therefore expect the gap to go to 0 at higher percentiles. To compute values, we run all simulations with load below 0.975 with 200 million samples, and all simulations with load 0.975 and above with 2 billion samples. The job size distributions, from left to right, are Exp(1), Hyperexponential with branches drawn from Exp(2) and Exp(3) and first branch probability 0.8, and Uniform(0, 2).

proving that $\mathbf{E}[\sup_{0 \le i < S} \mathbf{E}[S \mid X_{0:i}]] < \infty$, and then using the dominated convergence theorem. We prove this in Lemma D.11 using classical martingale results.

With these analogues in hand, the proof of Theorem 4.1 is identical to that of Theorem 3.1.

5 Simulations

We have shown that in the heavy-traffic regime, γ-Boost is tail constant optimal among all policies and γ-Surrogate is tail constant optimal in the unknown size setting. Our results are asymptotic in nature, so the question of whether γ-Boost and γ-Surrogate perform well outside of this regime remains open. Analyzing scheduling policies theoretically in the multiserver setting outside of the heavy-traffic regime is difficult [16], so we perform an empirical study with simulations. For mean response times, SRPT, which is heavy-traffic optimal, is still among the best performing policies outside of the heavy-traffic regime [16, 18], so one would hope that the same would hold for tails. Surprisingly, our simulations show this is not the case; γ-Boost can often perform poorly even compared to FCFS. Specifically, we study the performance of policies in the following regimes. These are not formal definitions, but roughly characterize system behaviors that we have seen:

- The *heavy-traffic regime*, i.e. in the $\rho \to 1$ limit.
- The *low-load regime*. In this case, the M/G/k almost always has free servers, so one can think of the system as acting like an M/G/∞, where each server has speed $1/k$.
- The *moderate load regime*, which falls between the low-load and heavy-traffic regimes.

As we see in Section 5.3, where each regime begins and ends will depend on system details, such as the number of servers.

5.1 Heavy Traffic Performance

In Fig. 5.1, we evaluate the performance of γ-Boost as $\rho \to 1$. Namely, we compare the tail performance in the M/G/k against its performance in the M/G/1. Our theory predicts that as load increases, performance in the M/G/k should approach performance in the M/G/1. We see that at

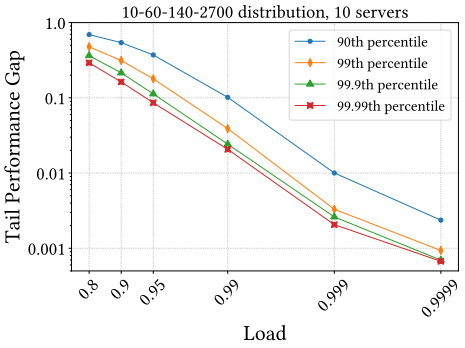


Fig. 5.2. (Lower is better.) γ-Gittins's heavy-traffic performance for $k = 10$ servers. We plot the Tail Performance Gap (TPG) against several loads and percentiles. For unknown sizes, the TPG at the qth percentile for load ρ is given by $1 - t_q^1/t_q^k$, where t_q^1 denotes the qth percentile response time of γ-Gittins in the M/G/1 and t_q^k denotes the qth percentile response time in the M/G/k. We run all simulations with load below 0.975 with 200 million samples, and all simulations with load 0.975 and above with 2 billion samples. The job size distribution is Unif$\{10, 60, 140, 2700\}$. As computing boost changes at every timestep would be computationally expensive, we use an insulated variant of γ-Gittins for simulation.

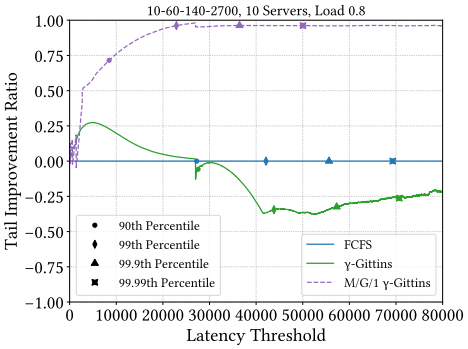


Fig. 5.3. (Higher is better.) γ-Gittins's performance in the low-load regime. Similar to γ-Boost, γ-Gittins significantly underperforms FCFS in this regime. We plot the Tail Improvement Ratio (TIR) against thresholds t. The TIR of a policy π is given by $1 - \mathbf{P}[T_\pi > t]/\mathbf{P}[T_{\text{FCFS}} > t]$. Simulations are run using 200 million jobs. The job size distribution is Unif$\{10, 60, 140, 2700\}$. As computing boost changes at every timestep would be computationally expensive, we use an insulated variant of γ-Gittins for simulation.

low loads, γ-Boost's performance in the M/G/k can be far from M/G/1 optimal, but at higher and higher loads, as our theory suggests, performance in the M/G/k converges to performance in the M/G/1. This suggests that in heavy traffic, scheduling with γ-Boost leads to the best achievable performance for the tail. We perform a similar evaluation in Fig. 5.2 when sizes are unknown, with similar results.

5.2 What Happens Outside of Heavy Traffic?

Outside of heavy traffic, the question of what to do for the tail becomes more complicated. In particular, under both the low-load and moderate-load regimes, we find that γ-Boost's performance suffers. The degradation can be large: as Fig. 5.4 shows, γ-Boost can *significantly underperform* FCFS outside the heavy-traffic regime. Figure 5.3 shows that γ-Gittins experiences a similar such degradation outside the heavy-traffic regime, relative to FCFS.

5.2.1 Hypotheses for Poor γ-Boost Performance. Why might γ-Boost underperform? There are several reasons this might be the case. The first has to do with jobs receiving implicit prioritization due to the service mechanics of an M/G/k system. Namely, jobs can enter service before work with priority over them completes, i.e., they receive some extra priority due to how server resources are partitioned in an M/G/k. Yu and Scully [49, Section 6.4] have observed that boosts which are too aggressive can lead to degraded tail performance, so if a similar over-prioritization is occurring here, it could negatively impact performance in the M/G/k.

Another factor that may arise in lower load settings is that of *job packing*; namely, that we may want to pack jobs so that they finish with a smaller makespan. Consider an example where we have 2 servers and a batch of three jobs, of size 1, 1, and 2, which have arrival times $-2, -1, -0$, respectively. Recall that with a single server, the rough intuition behind γ-Boost is that it attempts to minimize the exponential cost $\mathbf{E}[e^{\gamma T}]$,[9] which for this batch, is equivalent to minimizing: $\sum_{i=1}^{3} e^{\gamma(d_i - a_i)}$. With multiple servers, this is no longer the case. The boosted arrival times of these jobs would suggest serving both size 1 jobs, followed by the size 2 job. Because our servers run at speed $1/2$, the two size 1 jobs finish at time 2 and the size 2 job finishes at time 6. The cost of this schedule is $e^{\gamma(2-(-2))} + e^{\gamma(2-(-1))} + e^{\gamma(6-0)} = e^{4\gamma} + e^{3\gamma} + e^{6\gamma}$. We would do better, however, by prioritizing the larger job in this case: by serving jobs 1 and 3 first, then serving job 2 once job 1 finishes, we obtain a schedule with cost $e^{\gamma(2-(-2))} + e^{\gamma(4-(-1))} + e^{\gamma(4-0)} = e^{4\gamma} + e^{5\gamma} + e^{4\gamma}$. The difference between the two schedules is $e^{3\gamma} + e^{6\gamma} - (e^{4\gamma} + e^{5\gamma})$, which is positive for all $\gamma > 0$.

5.2.2 Corrective Boosting with γ-CombinedBoost. These factors suggest that we might try and compensate by *corrective boosting*: that is, boosting larger jobs more than small jobs. Namely, we consider the γ-CombinedBoost function, which combines γ-Boost with the corrective SizeBoost function:

$$b_{\text{SizeBoost}}(s) = (k-1)s, \quad b_{\gamma\text{-CombinedBoost}}(s) = b_{\gamma\text{-Boost}}(s) + b_{\text{SizeBoost}}(s).$$

The corrective $(k-1)s$ term gives large jobs more priority than small jobs. The choice of $k-1$ is somewhat arbitrary, but means that γ-CombinedBoost naturally reduces to γ-Boost when $k = 1$. It turns out that γ-CombinedBoost is also heavy-traffic optimal. We give the proof in Appendix F. It is almost identical to that of Theorem 3.1, with one key difference: in Theorem 3.1, γ-Boost has crossing work term (Notation 3.2(c)) identical to that of M/G/1 γ-Boost, in exchange for an altered boosting term, $\mathbf{E}[e^{\gamma(kS-B)}]$. The effect of the corrective boost in γ-CombinedBoost is to instead preserve the boosting term, in exchange for additional crossing work. Intuitively, at lower loads, one would expect the amount of crossing work jobs experience to be lower, so the effects of other terms might be more dominant.

Empirically, γ-CombinedBoost performs well at all load regimes we study. At low loads, the SizeBoost term dominates the γ-Boost term, and vice versa at high loads. We can see this in Fig. 5.4:

- In the low-load regime (left column), γ-CombinedBoost's performance is similar to that of SizeBoost's performance. For the unbounded distributions,[10] it actually approaches that of the M/G/∞ with speed $1/k$ servers, which provides a bound on how well any policy can do in the M/G/k.
- In the high-load regime (right column), γ-CombinedBoost's performance mirrors that of γ-Boost. This makes sense, as both policies are heavy-traffic optimal, and so will converge to the performance of M/G/1 γ-Boost.
- Finally, at moderate load (middle column), neither γ-Boost nor SizeBoost alone can outperform FCFS. γ-CombinedBoost, on the other hand, is able to achieve a performance improvement in all settings.

In summary, γ-CombinedBoost seems to not only obtain the best of both worlds in the light- and heavy-load regimes, but also exhibits performance that is more than the sum of its parts in moderate load, outperforming FCFS where neither of its constituent boost functions alone produces an improvement.

[9]This is of course only an informal statement, as $\mathbf{E}[e^{\gamma T_\pi}] = \infty$, but for a finite batch, it captures the right scheduling decisions. See [49, Section 1.5 and Section 4] for full details.

[10]For bounded distributions, such as the uniform distribution, the M/G/∞ effectively has $\mathbf{P}[T > s_{\max}] = 0$, so one should not expect policies to be able to match its performance.

