Strongly Tail-Optimal Scheduling in the Light-Tailed M/G/1

George Yu
Cornell University
School of Operations Research and Information Engineering
Ithaca, NY, USA

Ziv Scully
Cornell University
School of Operations Research and Information Engineering
Ithaca, NY, USA

ABSTRACT

We study the problem of scheduling jobs in a queueing system, specifically an M/G/1 with light-tailed job sizes, to asymptotically optimize the response time tail. This means scheduling to make $\mathbf{P}[T > t]$, the chance a job's response time exceeds t, decay as quickly as possible in the $t \to \infty$ limit. For some time, the best known policy was First-Come First-Served (FCFS), which has an asymptotically exponential tail: $\mathbf{P}[T > t] \sim Ce^{-\gamma t}$. FCFS achieves the optimal *decay rate* γ, but its *tail constant* C is suboptimal.

We derive a closed-form expression for the optimal tail constant, and we introduce γ-*Boost*, a new policy that achieves this optimal tail constant. We also show via simulation that γ-Boost has excellent practical performance. This abstract summarizes our full paper [14].

ACM Reference Format:
George Yu and Ziv Scully. 2024. Strongly Tail-Optimal Scheduling in the Light-Tailed M/G/1. In *Abstracts of the 2024 ACM SIGMETRICS/IFIP PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS/PERFORMANCE Abstracts '24), June 10–14, 2024, Venice, Italy*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3652963.3655084

1 PROBLEM: MINIMIZING TAIL LATENCY

Service Level Objectives (SLOs) for practical queueing systems often relate to the *tail* of the system's response time distribution T. The tail is the function mapping t to $\mathbf{P}[T > t]$, the probability that a job's response time T exceeds t, where a job's *response time* is the amount of time between its arrival and departure.

Motivated by the need to meet SLOs, we consider scheduling jobs to minimize $\mathbf{P}[T > t]$ in the M/G/1 queue. We actually aim to *asymptotically* minimize the tail, optimizing the decay of $\mathbf{P}[T > t]$ in the $t \to \infty$ limit. In this abstract, we focus on the setting where job sizes (a.k.a. service times) are known to the scheduler, but we consider settings with less information in our full paper [14].

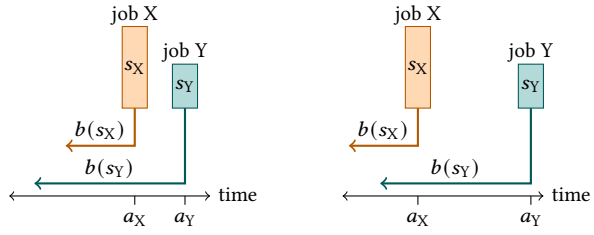
Let T_π denote the response time distribution under policy π. We say π is *weakly tail-optimal* [3] if there exists $c \geq 1$ such that

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

If additionally $c = 1$, we say π is *strongly tail-optimal*.

Whether a policy is weakly tail-optimal depends critically on whether the job size distribution is heavy-tailed or light-tailed.

SIGMETRICS/PERFORMANCE Abstracts '24, June 10–14, 2024, Venice, Italy



(a) Boost serves X after Y if their arrival times are close together.

(b) Boost serves X before Y if their arrival times are far apart.

Figure 2.1: Boost combines a job's arrival time with a size-dependent *boost* to determine its priority. Notation: job i's arrival time is a_i, its size is s_i, and its boost is $b(s_i)$.

In the heavy-tailed setting, specifically for *regularly varying* size distributions, several preemptive policies, such as Shortest Remaining Processing Time (SRPT), are known to be weakly tail-optimal [7, 9, 10]. We observe in our full paper [14, Appendix A] that many of these policies are, in fact, strongly tail-optimal. Strong tail-optimality is thus settled in the heavy-tailed case.

However, strong tail optimality remains an open problem in the light-tailed case, specifically for *class I* size distributions [1, 2]. Even for weak tail optimality, for some time, the only common policy known to be weakly tail-optimal was First-Come First-Served (FCFS). It is known that [3]

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

where $\gamma > 0$ is a constant called the *decay rate*, and $C_{\text{FCFS}} > 0$ is a constant we call FCFS's *tail constant*. Both γ and C_{FCFS} depend on the size distribution and arrival rate.

