Research Statement

Isaac Grosof, isaacg1.github.io

1 Overview

Modern computer systems are composed of jobs contending for scarce resources. Jobs wait in queues for network switches, for databases, for processors, for caches, for web servers, and far more. Scheduling the jobs – deciding which jobs run at which times – is therefore vital to improving the performance of modern computer systems. I build and analyze **theoretical models of scheduling** to **understand and optimize scheduling decisions** in modern computing systems.

I focus on *stochastic* models of scheduling which incorporate general distributional assumptions, within the framework of queueing theory. Such models differ from many traditional approaches, which are often focused on either overly pessimistic worst-case models, or stochastic models with rigid distributional assumptions. I seek a middle path: enough flexibility to capture real systems, enough specificity to prove strong positive results.

I have proven breakthrough optimality results in a variety of scheduling settings:

- The first result on optimal scheduling in a stochastic multiserver[1] system (M/G/k) [7] (Section 2.1). This paper received the **IFIP Performance 2018 Best Student Paper Award**.
- The first result on optimal scheduling and dispatching in a load-balancing system [8] (Section 2.2). This paper received the **ACM SIGMETRICS 2019 Best Student Paper Award**.
- The first result on optimal multiserver scheduling with limited information [14] (Section 2.3). This paper received the **INFORMS 2022 George Nicholson Award**.
- The first result on optimal scheduling of jobs which span multiple servers [6, 9] (Section 2.4).
- The first robustly optimal scheduling policy in the setting of stochastic scheduling with predictions [16] (Section 3.2).
- The first policy with superior tail latency to First-Come First-Served (FCFS), answering in the negative the long-standing open problem of *strong tail optimality* [10] (Section 3.3). This paper was awarded the **ACM SIGMETRICS 2021 Best Paper Award**.

I see my work in optimal scheduling as bridging the worst-case-focused theoretical computer science community and the stochastic-focused operations research community. I have studied many aspects of modern computing systems, including jobs requiring many servers, dispatching systems, and limited information settings through the Gittins policy. In the future, I will study scheduling for energy efficiency, scheduling with predictions from machine learning, and scheduling for tail latency.

2 Past work

The vast majority of stochastic scheduling theory focuses on single-server systems, where only a single job is processed at a time. Much less is known about scheduling in multiserver systems, even though multiserver systems are the norm in modern computing. I prove the first bounds on the performance of a variety of important policies in the multiserver setting, deriving bounds that hold across all arrival rates and requirement distributions. My bounds imply the first optimality results in the asymptotic regime where the job arrival rate approaches the capacity of the system.

[1] A “server” refers to a resource capable of processing a single job, which might be one core or an entire machine.

2.1 Optimal Multiserver Scheduling. Prior work on optimal scheduling has predominantly focused on single-server systems. It has long been known that optimizing mean response time (a.k.a. latency, delay) in a single-server system is simply a matter of favoring short jobs, resulting in the optimal Shortest Remaining Processing Time (SRPT) policy [13]. In the single-server system, scheduling is comparatively simple: The entire system is devoted to serving the single best job. The multiserver system is much more complex, because the system spreads its effort among many jobs, not just the best single job. *How can we understand such an imperfect system? Is SRPT still optimal?* It is known that in the deterministic worst-case setting, the competitive ratio between multiserver SRPT and the optimal clairvoyant policy is unbounded [11]. Moreover, such a lower bound exists for all online policies in the worst-case setting. We take a different approach, studying the stochastic version of the question.

We prove the *first optimality result for the stochastic multiserver system.* Specifically, we show that multiserver SRPT achieves asymptotically optimal mean response time in the multiserver system: As the job arrival rate approaches the capacity of the system, the ratio between SRPT's mean response time and that of the optimal policy converges to 1. This paper won the IFIP Performance 2018 Best Student Paper Award.

2.2 Optimal Dispatching. What if we are operating in an immediate-dispatch model, rather than the central-queue model studied in Section 2.1? In the worst case, the same lower bound from Section 2.1 also applies [1]. We therefore again explore the stochastic setting.

We create a new class of dispatching policies for which we prove the *first optimality result for the stochastic immediate-dispatch system* [8]. We prove that our policies, when combined with SRPT scheduling at the servers, achieve asymptotically optimal mean response time, in the limit as arrival rate approaches capacity. This paper won the ACM SIGMETRICS/IFIP Performance 2019 Best Student Paper Award, and was presented as a mini-plenary talk at ACM STOC 2021.

2.3 Scheduling with limited information. What if the scheduler has limited information about the jobs, and can't implement the SRPT policy, which requires perfect information? Just as SRPT is the optimal single-server scheduling policy, the Gittins policy has long been known to minimize mean response time in a single-server system where job sizes are estimated or unknown [5, 17]. Unfortunately, the Gittins policy is even more complex in general than multiserver SRPT, making its analysis even more difficult. In fact, prior to our work, nothing was known about the multiserver Gittins policy, or optimal multiserver scheduling under limited information.

