Performance of the Gittins Policy in the G/G/1 and G/G/*k*, With and Without Setup Times

Yige Hong*
Carnegie Mellon University
yigeh@andrew.cmu.edu

Ziv Scully†
Cornell University
zivscully@cornell.edu

1. INTRODUCTION

We consider the classic problem of preemptively scheduling jobs of unknown size (a.k.a. service time) in a queue to minimize mean number-in-system, or equivalently mean response time (a.k.a. sojourn time). We know how to solve this problem in an M/G/1, provided the job size distribution is known to the scheduler. In this case, the optimal policy is the *Gittins policy* (a.k.a. *Gittins index policy*) [1].

The Gittins policy works by assigning each job a numeric *rank*, or priority level, then always serving the job of best rank. A job's rank depends on two things: (1) the overall size distribution; and (2) the job's *age*, namely the amount of time the job has been served so far.

While Gittins solves the scheduling problem in the M/G/1 case, plenty of systems have features that require models beyond the M/G/1 to faithfully capture, including:

(a) *Multiple servers*, e.g. the M/G/k with $k \geq 2$.
(b) *Non-Poisson arrival processes*, e.g. the G/G/1.[1]
(c) *Periods of server unavailability*, e.g. setup time models.

Optimal scheduling is an open problem in all of these cases. While one may still use Gittins in these cases, e.g. by serving the k jobs of k best ranks, Gittins is known to be suboptimal for (a) and (b), and it is unknown whether Gittins is optimal for (c). Combining multiple features, as in the G/G/k/setup (G/G/k with setup times), makes optimal scheduling still more challenging. Nevertheless, we might hope that Gittins is a good heuristic, even if it is suboptimal. We therefore ask:

> *How good is Gittins's mean number-in-system in systems with features (a), (b), and (c)?*

1.1 Recent Progress on Gittins in the M/G/*k*

Only feature (a) has been addressed in full generality in prior work [2, 6, 8]. Specifically, it is known that in the M/G/k, the additive suboptimality gap of Gittins is bounded by [6][2]

$$\mathbf{E}[N]^{\mathrm{Gtn}} - \inf_{\pi} \mathbf{E}[N]^{\pi} \leq C(k-1)\log\frac{1}{1-\rho}. \qquad (1.1)$$

Let us briefly explain the notation used in (1.1):

- $\mathbf{E}[N]^{\pi}$ is to the mean number-in-system under policy π;
- k is the number of servers;
- $\rho \in [0, 1)$ is the *load* (a.k.a. utilization), namely the average fraction of servers that are busy; and
- $C = \frac{9}{8\log(3/2)} \approx 3.775$ is a constant.

Under mild conditions [8], the right-hand side is dominated by $\inf_\pi \mathbf{E}[N]^\pi$ as $\rho \to 1$. That is, as the M/G/k gets busier and busier, Gittins's suboptimality gap becomes negligible. Gittins is thus considered *heavy-traffic optimal* in the M/G/k.

The above progress on analyzing Gittins in the multiserver M/G/k is certainly promising for handling (a). But key steps of the analysis rely on Poisson arrivals and constant server availability, leaving (b) and (c) out of reach (Section 4).

1.2 Our Contribution

We give the first analysis of Gittins that handles any combination of (a) multiple servers, (b) G/G arrivals, and (c) setup times. Our main G/G/k/setup result (Theorem 3.1) states

$$\mathbf{E}[N]^{\mathrm{Gtn}} - \inf_{\pi} \mathbf{E}[N]^{\pi} \leq \ell_{(\mathrm{a})} + \ell_{(\mathrm{b})} + \ell_{(\mathrm{a})\&(\mathrm{c})}, \qquad (1.2)$$

where each term on the right-hand side is a "suboptimality loss" caused by the features in the subscript. For example, in the M/G/k, the gap is at most $\ell_{(\mathrm{a})}$. It turns out $\ell_{(\mathrm{a})}$ is the right-hand side of (1.1), so (1.2) strictly generalizes (1.1).