It is known that no policy can achieve asymptotic decay rate greater than γ [3, 11], so we can measure the performance of a weakly tail-optimal policy π by its tail constant

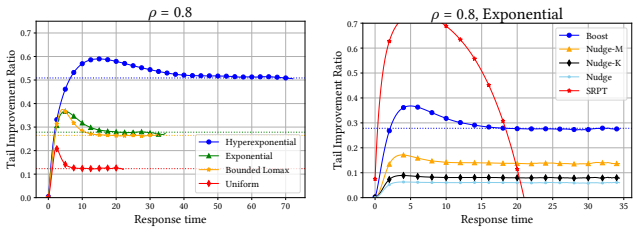
$$C_\pi = \lim_{t\to\infty} \exp(\gamma t)\, \mathbf{P}[T > t]. \tag{1.1}$$

The question of finding a strongly tail-optimal policy thus amounts to minimizing C_π over all policies π. It was previously conjectured that FCFS may be strongly tail-optimal [3, 13], but recent progress has improved upon FCFS's tail constant [4, 5, 12]. We thus ask:

> *What is the smallest possible tail constant C_π, and what policy π achieves it?*

2 OUR ANSWER: BOOST

We introduce *Boost*, a new family of scheduling policies, and γ-*Boost*, an instance of Boost that achieves strong tail optimality.



(a) Several size distributions. **(b) Several policies.**

Figure 2.2: Empirical tail improvement (higher is better) of γ-Boost (a) on several job size distributions, and (b) compared to Nudge variants [4, 5, 12] and SRPT. Plots show *tail improvement ratio* $1 - \mathbf{P}[T_\pi > t]/\mathbf{P}[T_{\text{FCFS}} > t]$ as a function of t. Dotted lines indicate asymptotic improvement $1 - C_\pi/C_{\text{FCFS}}$. Load is $\rho = 0.8$, and mean job size is $\mathbf{E}[S] = 1$. See our full paper [14] for additional details and more simulations.

An instance of Boost is specified by a *boost function* $b : \mathbb{R}_+ \to \mathbb{R}$, where $b(s)$ is called the *boost* of a job of size s. The rough idea is that Boost acts like FCFS, except it pretends that a job of size s arrives $b(s)$ time earlier than it actually does. Specifically, if a job of size s arrives at time a, we define its *boosted arrival time* to be

$$\text{boosted arrival time} = \text{arrival time} - \text{boost} = a - b(s).$$

Boost then follows one scheduling rule: *prioritize jobs from least to greatest boosted arrival time*. See Figure 2.1 for an illustration.

We prove two main theoretical results about Boost. First, we find an *explicit formula* for its tail constant C_{Boost} in terms of the boost function b. Second, we study a particular version of Boost, which we call γ*-Boost*, which has boost function

$$b_\gamma(s) = \frac{1}{\gamma} \log \frac{1}{1 - \exp(-\gamma s)}. \tag{2.1}$$

We show that $C_{\gamma\text{-Boost}} \le C_\pi$ for every other scheduling policy π, so γ-Boost is *strongly tail-optimal*. These results together resolve the question at the end of Section 1. We complement our theoretical results with simulations showing that γ-Boost has excellent practical performance, with Figure 2.2 showing one example.

We have focused above on the case of full job size information, but Boost and γ-Boost can also be defined in settings with partial job size information, and described in our full paper [14].

2.1 Why γ-Boost achieves strong tail optimality

Where does the boost function in (2.1) come from, and why is the resulting γ-Boost policy is strongly-tail optimal? Our key idea is to relate the problem of minimizing the tail constant C_π to a more traditional scheduling problem involving a type of weighted cost.

