

PERFORMANCE EVALUATION *review*

TABLE OF CONTENTS

- 1** Editor's note *B. Ji*

Special Issue on the 43rd International Symposium on Computer Performance, Modeling, Measurements and Evaluation (IFIP Performance 2025)

- 2** Foreword from the General Chair *R. van der Mei, S. Bhulai*
3 Message from the Program Committee Chairs *K. Avrachenkov, H. van den Berg, C. Xia*

Session 1.A: Queueing Systems and Scheduling Policies I

- 4** The last, the least, and the urgent: fluid modeling and performance equivalence for scheduling policies in partial service queues with abandonment *A. Ferragut, D. Goldszajn, F. Paganini*
6 Heavy-traffic Optimality of Skip-the-Longest-Queues in Heterogeneous Service Systems *Y. Luo, M. Zubeldia*
8 γ -CounterBoost: Optimizing Response Time Tails using Job Type Information Only *N. Charlet, B. Van Houdt*
10 Improving Nonpreemptive Multiserver Job Scheduling with Quickswap *Z. Chen, A. Anggraito, D. Olliaro, A. Marin, M. Marsan, B. Berg, I. Grosof*

Session 1.B: AI/ML Systems and Performance Evaluation

- 12** Optimizing Resource Allocation for Geographically-Distributed Inference by Large Language Models *T. Sun, T. He, B. Ji, P. Parag*
14 LLMemu: A Lightweight Performance Emulator for High-Fidelity Distributed LLM Training *S. Yang, E. Yu, P. Lu, D. Dong*
16 ScaleGS: Scalable Distributed Framework for Large-Scale 3D Gaussian Splatting with Edge Communication *Y. Kou, J. He, X. Yuan, D. Luo, Y. Zhang*
18 TiFSN: A Wavelet-EC-TCN Model for Quadrotor UAV Trajectory Prediction Based on Time-Frequency-Spatial Feature Fusion *H. Zhao, Y. Kou, Y. Xue, S. Wang, Z. Gu*

Session 3.A: Optimal Control, Pricing, and Learning

- 20** Switching Constrained OCO with Predictions and Feedback Delays *W. Pan, Z. Liu*
22 Bayesian Optimization For Dynamic Pricing and Learning *A. Anand, P. Agrawal, T. Bodas*
24 Designing asymptotically optimal policies for continuous-time weakly coupled MDPs *M. Perbal, B. Prabhu, M. Verloop*
26 Strategic Pricing and Ranking in Recommendation Systems with Seller Competition *T. Walunj, V. Kavitha, J. Nair, P. Agarwal*

Session 3.B: Analysis of Networking Protocols and Systems

- 28** Near-Optimal PCM Wear-Leveling Under Adversarial Attacks *T. Lange, J. Naor, G. Yadgar*
30 CommonSense: Efficient Set Intersection (SetX) Protocol Based on Compressed Sensing *J. Meng, T. Yang, J. Xu*
32 Content and Access Networks Synergies: Tradeoffs in Public and Private Investments by Content Providers *P. Agarwal, D. Manjunath*
34 A Control-Theoretic Perspective on BBR/CUBIC Congestion-Control Competition *S. Scherrer, A. Perrig, S. Schmid*

(Table of Contents continued on back cover)



Association for
Computing Machinery

Advancing Computing as a Science & Profession

SIGMETRICS People

Chair

Mor Harchol-Balter
Department of Computer Science
Carnegie Mellon University
harchol@cs.cmu.edu

Vice Chair

Niklas Carlsson
Dept. of Comp & Info Science (IDA)
Linköping University
niklas.carlsson@liu.se

Secretary/Treasurer

Anshul Gandhi
Department of Computer Science
Stony Brook University
anshul@cs.stonybrook.edu

Board

Giulia Fanti
Dept. of Electrical and Computer Engineering
Carnegie Mellon University
gfanti@andrew.cmu.edu

Board

Athina Markopoulou
Department of EECS
University of California, Irvine
athina@uci.edu

Board

Devavrat Shah
Department of EECS
Massachusetts Institute of Technology
devavrat@mit.edu

Board

Benny Van Houdt
Dept. of Mathematics and Computer Science
University of Antwerp
benny.vanhoudt@uantwerpen.be

Newsletter Editor

Bo Ji
Department of Computer Science
Virginia Tech
boji@vt.edu

Immediate Past Chair

Giuliano Casale
Department of Computing
Imperial College London
g.casale@imperial.ac.uk

SIGMETRICS Performance Evaluation Review

Performance Evaluation Review is a publication of SIGMETRICS, the Association for Computing Machinery's Special Interest Group concerned with computer system performance. Subscriptions, address changes and other business communications should be sent to ACM SIGMETRICS, c/o Association for Computing Machinery, 1601 Broadway, 10th Floor, New York, NY 10019-7434, (212) 869-7440. Contributions should be sent to the Editor in PDF format, with one inch margins. There is limited space in each issue of PER, and high standards are expected from submitted paper.

SIGMETRICS Electronic Bulletin Board

This service is intended for the exchange of technical information among members of the performance evaluation community. To have your name added, please visit <http://www.sigmetrics.org/maillinglist.shtml> and subscribe. To send an item, mail it to performance@merlot.usc.edu.

Notice to Past Authors of ACM-Published Articles

ACM intends to create a complete electronic archive of all articles and/or other material previously published by ACM. If you have written a work that has been previously published by ACM in any journal or conference proceedings prior to 1978, or any SIG Newsletter at any time, and you do NOT want this work to appear in the ACM Digital Library, please inform permissions@acm.org, stating the title of the work, the author(s), and when it was published.

Notice to Contributing Authors to SIG Newsletters

By submitting your article for distribution in this Special Interest Group publication, you hereby grant to ACM the following non-exclusive, perpetual, worldwide rights:

- to publish in print on condition of acceptance by the editor
- to digitize and post your article in the electronic version of this publication
- to include the article in the ACM Digital Library and in any Digital Library related services
- to allow users to make a personal copy of the article for noncommercial, educational or research purposes

However, as a contributing author, you retain copyright to your article and ACM will refer requests for republication directly to you.

SIGMETRICS Mission Statement

SIGMETRICS is the ACM Special Interest Group (SIG) for the computer/communication systems performance evaluation community.

