A Gittins Policy for Optimizing Tail Latency

Amit Harlev*
Cornell University
ah843@cornell.edu

George Yu*
Cornell University
gy45@cornell.edu

Ziv Scully
Cornell University
zivscully@cornell.edu

1. INTRODUCTION

Service level objectives (SLOs) for queueing systems typically relate to the tail of the system's response time distribution T. The tail is the function mapping a time t to the probability $\mathbf{P}[T > t]$. SLOs typically ask that high percentiles of T are not too large, i.e. that $\mathbf{P}[T > t]$ is small for large t.

Motivated by the problem of optimizing SLOs in settings where job sizes are often unknown, we look at the problem of asymptotically minimizing $\mathbf{P}[T > t]$ in the $t \to \infty$ limit with unknown job sizes. For light-tailed job size distributions, this problem was open for some time [8], until recent work [9] developed an optimal policy for known sizes. In this paper, we study the M/G/1 with *unknown sizes*. We propose a new (Gittins) index policy, and prove it has asymptotically optimal response time tail among all policies that schedule without job size information.

1.1 Background on tail optimality

To understand what it means to optimize the response time tail, we first define the notion of asymptotic optimality. Consider an M/G/1 setting with job size distribution S. Let T_π denote the response time distribution under a scheduling policy π. We say that a policy π is *weakly tail-optimal* if there exists a constant $c > 0$ such that,

$$\sup_{\pi'} \limsup_{t\to\infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]} \le c.$$

We further say π is *strongly tail-optimal* if $c = 1$. In the known-size case we take the supremum over all policies, but in the unknown-size setting, we limit to non-clairvoyant, age-based scheduling policies (Section 1.2). We assume a preempt-resume model: the job in service may be paused and resumed at a later point without loss of progress.

The asymptotic tail behavior under a policy π depends on whether the job's distribution is light- or heavy-tailed; Wierman and Zwart [8] showed that a policy cannot be tail-optimal for both heavy-tailed and light-tailed distributions.

Recent work by Yu and Scully [9, Appendix A] observes that, for an important class of heavy-tailed distributions, there exist many policies that are strongly tail-optimal. Several of these policies, such as Least Attained Service and Processor Sharing, do not use job size information, so the problem of strong tail optimality for unknown sizes is, like known sizes, largely solved in the heavy-tailed setting.

In the light-tailed setting, First-Come-First-Served (FCFS) was the best performing policy for some time in both the known and unknown size cases. FCFS was known to be weakly tail-optimal and conjectured to be strongly tail-optimal. In particular, the tail of FCFS is asymptotically exponential for light-tailed distributions, that is,

$$\mathbf{P}[T_{\text{FCFS}} > t] \sim C_{\text{FCFS}} \exp(-\gamma t),$$

where γ is a constant called the decay rate and C_{FCFS} is FCFS's tail constant. No policy has decay rate better than γ [1, 5], so strong tail optimality amounts to minimizing

$$C_\pi = \lim_{t\to\infty} \exp(\gamma t)\mathbf{P}[T_\pi > t].$$

Recently, new policies have emerged with better tail constant than FCFS, disproving the conjecture that it was strongly tail-optimal [2, 4, 6]. This line of work culminated in a strongly tail-optimal policy, *Boost*, which optimizes the tail constant for class I light-tailed distributions when job sizes are known [9]. However, unlike in the heavy-tailed setting, relying on job size information to determine which jobs to prioritize is essential to achieving strong tail optimality. Thus, strong tail optimality for unknown job sizes in the light-tailed setting is still an open question. We thus ask:

> *In the light-tailed M/G/1 with unknown job sizes, what age-based scheduling policy minimizes the tail constant C_π?*

1.2 How much preemption is too much?

We must first understand how size-based scheduling differs from age-based scheduling. With size-based scheduling, preemption is unnecessary: both preemptive and non-preemptive Boost have identical tail constants [9]. In the unknown-size setting, we cannot schedule based on size, but instead must schedule based on a job's attained service, i.e. its *age*. In this setting, preemption is required to do any better than FCFS, because our only tool for age-based scheduling is to preempt jobs we believe to be large.