We begin by considering the following alternative characterization of C_π, which follows from final value theorem [5, Theorem 4.3]:

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

There is thus a vague sense in which minimizing C_π is equivalent to minimizing $\mathbf{E}[\exp(\gamma T_\pi)]$. This is only an informal statement because, as one can deduce from (1.1), we have $\mathbf{E}[\exp(\gamma T_\pi)] = \infty$ for all policies π, even those that are weakly tail-optimal.

While minimizing the always-infinite quantity $\mathbf{E}[\exp(\gamma T_\pi)]$ is not a well-posed problem in the M/G/1, it is analogous to a well-posed problem in *deterministic single-machine scheduling* [6, 8]. Consider an arbitrary finite batch of jobs $\mathcal{I} = \{(a_1, s_1), \ldots, (a_n, s_n)\}$. Here a_i is the arrival time of job i, and s_i is its size. Let $d_{\pi,i}$ be the departure time of job i under policy π, and let the θ*-cost* of policy π be $K_\pi(\theta, \mathcal{I}) = \sum_{i=1}^n \exp(\theta(d_{\pi,i} - a_i))$. Minimizing $\mathbf{E}[\exp(\gamma T_\pi)]$ is analogous to minimizing γ-cost $K_\pi(\gamma, \mathcal{I})$ in the batch setting.

For $\theta < 0$, minimizing θ-cost is actually a variation of a classic single-machine scheduling problem: minimizing total weighted discounted completion time [6, 8], where job i's weight is $\exp(-\theta a_i)$. This problem is hard, but only because of arrival times. In the *batch relaxation*, in which we allow job i to be served even before time a_i, the optimal policy is an index policy called *Weighted Discounted Shortest Processing Time* (WDSPT) [8, Theorem 3.1.6]. To clarify, the arrival times a_i still matter in the batch relaxation, because they determine the weights $\exp(-\theta a_i)$.

Because $\gamma > 0$, one can view minimizing γ-cost as an instance of minimizing total weighted discounted completion time, but with a *negative discount rate*. Fortunately, essentially the same proof as in the standard positive-discount case shows that a version of WDSPT is optimal in the negative-discount case. The γ-Boost policy arises from finding a function b_γ such that WDSPT is equivalent to serving jobs in order of increasing boosted arrival time $a_i - b_\gamma(s_i)$.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under grant nos. CMMI-2307008, DMS-2023528, and DMS-2022448.

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.
[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.
[3] Onno J. Boxma and Bert Zwart. 2007. Tails in Scheduling. *ACM SIGMETRICS Performance Evaluation Review* 34, 4 (March 2007), 13–20.
[4] Nils Charlet and Benny Van Houdt. 2024. Tail Optimality and Performance Analysis of the Nudge-M Scheduling Algorithm. arXiv:2403.06588 [cs, math]
[5] 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.
[6] Jan Karel Lenstra and David B. Shmoys. 2020. Elements of Scheduling. arXiv:2001.06005 [cs]
[7] Misja Nuyens, Adam Wierman, and Bert Zwart. 2008. Preventing Large Sojourn Times Using SMART Scheduling. *Operations Research* 56, 1 (Feb. 2008), 88–101.
[8] Michael Pinedo. 2016. *Scheduling: Theory, Algorithms, and Systems* (5 ed.). Springer, Cham, Switzerland.
[9] Ziv Scully and Lucas van Kreveld. 2024. When Does the Gittins Policy Have Asymptotically Optimal Response Time Tail in the M/G/1? *Operations Research* 72, 2 (Feb. 2024).
[10] Ziv Scully, Lucas van Kreveld, Onno J. Boxma, Jan-Pieter Dorsman, and Adam Wierman. 2020. Characterizing Policies with Optimal Response Time Tails under Heavy-Tailed Job Sizes. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 4, 2, Article 30 (June 2020), 33 pages.
[11] Alexander L. Stolyar and Kavita Ramanan. 2001. Largest Weighted Delay First Scheduling: Large Deviations and Optimality. *The Annals of Applied Probability* 11, 1 (Feb. 2001), 1–48.
[12] 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.
[13] Adam Wierman and Bert Zwart. 2012. Is Tail-Optimal Scheduling Possible? *Operations Research* 60, 5 (Oct. 2012), 1249–1257.
[14] 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.