SIGMETRICS promotes research in performance analysis techniques as well as the advanced and innovative use of known methods and tools. It sponsors conferences, such as its own annual “SIGMETRICS” conference, publishes a newsletter “Performance Evaluation Review”, operates a web site (<http://www.sigmetrics.org>) and a mailing list linking researchers, students and practitioners interested in performance evaluation.

Target areas of performance analysis include file and memory systems, database systems, computer networks, operating systems, architecture, distributed systems, fault tolerant systems, and real-time system. In addition, members are interested in developing new performance methodology including mathematical modeling, analysis, instrumentation techniques, model verification and validation, workload characterization, simulation, statistical analysis, stochastic modeling, experimental design, reliability analysis, optimization, and queuing theory.

Editor’s Note

The current issue contains papers (including extended abstracts of long papers, short papers, and posters) from the 43rd International Symposium on Computer Performance, Modeling, Measurements and Evaluation (IFIP Performance 2025) held in Amsterdam, The Netherlands on November 11-13, 2025.

I would like to thank the authors for their contributions and Rob van der Mei, Sandjai Bhulai, Konstantin Avrachenkov, Hans van den Berg, and Cathy Xia for the preparation of the special issue.

PER preprints can be downloaded from the preprint queue page (http://www.sigmetrics.org/per_queue.shtml). You may also consider following the SIGMETRICS LinkedIn group (<http://www.linkedin.com/groups/ACM-SIGMETRICS-4045835/about>) for PER-related announcements.

I welcome your feedback to improve our community’s newsletter (boji@vt.edu).

Bo Ji
March 2026

Foreword from the General Chairs

It is our pleasure to welcome you to the 43rd IFIP PERFORMANCE International Conference, held at Centrum Wiskunde & Informatica (CWI) in Amsterdam, The Netherlands, from 11 to 13 November 2025. This edition of the conference was organized as a fully in-person event, providing an excellent opportunity for researchers and practitioners to meet, exchange ideas, and engage in scientific discussion.

The IFIP PERFORMANCE conference continues to be one of the leading international venues for research on the performance evaluation of computing and communication systems, encompassing modeling, measurement, and analytical and methodological advances. The technical program reflects the breadth and depth of current research in the field and is described in more detail in the program chairs' welcome that follows.

The conference program included four tutorials, which took place alongside the main sessions. These tutorials covered a diverse range of timely topics: *A gentle introduction to quantum communication and networking* by Samuel Oslovich and Sounak Kar; *From multiarmed bandits to weakly coupled Markov decision processes* by Nicolas Gast and Bruno Gaujal; *Performability and survivability quantification* by Poul Heegaard; and *Performance analysis of future smart grids* by Bert Zwart. We thank the tutorial speakers for sharing their expertise and contributing to the educational dimension of the conference.

We were also honored to host four distinguished keynote speakers. Mark Squillante (IBM Research) presented a keynote on the intersection of applied probability, performance evaluation, and quantum computing. Pierre Pinson (Imperial College London) spoke on analytics of energy markets and networks. Nelly Litvak (TU Eindhoven) delivered a keynote on random walks on complex networks. Remco Litjens (TNO and TU Delft) addressed resource management challenges for 6G mobile networks. We are grateful to all keynote speakers for their insightful and inspiring contributions.

We sincerely thank the Technical Program Committee Chairs, Konstantin Avrachenkov, Hans van den Berg, and Cathy Xia, for their careful and professional management of the review process. We also thank the members of the Technical Program Committee and all reviewers for their dedication and effort, which ensured the high quality of the conference program. Our thanks further extend to the Publicity Chair, Alessandro Zocca, for his work in promoting the conference.

We gratefully acknowledge the support of our sponsors: CWI, Vrije Universiteit Amsterdam, STAR, Networks, ACM SIGMETRICS, and IFIP. Their support was essential for the successful organization of this conference.

Finally, we thank all authors and participants for their contributions and for making the 43rd IFIP PERFORMANCE conference a stimulating and rewarding event. We hope that you find the proceedings both valuable and inspiring, and that the discussions initiated during the conference will continue to foster progress in our research community.

General Chairs

Rob van der Mei (Centrum Wiskunde & Informatica)

Sandjai Bhulai (Vrije Universiteit Amsterdam)

Message from the Technical Program Committee Chairs

Konstantin Avrachenkov INRIA konstantin.avrachenkov@inria.fr	Hans van den Berg University of Twente j.l.vandenberg@utwente.nl	Cathy Xia Ohio State University xia.52@osu.edu
---	---	---

We are very pleased to share with you the proceedings from the IFIP Performance 2025 conference. This special issue features the full short papers and extended abstracts of all accepted regular papers, along with brief descriptions of the poster presentations. Please note that the full versions of the accepted regular papers will also appear in a special issue of the Performance Evaluation journal. The program reflects a wide range of high-quality contributions in the area of performance modelling and analysis of computer communications systems, covering topics such as queuing theory, AI/ML-based techniques for system optimization, pricing, networking protocol analysis, and Markov process modeling in quantum computational settings. The emerging role of quantum computing in performance evaluation was further highlighted through a keynote presentation and a dedicated tutorial delivered on the final day of the conference.

Renowned for attracting top-quality research in performance evaluation, the Performance conference series continued that tradition this year. We received an excellent set of contributions that reflected the latest advances in the field. The selection process for the 2025 program was rigorous and highly competitive. Each submission first underwent a rigorous double-blind review by three independent reviewers from the technical program committee (TPC). This was followed by a discussion phase among the reviewers, who had no conflicts of interest with the authors, where the authors could be consulted to provide any clarification if needed. The final decisions were made during the online TPC meeting held on July 28, resulting in the acceptance of 19 regular papers and 10 short contributions.

We extend our sincere appreciation to the more than two hundred esteemed authors from across the globe who have chosen to submit their manuscripts to IFIP Performance 2025. Their scientific contributions and genuine enthusiasm form the cornerstone of this conference. We also wish to heartily thank the 63 TPC members who provided careful reviews, engaged in extensive online discussions, and offered invaluable assistance throughout the shepherding process to maintain the high quality of the conference.

We are excited to share the recipients of two distinguished awards among the selected papers. The Best Paper Award is given to the paper titled “A Control-Theoretic Perspective on BBR/CUBIC Congestion-Control Competition.” This exceptional work was crafted by Simon Scherrer, Adrian Perrig (both from ETH Zurich), and Stefan Schmid (TU Berlin). The Stephen S. Lavenberg Best Student Paper Award recognized the exceptional work of Tushar Walunj, Veeraruna Kavitha, Jayakrishnan Nair and Priyank Agarwal (all IIT Bombay) on “Strategic Pricing and Ranking in Recommendation Systems with Seller Competition”. On behalf of the technical program committee, we sincerely congratulate these authors on their extraordinary achievements and the well-earned recognition they have received.