We prove the *first optimality result for the unknown-size and estimated-size multiserver systems* for the multiserver Gittins policy [14], and for a simplified variant of Gittins in the unknown size setting [15]. We show that multiserver Gittins achieves asymptotically optimal mean response time in the limit as arrival rate approaches capacity. Our multiserver Gittins paper received the INFORMS 2022 George Nicholson Award.

2.4 Today's complicated jobs. Today's datacenter and supercomputing jobs each occupy multiple servers simultaneously. A machine learning job might require thousands of cores across hundreds of machines, running alongside other jobs with far smaller resource requirements. *How does one schedule such jobs, where each job may require a different number of servers?* In the corresponding theoretical model, there are currently no scheduling policies for which mean response time is understood, and optimality is not even in the picture.

We create the *first scheduling policy for which we can characterize mean response time* [6]. We build on that result to create a new scheduling policy for which we prove the *first optimality result* *for jobs requiring different numbers of servers* [9]. Our optimality result is an asymptotic result which holds both in the perfect information setting discussed in Sections 2.1 and 2.2, as well as as in the limited information setting of Section 2.3.

3 Ongoing and Future work

3.1 Scheduling for energy efficiency. Data centers require an enormous amount of energy, estimated at 205 TWh or 1% of global electricity use in 2018, with corresponding implications for CO_2 emissions [12]. *How should we schedule large-scale computing systems to optimize efficiency?* Queueing-theoretic study of energy efficiency in data centers is an emerging field [4]; little is known about optimizing scheduling policies for efficiency.

In the future, I plan to study scheduling for efficiency in supercomputing systems, in collaboration with Oak Ridge National Laboratory. I also plan to study optimizing scheduling for **reducing excess energy** currently used to achieve latency guarantees, increasing opportunities to put machines into **low-energy states** during lulls in traffic. I also plan to explore using dispatching and scheduling to increase utilization of **clean energy sources** by rerouting jobs to locations where such energy is available, particularly when excess power is generated due to environmental conditions. Finally, the growth of **heterogeneous computing hardware**, including low- and high-power CPUs, GPUs, machine learning accelerators, and FPGAs, presents new opportunities and challenges when scheduling for energy efficiency.

3.2 Scheduling with Predictions. Thanks to the rise of modern machine learning, we have more ability than ever to make predictions about jobs in advance. *How should we optimally use predictions to improve scheduling?* There are prior results based on the Gittins policy of relevance to this area, including my work discussed in Section 2.3, but such results require exact knowledge of the estimate quality distribution, which is often unrealistic. In the stochastic setting, little is known about scheduling with predictions where the scheduling policy is robust against small estimation errors. In the worst-case setting, robust optimality has recently been proven impossible [2].

This year, we took a first step towards answering these questions by creating SRPT-B, *the first robustly optimal stochastic scheduling policy* [16]. We proved that naive policies can have arbitrarily poor performance under arbitrarily small estimation errors. We then proved that SRPT-B yields optimal mean response time in the limit as estimates converge to perfectly accurate, with smooth improvement of performance as the estimates become more accurate. This paper was the subject of an invited talk at ACM STOC 2022.

I am also currently studying **incentive compatible** scheduling with estimates, where the scheduling policy must both incentivize users to provide accurate estimates as well as optimize mean response time using those estimates. This is of importance in large systems with many users with disparate priorities, such as a corporate datacenter shared by teams with differing goals.

In the future, I plan to study tighter couplings of scheduling with **machine learning**, using both empirical estimate-quality distributions and theoretical estimate-quality guarantees from machine learning. On the empirical side, I hope to design scheduling policies that can cope with the range of predictions qualities delivered by ML models in practice. On the theoretical side, scheduling presents interesting challenges for learning algorithms, because the consequences of poor scheduling decisions can have long-term impacts. Finally, I plan to study scheduling using predictions of future workload intensity, supplementing predictions of job characteristics.

3.3 Tail latency. Most quality of service guarantees in modern latency-sensitive computing focus on tail latency metrics, such as the 99th percentile of latency. Unfortunately, the primary performance metric in traditional queueing theory is mean latency, in large part due to the ease of studying the mean with tools such as Little's law. Tail metrics are much more difficult to study and optimize. *How should we schedule to optimize tail latency?*

This past year, I took a first step in optimizing tail latency by creating Nudge, the *first scheduling policy with superior asymptotic tail latency to FCFS* [10], overturning the prior conjecture of FCFS's strong tail optimality [3, 20]. Moreover, we proved that Nudge stochastically improves upon FCFS for all tail metrics, including all percentiles and all moments of latency. This paper won the ACM SIGMETRICS 2021 Best Paper Award.

In the future, I hope to optimize more tail metrics of importance to real systems, such as the 99th percentile of latency, or the fraction of jobs meeting quality-of-service guarantees, as well as fairness-aware metrics of tail latency, which focus on the tail latency of subgroups of jobs in addition to aggregate metrics.

One particular class of scheduling policies I plan to study are policies where jobs become linearly more important as they spend more time in the system [19]. Policies which incorporate time-in-system information into scheduling decisions are notoriously intractable, resulting in many open problems. The specific setting of linearly increasing priority is one of the only exceptions to that rule. Effective use of time-in-system information has great importance for optimizing tail latency.