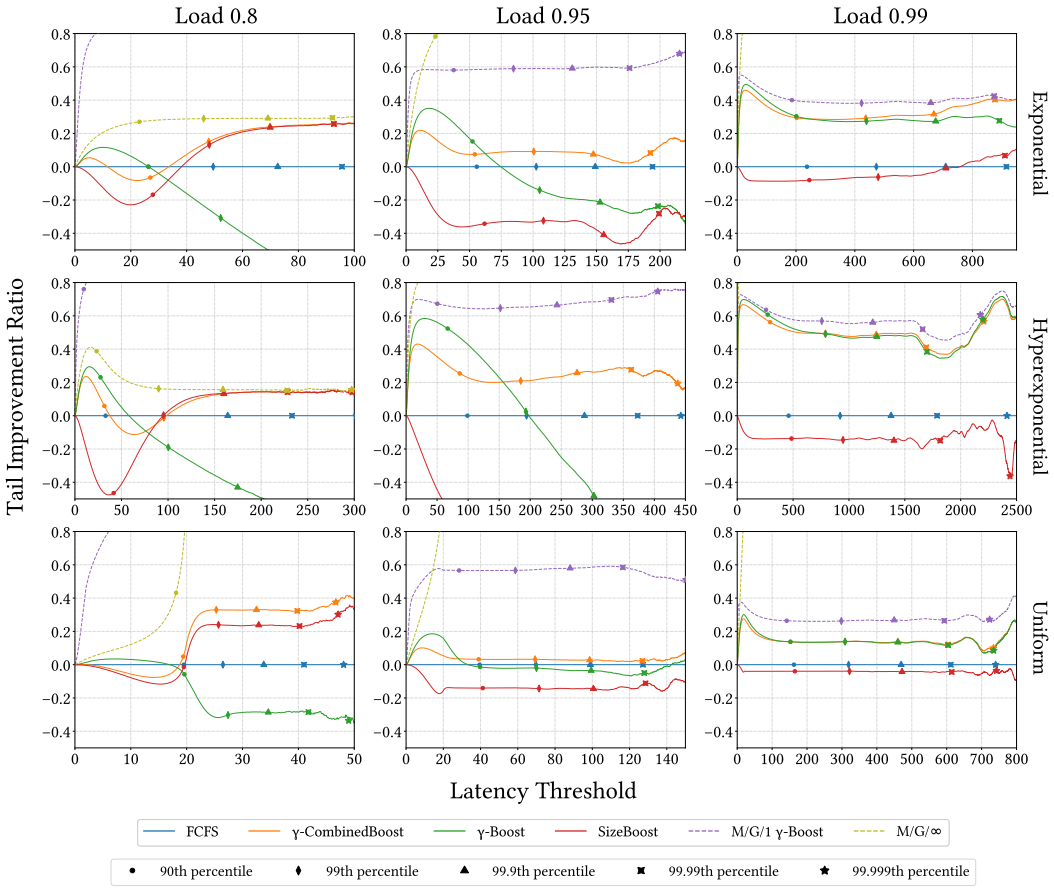


Fig. 5.4. (Higher is better.) Plot of performance of policies for $k = 10$ servers for different load regimes and job size distributions. We plot the Tail Improvement Ratio (TIR) of policies against thresholds t. The TIR of a policy π is given by $\text{TIR}(t) = 1 - \mathbf{P}[T_\pi > t]/\mathbf{P}[T_{\text{FCFS}} > t]$, where higher TIR means better performance. Simulations are run using 200 million jobs for loads 0.8 and load 0.95. For load 0.99, we run 2 billion jobs for convergence. The job size distributions are, from top row to bottom row, $\text{Exp}(1)$, Hyperexponential with branches drawn from $\text{Exp}(2)$ and $\text{Exp}(1/3)$ and first branch probability 0.8, and $\text{Uniform}(0, 2)$.

5.3 The Effect of More Servers

We run simulations under many more servers ($k = 100$), with results in Fig. 5.5. We find that, with this number of servers, there are still three different load regimes, but the thresholds at which those regimes change is different than that of the $k = 10$ server setting. Namely, we find that the low-load regime ends at much higher load than it did for $k = 10$ servers. In Fig. 5.5, even at load 0.975, our policies exhibit the same qualities as they did at load 0.8 for 10 servers:[11] γ-Boost performs poorly, while SizeBoost performs well, as does γ-CombinedBoost. At load 0.99, where γ-Boost used to perform well for $k = 10$ servers, it now performs poorly. The heavy-traffic regime kicks in at load 0.999, and we see that γ-Boost and γ-CombinedBoost approach the performance of M/G/1 γ-Boost, as theory predicts.

[11]In particular, we omit the plot at load 0.8 for $k = 100$ as the difference in performance across policies is negligible.

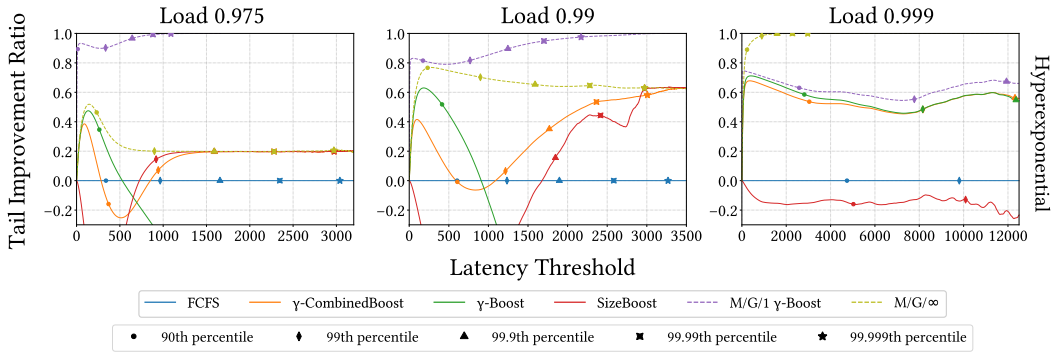


Fig. 5.5. (Higher is better.) Plot of performance of policies for $k = 100$ servers for different load regimes and job size distributions. The tail improvement ratio is as defined in Fig. 5.4. Simulations are run using 2 billion jobs for all loads. The job size distribution is Hyperexponential with branches drawn from Exp(2) and Exp(1/3) with first branch probability 0.8.

5.4 Is There Still Room For Improvement?

Our experiments from Section 5.2 suggest that using γ-CombinedBoost works well at all loads. A natural question to ask is the following: how much additional room for improvement is there? In Fig. 1.1, we present early experiments that suggest that additional improvements may be difficult to achieve. Due to computational requirements, however, we are only able to obtain results under limited samples, likely before convergence.

Recall that in the single-server setting, the key idea for γ-Boost comes from solving a deterministic scheduling problem [49, Section 4]. In the single server-setting, this problem can be solved easily by scheduling in boosted arrival time order under γ-Boost. In the multiserver setting, this scheduling problem becomes a mixed-integer nonlinear program that does not, to the best of our knowledge, have a simple solution. We can, however, attempt to optimize the objective function numerically. We do so using Gurobi [22], solving for the optimal nonpreemptive schedule of all jobs in the system on every new arrival. Our formulation is detailed in Appendix E. Because Gurobi is so computationally expensive, we are only able to obtain limited samples. From these samples, we observe in Fig. 1.1 that, despite γ-CombinedBoost's relative simplicity, it achieves performance roughly equivalent to scheduling according to solutions found by Gurobi. Because we have only limited samples, we leave a more detailed study of mathematical-programming-powered scheduling to future work.

6 Conclusion

In this work, we analyze how to optimize the tail of response time via scheduling in the M/G/k queueing model. We show that γ-Boost is tail constant optimal in the heavy-traffic limit; however, we also find that empirically, unlike for mean response time, good heavy-traffic performance is insufficient as a proxy for good performance at different loads. In particular, γ-Boost can significantly underperform FCFS at lighter loads. We take the first steps towards closing this gap with a new state-of-the-art scheduling policy, γ-CombinedBoost, which is theoretically heavy-traffic optimal, demonstrates state-of-the-art empirical performance at lighter loads. Our findings suggest some interesting questions to explore for future work.

For practitioners, the most immediate next step would be a more comprehensive empirical study. While FCFS already optimizes the decay rate, the additional improvements from optimizing the tail

constant can still be important. Yu and Scully [49] showed in the single server case that empirically, γ-Boost improves over FCFS for many common distributions, across all thresholds t, not just large ones. In the multiserver setting, our experiments in Section 5 are a first step towards quantifying what performance improvements are possible, but our exploration across server scales, distributions, and loads are far from a complete picture of multiserver tail scheduling. A more detailed study of Gurobi's performance would also help better quantify how much improvement is left on the table. Another practical consideration would be to reexamine the single-central-queue assumption, studying systems that combine immediate dispatching with scheduling.

On the theoretical side, all of our results are upper bounds, and we generally expect them to only be tight in heavy traffic. Whether one can prove complementary lower bounds remains open. We conjecture that one can prove lower bounds that match our Theorem 3.3 in heavy traffic. However, this is nontrivial even for FCFS, the simplest special case of Boost: existing results either show the existence of a tail constant without characterizing it [39, 41], or they show heavy-traffic distributional limit theorems that are a limit interchange away from characterizing the tail constant [32]. We suspect the techniques of Jhunjhunwala et al. [28] for the M/M/k could be extended to handle the FCFS case. Another avenue of exploration would be to study the behavior of γ-Boost and γ-CombinedBoost under different asymptotic regimes. In particular, understanding how γ-CombinedBoost and γ-Boost behave differently under regimes such as the Halfin-Whitt scaling [23] could provide new insight into the correct design decisions for multiserver tail scheduling.

Acknowledgments

This work was supported by the National Science Foundation (NSF) under grant nos. CMMI-2307008 and CNS-1955997. Amit Harlev was supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate (NDSEG) Fellowship Program (https://ndseg.sysplus.com/).

Code underlying plots and simulations was prepared in part using generative AI tools.

References

[1] Joseph Abate, Gagan L. Choudhury, and Ward Whitt. 1994. Waiting-Time Tail Probabilities in Queues with Long-Tail Service-Time Distributions. *Queueing Systems* 16, 3-4 (Sept. 1994), 311–338. doi:10.1007/BF01158960

[2] Joseph Abate and Ward Whitt. 1997. Asymptotics for $M/G/1$ Low-Priority Waiting-Time Tail Probabilities. *Queueing Systems* 25, 1 (June 1997), 173–233. doi:10.1023/A:1019104402024

[3] Søren Asmussen. 2003. *Applied Probability and Queues* (2 ed.). Number 51 in Stochastic Modelling and Applied Probability. Springer, New York, NY. doi:10.1007/b97236

[4] Peter Bank and Christian Küchler. 2007. On Gittins' Index Theorem in Continuous Time. *Stochastic Processes and their Applications* 117, 9 (Sept. 2007), 1357–1371. doi:10.1016/j.spa.2007.01.006

[5] Dimitris Bertsimas and José Niño-Mora. 1996. Conservation Laws, Extended Polymatroids and Multiarmed Bandit Problems; a Polyhedral Approach to Indexable Systems. *Mathematics of Operations Research* 21, 2 (May 1996), 257–306. doi:10.1287/moor.21.2.257

[6] Onno J. Boxma and Bert Zwart. 2007. Tails in Scheduling. *ACM SIGMETRICS Performance Evaluation Review* 34, 4 (March 2007), 13–20. doi:10.1145/1243401.1243406

[7] Nils Charlet and Benny Van Houdt. 2024. Tail Optimality and Performance Analysis of the Nudge-M Scheduling Algorithm. arXiv:2403.06588 [cs, math] doi:10.48550/arXiv.2403.06588

[8] Jie Chen, Kent H. Lundberg, Daniel E. Davison, and Dennis S. Bernstein. 2007. The Final Value Theorem Revisited - Infinite Limits and Irrational Functions. *IEEE Control Systems Magazine* 27, 3 (2007), 97–99. doi:10.1109/MCS.2007.365008

[9] Awi Federgruen and Harry Groenevelt. 1988. $M/G/c$ Queueing Systems with Multiple Customer Classes: Characterization and Control of Achievable Performance under Nonpreemptive Priority Rules. *Management Science* 34, 9 (1988), 1121–1138. doi:10.1287/mnsc.34.9.1121

[10] S. W. Fuhrmann and Robert B. Cooper. 1985. Stochastic Decompositions in the $M/G/1$ Queue with Generalized Vacations. *Operations Research* 33, 5 (Oct. 1985), 1117–1129. doi:10.1287/opre.33.5.1117

[11] 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. doi:10.1111/j.2517-6161.1979.tb01068.x

[12] John C. Gittins, Kevin D. Glazebrook, and Richard R. Weber. 2011. *Multi-Armed Bandit Allocation Indices* (2 ed.). Wiley, Chichester, UK. doi:10.1002/9780470980033

[13] Kevin D. Glazebrook. 2003. An Analysis of Klimov's Problem with Parallel Servers. *Mathematical Methods of Operations Research* 58, 1 (Sept. 2003), 1–28. doi:10.1007/s001860300278

[14] Kevin D. Glazebrook, David J. Hodge, Christopher Kirkbride, and R. J. Minty. 2014. Stochastic Scheduling: A Short History of Index Policies and New Approaches to Index Generation for Dynamic Resource Allocation. *Journal of Scheduling* 17, 5 (Oct. 2014), 407–425. doi:10.1007/s10951-013-0325-1