Finally, we wish to express our gratitude to the two general chairs of the conference, Rob van der Mei and Sandjai Bhulai, and the Publicity Chair, Alessandro Zocca, as well as the IFIP WG 7.3 steering group members Leana Golubchik, John C.S. Lui and Urtzi Ayesta, for their great support and seamless collaboration.

Konstantin Avrachenkov,
INRIA

Hans van den Berg,
University of Twente

Cathy Xia,
Ohio State University

Technical Programme Committee Co-chairs

The last, the least, and the urgent: Fluid modeling and performance equivalence for scheduling policies in partial service queues with abandonment

Andres Ferragut
Universidad ORT Uruguay
Montevideo, Uruguay
ferragut@ort.edu.uy

Diego Goldszajn
Universidad ORT Uruguay
Montevideo, Uruguay
goldszajn@ort.edu.uy

Fernando Paganini *
Universidad ORT Uruguay
Montevideo, Uruguay
paganini@ort.edu.uy

ABSTRACT

In several queueing systems, arriving tasks have both service and timing requirements, and depart whenever one of them is attained. In several applications, the partial work performed by the system on tasks during their stay is still useful, even if the task is not completed. We analyze these *partial service queues with abandonment* under a many-server setting and characterize the equilibrium performance of several policies through fluid models. In particular, we show that deadline-oblivious policies like Least-Attained-Service and Last-Come-First-Served exhibit the same equilibrium performance as Earliest-Deadline-First. This striking property means that designers may avoid the difficult job of estimating deadlines without performance penalties. The conclusions are validated by numerical experiments.

Keywords

Deadline queues, EDF, LAS, LCFS.

1. INTRODUCTION

In many queueing systems, it is necessary to consider not only the service time requested by the task or customer, but also the total amount of time that each task is willing to stay in the system, i.e., its deadline. These systems with *customer reneging* or *abandonment* have a long and rich history in queueing theory [1]. Also, they have several applications in operations research problems (e.g., emergency assistance assignment), networking and processor scheduling environments, among many other examples.

One of the natural methods to incorporate deadlines is the *Earliest-Deadline-First (EDF)* scheduling policy. This policy has been extensively analyzed through fluid limits (cf. [4, 5]) that require measure-valued processes and transport equations, which are needed to keep track of the system state. This is due to the fact that, for the problem to be interesting for applications, it is necessary to consider general service and deadline or sojourn time distributions. Two

approaches have evolved: [4] tracks *elapsed* times in the system, and the approach of [2] tracks the *residual* times. Each approach has its advantages and both are required for the results of this paper.

Most papers concerning reneging queues deal with the “call-center scenario”, where customers only abandon while waiting in the queue, and never renege (nor are interrupted) after service starts. However, in recent years [6], some applications like EV charging and AI inference provide examples where tasks may abandon *during service*, leaving the system with partially completed work, and this *partial service* received remains useful for the customer. In this paper, we analyze these *partial service queues with abandonment* in the many-server setting. For this purpose, we develop fluid models based on partial differential equations (PDEs) in the two relevant coordinates of service and sojourn times. We admit general and possibly correlated service and patience requirements.

Our main contributions are fluid models for deadline-aware and deadline-oblivious policies. In particular, we highlight that the deadline-oblivious policies Least-Attained-Service (LAS) and Last-Come-First-Served (LCFS) *yield the same equilibrium performance* as EDF despite not using any information about deadlines. This striking property breaks away from the call-center scenario and has important practical implications, since estimating deadlines is considered a hard problem. Our result implies that system designers may avoid the difficult job of estimating deadlines without incurring a performance penalty.

2. MAIN CONTRIBUTIONS

We consider service systems with Poisson arrivals at rate λ and where each arriving task i has two associated random variables:

1. S_i is the amount of work required by task i (in time units).
2. T_i is the maximum amount of time that the task may stay in the system.

We assume that (S_i, T_i) are independent across customers, with a common density $g(\sigma, \tau)$, allowing for correlation between service and patience requirements. We consider a many-server setting where at most C customers can be served simultaneously. There are two different ways in which task i may leave the system: if it has received S_i units of service time or if it has spent T_i units of time in the system; in the latter case, the task leaves the system even if

*This work was partially supported by ANII Uruguay under grant PD_NAC_2024.1.182118 and AFOSR US under grant #FA9550-23-1-0350.

it is currently in service. The load of the system is thus $\rho := \lambda E[\min\{S, T\}]$, and the amount of service attained by task i by this time is denoted by $S_{a,i} \leq S_i$.

We propose a fluid model for the dynamics in terms of two state descriptors: $\mu_t(d\sigma, d\tau)$ is a measure that stores the current *remaining service and sojourn times* of the customers in service, while $\nu_t(dx, dy)$ stores the *attained service and elapsed times* of the customers in the system. Building upon [2, 4, 6], we derive suitable dynamic equations for both state descriptors, written in terms of the projections of the measure against a family of test functions in the positive orthant that completely characterize such measures. In particular, we generalize the approach of [4] to enable tasks to depart while in service, which requires resorting to a suitable generalization of the concept of hazard rate from survival analysis [3].

Motivated by this analysis, we solve the corresponding equilibrium equations for the EDF policy (which requires the use of the remaining service/deadline state descriptor) and also for the deadline-oblivious policies LAS and LCFS (which require the attained service/elapsed time description). The main result of the paper is:

THEOREM 1. *Consider a partial service system with Poisson arrivals at rate λ , $(S, T) \sim g(\sigma, \tau)$ and capacity C . Assume that $\rho > C$ (i.e., the system is in overload) and let z^* be defined as the unique solution of:*

$$\lambda E[\min\{S, T, z^*\}] = C. \quad (1)$$

Then in the fluid model, the attained service S_a of each customer is the same under EDF, LAS and LCFS, and is distributed as:

$$S_a \sim \min\{S, T, z^*\}.$$

The equivalence of the attained service distribution for all three policies is a striking property: the main performance metric of the system can be completely emulated, in the fluid models, by deadline-oblivious policies. In particular, LCFS does not need more information than the order of arrival, so it can be easily implemented in practice.