Remarkably, both $\ell_{(\mathrm{b})}$ and $\ell_{(\mathrm{a})\&(\mathrm{c})}$ are uniformly bounded at all loads. This implies Gittins is heavy-traffic optimal in the G/G/k/setup under lenient assumptions (Theorem 3.2). Note also that (1.2) has no $\ell_{(\mathrm{c})}$ term, implying another new result: Gittins is optimal in the M/G/1/setup (Corollary 3.3).

While we focus for concreteness on the case where job sizes are unknown, our techniques and results generalize to cases where job sizes are known or partially known. In particular, in the known job size case, Gittins reduces to SRPT (Shortest Remaining Processing Time), yielding the first mean number-in-system bounds on SRPT with G/G arrivals. See the full version of this paper for details [3].

1.3 Main Challenge: G/G Arrivals

Analyzing $\mathbf{E}[N]^{\mathrm{Gtn}}$ is not easy. Until 2018, even the M/G/1 case was open [7]. Since then, there has been much progress understanding Gittins under M/G arrivals [2, 6, 8].

But under G/G arrivals, we know little beyond the fact that Gittins is suboptimal in the G/G/1 [1]. This is likely because G/G arrivals are significantly harder to work with than M/G arrivals. For example, we can no longer use tools like PASTA [9] or analyses based on M/G/1 busy periods [7].

We introduce a *new work decomposition law* for working with G/G arrivals. It allows us to generalize certain prior bounds for M/G arrivals to G/G arrivals (Section 4).

*Supported by NSF grant no. ECCS-2145713.
†Research done in part while visiting the Simons Institute for Theoretical Computer Science at UC Berkeley, and in part while a FODSI postdoc at Harvard and MIT supported by NSF grant nos. DMS-2023528 and DMS-2022448.
[1]Throughout, the G/G arrival notation refers to i.i.d. interarrival and service times (a.k.a. GI/GI arrivals).
[2]Throughout, log is the natural logarithm.

2. MODEL

We consider a preempt-resume G/G/k with a single central queue and k identical servers. The system experiences *G/G arrivals* (a.k.a. *GI/GI arrivals*). Jobs arrive one-by-one with i.i.d. *interarrival times* drawn from distribution A; each job has an i.i.d. *size*, or service requirement, drawn from distribution S; and interarrival times and job sizes are independent. The M/G/k is the special case where A is exponential.

At any moment of time, a job in the system can be served by one server. Any jobs not in service wait in the queue. Once a job's service is finished, it departs. We follow the convention that each of the k servers has service rate $1/k$. This means a job of size S requires kS time in service to finish. This convention gives all systems we study the same maximum total service rate, namely $k \cdot 1/k = 1$, and thereby the same stability condition.

We write $\lambda = 1/\mathbf{E}[A]$ for the arrival rate and $\rho = \lambda \mathbf{E}[S]$ for the system's *load*, or utilization. One can think of ρ as the average fraction of servers that are busy. We assume $\rho < 1$, which is necessary for stability. However, it has never been proven that $\rho < 1$ is *sufficient* for stability in the G/G/k under policies like Gittins. Stability is not trivial to show because the G/G/k may lack the regenerative busy period structure of the G/G/1 [5]. Preemptive scheduling and setup times only further complicate matters. We simply assume the system is stable for $\rho < 1$, which we conjecture is indeed the case, but leave proving it for future work.

We make one additional assumption on interarrival times A.

Assumption 2.1. There exist $A_{\min}, A_{\max} \in \mathbb{R}_{\geq 0}$ such that for all $a \geq 0$,

$$\mathbf{E}[A - a \mid A > a] \in [A_{\min}, A_{\max}].$$

That is, no matter when the last arrival was, the expected time until the next arrival is between $A_{\min}$ and $A_{\max}$.

Specifically, the quantity $\lambda(A_{\max} - A_{\min})$ appears in our results. We can think of it as measuring "how non-Poisson" arrival times are. For instance, when A is exponential, one may use $A_{\min} = A_{\max} = 1/\lambda$, so $\lambda(A_{\max} - A_{\min}) = 0$. More generally, if A has hazard rate bounded in $[\lambda_{\min}, \lambda_{\max}]$, one may use $A_{\min} = 1/\lambda_{\max}$ and $A_{\max} = 1/\lambda_{\min}$.