[15] Kevin D. Glazebrook and José Niño-Mora. 2001. Parallel Scheduling of Multiclass $M/M/m$ Queues: Approximate and Heavy-Traffic Optimization of Achievable Performance. *Operations Research* 49, 4 (Aug. 2001), 609–623. doi:10.1287/opre.49.4.609.11225

[16] Isaac Grosof. 2019. Open Problem—M/G/k/SRPT under Medium Load. *Stochastic Systems* 9, 3 (Sept. 2019), 297–298. doi:10.1287/stsy.2019.0042

[17] Isaac Grosof. 2023. *Optimal Scheduling in Multiserver Queues.* Ph. D. Dissertation. Carnegie Mellon University, Pittsburgh, PA. https://isaacg1.github.io/assets/isaac-thesis.pdf

[18] Isaac Grosof, Ziv Scully, and Mor Harchol-Balter. 2018. SRPT for Multiserver Systems. *Performance Evaluation* 127–128 (Nov. 2018), 154–175. doi:10.1016/j.peva.2018.10.001

[19] Isaac Grosof, Ziv Scully, Mor Harchol-Balter, and Alan Scheller-Wolf. 2022. Optimal Scheduling in the Multiserver-Job Model under Heavy Traffic. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 6, 3, Article 51 (Dec. 2022), 32 pages. doi:10.1145/3570612

[20] Isaac Grosof and Ziyuan Wang. 2024. Bounds on M/G/k Scheduling under Moderate Load Improving on SRPT-k and Tightening Lower Bounds. *ACM SIGMETRICS Performance Evaluation Review* 52, 2 (Sept. 2024), 24–26. doi:10.1145/3695411.3695421

[21] Isaac Grosof, Kunhe Yang, Ziv Scully, and Mor Harchol-Balter. 2021. Nudge: Stochastically Improving upon FCFS. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 5, 2, Article 21 (June 2021), 29 pages. doi:10.1145/3460088

[22] Gurobi Optimization, LLC. 2024. Gurobi Optimizer Reference Manual. https://www.gurobi.com

[23] Shlomo Halfin and Ward Whitt. 1981. Heavy-Traffic Limits for Queues with Many Exponential Servers. *Operations Research* 29, 3 (June 1981), 567–588. doi:10.1287/opre.29.3.567

[24] Mor Harchol-Balter. 2013. *Performance Modeling and Design of Computer Systems: Queueing Theory in Action.* Cambridge University Press, Cambridge, UK. doi:10.1017/CBO9781139226424

[25] Amit Harlev, George Yu, and Ziv Scully. 2025. A Gittins Policy for Optimizing Tail Latency. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 9, 2, Article 17 (June 2025), 40 pages. doi:10.1145/3727109

[26] Yige Hong and Ziv Scully. 2023. Performance of the Gittins Policy in the G/G/1 and G/G/k, with and without Setup Times. *ACM SIGMETRICS Performance Evaluation Review* 51, 2 (Sept. 2023), 33–35. doi:10.1145/3626570.3626583

[27] Donald L. Iglehart. 1972. Extreme Values in the GI/G/1 Queue. *The Annals of Mathematical Statistics* 43, 2 (April 1972), 627–635. doi:10.1214/aoms/1177692642

[28] Prakirt Raj Jhunjhunwala, Daniela Hurtado-Lange, and Siva Theja Maguluri. 2023. Exponential Tail Bounds on Queues: A Confluence of Non-Asymptotic Heavy Traffic and Large Deviations. arXiv:2306.10187 [math] doi:10.48550/arXiv.2306.10187

[29] Nicole El Karoui and Ioannis Karatzas. 1994. Dynamic Allocation Problems in Continuous Time. *The Annals of Applied Probability* 4, 2 (May 1994), 255–286. doi:10.1214/aoap/1177005062

[30] Haya Kaspi and Avishai Mandelbaum. 1998. Multi-Armed Bandits in Discrete and Continuous Time. *The Annals of Applied Probability* 8, 4 (Nov. 1998), 1270–1290. doi:10.1214/aoap/1028903380

[31] John F. C. Kingman. 1993. *Poisson Processes.* Number 3 in Oxford Studies in Probability. Oxford University Press, Oxford.

[32] Julian Köllerström. 1974. Heavy Traffic Theory for Queues with Several Servers. I. *Journal of Applied Probability* 11, 3 (Sept. 1974), 544–552. doi:10.2307/3212698

[33] Jan Karel Lenstra and David B. Shmoys. 2020. Elements of Scheduling. arXiv:2001.06005 [cs] doi:10.48550/arXiv.2001.06005

[34] Stefano Leonardi and Danny Raz. 2007. Approximating Total Flow Time on Parallel Machines. *J. Comput. System Sci.* 73, 6 (Sept. 2007), 875–891. doi:10.1016/j.jcss.2006.10.018

[35] Avi Mandelbaum. 1987. Continuous Multi-Armed Bandits and Multiparameter Processes. *The Annals of Probability* 15, 4 (Oct. 1987), 1527–1556. doi:10.1214/aop/1176991992

[36] Michel Mandjes and Onno Boxma. 2023. *The Cramér–Lundberg Model and Its Variants: A Queueing Perspective.* Springer, Cham, Switzerland. doi:10.1007/978-3-031-39105-7

[37] Masakiyo Miyazawa. 1994. Decomposition Formulas for Single Server Queues with Vacations : A Unified Approach by the Rate Conservation Law. *Communications in Statistics. Stochastic Models* 10, 2 (Jan. 1994), 389–413. doi:10.1080/

15326349408807301
[38] Masakiyo Miyazawa. 1994. Rate Conservation Laws: A Survey. *Queueing Systems* 15, 1 (March 1994), 1–58. doi:10.1007/BF01189231
[39] Masakiyo Miyazawa. 2017. A Unified Approach for Large Queue Asymptotics in a Heterogeneous Multiserver Queue. *Advances in Applied Probability* 49, 1 (March 2017), 182–220. doi:10.1017/apr.2016.84
[40] Michael Pinedo. 2016. *Scheduling: Theory, Algorithms, and Systems* (5 ed.). Springer, Cham, Switzerland.
[41] John S. Sadowsky and Wojciech Szpankowski. 1995. The Probability of Large Queue Lengths and Waiting Times in a Heterogeneous Multiserver Queue I: Tight Limits. *Advances in Applied Probability* 27, 2 (June 1995), 532–566. doi:10.2307/1427838
[42] Ziv Scully. 2022. Bounding Mean Slowdown in Multiserver Systems. *ACM SIGMETRICS Performance Evaluation Review* 49, 2 (Jan. 2022), 36–38. doi:10.1145/3512798.3512812
[43] Ziv Scully. 2022. *A New Toolbox for Scheduling Theory.* Ph. D. Dissertation. Carnegie Mellon University, Pittsburgh, PA. https://ziv.codes/pdf/scully-thesis.pdf
[44] 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 (Dec. 2020), 29 pages. doi:10.1145/3428328
[45] Ziv Scully, Isaac Grosof, and Mor Harchol-Balter. 2021. Optimal Multiserver Scheduling with Unknown Job Sizes in Heavy Traffic. *Performance Evaluation* 145, Article 102150 (Jan. 2021), 31 pages. doi:10.1016/j.peva.2020.102150
[46] Daniel W. Stroock. 2011. *Probability Theory: An Analytic View* (2 ed.). Cambridge University Press, Cambridge, UK.
[47] Benny Van Houdt. 2022. On the Stochastic and Asymptotic Improvement of First-Come First-Served and Nudge Scheduling. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 6, 3 (Dec. 2022), 1–22. doi:10.1145/3570610
[48] Adam Wierman and Bert Zwart. 2012. Is Tail-Optimal Scheduling Possible? *Operations Research* 60, 5 (Oct. 2012), 1249–1257. doi:10.1287/opre.1120.1086
[49] George Yu and Ziv Scully. 2024. Strongly Tail-Optimal Scheduling in the Light-Tailed M/G/1. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 8, 2, Article 27 (June 2024), 33 pages. doi:10.1145/3656011

A Miscellaneous proofs

Lemma A.1. *The decay rate γ satisfies $1/\rho = \mathbf{E}[e^{\gamma S_e}]$.*

Proof. Since the job size distribution is class I, we have $\mathbf{E}[e^{\gamma S_e}] = \frac{\mathbf{E}[e^{\gamma S}]-1}{\gamma \mathbf{E}[S]}$ [24, Chapter 25]. Then this follows immediately from (2.2). □

Lemma A.2. *For any non-idling scheduling policy π in the M/G/k,*

$$\mathbf{E}[I_\pi^k] = 1 - \rho.$$

Proof. We apply Miyazawa's rate conservation law [38] to W. Work is done continuously whenever available, at rate $1/k$ for each occupied server. Therefore, by the definition of I_π^k, the average continuous change from work being completed is $\mathbf{E}[-1 + I_\pi^k]$. The rate conservation law gives

$$\mathbf{E}[-1 + I_\pi^k] + \lambda \mathbf{E}[(W + S) - W] = 0,$$

so, recognizing that $\rho = \lambda \mathbf{E}[S]$ and rearranging yields the desired result. □

Lemma A.3. *For any random variable $X \geq 0$,*

$$\mathbf{C}^-[X] \leq \tilde{\mathbf{C}}^-[X] \leq \tilde{\mathbf{C}}^+[X] \leq \mathbf{C}^+[X].$$

Proof. First, we show that $\tilde{\mathbf{C}}^+[X] \leq \mathbf{C}^+[X]$. By the definition of lim sup, for any $\delta > 0$, there exists $m_\delta > 0$ s/t $\mathbf{C}^+[X] + \delta > e^{\gamma t}\mathbf{P}[X > t]$ for all $t > m_\delta$. Therefore, fixing a $\delta > 0$ and applying

the tail integral formula (Lemma A.4) yields

$$\begin{aligned}\frac{\varepsilon \mathbf{E}[e^{(\gamma-\varepsilon)X}-1]}{\gamma-\varepsilon} &= \varepsilon\int_0^\infty \mathbf{P}[X>t]e^{(\gamma-\varepsilon)t}\,\mathrm{d}t\\ &\le \varepsilon\frac{e^{(\gamma-\varepsilon)m_\delta}-1}{\gamma-\varepsilon}+\varepsilon\int_{m_\delta}^\infty (\mathbf{C}^+[X]+\delta)e^{-\varepsilon t}\,\mathrm{d}t\\ &= \varepsilon\frac{e^{(\gamma-\varepsilon)m_\delta}-1}{\gamma-\varepsilon}+(\mathbf{C}^+[X]+\delta)e^{-\varepsilon m_\delta}.\end{aligned}$$

Taking the lim sup as $\varepsilon\to 0$ on both sides yields

$$\tilde{\mathbf{C}}^+[X]\le \mathbf{C}^+[X]+\delta,$$

and since our choice of $\delta>0$ was arbitrary, we have $\tilde{\mathbf{C}}^+[X]\le \mathbf{C}^+[X]$, as desired.