3. EXPERIMENTAL VALIDATION

We consider a setup where S and T are correlated. Specifically, we set

$$S = e^U, T = e^V \quad \text{with} \quad (U, V) \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}\right).$$

In particular, the random variables U and V are correlated with normal distributions, and therefore S and T are correlated with log-normal distributions. We fed the system with the same stream of incoming tasks, i.e., with the same arrival, service and patience time under all three policies.

In this case, we have that $E[\min\{S, T\}] \approx 1.36$ (numerically estimated). In order to satisfy the overload condition, We choose $\lambda = 120$ and $C = 100$, so the system is in overload as required by Theorem 1, with $\rho \approx 160 > 100 = C$. The threshold z^* solving eq. (1) can be numerically estimated to be $z^* \approx 1.322$. Figure 1 shows a snapshot of the state space in steady state for EDF and LCFS, highlighting the threshold behavior of the fluid model.

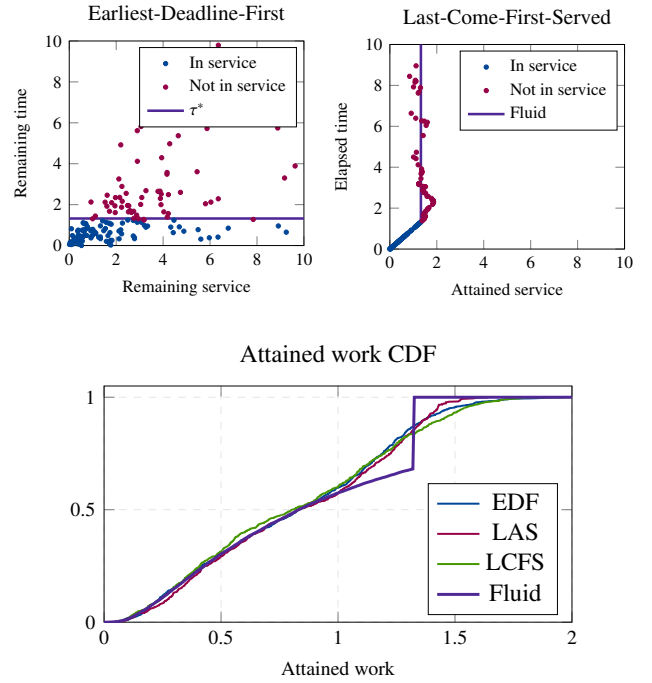


Figure 1: Steady state snapshot of the state space for the EDF and LCFS policies under correlated traffic and attained work empirical CDF under the three policies compared to the fluid model.

Finally, Figure 1 also shows the empirical distribution of the attained service and compares them with the distribution predicted by the fluid models, i.e., the distribution of $\min\{S, T, z^*\}$. The empirical CDFs approximately follow the fluid prediction, and more importantly, the three policies exhibit the same performance for the discrete stochastic system, as claimed. Further analysis is needed to characterize the attained work around z^* in the discrete system.

4. REFERENCES

- [1] D. Barrer. Queuing with impatient customers and ordered service. *Operations Research*, 5(5):650–656, 1957.
- [2] H. C. Gromoll, P. Robert, and B. Zwart. Fluid limits for processor-sharing queues with impatience. *Mathematics of Operations Research*, 33(2):375–402, 2008.
- [3] N. L. Johnson and S. Kotz. A vector multivariate hazard rate. *Journal of Multivariate Analysis*, 5(1):53–66, 1975.
- [4] W. Kang and K. Ramanan. Fluid limits of many-server queues with reneging. *Annals of Applied Probability*, 20(6):2204–2260, Dec. 2010.
- [5] L. Kruk, J. Lehoczy, K. Ramanan, and S. Shreve. Heavy traffic analysis for EDF queues with reneging. *Annals of Applied Probability*, 21(2):484–545, 2011.
- [6] M. Zeballos, A. Ferragut, and F. Paganini. Proportional fairness for EV charging in overload. *IEEE Transactions on Smart Grid*, 10(6):6792–6801, 2019.

Heavy-traffic Optimality of Skip-the-Longest-Queues in Heterogeneous Service Systems

[Extended Abstract]^{*}

Yishun Luo
University of Minnesota
luo00329@umn.edu

Martin Zubeldia
University of Minnesota
zubeldia@umn.edu

ABSTRACT

We consider a discrete-time parallel service system consisting of n heterogeneous single-server queues with infinite capacity. Jobs arrive to the system as an i.i.d. process with rate proportional to n , and must be immediately dispatched in the time slot that they arrive. The dispatcher is assumed to be able to exchange messages with the servers to obtain their queue lengths and make dispatching decisions, introducing an undesirable communication overhead. In this setting, we propose an ultra-low communication overhead load balancing policy dubbed k -Skip-the- d -Longest-Queues (k -SLQ- d), where queue lengths are only observed every $k(n-d)$ time slots and, between observations, incoming jobs are sent to a queue that is not one of the d longest ones at the time that the queues were last observed. For this policy, we establish conditions on d for it to be throughput optimal and we show that, under that condition, it is asymptotically delay-optimal in heavy-traffic for arbitrarily low communication overheads (i.e., for arbitrarily large k).

Keywords

load balancing; sparse communication; heavy-traffic

Overview

Parallel service systems are ubiquitous in many applications, from checkout lines at the supermarket, to server farms for cloud computing. These systems involve a single stream of incoming jobs that are dispatched to one of many single-server queues operating in parallel. In these systems, the main performance metrics of interest are: the maximum throughput that can be achieved, and the average delay of a typical job. In this setting, a throughput and delay optimal policy is the well-know Join-the-Shortest-Queue (JSQ) policy, where incoming jobs are sent to one of the shortest queues. Despite its outstanding performance, in practice this policy does not scale well to large systems, as the communication overhead required to implement the JSQ policy becomes prohibitive. In order to make dispatching policies

scalable to large systems, many policies have been proposed in the past. In particular, it was discovered that the use of memory at the dispatcher can greatly decrease the communication overhead required to implement policies with excellent throughput and delay performance, with this communication overhead being the more expensive requirement in most applications. Since an exact analysis for these more sophisticated policies is usually intractable, it is common to consider the performance in several asymptotic regimes.

In this paper, we focus on the throughput and delay performance in classical and many-server heavy-traffic regimes, where the arrival rate converges to the maximum capacity of the system. In such regimes, the multi-dimensional state of the queuing system is reduced to a single dimensional state. This behavior of the queuing system is known as the State Space Collapse (SSC) and it is critical to achieve delay optimality. In particular, we introduce and analyze a novel dispatching policy that attains throughput and delay optimality while having an arbitrarily small communication overhead. Before presenting our main contributions, we briefly outline prior work in this setting.