To optimize the tail, we wish to use age information to prioritize jobs believed to be small, without overly delaying jobs we believe to be large. Just as the Gittins policy is the age-based generalization of SRPT for minimizing mean response time, we wish to find an an analogous age-based generalization of Boost that optimizes the tail in the unknown size setting. Since Boost uses arrival time and size information, we ask: can we construct a variant of Boost that uses age and arrival time instead of size and arrival time?

In this work, we show that doing so yields a strongly tail-optimal policy. Specifically, we introduce a novel variant of the Gittins index policy that incorporates arrival time and

*Authors contributed equally to this research.

This work was supported by the NSF under grant no. CMMI-2307008.

job age information, and prove its optimality in the M/G/1 with light-tailed job size distribution and unknown job sizes.

Formally, we consider the setting of the M/G/1 queue with arrival rate λ, job size distribution S, and load $\rho = \lambda\mathbf{E}[S]$, where we take $\rho < 1$ so that the system is stable. We assume that S is light-tailed, specifically that it is a class I distribution [9, Definition 2.1].

2. PRIOR WORK ON BOOST POLICIES

The policy that achieves strong-tail optimality in the known-size case belongs to the family of policies known as Boost policies, which are introduced and analyzed in [9]. We give a brief overview of the main ideas of [9] below, explaining how we adapt them to unknown sizes in Section 3.

Boost policies work by assigning every job a *boosted arrival time* and then serving jobs in order of increasing boosted arrival time. A job's boosted arrival time is given by

$$\text{boosted arrival time} = \text{arrival time} - \text{boost},$$

where the *boost* of a job is given by a boost function $b(s)$ that maps each job size to a non-negative boost. The boost policy that achieves strong tail optimality uses a boost function that strikes the right balance between prioritizing short jobs vs. prioritizing jobs that have been in system for a long time.

The key idea in [9] is to relate the problem of strong tail optimality in the M/G/1 to a deterministic batch scheduling problem. This idea follows from an alternative expression for the tail constant, namely

$$C_\pi = \lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[\exp(\theta T_\pi)], \tag{2.1}$$

which comes from final value theorem [4, Theorem 4.3]. Informally, (2.1) tells us that minimizing C_π is morally equivalent to "minimizing $\mathbf{E}[\exp(\gamma T_\pi)]$"—even though this expectation is infinite. This suggests the following construction:

- Instead of studying $\mathbf{E}[e^{\gamma T_\pi}]$ in the queueing setting, consider minimizing the sum $\sum_{i=1}^n \mathbf{E}[e^{\gamma T_\pi(i)}]$ for a *finite batch* of n jobs, where $T_\pi(i)$ is the response time of job i under π.
- View $e^{\gamma T_\pi(i)}$ as $e^{-\gamma A(i)}e^{\gamma D_\pi(i)}$, where $A(i)$ and $D_\pi(i)$ are a job's arrival time (policy-invariant) and departure time (depends on π), respectively.
- Relax the problem by allowing job i to be served prior to its arrival time $A(i)$.

Given a fixed set of jobs and arrival times, this process results in a deterministic weighted batch scheduling problem with *inflation rate* γ. That is, we pay $e^{-\gamma A(i)}$ times an inflation factor $e^{\gamma D_\pi(i)}$ when job i completes at time $D_\pi(i)$. That is, inflation is "negative discounting". Yu and Scully [9] show that the solution to this batch problem is a boost policy that when applied in the queueing setting is strongly tail-optimal.

3. KEY IDEAS

We look to establish similar optimality results for unknown sizes. Because job sizes are no longer known to the scheduler, we consider age-based scheduling policies, asking:

(a) What is the optimal age-based scheduling policy?
(b) How do we prove its optimality?