Next, we show that $\mathbf{C}^-[X]\le \tilde{\mathbf{C}}^-[X]$. By the definition of lim inf, for any $\delta>0$, there exists $m_\delta>0$ s/t $\mathbf{C}^-[X]-\delta<e^{\gamma t}\mathbf{P}[X>t]$ for all $t>m_\delta$. Therefore, fixing a $\delta>0$ and applying the tail integral formula (Lemma A.4) yields:

$$\begin{aligned}\frac{\varepsilon \mathbf{E}[e^{(\gamma-\varepsilon)X}-1]}{\gamma-\varepsilon} &= \varepsilon\int_0^\infty \mathbf{P}[X>t]e^{(\gamma-\varepsilon)t}\,\mathrm{d}t\\ &\ge \varepsilon\int_{m_\delta}^\infty (\mathbf{C}^-[X]-\delta)e^{-\varepsilon t}\,\mathrm{d}t\\ &= (\mathbf{C}^-[X]-\delta)e^{-\varepsilon m_\delta},\end{aligned}$$

and taking the lim inf as $\varepsilon\to 0$ on both sides yields

$$\tilde{\mathbf{C}}^-[X]\ge \mathbf{C}^-[X]-\delta,$$

and since our choice of $\delta>0$ was arbitrary, we have $\tilde{\mathbf{C}}^-[X]\ge \mathbf{C}^-[X]$, as desired. Finally, $\tilde{\mathbf{C}}^-[X]\le \tilde{\mathbf{C}}^+[X]$ is immediate from the definitions of lim inf and lim sup. □

Lemma A.4 (Tail Integral Formula). *Let X be a nonnegative random variable and $f:[0,\infty)\to[0,\infty)$ be an increasing differentiable function, i.e. $f'(t)\ge 0$ for all $t\ge 0$. Then*

$$\mathbf{E}[f(X)] = f(0)+\int_0^\infty f'(t)\,\mathbf{P}[X>t]\,\mathrm{d}t.$$

Proof. We can write $f(x)$ as

$$f(x) = f(0)+\int_0^\infty f'(t)\,\mathbb{1}(t<x)\,\mathrm{d}t,$$

from which we get

$$\begin{aligned}\mathbf{E}[f(X)] &= f(0)+\mathbf{E}\left[\int_0^\infty f'(t)\,\mathbb{1}(t<X)\,\mathrm{d}t\right]\\ &= f(0)+\int_0^\infty \mathbf{E}[f'(t)\,\mathbb{1}(t<X)]\,\mathrm{d}t\\ &= f(0)+\int_0^\infty f'(t)\,\mathbf{P}[X>t]\,\mathrm{d}t,\end{aligned}$$

where the interchange of integral and expectation is justified by Tonelli's theorem. □

B Proofs for Results on Wasted Work

Lemma 3.8. *Let π be a non-idling policy. For a fixed $\varepsilon > 0$ and assuming $\mathbf{E}[e^{(\gamma+\varepsilon)kS_e}] < \infty$,*

$$\frac{\mathbf{E}[I_\pi^k e^{\gamma W_\pi^k}]}{1-\rho} \le \frac{\gamma+\varepsilon}{\varepsilon}\left(\frac{k\mathbf{E}[e^{(\gamma+\varepsilon)S_e}]}{1-\rho}\right)^{\gamma/(\gamma+\varepsilon)}.$$

Proof. First, observe that $\mathbf{E}[I_\pi^k e^{\gamma W_\pi^k}] \le \mathbf{E}[I_\pi^k e^{\gamma L}]$, where L is $k \times$ (max job size in the system). This is because both terms are 0 when $I_\pi^k = 0$, and $L \ge W_\pi^k$ whenever $I \ne 0$. It therefore suffices to bound $\mathbf{E}[I_\pi^k e^{\gamma L}]$ by applying Lemma 3.7.

We have

$$\frac{1}{1-\rho}\mathbf{E}[I_\pi^k \mathbb{1}(L > x)] \le \frac{k}{1-\rho}\mathbf{P}[kS_e > x].$$

We know that $\mathbf{E}[I_\pi^k] = 1-\rho$ by Lemma A.2, so that the LHS can be thought of as an expectation under a change of measure, namely under the probability measure $\mathbf{P}_I[\cdot]$, where $\mathbf{P}_I[A] = \frac{\mathbf{E}[I_\pi^k \mathbb{1}(A)]}{1-\rho}$ for any event A.

Therefore, we have that $\mathbf{P}_I[L > x] \le \frac{k}{1-\rho}\mathbf{P}[kS_e > x]$. By assumption, we have that $\mathbf{E}[e^{(\gamma+\varepsilon)kS_e}] < \infty$, and a Chernoff bound argument on the RHS yields the bound

$$\mathbf{P}_I[L > x] \le \min\left\{\frac{k}{1-\rho}\mathbf{E}[e^{(\gamma+\varepsilon)kS_e}]e^{-(\gamma+\varepsilon)x}, 1\right\}.$$

Let $b = \frac{1}{\gamma+\varepsilon}\log\left(\frac{k\mathbf{E}[e^{(\gamma+\varepsilon)S_e}]}{1-\rho}\right)$. Observe that for $x \le b$, $\frac{k}{1-\rho}\mathbf{E}[e^{(\gamma+\varepsilon)kS_e}]e^{-(\gamma+\varepsilon)x} \ge 1$, and for $x > b$, it is less than 1. For any $\theta < \gamma$, we have the bounds

$$e^{\theta x}\mathbf{P}_I[L > x] \le \min\left\{\frac{k}{1-\rho}\mathbf{E}[e^{(\gamma+\varepsilon)kS_e}]e^{-(\gamma+\varepsilon)x}e^{\theta x}, e^{\theta x}\right\}.$$

Integrating both sides and applying the tail integral formula (Lemma A.4) to the LHS yields

$$\frac{\mathbf{E}_I[e^{\theta L}-1]}{\theta} \le \int_0^b e^{\theta x}\,\mathrm{d}x + \frac{k\mathbf{E}[e^{(\gamma+\varepsilon)kS_e}]}{1-\rho}\int_b^\infty e^{(\theta-(\gamma+\varepsilon))x}\,\mathrm{d}x.$$

Now computation yields

$$\mathbf{E}_I[e^{\theta L}] - 1 \le e^{\theta b} - 1 - \frac{k\mathbf{E}[e^{(\gamma+\varepsilon)kS_e}]}{1-\rho}\frac{\theta}{\theta-(\gamma+\varepsilon)}e^{(\theta-(\gamma+\varepsilon))b},$$

and, taking the $\theta \to \gamma$ limit, monotone convergence theorem yields

$$\mathbf{E}_I[e^{\gamma L}] - 1 \le e^{\gamma b} - 1 + \frac{k\mathbf{E}[e^{(\gamma+\varepsilon)kS_e}]}{1-\rho}\frac{\gamma}{\varepsilon}e^{-\varepsilon b}.$$

Finally, we have:

$$\mathbf{E}_I[e^{\gamma L}] \le e^{\gamma b} + \frac{k\mathbf{E}[e^{(\gamma+\varepsilon)kS_e}]}{1-\rho}\frac{\gamma}{\varepsilon}e^{-\varepsilon b}$$

$$\mathbf{E}_I[e^{\gamma L}] \le e^{\gamma b}\left(1 + \frac{k\mathbf{E}[e^{(\gamma+\varepsilon)kS_e}]}{1-\rho}\frac{\gamma}{\varepsilon}e^{-(\gamma+\varepsilon)b}\right).$$

Plugging in for the value of b yields the desired inequality. □

C Proof of Bound on Boost Policy Response Time Transform

Theorem 3.3. *Let π be a boost policy and suppose that $\mathbf{E}[e^{\gamma k V(\infty)}] < \infty$, $\mathbf{E}[e^{\gamma k S}] < \infty$, and $\mathbf{E}[I_\pi^k e^{\gamma W_\pi^k}] < \infty$. The tail constant of π can be bounded above as follows:*

$$\mathbf{C}^+[T_\pi^k] \le \mathbf{C}^+[W_{\mathrm{M/G/1}}] \frac{\mathbf{E}[I_\pi^k e^{\gamma W_\pi^k}]}{1-\rho} \mathbf{E}[e^{\gamma(kS-B)}] \mathbf{E}[e^{\gamma V(\infty)}].$$

Proof. Fix an arbitrary $u \ge 0$, and let $\hat{B} = \min\{B, u\}$. First we analyze the transforms of $W_\pi^k - \hat{B} + kS + V_\pi(\infty)$ and $k(S + V_\pi(\infty) + \overline{V}(\hat{B}))$. We have:

$$\begin{aligned}
\mathbf{E}[e^{\theta(W_\pi^k - \hat{B} + V_\pi(\infty) + kS)}] &= \mathbf{E}[e^{\theta W_\pi^k}] \mathbf{E}[e^{\theta(kS-\hat{B})}] \mathbf{E}[e^{\theta V_\pi(\infty)}] && \text{(Independence of } W_\pi^k, S, \text{ and } V) \\
&= \mathbf{E}[e^{\theta W_{\mathrm{M/G/1}}}] \frac{\mathbf{E}[I_\pi^k e^{\theta W_\pi^k}]}{1-\rho} \mathbf{E}[e^{\theta(kS-\hat{B})}] \mathbf{E}[e^{\theta V_\pi(\infty)}]. && \text{(Theorem 2.4.)}
\end{aligned}$$

$$\mathbf{E}[e^{\theta k(S + V_\pi(\infty) + \overline{V}(\hat{B}))}] = \mathbf{E}[e^{\theta k V_\pi(\infty)}] \mathbf{E}[e^{\theta k \overline{V}(\hat{B})}] \mathbf{E}[e^{\theta k S}] \qquad \text{(Independence of } V, \overline{V}, \text{ and } S.)$$

By our finiteness assumptions on $\mathbf{E}[e^{\gamma k V_\pi(\infty)}], \mathbf{E}[e^{\gamma k S}], \frac{\mathbf{E}[I_\pi^k e^{\gamma W_\pi^k}]}{1-\rho}$, the transforms above can only have a pole where the transform of W_π^k has a pole, so a final value theorem implies that

$$\begin{aligned}
&\mathbf{C}^+[W_\pi^k - \hat{B} + kS + V_\pi(\infty)] \\
&\quad = \tilde{\mathbf{C}}^+[W_\pi^k - \hat{B} + kS + V_\pi(\infty)] \\
&\quad = \lim_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma} \mathbf{E}[e^{\theta(W_\pi^k + V(\infty) - \hat{B} + kS)}] \\
&\quad = \lim_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma} \mathbf{E}[e^{\theta W_{\mathrm{M/G/1}}}] \frac{\mathbf{E}[I_\pi^k e^{\theta W_\pi^k}]}{1-\rho} \mathbf{E}[e^{\theta(kS-\hat{B})}] \mathbf{E}[e^{\theta V(\infty)}] \\
&\quad = \left(\lim_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma} \mathbf{E}[e^{\theta W_{\mathrm{M/G/1}}}] \frac{\mathbf{E}[I_\pi^k e^{\theta W_\pi^k}]}{1-\rho}\right) \left(\lim_{\theta \to \gamma} \mathbf{E}[e^{\theta(kS-\hat{B})}]\right) \left(\lim_{\theta \to \gamma} \mathbf{E}[e^{\theta V(\infty)}]\right) \\
&\quad = \mathbf{C}^+[W_{\mathrm{M/G/1}}] \frac{\mathbf{E}[I_\pi^k e^{\gamma W_\pi^k}]}{1-\rho} \mathbf{E}[e^{\gamma(kS-\hat{B})}] \mathbf{E}[e^{\gamma V(\infty)}] && \text{(Monotone convergence.)}
\end{aligned}$$

and

$$\begin{aligned}
&\mathbf{C}^+[k(V(\infty) + \overline{V}(\hat{B}) + kS)] \\
&\quad = \tilde{\mathbf{C}}^+[k(V(\infty) + \overline{V}(\hat{B}) + kS)] \\
&\quad = \lim_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma} \mathbf{E}[e^{\theta k(V(\infty) + \overline{V}(\hat{B}) + kS)}] \\
&\quad = \lim_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma} \mathbf{E}[e^{\theta k V(\infty)}] \mathbf{E}[e^{\theta k \overline{V}(\hat{B})}] \mathbf{E}[e^{\theta k S}] \\
&\quad = 0 \cdot \mathbf{E}[e^{\gamma k V(\infty)}] \mathbf{E}[e^{\gamma k \overline{V}(\hat{B})}] \mathbf{E}[e^{\gamma k S}] && \text{(Monotone convergence.)} \\
&\quad = 0.
\end{aligned}$$

The final line follows from the fact that all three terms $\mathbf{E}[e^{\gamma k V(\infty)}], \mathbf{E}[e^{\gamma \overline{V}(\hat{B})}], \mathbf{E}[e^{\gamma k S}]$ are all finite:

- $\mathbf{E}[e^{\gamma k V(\infty)}]$ is finite by assumption.
- $\mathbf{E}[e^{\gamma k S}]$ is finite by assumption.