There is a large body of literature on load balancing policies for parallel service systems. On one extreme, Round Robin has the lowest delay among all policies with no communication overhead, although its delay performance is far from optimal, and its stability region is reduced. On the other extreme, Join-the-Shortest-Queue (JSQ) policy is known to be delay optimal for homogeneous systems and throughput optimal in general, but it requires prohibitive amounts of communication overhead. Motivated by the need to reduce communication overhead, there are many policies that fall between Round Robin and JSQ, such as the Power-of- d -Choices (Pod) [1, 2], Join-Idle-Queue (JIQ) [3], and other memory enhanced policies that make dispatching decisions based on stored information, such as server ID, queue length, etc. [4, 5, 6]. However, policies like Pod and JIQ may exhibit poor performance in heterogeneous systems [7]. To address this issue and improve performance, modified versions of JSQ, Pod, and JIQ have been developed that incorporate service speed information into the dispatching decisions. Furthermore, if the servers themselves are capable of performing analysis, there are policies such as CARE [8], that make dispatching decisions based on estimated queues.

Since most load balancing policies are not amenable to exact analysis, several asymptotic regimes have been considered in the literature. One of the most popular is classic heavy-traffic, where the load of the queueing system is made to converge to one. For this regime, JSQ has been shown

^{*}A full version of this paper is available at <https://arxiv.org/pdf/2503.02982>

to be throughput and delay optimal in classical and many-server heavy-traffic in continuous and discrete time setting. Moreover, the *Pod* policy has also been shown to be delay optimal in heavy-traffic [9]. While the original JIQ policy is not delay-optimal in heavy-traffic, a modified version of it, dubbed JBT- r , has been shown to be optimal [10]. JBT- r has also been shown numerically to achieve a low communication overhead, although it lacks a closed-form expression for it. Finally, it has been shown in [11] that any load balancing policy is heavy-traffic delay optimal as long as it satisfies certain conditions. Nevertheless, these conditions are hard to verify in practice for many policies, especially for heterogeneous systems and for ultra-low communication overhead policies like the one we introduce in this paper. Another important regime is many-server heavy-traffic regime where both the number of servers and arrival rate increase together. JSQ and *Pod* have been shown to be delay optimal in this regime under certain conditions [12].

Our contribution

In this paper we focus on developing a family of dispatching policies that are throughput and delay optimal in heavy traffic, while requiring extremely low communication overhead. Compared with the policies mentioned above, our family of policies is not only easy to implement but also guarantee arbitrarily low communication overhead. Such policies are born from the following insights:

- (i) In heavy traffic, queues are so long that they appear to be of an approximately constant length for long intervals of time. Therefore, there is no need to sample them in each time slot in order to make good dispatching decisions.
- (ii) When making repeated dispatching decisions over many consecutive time slots using the same queue length vector, policies that work well over one time slot (like Joint-the-Shortest-Queue) do not work so well. For repeated decisions, it is better to spread out the arrivals among different queues as much as possible in order to maintain the balancedness of queues.

With these insights in mind, we introduce a new load balancing policy called k -Round-Robin-Skip-the-Longest- d -Queues (k -SLQ- d), where k and d are positive integers. Under this policy, the queue lengths are sampled every $k(n-d)$ time slots, and incoming jobs between the sampling times are sent in a Round-Robin fashion to the $n-d$ queues that were shortest at the last time that they were sampled. This Round-Robin loop is done for k rounds, which fills the $k(n-d)$ time slots in between queue length samplings. For this policy:

1. We obtain its stability region and establish necessary and sufficient conditions on the number of skipped servers d for throughput optimality when the servers are heterogeneous. In particular, we show that these do not depend on the number of rounds k .
2. We show that, under almost the same condition on d as for throughput optimality, the k -SLQ- d policy is delay-optimal in classical heavy-traffic for any k . In particular, this implies that we can obtain delay optimality with an arbitrarily low communication overhead. Moreover, if the message rate can be made to

depend on the stability slack ϵ , we show that delay optimality can be achieved with a vanishing message rate of order $o(\epsilon)$.

3. We extend our results to the many-server heavy-traffic regime. We show that the number of skipped servers d in order to be throughput optimal is always of order $\Theta(n)$ unless the system is almost homogeneous. Moreover, we show that our policy is delay optimal in many-server heavy-traffic as long as $\epsilon \in o(n^{-11})$.
4. For the homogeneous case, there is a phase transition in the delay performance in both classical and many-server heavy-traffics between using a pure Round-Robin policy, and doing Round-Robin skipping just the longest queue at the beginning of each round.

From the methodological side, we use Lyapunov drift arguments to establish our results. However, the nature of our policy makes it so the one-step drift of the queue lengths can be positive at times. To overcome this challenge, we look at the drift of the queues over a whole cycle of $k(n-d)$ time slots. This brings new challenges, as usual tricks in the literature that are used to avoid problematic terms do not work over multiple time slots.

1. REFERENCES

- [1] M. Mitzenmacher, The power of two choices in randomized load balancing, Ph.D. thesis, U.C. Berkeley (1996).
- [2] N. D. Vvedenskaya, R. L. Dobrushin, F. I. Karpelevich, Queueing system with selection of the shortest of two queues: an asymptotic approach, Problems of Information Transmission 32 (1) (1996) 15–27.
- [3] Y. Lu, Q. Xie, G. Klier, A. Geller, J. R. Larus, A. Greenberg, Join-idle-queue: A novel load balancing algorithm for dynamically scalable web services, Performance Evaluation 68 (11) (2011) 1056–1071.
- [4] D. Gamarnik, J. N. Tsitsiklis, M. Zubeldia, Delay, memory, and messaging tradeoffs in distributed service systems, Stochastic Systems 8 (1) (2018) 45–74.
- [5] D. Gamarnik, J. N. Tsitsiklis, M. Zubeldia, Stability, memory, and messaging trade-offs in heterogeneous service systems, Mathematics of Operations Research 47 (3) (2022) 1862–1874.
- [6] M. van der Boer, M. Zubeldia, S. Borst, Zero-wait load balancing with sparse messaging, Operations Research Letters 48 (3) (2020) 368–375.
- [7] K. Gardner, J. Abdul Jaleel, A. Wickeham, S. Doroudi, Scalable load balancing in the presence of heterogeneous servers, ACM SIGMETRICS Performance Evaluation Review 48 (3) (2021) 37–38.
- [8] G. Mendelson, X. Kuang, Load balancing using sparse communication, arXiv preprint arXiv:2206.02410 (2022).
- [9] A. Eryilmaz, R. Srikant, Asymptotically tight steady-state queue length bounds implied by drift conditions, Queueing Systems 72 (2012) 311–359.
- [10] X. Zhou, J. Tan, N. Shroff, Heavy-traffic delay optimality in pull-based load balancing systems: Necessary and sufficient conditions, Proceedings of the ACM on Measurement and Analysis of Computing Systems 2 (3) (2018) 1–33.
- [11] X. Zhou, F. Wu, J. Tan, K. Srinivasan, N. Shroff, Degree of queue imbalance: Overcoming the limitation of heavy-traffic delay optimality in load balancing systems, Proceedings of the ACM on Measurement and Analysis of Computing Systems 2 (1) (2018) 1–41.
- [12] D. Hurtado-Lange, S. T. Maguluri, A load balancing system in the many-server heavy-traffic asymptotics, Queueing Systems 101 (3) (2022) 353–391.