Many interarrival distributions A satisfy Assumption 2.1, such as all phase-type distributions. One can think of Assumption 2.1 as a relaxation of the well-known *New Better than Used in Expectation* (NBUE) property, which is the special case where $A_{\max} = \mathbf{E}[A]$. The main distributions ruled out by Assumption 2.1 are various classes of heavy-tailed distributions, e.g. power-law tails.

2.1 Scheduling and Performance Objective

The system's scheduling policy determines which subset of jobs receive service at every moment in time. Job sizes and interarrival times are unknown to the scheduler, but the job size *distribution* S is known. The full version of this paper also considers known and partially known sizes [3].

We consider only *non-idling* policies, meaning those that never unnecessarily leave a server idle (see also Section 2.2). We discuss idling policies in the full version of this paper [3].

We focus on minimizing the *mean number-in-system*, i.e. the mean number of in the system. We denote the mean number-in-system under policy π by $\mathbf{E}[N]^{\pi}$, omitting the subscript if there is no ambiguity. By Little's law, minimizing $\mathbf{E}[N]$ is equivalent to minimizing *mean response time*, the average amount of time between a job's arrival and departure.

The main policy we focus on is the *Gittins policy* (abbreviated Gtn). Based on the job size distribution S, Gittins constructs a *rank function* $\text{rank}_{\text{Gtn}} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. When a job's *age*[3] (a.k.a. attained service) is x, the Gittins policy assigns it *rank*, or priority level, $\text{rank}_{\text{Gtn}}(x)$. Lower rank denotes better priority, so the Gittins policy always serves the job or jobs of *least* rank, breaking ties arbitrarily.

The most important fact about the Gittins policy is that thanks to the way rank_{Gtn} is defined, Gittins minimizes $\mathbf{E}[N]$ in the M/G/1 among all nonanticipating policies. But beyond this fact, the details of how rank_{Gtn} is defined are not essential to this work. We thus omit them for brevity, referring the curious reader to the literature [1, 8].

2.2 Setup Times

We consider models in which servers require *setup times* to transition from idle to busy. We denote this with an extra "/setup", as in G/G/k/setup. Whenever a server switches from idle to busy, it must first complete an i.i.d. amount of *setup work*, distributed as U. Like work from jobs, servers complete setup work at rate $1/k$, so setup *work* U results in setup *time* kU. Setup work amounts are independent of interarrival times and job sizes.

For the purposes of stating our results in a unified manner, we consider the G/G/k without setup times to be the special case of the G/G/k/setup where $\mathbf{P}[U = 0] = 1$.

In multiserver systems, there are nontrivial modeling and design choices to make about setup times. To simplify our presentation, we assume the following specific model of setup times, but the full version of this paper considers a broader class of setup time models [3].

In our model, each server can be in one of three states:

- *Setting-up*, i.e. doing setup work.
- *Busy*, i.e. serving a job.
- *Idle*, i.e. neither serving a job nor doing setup work.

Servers transition between states as follows:

- When a setting-up server finishes its setup work, it becomes busy.
- When a busy server is no longer assigned a job to serve, e.g. due to having just completed a job, it becomes idle.
  - If a setting-up server becomes busy but is not immediately assigned a job, it immediately becomes idle, effectively skipping the busy state.
- When an arrival causes the number of jobs in the queue to strictly exceed the number of idle servers, an idle server becomes setting-up.

In the context of setup times, *non-idling* means only letting a server transition to idle if the queue is empty.

3. MAIN RESULTS

As in Section 1, we can describe a G/G/k/setup system by whether it has (a) multiple servers, (b) non-Poisson arrivals, and (c) setup times. Our main result bounds the suboptimality gap in terms of the features it has.