- $\mathbf{E}[e^{\gamma k\overline{V}(\hat{B})}] \le \mathbf{E}[e^{\gamma kA(u)}]$, where $A(u)$ is the amount of work that arrives to the system in an interval of length u. $\mathbf{E}[e^{\gamma kA(u)}] = \exp(\lambda u(\mathbf{E}[e^{\gamma kS}] - 1))$ by a standard M/G/1 result [24, Chapter 25.6]. This is finite from the above assumption.

Applying Lemma 3.5 then yields:

$$\begin{aligned}
\mathbf{C}^+[T_\pi^k] &= \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T_\pi^k > t] \\
&\le \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[W_\pi^k - \min\{B,u\} + kS + V_\pi(\infty) > t] + e^{\gamma t}\mathbf{P}[k(S + V_\pi(\infty) + \overline{V}(\min\{B,u\})) > t] \\
&= \mathbf{C}^+[W_\pi^k - \hat{B} + kS + V_\pi(\infty)] + \mathbf{C}^+[k(S + V_\pi(\infty)) + \overline{V}(\hat{B})] \\
&= \tilde{\mathbf{C}}^+[W_\pi^k - \hat{B} + kS + V_\pi(\infty)] + \tilde{\mathbf{C}}^+[k(S + V_\pi(\infty)) + \overline{V}(\hat{B})] \\
&= \mathbf{C}^+[W_{\mathrm{M/G/1}}]\frac{\mathbf{E}[I_\pi^k e^{\gamma W_\pi^k}]}{1-\rho}\mathbf{E}[e^{\gamma(kS-\hat{B})}]\mathbf{E}[e^{\gamma V(\infty)}].
\end{aligned}$$

Because this holds for all u, it also holds in the $u \to \infty$ limit. Monotone convergence therefore yields

$$\mathbf{C}^+[T_\pi^k] \le \mathbf{C}^+[W_{\mathrm{M/G/1}}]\frac{\mathbf{E}[I_\pi^k e^{\gamma W_\pi^k}]}{1-\rho}\mathbf{E}[e^{\gamma(kS-B)}]\mathbf{E}[e^{\gamma V(\infty)}]. \qquad \square$$

D Proof of γ-Surrogate Heavy-Traffic Optimality in the M/G/k Unknown-Size Setting

In this appendix we provide a complete proof of Theorem 4.1, following the outline in Section 4.1. To do this we first introduce the system model used in Harlev et al. [25] and summarize their main result. This is done in Appendices D.1 and D.2. Then in Appendix D.3, we prove the lemmas described in Section 4.1 and use them to prove Theorem 4.1.

D.1 System Model

Each job is modeled as an *absorbing discrete-time Markov chains with countable state space* that is independent of the arrival process and all other jobs in the system. The state of the job contains all information about the job relevant to the scheduler and advances once per unit of service. All jobs are assumed to share a state space $\mathbb{X} \sqcup \{x_{\mathrm{done}}\}$ and have the same Markovian dynamics. Each job is initialized at a state drawn from distribution X_{new}, and completes and exits the system when it reaches x_{done}.

We use the following notation for the Markovian job model:

- We write X_u for the random state of the job after u units of service.
- We let S represent a job's size, which is the hitting time of the completion state.[12] That is, $S = \min\{u \ge 0 : X_u = x_{\mathrm{done}}\}$.
- We denote a job's trajectory during its first u units of service as $X_{0:u} = (X_0, X_1, \ldots, X_u)$. A job's trajectory is only defined up to S units of service and it is always the case that $X_S = x_{\mathrm{done}}$.

Just as in the known-size model, the job size distribution S is assumed to be class I (Assumption 2.5). Additionally, we assume without loss of generality that for every state $y \in \mathbb{X}$, there is some state x with non-zero probability mass in X_{new} such that there is positive probability of reaching y starting from x. If a state does not satisfy this condition, it is unreachable by all jobs and has no impact on the system.

[12] Just as in [25], we mildly abuse terminology by writing S for both the job size *distribution* and, when convenient, the *random variable* with that distribution corresponding to a generic job's random size. We do the same for other distributions in this section without further comment.

Note that while this *job model* is discrete, we are still considering a continuous-time M/G/k. In practice, the discrete time model just means that each job's service is divided into time units of length 1 and jobs cannot be preempted during a unit of service. Arrivals can still occur at any time.

D.2 Boost Policies for Markov Jobs

Boost policies can no longer depend on the size of the job, as sizes are unknown to the scheduler. Instead, boost policies for Markov jobs map each job's trajectory to a boost. They then operate in much the same way as in the known-size model: serve the jobs in order from least to greatest boosted arrival time, which is defined as a job's arrival time minus its boost. Notably, since a job's boost depends on its trajectory, its boost may change with service, which does not happen in the known-size model. If a job's boosted arrival time exceeds that of a job in the queue, the boost policy will preempt it and replace it with the job in the queue. Markov job boost policies are required to assign every job that has not yet attained any service a finite boost with probability 1. This is ensure that at most k jobs at a time have boost ∞ and thus that $(k+1)$-way ties are probability-zero events.

Harlev et al. [25] introduced three related boost policies for Markov jobs:

- The γ-Gittins boost policy has boost function

$$b_{\gamma\text{-Gittins}}(X_{0:u}) = \frac{1}{\gamma}\log\Gamma_\gamma(X_u) + \frac{1}{\gamma}\log\frac{e^\gamma}{e^\gamma - 1}.$$

  Here γ is the same solution to (2.2) as in the known-size model, $\Gamma_\gamma(x)$ is a variant of the *Gittins index* [11, 12] and defined below (Definition D.1), and the $\frac{1}{\gamma}\log\frac{e^\gamma}{e^\gamma-1}$ term is added by convention to ensure boosts are nonnegative.
- The γ-Surrogate boost policy is a version of γ-Gittins with decreasing boost function:

$$b_{\gamma\text{-Surrogate}}(X_{0:u}) = \min_{t\in\{0,\dots,u\}} b_{\gamma\text{-Gittins}}(X_{0:t}).$$

- The γ-Insulated boost policy is a "minimally preemptive" version of γ-Gittins:

$$b_{\gamma\text{-Insulated}}(X_{0:u}) = \begin{cases} b_{\gamma\text{-Gittins}}(X_{0:u}) & b_{\gamma\text{-Gittins}}(X_{0:u}) = b_{\gamma\text{-Surrogate}}(X_{0:u}) \\ \infty & \text{otherwise.} \end{cases}$$

The primary result of Harlev et al. [25] is that all three of these policies are tail constant optimal in the M/G/1 among all *non-clairvoyant* policies, that is, policies that choose which job to serve using only the information available at the time: the trajectories of all jobs in the system up to their current states. Formally, they proved that for all non-clairvoyant policies π,

$$\mathbf{C}^+[T^1_\pi] \geq \mathbf{C}^+[T^1_{\gamma\text{-Gittins}}] = \mathbf{C}^+[T^1_{\gamma\text{-Surrogate}}] = \mathbf{C}^+[T^1_{\gamma\text{-Insulated}}].$$

D.3 Heavy-Traffic Optimality Proof

We now follow the outline in Section 4.1 to prove Theorem 4.1. To do so, we must first define some notation.

Definition D.1.

(a) For all $x \in \mathbb{X}$ and $\mathbb{Y} \subseteq \mathbb{X}$, define the following distributions:

$$\mathrm{S}(x,\mathbb{Y}) = (\text{service needed for a job starting at state } x \text{ to exit } \mathbb{Y}),$$
$$\mathrm{Completed}(x,\mathbb{Y}) = \mathbb{1}(\text{job starting at state } x \text{ is at } x_{\text{done}} \text{ after exiting } \mathbb{Y}).$$

(b) For all $x \in \mathbb{X}$, the *γ-Gittins index*, $\Gamma_\gamma(x)$, is defined as,

$$\Gamma_\gamma(x) = \sup_{\{x\} \subseteq \mathbb{Y} \subseteq \mathbb{X}} \frac{\mathbf{E}[e^{\gamma\, \mathrm{S}(x,\mathbb{Y})}\, \mathrm{Completed}(x, \mathbb{Y})]}{\frac{e^\gamma}{e^\gamma - 1}\mathbf{E}[e^{\gamma\, \mathrm{S}(x,\mathbb{Y})} - 1]}.$$

Definition D.2. Let $\underline{B}_\gamma$ be the distribution of worst ever boost experienced by a job under the γ-Gittins, γ-Surrogate, and γ-Insulated policy (the worst ever boost of a job is the same under all three). That is,

$$\underline{B}_\gamma = \min_{0 \le u < S} b_{\gamma\text{-Surrogate}}(X_{0:u}).$$

Definition D.3 (Crossing work, as defined in Harlev et al. [25]).

(a) The *u-crossing work* of a job is the amount of service until the first time its boosted arrival time is after u.
(b) The *u-non-crossing work* of a job is its size minus its u-crossing work.
(c) The *crossing work*, $V(u, v)$, is the sum of u-crossing work of each job that arrives in the system after time u and up to time $u + v$.
(d) The *non-crossing work*, $\overline{V}(u, v)$ is the sum of u-non-crossing work of each job that arrives in the system after time u and up to time $u + v$. Equivalently, this is the amount of work that arrived in the system after time u and up to time $u + v$ minus the crossing work $V(u, v)$.

We start by extending Lemma 3.4 to the unknown-size setting.

Lemma D.4. *Let Q be the event that at at all times between the tagged job's worst boosted arrival time, $-\underline{B}_\gamma$ (assume without loss of generality that the arrival time is 0) and its true arrival time 0, there are at least k jobs in the system with arrival time earlier than time $-\underline{B}_\gamma$. Let $u \ge 0$. The tagged job's response time under γ-Surrogate can be upper bounded by*

$$T^k_{\gamma\text{-Surrogate}} \le \begin{cases} W^k_{\gamma\text{-Surrogate}} - \min\{\underline{B}_\gamma, u\} + kS + V(-\min(\underline{B}_\gamma, u), \infty) & \text{if } Q \text{ holds} \\ k(S + V(-\min(\underline{B}_\gamma, u), \infty) + \overline{V}(-\min(\underline{B}_\gamma, u), \min\{\underline{B}_\gamma, u\})) & \text{if } Q \text{ does not hold.} \end{cases}$$

Proof. First observe that assuming the tagged job has constant boost $\hat{B} = \min(\underline{B}_\gamma, u)$ can only increase its response time, so it suffices to analyze the response time under this assumption. Thus, throughout this proof, we can simply refer to the tagged job's boosted arrival time without worrying about it changing with service. In both cases, we will bound the amount of time the tagged job can spend in the system.

If Q holds: by our definition of Q, between the tagged job's boosted arrival time and its actual arrival time, all servers have only worked on work with boosted arrival time better than the tagged job's boosted arrival time.

An upper bound on the amount of work with boosted arrival time better than the tagged job's boosted arrival time is $W^k_{\gamma\text{-Surrogate}} + V(-\hat{B}, \infty)$. Between the tagged job's boosted arrival time and actual arrival time, all servers only worked on work belonging to this $W^k_{\gamma\text{-Surrogate}} + V(-\hat{B}, \infty)$ amount of work. Therefore, the remaining amount of such work at the tagged job's actual arrival time is $W^k_{\gamma\text{-Surrogate}} + V(-\hat{B}, \infty) - \hat{B}$. Once the tagged job arrives, one of the following must be true until it departs the system:

(1) All servers are occupied with work that has boosted arrival time better than the tagged job's boosted arrival time, i.e., this work leaves the system at rate 1.
(2) If not all servers are occupied with such work, the tagged job must be in service, because it has the best boosted arrival time among all remaining jobs. The tagged job is served at rate $1/k$.