Finally, I hope to explore the stochastic-dominance-ordering Pareto frontier of latency distributions achievable by any scheduling policy. My work on Nudge proves that FCFS is not on this frontier, while SRPT is provably on the frontier, but little else is known.

3.4 Bridging worst-case and stochastic communities. Currently, there are two mostly-separate communities studying scheduling theory: the worst-case community, which mostly publishes in conferences such as STOC and FOCS, and the stochastic community, which mostly publishes in conferences such as SIGMETRICS and Performance and journals such as Operations Research and Queueing Systems. I have a foot in both of these worlds: I study stochastic scheduling and have primarily published in SIGMETRICS, Performance and Queueing Systems, while I have a computer science background emphasizing worst-case results, and have also published in ITCS and presented at STOC. *How can we bridge and unify these two communities?*

I have taken a first step towards this goal in my papers on multiserver SRPT and optimal dispatching discussed in Sections 2.1 and 2.2 [7, 8]. Both papers examine policies which had previously appeared in the worst-case literature [1, 11]. I prove very different results in the stochastic setting: asymptotically optimal mean response time, in contrast to tight, but unbounded, competitive ratio results in the worst-case setting.

In the future, I hope to prove crossover results or unifying principles between the two settings. I also hope to study optimal scheduling in settings which lie between the worst-case and stochastic settings, such as setting of smoothed analysis [18]. Finally, I hope to help the worst-case and stochastic communities work more closely together, with events like joint workshops and tutorials to help members of one community get up to speed on the techniques of the other.

References

[1] Nir Avrahami and Yossi Azar. Minimizing total flow time and total completion time with immediate dispatching. In *Proceedings of the Fifteenth Annual ACM Symposium on Parallel Algorithms and Architectures*, SPAA '03, pages 11–18, New York, NY, USA, 2003.

[2] Yossi Azar, Stefano Leonardi, and Noam Touitou. Flow time scheduling with uncertain processing time. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 1070–1080, New York, NY, USA, 2021.

[3] Onno J. Boxma and Bert Zwart. Tails in scheduling. *SIGMETRICS Performance Evaluation Review*, 34(4):13–20, 2007.

[4] Anshul Gandhi, Kanad Ghose, Kartik Gopalan, Syed Rafiul Hussain, Dongyoon Lee, David Liu, Zhenhua Liu, Patrick McDaniel, Shuai Mu, and Erez Zadok. Metrics for sustainability in data centers. *HotCarbon*, 2022.

[5] John Gittins, Kevin Glazebrook, and Richard Weber. *Multi-armed bandit allocation indices.* John Wiley & Sons, 2011.

[6] Isaac Grosof, Mor Harchol-Balter, and Alan Scheller-Wolf. WCFS: A new framework for analyzing multiserver systems. *Queueing Systems*, 2022.

[7] Isaac Grosof, Ziv Scully, and Mor Harchol-Balter. SRPT for multiserver systems. *Performance Evaluation*, 127-128:154–175, 2018.

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

[9] Isaac Grosof, Ziv Scully, Mor Harchol-Balter, and Alan Scheller-Wolf. Optimal scheduling in the multiserver-job model under heavy traffic. 2022. Under submission. `https://isaacg1.github.io/assets/msj-srpt.pdf`.

[10] Isaac Grosof, Kunhe Yang, Ziv Scully, and Mor Harchol-Balter. Nudge: Stochastically improving upon FCFS. In *Abstract Proceedings of the 2021 ACM SIGMETRICS / International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '21, page 11–12, New York, NY, USA, 2021.

[11] Stefano Leonardi and Danny Raz. Approximating total flow time on parallel machines. *Journal of Computer and System Sciences*, 73(6):875 – 891, 2007.

[12] Eric Masanet, Arman Shehabi, Nuoa Lei, Sarah Smith, and Jonathan Koomey. Recalibrating global data center energy-use estimates. *Science*, 367(6481):984–986, 2020.

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

[14] Ziv Scully, Isaac Grosof, and Mor 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), November 2020.

[15] Ziv Scully, Isaac Grosof, and Mor Harchol-Balter. Optimal multiserver scheduling with unknown job sizes in heavy traffic. *Performance Evaluation*, 145:102150, 2021.

[16] Ziv Scully, Isaac Grosof, and Michael Mitzenmacher. Uniform bounds for scheduling with job size estimates. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022).* Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

[17] Ziv Scully and Mor Harchol-Balter. The Gittins policy in the M/G/1 queue. In *2021 19th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)*, pages 1–8, 2021.

[18] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *J. ACM*, 51(3):385–463, May 2004.

[19] David A Stanford, Peter Taylor, and Ilze Ziedins. Waiting time distributions in the accumulating priority queue. *Queueing Systems*, 77(3):297–330, 2014.

[20] Adam Wierman and Bert Zwart. Is tail-optimal scheduling possible? *Operations Research*, 60(5):1249–1257, October 2012.