Theorem 3.1. *In the G/G/k/setup, under Assumption 2.1, the suboptimality gap of Gittins is bounded by*

$$\mathbf{E}[N]^{\text{Gtn}} - \inf_{\pi} \mathbf{E}[N]^{\pi} \leq \ell_{(\text{a})} + \ell_{(\text{b})} + \ell_{(\text{a})\&(\text{c})},$$

[3]Specifically, in the G/G/k, a job's age increases at rate $1/k$ during service, staying constant while waiting in the queue.

where

$$\ell_{(\mathrm{a})} = \left(\frac{9}{8\log\frac{3}{2}} + 1\right)(k-1)\log\frac{1}{1-\rho},$$
$$\ell_{(\mathrm{b})} = \lambda(A_{\max} - A_{\min}),$$
$$\ell_{(\mathrm{a})\&(\mathrm{c})} = \mathbb{1}\big(k \geq 2 \text{ and } \mathbf{P}[U>0] > 0\big) \times \left(2(k-1) + \lambda\left(A_{\max} + \frac{k\mathbf{E}[U^2]}{2\mathbf{E}[U]}\right)\right).$$

Note that $\ell_{(\mathrm{a})}$ is nonzero only if feature (a) is present, and similarly for $\ell_{(\mathrm{b})}$ and $\ell_{(\mathrm{a})\&(\mathrm{c})}$. The suboptimality gap is often negligible in heavy traffic, and it is zero in the M/G/1/setup.

Theorem 3.2. *Consider a G/G/k/setup, and suppose that either S or A is not deterministic. Under Assumption 2.1, if $k = 1$ or $\mathbf{E}[S^2(\log S)^+] < \infty$, then*[4]

$$\lim_{\rho\to 1} \frac{\mathbf{E}[N]^{\mathrm{Gtn}}}{\inf_\pi \mathbf{E}[N]^\pi} = 1.$$

Corollary 3.3. *In the M/G/1/setup, the Gittins policy minimizes $\mathbf{E}[N]$.*

4. OBSTACLES AND KEY IDEAS

This section discusses how we prove Theorems 3.1 and 3.2. We use the same overall strategy as prior work on Gittins in the M/G/k [2, 6, 8], but applying it to the G/G/k/setup presents several obstacles. We discuss two of the biggest:

- Bounding mean work under G/G arrivals (Section 4.1).
- Heavy-traffic analysis under G/G arrivals (Section 4.2).

4.1 Work Decomposition with G/G Arrivals

Prior work analyzes the performance of the Gittins policy using the following identity, known as *WINE* (Work Integral Number Equality) [6]:

$$\mathbf{E}[N]^\pi = \int_0^\infty \frac{\mathbf{E}[W(r)]^\pi}{r^2}\,\mathrm{d}r. \tag{4.1}$$

WINE expresses mean number-in-system in terms of *mean r-work* $\mathbf{E}[W(r)]$. A system's r-work is the total service required to serve all jobs in the system until they all either complete or reach rank greater than r, as determined by $\mathrm{rank}_{\mathrm{Gtn}}$ (Section 2.1). For example, ordinary work is the $r = \infty$ case.

Of course, to use WINE (4.1), we must compute mean r-work. Prior work on M/G arrivals typically does so using a *work decomposition law*. In the context of mean r-work, this is a result of the form

$$\mathbf{E}[W(r)]^\pi = \mathbf{E}[W(r)]^{\mathrm{M/G/1\text{-}min}} + \Delta^\pi(r). \tag{4.2}$$

Above, $\mathbf{E}[W(r)]^{\mathrm{M/G/1\text{-}min}}$ is the minimum mean r-work possible in an M/G/1,[5] and $\Delta^\pi(r)$ quantifies the degree to which π prioritizes r-work over other work. Combining (4.2) with WINE (4.1), bounding $\mathbf{E}[N]^\pi$ reduces to bounding $\Delta^\pi(r)$.

We would like to take the same approach for G/G arrivals. WINE (4.1) already holds under G/G arrivals. However, nearly all work decomposition laws like (4.2) in the literature require M/G arrivals [6, Section 2.4.1]. For instance, Miyazawa [4] uses Palm calculus to prove work decomposition laws for M/G/1 systems with vacations, setup times, and similar features. But the proofs rely crucially on PASTA [9], the fact that Poisson arrivals observe a steady-state system.