The maximum amount of time that (1) can hold is $W^k_{\gamma\text{-Surrogate}} + V(-\hat{B}, \infty) - \hat{B}$, and the maximum amount of time that (2) can hold is kS, so the maximum amount of time that either can hold is $W^k_{\gamma\text{-Surrogate}} + V(-\hat{B}, \infty) - \hat{B} + kS$, which yields an upper bound on the amount of time the tagged job can spend in the system.

If Q does not hold: At the tagged job's true arrival time, there are at most $k-1$ jobs in the system with arrival time before the tagged job's boosted arrival time. In particular, even if all such jobs are in service, they only occupy $k-1$ of the k available servers. Then, once the tagged job arrives, one of the following must be true until it departs the system:

(1) There is a server working on a job from $\overline{V}(-\hat{B}, \hat{B})$, and nothing from $V(-\hat{B}, \infty)$ is in service, nor is the tagged job in service.
(2) There is at least one server working on work from $V(-\hat{B}, \infty)$, i.e. such work leaves the system at rate at least $1/k$.
(3) The tagged job is served at rate $1/k$.

(1) can only hold at the tagged job's arrival time. For each server that is serving a job from $\overline{V}(-\hat{B}, \hat{B})$, after that job's completion, the server will take either the tagged job, or work from $V(-\hat{B}, \infty)$ into service. Namely, after a job from $\overline{V}(-\hat{B}, \hat{B})$ completes, either event (2) or (3) will hold until the tagged job departs the system. A simple upper bound on how long (1) can hold is to imagine that all the work in $\overline{V}(-\hat{B}, \hat{B})$ is served on a single server, leaving at the rate $1/k$, before the other two events hold. (2) can hold for at most $kV(-\hat{B}, \infty)$, and (3) can only hold for at most kS. Therefore, the tagged job will spend at most $k(\overline{V}(-\hat{B}, \hat{B}) + V(-\hat{B}, \infty) + S)$ time in the system, as desired. □

Using Lemma D.4, we can now prove an analogue of Theorem 3.3 in the unknown-size setting.

Theorem D.5. *Assume that* $\mathbf{E}[e^{\gamma k V(0,\infty)}] < \infty$, $\mathbf{E}[e^{\gamma k S}] < \infty$, *and* $\mathbf{E}[I^k_{\gamma\text{-Surrogate}} e^{\gamma W^k_{\gamma\text{-Surrogate}}}] < \infty$. *Let* $\mathbf{C}^+[W_{\mathrm{M/G/1}}] = \lim_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma} \mathbf{E}[e^{\theta W^1}]$. *The tail constant of* γ-Surrogate *can be bounded above as follows:*

$$\limsup_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma} \mathbf{E}[e^{\theta T^k_{\gamma\text{-Surrogate}}}] \le \mathbf{C}^+[W_{\mathrm{M/G/1}}] \frac{\mathbf{E}[I^k_{\gamma\text{-Surrogate}} e^{\gamma W^k_{\gamma\text{-Surrogate}}}]}{1-\rho} \mathbf{E}[e^{\gamma(kS-B)}] \mathbf{E}[e^{\gamma V(0,\infty)}].$$

Proof. The steps in the proof of Theorem 3.3 all hold in this case, except that we replace uses of Lemma 3.4 with uses of Lemma D.4 and must use the fact that $V(u, v)$ is stationary in u. □

Following our outline in Section 4.1, the next step is to prove an analogue of Lemma 3.9 for the unknown-size setting.

Lemma D.6. *If* $\mathbf{E}[e^{\gamma k S}] < \infty$, *then* $\mathbf{E}[e^{\gamma k V(0,\infty)}] < \infty$.

However, this follows directly from the proof of [25, Lemma 4.7] for all boost policies in the unknown-size setting. Finally, we must prove an analogue of Lemma 3.10 for the unknown-size setting. Note that once we do so, Theorem 4.1 follows immediately from the proof of Theorem 3.1 with lemmas and theorems appropriately replaced with their analogues, including replacing [49, Theorem 3.1] with [25, Theorem 4.11]. Thus, once we prove this next lemma, we are done. Before doing so, we introduce the following notation for convenience.

Definition D.7.
(a) $s(x) = \mathbf{E}[\mathrm{S}(x, \mathbb{X})]$,
(b) $G(x) = \sup_{\{x\} \subseteq \mathbb{Y} \subseteq \mathbb{X}} \frac{\mathbf{E}[\mathrm{Completed}(x, \mathbb{Y})]}{\mathbf{E}[\mathrm{S}(x, \mathbb{Y})]}$.

Lemma D.8. *Under γ-Surrogate,*

$$\lim_{\rho\to 1}\frac{\mathbf{E}[e^{\gamma(kS-\underline{B}_\gamma)}]}{\mathbf{E}[e^{\gamma(S-\underline{B}_\gamma)}]}=1$$

Proof. First observe that if we expand $\underline{B}_\gamma$ using its definition, we get,

$$\mathbf{E}[e^{\gamma(kS-\underline{B}_\gamma)}]=\mathbf{E}\left[\frac{1}{\inf_{x\in X_{0:S}}\frac{e^\gamma}{e^\gamma-1}\Gamma_\gamma(x)}e^{\gamma kS}\right].$$

By Lemmas D.9 and D.10,

$$\lim_{\gamma\to 0}\frac{1}{\inf_{x\in X_{0:S}}\frac{\gamma e^\gamma}{e^\gamma-1}\Gamma_\gamma(x)}=\frac{1}{\inf_{x\in X_{0:S}}G(x)}\le\sup_{x\in X_{0:S}}s(x).$$

Then, since $e^{\gamma kS}\to 1$ almost surely as $\gamma\to 0$, there exists a $\delta>0$ such that, almost surely, for all $\gamma<\delta$,

$$\frac{e^{\gamma kS}}{\inf_{x\in X_{0:S}}\frac{\gamma e^\gamma}{e^\gamma-1}\Gamma_\gamma(x)}\le\sup_{x\in X_{0:S}}s(x)+1. \tag{D.1}$$

Lemma D.11 shows that $\mathbf{E}[\sup_{x\in X_{0:S}}s(x)]<\infty$, so (D.1) justifies the following use of the dominated convergence theorem:[13]

$$\begin{aligned}\lim_{\rho\to 1}\frac{\mathbf{E}[e^{\gamma(kS-\underline{B}_\gamma)}]}{\mathbf{E}[e^{\gamma(S-\underline{B}_\gamma)}]}&=\lim_{\gamma\to 0}\frac{\mathbf{E}[\frac{1}{\gamma}e^{\gamma(kS-\underline{B}_\gamma)}]}{\mathbf{E}[\frac{1}{\gamma}e^{\gamma(S-\underline{B}_\gamma)}]}\\&=\frac{\mathbf{E}[\lim_{\gamma\to 0}\frac{1}{\gamma}e^{\gamma(kS-\underline{B}_\gamma)}]}{\mathbf{E}[\lim_{\gamma\to 0}\frac{1}{\gamma}e^{\gamma(S-\underline{B}_\gamma)}]}\\&=\frac{\mathbf{E}[\lim_{\gamma\to 0}e^{\gamma kS}\cdot\frac{1}{\inf_{x\in X_{0:S}}G(x)}]}{\mathbf{E}[\lim_{\gamma\to 0}e^{\gamma S}\cdot\frac{1}{\inf_{x\in X_{0:S}}G(x)}]}\\&=1.\end{aligned}$$

□

Lemma D.9. *For all $x\in\mathbb{X}$, $G(x)\ge\frac{1}{s(x)}$.*

Proof. Recall that

$$G(x)=\sup_{\{x\}\subseteq\mathbb{Y}\subseteq\mathbb{X}}\frac{\mathbf{E}[\text{Completed}(x,\mathbb{Y})]}{\mathbf{E}[\text{S}(x,\mathbb{Y})]}.$$

The bound follows by considering $\mathbb{Y}=\mathbb{X}$:

$$G(x)\ge\frac{\mathbf{E}[\text{Completed}(x,\mathbb{X})]}{\mathbf{E}[\text{S}(x,\mathbb{X})]}=\frac{1}{s(x)}.$$

□

Lemma D.10.

$$\lim_{\gamma\to 0}\inf_{x\in X_{0:S}}\frac{\gamma e^\gamma}{e^\gamma-1}\Gamma_\gamma(x)=\inf_{x\in X_{0:S}}G(x).$$

[13]The justification for the use of dominated convergence theorem in the denominator is identical.

Proof. We start by plugging in the definition of $\Gamma_\gamma(x)$ and then applying Taylor's theorem:

$$\inf_{x\in X_{0:S}} \frac{\gamma e^\gamma}{e^\gamma - 1}\Gamma_\gamma(x) = \inf_{x\in X_{0:S}} \sup_{\{x\}\subseteq\mathbb{Y}\subseteq\mathbb{X}} \frac{\gamma \mathbf{E}[e^{\gamma\,\mathrm{S}(x,\mathbb{Y})}\,\mathrm{Completed}(x,\mathbb{Y})]}{\mathbf{E}[e^{\gamma\,\mathrm{S}(x,\mathbb{Y})}] - 1}$$

$$= \inf_{x\in X_{0:S}} \sup_{\{x\}\subseteq\mathbb{Y}\subseteq\mathbb{X}} \frac{\mathbf{E}[\left(1+\gamma\,\mathrm{S}(x,\mathbb{Y}) + \frac{\gamma^2}{2}\,\mathrm{S}(x,\mathbb{Y})^2 e^{\xi\,\mathrm{S}(x,\mathbb{Y})}\right)\mathrm{Completed}(x,\mathbb{Y})]}{\mathbf{E}[\mathrm{S}(x,\mathbb{Y}) + \frac{\gamma}{2}\,\mathrm{S}(x,\mathbb{Y})^2 e^{\xi\,\mathrm{S}(x,\mathbb{Y})}]}$$

for some $\xi \in (0,\gamma)$. Now observe that since we assume that for all $x \in \mathbb{X}$ there is a positive probability of a trajectory containing x,

$$\infty > \mathbf{E}[e^{\gamma S}] \geq \mathbf{E}[e^{\gamma S} \mid x \in X_{0:S}]\mathbf{P}[x \in X_{0:S}] \geq \mathbf{E}[e^{\gamma\,\mathrm{S}(x,\mathbb{X})}]\mathbf{P}[x \in X_{0:S}]$$

implies that $\mathbf{E}[e^{\gamma\,\mathrm{S}(x,\mathbb{X})}] < \infty$ and so $\mathbf{E}[\mathrm{S}(x,\mathbb{X})^2 e^{\xi\,\mathrm{S}(x,\mathbb{X})}] < \infty$ since $\xi \in (0,\gamma)$. Moreover, since $\mathrm{S}(x,\mathbb{Y}) \leq \mathrm{S}(x,\mathbb{X})$ for all $\mathbb{Y} \subseteq \mathbb{X}$, it follows that $\mathbf{E}[\mathrm{S}(x,\mathbb{Y})^2 e^{\xi\,\mathrm{S}(x,\mathbb{Y})}] < \infty$ and

$$\lim_{\gamma\to 0} \inf_{x\in X_{0:S}} \frac{\gamma e^\gamma}{e^\gamma - 1}\Gamma_\gamma(x) = \inf_{x\in X_{0:S}} \sup_{\{x\}\subseteq\mathbb{Y}\subseteq\mathbb{X}} \frac{\mathbf{E}[\mathrm{Completed}(x,\mathbb{Y})]}{\mathbf{E}[\mathrm{S}(x,\mathbb{Y})]} = \inf_{x\in X_{0:S}} G(x). \qquad \square$$

Lemma D.11. $\mathbf{E}[\sup_{x\in X_{0:S}} s(x)] < \infty$.