It turns out that for (a), we can use the same overall approach as [9], but for (b), we require a new strategy.

3.1 Finding the optimal scheduling policy

Just as in [9], we expect the optimal policy to arise from a relaxation to a batch scheduling problem, which we construct by following the same steps as in Section 2. Due to unknown job sizes, the optimal policy for our resulting batch problem is different than that of [9]. Namely, the optimal policy must use preemption to minimize costs. While this batch problem is similar to problems for which the Gittins policy is optimal, there is no known optimal policy for settings with inflation instead of discounting. We show that a variant of the Gittins policy is optimal by adapting the proof of Weber [7].

3.2 Proving optimality

For (b), our approach differs significantly from that of [9]. Roughly speaking, [9] proves optimality in the queueing setting by directly relating it to the batch setting. The idea is to treat each *busy period* as a random instance of a deterministic batch problem. With unknown sizes, setting busy periods as batches yields random instances of stochastic batch problems *with non-independent job sizes* [9, Appendix B]. Because independence is a crucial assumption for Gittins policies [3], the busy-period approach of [9] seems unlikely to work with unknown sizes.

Our main technical contribution is a new approach for (b) that proves optimality *directly in the queueing setting*, without going via the batch problem. Like our approach to the batch problem, our approach is based on Weber's proof [7] of the Gittins policy's optimality, with one key difference: our proof is "quantitiative", rather than "qualitative". That is, Weber proves the Gittins policy is optimal without quantifying the performance it achieves. This qualitative approach does not work in the queueing setting for two main reasons.

The first problem is arrivals. Gittins policies are known to not be optimal in the presence of arrivals, except for in the special case of homogeneous Poisson arrivals [3]. While our arrivals are Poisson, they are unfortunately *not homogenous*: jobs that arrive later cost less.

The second problem is that we cannot reason directly about inflation rate γ because $\mathbf{E}[e^{\gamma T_\pi}] = \infty$ for all policies π. Instead, we consider policies under inflation rate θ and then let $\theta \to \gamma$. Due to the mismatch between inflation rates, we should not expect Gittins with inflation rate γ to minimize $\mathbf{E}[e^{\theta T_\pi}]$ for any fixed $\theta < \gamma$.

We get around these problems by using a quantitative approach. Namely, we quantify the performance of both Gittins and of a lower bound, and show that they match at the $\theta \to \gamma$ limit. A novel feature of our approach is the lower bound: it is based on quantitatively analyzing the lower bound that features in Weber's qualitative proof.

4. THE BATCH PROBLEM

We find the optimal policy for the batch problem from Section 3.1 by adapting the proof of Weber [7]. To recap, there are n i.i.d. jobs with unknown sizes, each with completion cost κ_i, and inflation rate θ. Let K_π^θ be the cost of a random job under policy π. We wish to find the policy that minimizes $\mathbf{E}[K_\pi^\theta]$. Though we only need $\theta = \gamma$ in Section 4, we consider all $0 < \theta \le \gamma$ as it is useful for Section 5.

Define the *Gittins index*, $\nu_i^\theta(x)$, of job i at age x to be,

$$\zeta_\theta(x) = \sup_{y>x} \frac{\theta\mathbf{E}[e^{\theta S}\mathbb{1}(S \le y) \mid S > x]}{\mathbf{E}[e^{\theta(\min(S,y)-x)} - 1 \mid S > x]}, \quad \nu_i^\theta(x) = \kappa_i\zeta_\theta(x).$$

Similarly, define the *surrogate cost* of job i at age x to be $\kappa_i \cdot \inf_{0\le t\le x} \zeta(x)$. Let L^θ_π represent the cost you would accrue from serving a random job (following policy π) if instead of paying each job's completion cost, you continuously pay a job's surrogate cost while it is in service. Just as in [7], our surrogate costs satisfy the following property.