γ -CounterBoost: Optimizing Response Time Tails using Job Type Information Only

Nils Charlet, Benny Van Houdt
Dept. Computer Science, University of Antwerp, Belgium
{nils.charlet,benny.vanhoudt}@uantwerpen.be

ABSTRACT

This paper focuses on the problem of minimizing the tail in a light-tailed M/G/1-queue in the partial job size information setting when the scheduler only makes use of the job types and *does not exploit arrival times*. Prior work showed that in case of 2 job types the so-called Nudge- M policy minimizes the tail in a large class of scheduling policies. In this paper we introduce the γ -CounterBoost policy in the partial information setting with $d \geq 2$ job types and prove that it minimizes the tail in an even broader class of scheduling policies called Contextual CounterBoost policies. The γ -CounterBoost policy reduces to the Nudge- M policy in case of $d = 2$ job types.

1. INTRODUCTION

For most light tailed distributions, it was long believed that the First Come First Served (FCFS) policy is tail optimal [1]. However, it was recently shown that the response time tail of FCFS can be improved upon using the Nudge scheduling policy [2]. More specifically, the response time under FCFS in an M/G/1 queue with a class I job size distribution decays exponentially, meaning

$$P[R_{FCFS} > t] \approx c_{FCFS} e^{-\gamma t},$$

for some constants c_{FCFS} and γ . The decay rate γ of FCFS is known to be optimal [3], but the *prefactor* c_{FCFS} can be further reduced by the Nudge scheduling policy [2]. The initial Nudge scheduling policy operates almost identical to FCFS, except that an arriving *short* job may pass/skip the last job in the queue, provided that the last job is a *long* job that was not passed by any other job.

It became immediately clear that it is not hard to reduce the prefactor of the initial Nudge scheduling policy by considering different Nudge variations [4, 5]. A key characteristic of these Nudge variations is that they all rely solely on job size (or type) information of the last few jobs that arrived in the queue to determine how many jobs an arriving job is allowed to pass.

It turns out that apart from the job sizes, the arrival times of jobs also play a key role in coming up with a response time tail optimal policy. This led to the introduction of the γ -Boost scheduling policy which was shown to have an optimal response time tail, that is, it minimizes the prefactor, for

class I job size distributions in case the scheduler knows the exact size z and arrival time of any incoming job [6]. The γ -Boost policy schedules jobs based on their *boosted arrival time*. The boosted arrival time of a job is defined as its actual arrival time t_a minus its boost $b(z)$, which depends on the exact job size z of the job. The γ -Boost policy can also be used when the job size information that is available to the scheduler is more coarse, for instance when jobs are partitioned into types and the scheduler only knows the type of an incoming job. In such case the boosted arrival time is defined as before, but the boost of a job now depends on its type i as opposed to its exact job size z . In this partial information setting, γ -Boost was shown to achieve the lowest prefactor among all boost policies, meaning the choice of the boost value b_i is optimal. The γ -Boost policy is believed to be tail optimal among all scheduling policies in this setting, but a proof is still lacking.

This paper instead looks at the partial information setting where arrival times are not known, and only type information is available to the scheduler, not exact sizes. The initial Nudge policy and the variations considered afterwards all share the property that they do not exploit arrival times. The current state-of-the-art Nudge policy for 2 job types, initially called Nudge- M [5], simply states that whenever a type-1 arrival joins the queue it is served before any type-2 job that (a) is still waiting in the queue and that (b) was one of the previous M arrivals (of any type). The Nudge- M policy was proven to minimize the prefactor in a large class of Nudge policies if the parameter M is chosen optimally. If we define the *arrival counter* of a job equal to n if it is the n -th job arrival, then Nudge- M can be reformulated as a scheduling algorithm that schedules jobs based on a boosted arrival counter (that is, type-1 jobs get boost M and type-2 jobs get boost 0).

This reformulation immediately yields a natural generalization to a context with $d > 2$ job types (possibly even infinitely many). In case we have d types, this generalization defines the counter boost of a type i job as

$$b_i^* = \frac{1}{\log \tilde{S}(-\gamma)} \log \left(\frac{\tilde{S}_i(-\gamma)}{\tilde{S}_i(-\gamma) - 1} \right),$$

for all i , where S is the job size distribution of a random job, S_i is the job size distribution of a type- i job, and $\tilde{S}$ and $\tilde{S}_i$ are their respective Laplace transforms. We refer to this policy as γ -CounterBoost.

The Nudge- M policy was shown to minimize the tail among a large class $\mathcal{F}$ of Nudge policies [5]. A first contribution in this paper is the introduction of a new class of Counter-

Boost policies called *Contextual CounterBoost* (CCB) policies, that is broader than the class $\mathcal{F}$. A CCB policy is characterized by a context window size M and a boost function b . The boost function maps the strings s holding the types of the previous and possibly M next arrivals to a boost value. Whenever the server needs to select a new job for service, it computes the boosted arrival counter for each waiting job by subtracting the boost of the job from its arrival counter. The job with the smallest boosted arrival counter is served next, where ties are broken in favor of jobs with a larger boost. Note that γ -CounterBoost is a CCB policy with a context window of size $M = 0$ as the types of the previous and next jobs do not impact its arrival counter boost.