Proof. Recall that we define S, the size of a job, as $S = \min\{t \geq 0 : X_t = x_{\mathrm{done}}\}$. Now define the martingale $M_n = \mathbf{E}[S \mid X_{0:n}]$, where we use the convention that $X_S = X_{S+1} = \cdots = x_{\mathrm{done}}$ so that M_n is defined for all $n \geq 0$. Observe that for any $n \geq 0$,

$$s(x) = \mathbf{E}[\mathrm{S}(x,\mathbb{X})] = \mathbf{E}[S - n \mid X_n = x],$$

which means that, for all $p > 1$,

$$\begin{aligned}
\mathbf{E}[\sup_{0\leq i\leq S} s(X_i)] &= \mathbf{E}[\sup_{0\leq i\leq S} \mathbf{E}[S - n \mid X_{0:i}]] \\
&\leq \mathbf{E}[\sup_{0\leq i\leq S} M_i] \\
&= \mathbf{E}[\sup_{n\geq 0} M_n] \\
&\leq \mathbf{E}[\sup_{n\geq 0} M_n^p]
\end{aligned}$$

where the last step follows from the fact that $S \geq 1$ almost surely, and thus $M_n \geq 1$. We now use Doob's inequality [46, Theorem 5.2.1], which tells us that for any $p \in (1,\infty)$,

$$\mathbf{E}[\sup_{n\geq 0} M_n^p]^{\frac{1}{p}} \leq \frac{p}{p-1} \sup_{n\geq 0} \mathbf{E}[M_n^p]^{\frac{1}{p}}.$$

Expanding the expectation in the right hand side we get

$$\mathbf{E}[M_n^p] = \mathbf{E}[\mathbf{E}[S \mid X_{0:n}]^p] \leq \mathbf{E}[\mathbf{E}[S^p \mid X_{0:n}]] = \mathbf{E}[S^p],$$

where the inequality follows from the conditional Jensen's inequality. Putting everything together, we get the following bound:

$$\mathbf{E}[\sup_{0\leq i\leq S} s(x)] \leq \left(\frac{p}{p-1}\right)^p \mathbf{E}[S^p],$$

which is finite since S is class I, and thus has all finite moments. $\square$

E Mathematical Program for Deterministic Multiserver Scheduling Problem

We use Gurobi to optimize the following mathematical program for scheduling, solving an instance using the current system state (i.e., the remaining size and arrival time of each job still in the system) on each new arrival. Namely, assume that we have n jobs in our instance, labeled $\{1, 2, \ldots, n\}$, each with processing time ks_i and arrival time a_i. We assume that the first k jobs are already in service, and the remaining $k+1, \ldots, n$ jobs are, WLOG, sorted in boosted arrival time order, where the boost is computed using the processing time of the job.

We have the following 0/1 decision variables:

- $x_{0,i}$ represents whether job i is the first to be scheduled on a machine.
- $x_{i,n+1}$ represents whether job i is the last job to be scheduled on a machine.
- $x_{i,j}$ represents whether job i precedes job j on a machine.

In particular, we only have $x_{i,j}$ decision variables whenever $i < j$, because on each machine, it is locally optimal to schedule in boosted arrival time order. We then solve the following program to compute a nonpreemptive schedule:

$$
\begin{aligned}
\underset{\boldsymbol{x},\boldsymbol{d}}{\text{minimize}} \quad & \sum_{i=1}^{n} e^{\gamma(d_j - a_j)} \\
\text{such that} \quad & \sum_{i=1}^{k} x_{0,i} = k \\
& \sum_{i=0}^{j-1} x_{i,j} = 1 && \text{for } j = 1, \ldots, k \\
& \sum_{i=1}^{j-1} x_{i,j} = 1 && \text{for } j = k+1, \ldots, n \\
& \sum_{i=0}^{j-1} x_{i,j} = \sum_{i=j+1}^{n+1} x_{j,i} && \text{for } j = 1, \ldots, k \\
& \sum_{i=1}^{j-1} x_{i,j} = \sum_{i=j+1}^{n+1} x_{j,i} && \text{for } j = k+1, \ldots, n \\
& d_j = ks_j && \text{for } j = 1, \ldots, k \\
& d_j = \sum_{i=1}^{j-1} (d_i + ks_j) x_{i,j} && \text{for } j = k+1, \ldots, n.
\end{aligned}
$$

Here the departure times d_j are represented as continuous decision variables, but they are fully constrained by the 0/1 decision variables $x_{i,j}$.

F γ-CombinedBoost's Heavy-Traffic Optimality

Throughout this section, we shorten γ-CombinedBoost to γ-CB in subscripts to reduce clutter. We first require the following lemmas:

Lemma F.1.

$$\lim_{\rho \to 1} \frac{\mathbf{E}[e^{\gamma(kS - B_{\gamma\text{-CB}})}]}{\mathbf{E}[e^{\gamma(S - B_{\gamma\text{-Boost}})}]} = 1$$

Proof.

$$\lim_{\gamma\to 0}\frac{\mathbf{E}[e^{\gamma(kS-B_{\gamma\text{-CB}})}]}{\mathbf{E}[e^{\gamma(S-B_{\gamma\text{-Boost}})}]}=\lim_{\gamma\to 0}\frac{\mathbf{E}[e^{\gamma kS-B_{\gamma\text{-Boost}}-(k-1)S}]}{\mathbf{E}[e^{\gamma(S-B)}]}$$
$$=\lim_{\gamma\to 0}\frac{\mathbf{E}[e^{\gamma(S-B_{\gamma\text{-Boost}})}]}{\mathbf{E}[e^{\gamma(S-B_{\gamma\text{-Boost}})}]}$$
$$=1.$$

□

Lemma F.2.

$$\lim_{\rho\to 1}\frac{e^{\lambda\mathbf{E}[B_{\gamma\text{-CB}}(e^{\gamma S}-1)]}}{e^{\lambda\mathbf{E}[B_{\gamma\text{-Boost}}(e^{\gamma S}-1)]}}=1.$$

Proof. Observe that

$$\lim_{\rho\to 1}e^{\lambda\mathbf{E}[(k-1)S(e^{\gamma S}-1)]}=e^{\lim_{\rho\to 1}\lambda\mathbf{E}[(k-1)S(e^{\gamma S}-1)]}=e^{\frac{1}{\mathbf{E}[S]}\lim_{\gamma\to 0}\mathbf{E}[(k-1)S(e^{\gamma S}-1)]}.$$

Since $(k-1)S(e^{\gamma S}-1)\le 2(k-1)S$ for sufficiently small γ, we can apply the dominated convergence theorem to get,

$$e^{\frac{1}{\mathbf{E}[S]}\lim_{\gamma\to 0}\mathbf{E}[(k-1)S(e^{\gamma S}-1)]}=e^{\frac{1}{\mathbf{E}[S]}\mathbf{E}[\lim_{\gamma\to 0}(k-1)S(e^{\gamma S}-1)]}=1.$$

Now,

$$\lim_{\rho\to 1}\frac{e^{\lambda\mathbf{E}[B_{\gamma\text{-CB}}(e^{\gamma S}-1)]}}{e^{\lambda\mathbf{E}[B_{\gamma\text{-Boost}}(e^{\gamma S}-1)]}}=\lim_{\rho\to 1}\frac{e^{\lambda\mathbf{E}[B_{\gamma\text{-Boost}}(e^{\gamma S}-1)]}e^{\lambda\mathbf{E}[(k-1)S(e^{\gamma S}-1)]}}{e^{\lambda\mathbf{E}[B_{\gamma\text{-Boost}}(e^{\gamma S}-1)]}}$$
$$=\lim_{\rho\to 1}e^{\lambda\mathbf{E}[(k-1)S(e^{\gamma S}-1)]}=1$$

□

Lemma F.3. *For any $u\in\mathbb{R}_+\cup\{\infty\}$, if $\mathbf{E}[e^{\gamma(k+1)S}]<\infty$, then*

$$\mathbf{E}[e^{\gamma kV_{\gamma\text{-CB}}(\infty)}]=e^{\lambda\mathbf{E}[\min\{B_{\gamma\text{-CB}},u\}(e^{\gamma kS}-1)]}<\infty.$$

Proof. As in Lemma 3.9, from Campbell's Theorem [49, Lemma 3.5], we have

$$\mathbf{E}[e^{\gamma kV_{\gamma\text{-CB}}(\infty)}]=e^{\lambda\mathbf{E}[\min\{B_{\gamma\text{-CB}},u\}(e^{\gamma kS}-1)]},$$

so long as the RHS is finite. It suffices to show that $\mathbf{E}[B_{\gamma\text{-CB}}(e^{\gamma kS}-1)]$ is finite. Using the definition of $B_{\gamma\text{-CB}}$, this is

$$\lambda\mathbf{E}[B_{\gamma\text{-CB}}(e^{\gamma kS}-1)]=\lambda\mathbf{E}[B_{\gamma\text{-Boost}}(e^{\gamma kS}-1)]+\lambda\mathbf{E}[(k-1)S(e^{\gamma kS}-1)].$$

The first term is finite from Lemma 3.9. The second term is finite because when $k=1$, the term is just 0. When $k>1$, we have:

$$\lambda\mathbf{E}[(k-1)S(e^{\gamma kS}-1)]=\frac{\lambda}{\gamma}(k-1)\mathbf{E}[\gamma S(e^{\gamma kS}-1)]$$
$$\le\frac{\lambda(k-1)}{\gamma}\mathbf{E}[e^{\gamma S}(e^{\gamma kS}-1)]$$
$$\le\frac{\lambda(k-1)}{\gamma}\mathbf{E}[e^{\gamma(k+1)S}]<\infty,$$

since by assumption, $\mathbf{E}[e^{\gamma(k+1)S}]<\infty$. □

With these lemmas, proving heavy-traffic optimality is similar to that of Theorem 3.1.

Theorem F.4. *Under γ-CombinedBoost,*

$$\lim_{\rho\to 1}\frac{\mathbf{C}^+[T^k_{\gamma\text{-CB}}]}{\mathbf{C}^+[T^1_{\gamma\text{-Boost}}]}=1.$$

Proof. Because γ-Boost is optimal in the M/G/1 across all policies, and any M/G/k policy can be replicated in the resource-pooled M/G/1, we know that for all ρ,

$$\frac{\mathbf{C}^+[T^k_{\gamma\text{-CB}}]}{\mathbf{C}^+[W_{\text{M/G/1}}]\mathbf{E}[e^{\gamma(S-B_{\gamma\text{-Boost}})}]\mathbf{E}[e^{\gamma V_{\gamma\text{-Boost}}(\infty)}]}\geq 1,$$

where the denominator is the tail constant of γ-Boost [49, Theorem 3.1]. Therefore, it suffices to show that

$$\frac{\mathbf{C}^+[T^k_{\gamma\text{-CB}}]}{\mathbf{C}^+[W_{\text{M/G/1}}]\mathbf{E}[e^{\gamma(S-B_{\gamma\text{-Boost}})}]\mathbf{E}[e^{\gamma V_{\gamma\text{-Boost}}(\infty)}]}\leq 1.$$

Since $1/\rho=\mathbf{E}[e^{\gamma S_e}]$ (Lemma A.1), as we take $\rho\to 1$, we have $\gamma\to 0$. Then, for sufficiently high load $\rho\in(\rho',1)$, because the job size distribution is class I, we can assume that $\mathbf{E}[e^{\gamma(k+1)S}]<\infty$ and, that there exists $\varepsilon>0$ such that $\mathbf{E}[e^{(\gamma+\varepsilon)kS_e}]<\infty$. Under these assumptions, Lemma 3.8 implies that $\frac{\mathbf{E}[I^k_{\gamma\text{-CB}}e^{\gamma W^k_{\gamma\text{-CB}}}]}{1-\rho}$ is bounded, and Lemma F.3 implies that $\mathbf{E}[e^{\gamma k V_{\gamma\text{-CB}}(\infty)}]<\infty$, so we can apply Theorem 3.3 to get:

$$\mathbf{C}^+[T^k_{\gamma\text{-CB}}]\leq\mathbf{C}^+[W_{\text{M/G/1}}]\frac{\mathbf{E}[I^k_{\gamma\text{-CB}}e^{\gamma W^k_{\gamma\text{-CB}}}]}{1-\rho}\mathbf{E}[e^{\gamma(kS-B_{\gamma\text{-CB}})}]\mathbf{E}[e^{\gamma V_{\gamma\text{-CB}}(\infty)}].$$