Lemma 4.1. *In the batch setting,*
(a) $\mathbf{E}[K^\theta_\pi] \ge \mathbf{E}[L^\theta_\pi]$ *for any policy* π.
(b) $\mathbf{E}[K^\theta_\pi] = \mathbf{E}[L^\theta_\pi]$ *for any insulated policy* π.

A policy is *insulated* if, under that policy, a job never leaves service on an interval where its surrogate cost is constant. Let θ-Gittins denote the policy which always serves the job with greatest Gittins index $\nu^\theta_i(x)$. Serving the job with greatest Gittins index is equivalent to serving the job with greatest surrogate cost, which means that θ-Gittins is insulated and pays surrogate costs in decreasing order. Since there is inflation, this minimizes the total surrogate cost.

Lemma 4.2. *In the batch setting,*
(a) $\mathbf{E}[L^\theta_{\theta\text{-Gittins}}] \le \mathbf{E}[L^\theta_\pi]$ *for any policy* π.
(b) θ*-Gittins is an insulated policy.*

Putting everything together, we get that θ-Gittins is optimal in the batch setting.

Theorem 4.3. *In the batch setting, for any policy* π*,*

$$\mathbf{E}[K^\theta_\pi] \ge \mathbf{E}[K^\theta_{\theta\text{-Gittins}}].$$

Proof. $\mathbf{E}[K^\theta_\pi] \ge \mathbf{E}[L^\theta_\pi] \ge \mathbf{E}[L^\theta_{\theta\text{-Gittins}}] = \mathbf{E}[K^\theta_{\theta\text{-Gittins}}]$.

5. PROVING STRONG TAIL OPTIMALITY

We now prove that the γ-Gittins policy as defined in Section 4—now with $\kappa = e^{-\gamma A}$ where A is the arrival time of the job—is a strongly tail-optimal boost policy. Recall that γ-Gittins is the policy that serves the job with greatest Gittins index $\nu^\gamma_i(x)$, and observe that γ-Gittins is indeed an age-based boost policy—the policy with boost function $b(x) = -\frac{1}{\gamma}\log\zeta(x)$ will always serve the job of greatest Gittins index $\nu^\gamma_i(x)$. As noted in Section 3.2, we cannot directly reason about the cost at inflation rate γ, so we consider the cost of policies for inflation rate $\theta < \gamma$, and then take the $\theta \to \gamma$ limit.

The proof begins as in the batch setting. Define fair and surrogate costs for jobs in the queueing setting just as we did in the batch setting. We have

Lemma 5.1. *In the queueing setting,*
(a) $\mathbf{E}[K^\theta_\pi] \ge \mathbf{E}[L^\theta_\pi]$ *for any policy* π.
(b) $\mathbf{E}[K^\theta_\pi] = \mathbf{E}[L^\theta_\pi]$ *for any insulated policy* π.

Unfortunately, the remainder of the proof does not immediately extend to the queueing setting. The primary difference is that with arrivals, serving the job with greatest Gittins index is no longer equivalent to serving the job with greatest surrogate cost, and neither of these results in an insulated policy. Therefore, we now have three variants of our policy,

(1) θ*-Gittins*: serve the job with greatest Gittins index.
(2) θ*-Surrogate*: serve the job with greatest surrogate cost.
(3) θ*-Insulated*: never preempt a job on an interval where its surrogate cost is constant, but otherwise serve the job with greatest surrogate cost.

Note that now, θ-Surrogate minimizes the total surrogate cost, θ-Insulated is insulated, and θ-Gittins—our policy of interest—satisfies neither property!