2. OPTIMALITY OF γ -COUNTERBOOST

The main contribution of this paper exists in proving that γ -CounterBoost minimizes the prefactor within the class of CCB policies as stated in the next theorem:

Theorem 1. *γ -CounterBoost has the lowest prefactor among all the policies in $\mathcal{G} = \cup_M \mathcal{G}_M$, where $\mathcal{G}_M$ is the set of Contextual CounterBoost policies with context window size M .*

This means that using a context window size M larger than zero does not offer any reduction in the prefactor.

To establish this result we show that starting from any CCB policy with boost function $b^{(1)}$, we can find a sequence of boost functions $(b^{(k)})_{k=1}^m$ with $b^{(m)}$ the boost function of γ -CounterBoost such that the prefactor of the CCB policy associated with boost function $b^{(k)}$ is non-increasing in k .

In order to make sure that the prefactor does not worsen when increasing k , it is key to make the difference between two boost functions $b^{(k)}$ and $b^{(k+1)}$ as small as possible. The Single Swap Theorem, stated below, gives a simple condition on when replacing a boost function b by b' improves the prefactor provided that the only difference between b and b' is that a type- i job that sees a specific context will pass a type- j job that also sees a specific (possibly overlapping) context when using b' , while it did not using b .

Theorem 2 (Single Swap Theorem). *Consider a CCB policy that is modified such that for some context s with $s_{M+1} = i$ and some other context s' with $s'_{M+1} = j$, $N_b(s, s') = ([b(s) - b(s')]^+)$ increases from $k-1$ to $k \geq 1$, while all other N_b -values remain the same. In other words, a type- j job with context s' that arrived k jobs before a type- i job with context s was initially not passed, but it is after the modification. If this modification changes the prefactor, then it reduces the prefactor if and only if $k \leq N^*(i, j) = ([b_i^* - b_j^*])^+$.*

The next step, which is the more challenging one, exists in showing that one can find the required sequence of boost functions to transform any CCB policy in γ -CounterBoost such that we can apply our Single Swap Theorem to show that the prefactor is non-increasing.

The key technical novelty required to find this sequence of boost functions lies in replacing the boost function b of a CCB policy by some other boost function $\hat{b}$ such that the associated CCB policy is still the same, but the boost values $\hat{b}(s)$ are all independent of each other over the field of the rational numbers. Further, as adding a constant to all the boost values does not alter a CCB policy, we can increase all the b_i^* values by a sufficiently large Δ such that the initial boost values of $b^{(1)}(s)$ are all below the boosts of

γ -CounterBoost. The sequence of boosts is then constructed such that all the boosts $b(s)$ simultaneously increase towards their target. The independence over the rationals makes sure that each time that the CCB policy changes, it is a change for which we can show that it does not worsen the prefactor due to the Single Swap Theorem.

Further, we show that the prefactor of γ -CounterBoost converges to that of γ -Boost as the load λ tends to one, showing that the gains γ -Boost achieves using arrival time information vanish in heavy traffic.

Theorem 3. *For λ tending to one we have $b_i^* - b_j^* \approx \log(E[S_j]/E[S_i])/\log(1 + \gamma/\lambda)$ and*

$$\lim_{\lambda \rightarrow 1^-} c_{\gamma\text{-CounterBoost}} = c_{FCFS} \prod_{j=1}^d E[S_j]^{-p_j E[S_j]},$$

where p_j is the probability of a type- j job. This is the same limit as for γ -Boost in case of unknown job sizes and d job types.

This property and its proof are a straightforward generalization of the results in [5] for $d = 2$ types. We also include a heavy traffic approximation b^{heavy} for the boost function b^* , which may be easier to estimate in practice as it does not rely on the decay rate γ or the Laplace transform of the job size distributions.

Acknowledgement

We would like to thank George Yu and Ziv Scully for some useful discussions and for introducing the class of Contextual CounterBoost policies. This work was supported by the FWO project G0A9823N.

3. REFERENCES

- [1] A. Wierman and B. Zwart, “Is tail-optimal scheduling possible?” *Operations Research*, vol. 60, no. 5, pp. 1249–1257, 2012. [Online]. Available: <http://www.jstor.org/stable/23323693>
- [2] I. Grosof, K. Yang, Z. Scully, and M. Harchol-Balter, “Nudge: Stochastically improving upon FCFS,” *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 5, no. 2, jun 2021. [Online]. Available: <https://doi.org/10.1145/3460088>
- [3] O. Boxma and B. Zwart, “Tails in scheduling,” *SIGMETRICS Perform. Eval. Rev.*, vol. 34, no. 4, p. 13–20, mar 2007. [Online]. Available: <https://doi.org/10.1145/1243401.1243406>
- [4] B. Van Houdt, “On the stochastic and asymptotic improvement of first-come first-served and nudge scheduling,” *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 6, no. 3, pp. 1–22, dec 2022. [Online]. Available: <https://doi.org/10.1145/3570610>
- [5] N. Charlet and B. Van Houdt, “Tail optimality and performance analysis of the Nudge-M scheduling algorithm,” *arXiv preprint arXiv:2403.06588*, 2024. [Online]. Available: <https://arxiv.org/abs/2403.06588>
- [6] G. Yu and Z. Scully, “Strongly tail-optimal scheduling in the light-tailed M/G/1,” *SIGMETRICS Perform. Eval. Rev.*, vol. 52, no. 1, p. 5–6, Jun. 2024. [Online]. Available: <https://doi.org/10.1145/3673660.3655084>

Improving Nonpreemptive Multiserver Job Scheduling with Quickswap

Zhongrui Chen^{*}
jcpwflor@cs.unc.edu

Andrea Marin[†]
marin@unive.it

Adityo Anggraito[†]
adityo.anggraito@unive.it

Marco Ajmone Marsan[‡]
marco.ajmone@imdea.org

Diletta Olliaro[†]
diletta.olliaro@unive.it

Benjamin Berg^{*}
ben@cs.unc.edu

Isaac Grosf[§]
izzy.grosf@northwestern.edu

ABSTRACT

Modern data center workloads are composed of multiserver jobs that require multiple servers in order to run. Many multiserver jobs can run in parallel as long as there are enough servers to satisfy the demand of each job. Given a stream of arriving jobs, a scheduling policy must determine which jobs to run at every moment in time to minimize the mean response time across jobs — the average time from when a job arrives to the system until it is complete. Because preemptions are often computationally expensive in data centers, the scheduling policy cannot preempt running jobs.