Therefore, it suffices to show that

$$\lim_{\rho\to 1}\frac{\mathbf{E}[I^k_{\gamma\text{-CB}}e^{\gamma W^k_{\gamma\text{-CB}}}]}{1-\rho}=1,$$
$$\lim_{\rho\to 1}\frac{\mathbf{E}[e^{\gamma(kS-B_{\gamma\text{-CB}})}]}{\mathbf{E}[e^{\gamma(S-B_{\gamma\text{-Boost}})}]}=1,$$
$$\lim_{\rho\to 1}\frac{\mathbf{E}[e^{\gamma V_{\gamma\text{-CB}}(\infty)}]}{\mathbf{E}[e^{\gamma V_{\gamma\text{-Boost}}(\infty)}]}=1,$$

where we have simplified the first ratio by dividing off the $\mathbf{C}^+[W_{\text{M/G/1}}]$ from both numerator and denominator. The first equality follows immediately from Theorem 3.6. The second follows from Lemma F.1. The third follows from Lemma F.2. □

G Amendments to Previous Results on Strong Tail Optimality

One may ask why we present results on tail constant optimality as opposed to the notion of strong tail optimality as it is defined in Boxma and Zwart [6], given the work on tail-optimal scheduling in Harlev et al. [25], Yu and Scully [49]. The reason is that [25, 49] prove bounds on $\mathbf{C}^+[T_\pi]$, but not $\mathbf{C}^-[T_\pi]$, via bounds on $\tilde{\mathbf{C}}^+[T_\pi],\tilde{\mathbf{C}}^-[T_\pi]$. However, bounds on $\mathbf{C}^-[T_\pi]$ are needed for strong tail optimality.

Namely, Yu and Scully [49] construct a lower bound using γ-Cheat s/t $\tilde{\mathbf{C}}^+[T_{\gamma\text{-Cheat}}]\leq\tilde{\mathbf{C}}^-[T_\pi]$ for any policy π ([49, Theorem 4.3]). They then show that γ-Boost, which satisfies the property $\mathbf{C}^-[T_\pi]=\mathbf{C}^+[T_\pi]$, attains $\tilde{\mathbf{C}}^+[T_{\gamma\text{-Boost}}]=\tilde{\mathbf{C}}^+[T_{\gamma\text{-Cheat}}]$ ([49, Theorem 5.1]). This, along with Lemma A.3, implies that $\mathbf{C}^+[T_{\gamma\text{-Boost}}]\leq\mathbf{C}^+[T_\pi]$, for any policy π. Harlev et al. [25] employ a similar approach, with γ-Surrogate acting as a lower bound ([25, Theorem 3.7]) for γ-Gittins ([25, Proposition 4.3]).

We know from Lemma A.3 that for any random variable $X \geq 0$,

$$\mathbf{C}^-[X] \leq \tilde{\mathbf{C}}^-[X] \leq \tilde{\mathbf{C}}^+[X] \leq \mathbf{C}^+[X],$$

so we have $\mathbf{C}^+[T_{\gamma\text{-Boost}}]/\mathbf{C}^+[T_\pi] \leq 1$ in the known-size setting and $\mathbf{C}^+[T_{\gamma\text{-Gittins}}]/\mathbf{C}^+[T_\pi] \leq 1$ in the unknown-size setting, i.e., γ-Boost and γ-Gittins are *tail constant optimal* in their respective settings. However, strong tail optimality is defined as follows:

Definition G.1. A policy π is *strongly tail-optimal* among a class of policies Π if it satisfies

$$R_\pi \leq 1,$$

where R_π is the *tail competitive ratio*

$$R_\pi = \sup_{\pi' \in \Pi} \limsup_{t \to \infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}.$$

In particular, for any two policies π, π',

$$\limsup_{t \to \infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]} \leq \frac{\mathbf{C}^+[T_\pi]}{\mathbf{C}^-[T_{\pi'}]},$$

so the results from [25, 49] are only meaningful as a bound on R_π when $\mathbf{C}^+[T_\pi] = \mathbf{C}^-[T_\pi]$ for all policies π under consideration. In particular, we cannot conclude strong tail optimality when we include policies where $\mathbf{C}^+[T_\pi] > \mathbf{C}^-[T_\pi]$.

We can, however, conclude a strong tail competitiveness of a different kind, namely, *strong tail constant competitiveness*. In particular, we let

$$\tilde{R}_\pi = \sup_{\pi' \in \Pi} \limsup_{\theta \to \gamma} \frac{\mathbf{E}[e^{\theta T_\pi}]}{\mathbf{E}[e^{\theta T_{\pi'}}]}$$

be the *tail constant competitive ratio*, and say that a policy is *strong tail constant competitive* if $\tilde{R}_\pi = 1$. Then we have:

$$\limsup_{\theta \to \gamma} \frac{\mathbf{E}[e^{\theta T_\pi}]}{\mathbf{E}[e^{\theta T_{\pi'}}]} \leq \frac{\limsup_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma} \mathbf{E}[e^{\theta T_\pi}]}{\liminf_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma} \mathbf{E}[e^{\theta T_{\pi'}}]} = \frac{\tilde{\mathbf{C}}^+[T_\pi]}{\tilde{\mathbf{C}}^-[T_{\pi'}]}.$$

But since $\mathbf{C}^+[T_{\gamma\text{-Boost}}] = \tilde{\mathbf{C}}^+[T_{\gamma\text{-Boost}}] \leq \inf_\pi \tilde{\mathbf{C}}^-[C_\pi]$, we know that $\tilde{\mathbf{C}}^+[T_{\gamma\text{-Boost}}]/\tilde{\mathbf{C}}^-[T_\pi] \leq 1$ for any policy π.

In summary, γ-Boost and γ-Gittins attain tail constant optimality, which is a different notion than that of strong tail optimality, but we believe it captures a useful notion of optimal tail performance: if a policy is tail constant optimal, then it achieves the best possible tail constant competitive ratio. We also note that γ-Boost and γ-Gittins achieves the best possible competitive ratio against the universal lower bound of the form $\mathbf{P}[T_\pi > t] \geq \frac{1-\rho}{\rho \mathbf{E}[S_e]} \mathbf{P}[Q > t]$ [6], where Q is a random variable representing the maximum amount of work in a busy period, which is known to have $\mathbf{C}^+[Q] = \mathbf{C}^-[Q]$ [27].

While the γ-Boost policy in Yu and Scully [49] and the γ-Gittins policy in Harlev et al. [25] are tail constant optimal among all policies in their respective settings, they are strongly tail optimal only among the class of policies for which $\mathbf{C}^-[T_\pi] = \mathbf{C}^+[T_\pi]$. We conjecture that among policies including those for which $\mathbf{C}^-[T_\pi] < \mathbf{C}^+[T_\pi]$, there may be no policy π which attains $R_\pi = 1$, and that γ-Boost may still be optimal in the sense that $R_{\gamma\text{-Boost}} \leq R_\pi$ for all policies π, even though it may be that $R_{\gamma\text{-Boost}} > 1$.

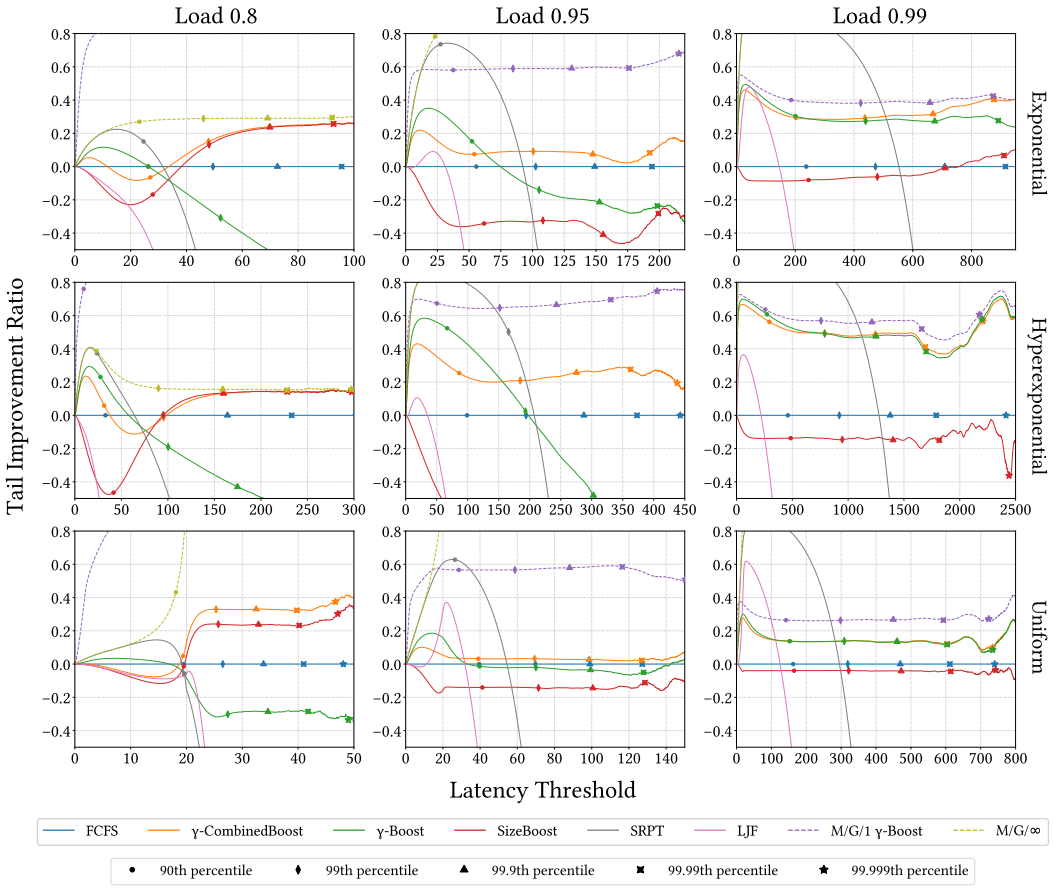


Fig. H.1. (Higher is better.) Plot of performance of policies for $k = 10$ servers for different load regimes and job size distributions. We plot the Tail Improvement Ratio (TIR) of policies against thresholds t. The TIR of a policy π is given by $\text{TIR}(t) = 1 - \mathbf{P}[T_\pi > t]/\mathbf{P}[T_{\text{FCFS}} > t]$, where higher TIR means better performance. Simulations are run using 200 million jobs for loads 0.8 and load 0.95. For load 0.99, we run 2 billion jobs for convergence. The job size distributions are, from top row to bottom row, $\text{Exp}(1)$, Hyperexponential with branches drawn from $\text{Exp}(2)$ and $\text{Exp}(1/3)$ and first branch probability 0.8, and $\text{Uniform}(0, 2)$. This figure is the same as Fig. 5.4, except with the SRPT and Largest Job First (LJF) policies added.

H Additional Simulations

In this section we present simulation results for the SRPT and Largest Job First (LJF) policies for a variety of loads and distributions. The results are presented in Fig. H.1 by adding each of these policies to Fig. 5.4, which was presented in Section 5.2. The primary takeaway from these simulations is that, although increasing the boost for large jobs can improve performance (as discussed in Section 5), strictly prioritizing large (or small) jobs leads to *terrible* asymptotic tail performance.

Received July 2025; revised September 2025; accepted October 2025