Lemma 5.2. *In the queueing setting,*
(a) $\mathbf{E}[L^\theta_{\theta\text{-Surrogate}}] \le \mathbf{E}[L^\theta_\pi]$ *for any policy* π.
(b) θ*-Insulated is an insulated policy.*

Since no policy is both insulated and minimizes surrogate costs, we cannot argue optimality as we did for Theorem 4.3. However, we only need the optimality to hold at the $\theta \to \gamma$ limit, which means it is sufficient for all three policies to perform equally at the limit, which is indeed the case. In fact, all three policies have the same performance in both real costs and surrogate costs. To state this rigorously we need a "surrogate cost version" of C_π: $D_\pi = \lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[L^\theta_\pi]$. Note that by Lemma 5.2(a), $C_\pi \ge D_\pi$ with equality if (but not only if) π is insulated. Thus, the following result shows that all three policies are "asymptotically insulated".

Lemma 5.3.
(a) $C_{\gamma\text{-Gittins}} = C_{\gamma\text{-Surrogate}} = C_{\gamma\text{-Insulated}}$.
(b) $D_{\gamma\text{-Gittins}} = D_{\gamma\text{-Surrogate}} = D_{\gamma\text{-Insulated}}$.
In fact, these are all equal since $C_{\gamma\text{-Insulated}} = D_{\gamma\text{-Insulated}}$.

At this point it may appear that we are done. However, Lemma 5.2(a) does not provide the lower bound on surrogate costs that we need. It only tells us that for all policies π,

$$D_\pi \ge D^* := \liminf_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[L^\theta_{\theta\text{-Surrogate}}] \tag{5.1}$$

while we wished to show that

$$D_{\gamma\text{-Surrogate}} = \liminf_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[L^\theta_{\gamma\text{-Surrogate}}]$$

is minimal. Fortunately, simultaneously taking the limit in both the parameter θ and the policy θ-Surrogate in (5.1) shows that $D_{\gamma\text{-Surrogate}} = D^*$.

Theorem 5.4. *For all policies* π*,*

$$C_\pi \ge C_{\gamma\text{-Gittins}} = C_{\gamma\text{-Surrogate}} = C_{\gamma\text{-Insulated}}.$$

That is, all three of γ*-Gittins,* γ*-Surrogate, and* γ*-Insulated are strongly tail optimal.*

References

[1] Onno J. Boxma and Bert Zwart. 2007. Tails in Scheduling. *SIGMETRICS Perform. Eval. Rev.* 34, 4, 13–20.
[2] Nils Charlet and Benny Van Houdt. 2024. Tail Optimality and Performance Analysis of the Nudge-M Scheduling Algorithm. arXiv:2403.06588 [cs, math]
[3] John C. Gittins, Kevin D. Glazebrook, and Richard R. Weber. 2011. *Multi-Armed Bandit Allocation Indices* (2 ed.). Wiley, Chichester, UK.
[4] Isaac Grosof, Kunhe Yang, Ziv Scully, and Mor Harchol-Balter. 2021. Nudge: Stochastically Improving upon FCFS. *Proc. ACM Meas. Anal. Comput. Syst.* 5, 2, Article 21, 29 pages.
[5] Alexander L. Stolyar and Kavita Ramanan. 2001. Largest Weighted Delay First Scheduling: Large Deviations and Optimality. *Ann. Appl. Probab.* 11, 1, 1–48.
[6] Benny Van Houdt. 2022. On the Stochastic and Asymptotic Improvement of First-Come First-Served and Nudge Scheduling. *Proc. ACM Meas. Anal. Comput. Syst.* 6, 3, 1–22.
[7] Richard R. Weber. 1992. On the Gittins Index for Multiarmed Bandits. *Ann. Appl. Probab.* 2, 4, 1024–1033.
[8] Adam Wierman and Bert Zwart. 2012. Is Tail-Optimal Scheduling Possible? *Oper. Res.* 60, 5, 1249–1257.
[9] George Yu and Ziv Scully. 2024. Strongly Tail-Optimal Scheduling in the Light-Tailed M/G/1. *Proc. ACM Meas. Anal. Comput. Syst.* 8, 2, Article 27, 33 pages.