We prove a new work decomposition law *without using PASTA*, so it holds for G/G arrivals. It implies

$$\mathbf{E}[W(r)]^\pi \leq \mathbf{E}[W(r)]^{\mathrm{G/G/1\text{-}min}} + \Delta^\pi(r) + \lambda(A_{\max} - A_{\min})\mathbf{E}[r\text{-work of a new arrival}], \tag{4.3}$$

where $\Delta^\pi(r)$ is essentially the same as it is in (4.2). The bound degrades gracefully as A deviates further from an exponential distribution, thanks to the $\lambda(A_{\max} - A_{\min})$ factor.

Combining (4.3) with WINE (4.1) and bounds on $\Delta^{\mathrm{Gtn}}(r)$ from prior work on the M/G/k [6] proves Theorem 3.1 for systems without setup times. To complete the proof, we prove new bounds on $\Delta^{\mathrm{Gtn}}(r)$ under setup times. This additional challenge is mostly orthogonal to the issue of G/G arrivals.

4.2 Heavy-Traffic Analysis with G/G Arrivals

To show prove Theorem 3.2 using Theorem 3.1, we need to show that $\ell_{(\mathrm{a})} + \ell_{(\mathrm{b})} + \ell_{(\mathrm{a})\&(\mathrm{c})}$ is small relative to $\inf_\pi \mathbf{E}[N]^\pi$ as $\rho \to 1$. Given that the former is $\Theta\big(\log\frac{1}{1-\rho}\big)$, we need to show that in the G/G/k/setup, $\inf_\pi \mathbf{E}[N]^\pi > \omega\big(\log\frac{1}{1-\rho}\big)$.

It actually suffices to prove such a bound for SRPT in the G/G/1. This is because removing setup times, merging the servers into one fast server, and revealing job sizes to the scheduler can only reduce $\mathbf{E}[N]$. However, while the heavy-traffic performance of SRPT has been characterized in the M/G/1, its G/G/1 performance is open.

By combining WINE (4.1) with our new work decomposition law for G/G arrivals (4.3), we were able to show that SRPT's heavy-traffic performance in the G/G/1 is only a constant factor away from its M/G/1 performance.

Theorem 4.1. *Under Assumption 2.1, letting $c_V^2 = \frac{\mathbf{Var}[V]}{\mathbf{E}[V]^2}$,*[4]

$$\lim_{\rho\to 1} \frac{\mathbf{E}[N]^{\mathrm{G/G/1\text{-}SRPT}}}{\mathbf{E}[N]^{\mathrm{M/G/1\text{-}SRPT}}} = \frac{c_S^2 + c_A^2}{c_S^2 + 1}.$$

[4]By $\rho \to 1$, we mean a limit in which interarrival times are scaled uniformly while job sizes remain fixed.

[5]The comparison is meaningful even if π is multiserver policy, because we normalize total service rate to 1 (Section 2).

References

[1] J. C. Gittins, K. D. Glazebrook, and R. R. Weber. *Multi-Armed Bandit Allocation Indices.* Wiley, Chichester, UK, second edition, 2011.

[2] I. Grosof, Z. Scully, M. Harchol-Balter, and A. Scheller-Wolf. Optimal scheduling in the multiserver-job model under heavy traffic. *Proc. ACM Meas. Anal. Comput. Syst.*, 6(3):1–32, Dec. 2022.

[3] Y. Hong and Z. Scully. Performance of the Gittins policy in the G/G/1 and G/G/k, with and without setup times, Apr. 2023. URL `http://arxiv.org/abs/2304.13231`.

[4] M. Miyazawa. Decomposition formulas for single server queues with vacations : A unified approach by the rate conservation law. *Commun. Statist.—Stochastic Models*, 10(2):389–413, Jan. 1994.

[5] E. Morozov and B. Steyaert. *Stability Analysis of Regenerative Queueing Models: Mathematical Methods and Applications.* Springer, Cham, Switzerland, 2021.

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

[7] Z. Scully, M. Harchol-Balter, and A. Scheller-Wolf. SOAP: One clean analysis of all age-based scheduling policies. *Proc. ACM Meas. Anal. Comput. Syst.*, 2(1), Apr. 2018.

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

[9] R. W. Wolff. Poisson arrivals see time averages. *Oper. Res.*, 30(2):223–231, 1982.