We propose a new class of non-preemptive policies called Most Servers First with Quickswap (MSFQ). MSFQ policies prioritize jobs that demand more servers. However, they keep mean response time low by periodically granting priority to other jobs in the system. We prove conditions under which MSFQ policies stabilize the system, and analyze their mean response time. We prove that the MSFQ priority switching mechanism allows these policies to greatly outperform state-of-the-art policies that rely on simpler priority mechanisms. We validate our theoretical results via extensive simulations using traces from the Google Borg system.

1. INTRODUCTION

Modern data centers serve multiserver jobs that occupy multiple servers simultaneously. Each multiserver job is associated with a server need, the number of servers the job requires to run, and a service duration, the amount of time the job must run to be completed. A set of multiserver jobs can run in parallel only if the system has enough dedicated servers for each job. A scheduling policy must select which jobs to run in parallel at every moment in time. Given a fixed number of servers, k , our goal is to choose a scheduling policy that minimizes the mean response time across jobs in a stream of arriving multiserver jobs — the average time from when a job arrives to the system until it is completed.

^{*}University of North Carolina at Chapel Hill, USA

[†]Università Ca' Foscari Venezia, Italy

[‡]IMDEA Networks Institute, Spain

[§]Northwestern University, USA

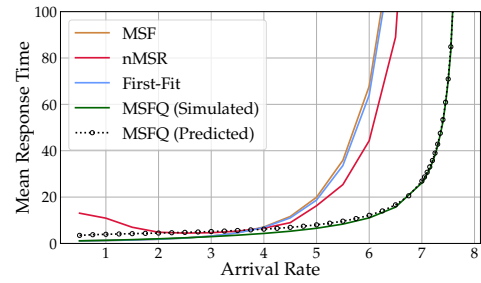


Figure 1: Mean response times as a function of arrival rates for MSFQ and competitor policies.

Designing scheduling policies for multiserver jobs is difficult for two reasons: (i) it is hard to utilize all servers in the system due to bin-packing effects; (ii) modern multiserver jobs are stateful, making preemptions costly or impossible. This raises the question of how to design and analyze non-preemptive scheduling policies that minimize the mean response time across a stream of multiserver jobs.

To achieve high server utilization, prior work on scheduling multiserver jobs has suggested prioritizing jobs with larger server needs. Specifically, [3] evaluated the Most-Servers-First (MSF) policy, a non-preemptive policy that prioritizes jobs with larger server needs. MSF considers all jobs in descending server need order. If there are enough servers to satisfy a job, MSF puts this job into service.

To understand the pros and cons of MSF, consider an example where jobs either need one server or k servers. We refer to this as the one-or-all setting. Here, MSF serves jobs in two alternating phases: (i) MSF serves k -server jobs until none remain in the system; (ii) MSF serves 1-server jobs until none remain before returning to serve k -server jobs. We show in our full paper [1] that MSF achieves optimal long-run average server utilization in the one-or-all setting. Unfortunately, this high server utilization does not lead to low mean response time. MSF suffers because, as the job arrival rate increases, it takes an increasingly long time to switch between phases. This creates a feedback loop where 1-server jobs accumulate while the system processes k -server jobs, leading to a long phase of serving 1-server jobs during which many k -server jobs accumulate.

To address the slow phase switching of MSF, we develop a class of scheduling policies for the one-or-all setting called

Most Server First with Quickswap (MSFQ). We show that MSFQ policies stabilize the system whenever it is possible to do so (Theorem 1), and we analyze the mean response time of MSFQ (Theorem 2). To evaluate MSFQ policies, we compare them against MSF, First-Fit (an optimized first-come-first-served policy), and a non-preemptive Markovian Service Rate (nMSR) policy [2] (see Figure 1). We also develop variants of MSFQ that process general multiserver job workloads, and evaluate these variants using traces from Google Borg [4] (see Figure 2).

2. OUR MODEL

We consider a system with k servers. A multiserver job is represented by an ordered pair (i, s) , where $i \in \{1, 2, \dots, k\}$ is the number of servers the job needs to run and s is the job's service duration, the time the job must run on the servers before completion. Jobs occupy a fixed number of servers throughout their time in service, and a job in service cannot be preempted. We refer to this job model as the Multiserver Job (MSJ) model.

We consider workloads composed of several job classes, where class- i jobs each require i servers. We consider serving a stream of multiserver jobs, where class- i jobs arrive according to an independent Poisson process with rate λ_i . We further assume the service durations of class- i jobs are i.i.d. exponentially distributed random variables such that $S_i \sim \exp(\mu_i)$.

A set of jobs can run in parallel if their aggregate server demand does not exceed the number of available servers. That is, let $\mathbf{u} = (u_1, u_2, \dots, u_k)$ denote a set of multiserver jobs containing u_i class- i jobs. These jobs can run in parallel if and only if $\sum_{i=1}^k i u_i \leq k$.

3. MAIN RESULTS

We introduce the class of Most Server First with Quickswap (MSFQ) policies in the one-or-all setting. MSFQ prioritizes jobs with high server needs, following similar service phases as the MSF policy. However, to shorten the durations of each phase, an MSFQ policy is associated with a threshold, ℓ . Instead of serving class-1 jobs in phase 2 until none remain, MSFQ serves class-1 jobs until there are ℓ jobs remaining in the system. At this point, MSFQ stops admitting new class-1 jobs into service and completes the ℓ class-1 jobs in service. Once these jobs are complete, MSFQ begins serving class- k jobs.

We show that, by using the threshold ℓ to abbreviate phase 2, MSFQ policies can reduce the feedback loop that leads to high mean response time under MSF. We first show that MSFQ policies stabilize the system whenever it is possible to do so. That is, if it is possible to achieve a finite mean response time, any MSFQ policy will have a finite mean response time.

Theorem 1. A Most Servers First with Quickswap (MSFQ) policy with any threshold ℓ , $0 \leq \ell < k$, stabilizes the system whenever it is possible to do so in the one-or-all system.

Proof. We first show that no scheduling policy is stable if $\lambda_1/k\mu_1 + \lambda_k/\mu_k \geq 1$ by comparing to a resource-pooled system. Next, we show that our MSFQ policy is stable given that $\lambda_1/k\mu_1 + \lambda_k/\mu_k < 1$ using a Lyapunov drift argument. $\square$

Next, we analyze the mean response time of an MSFQ policy under the one-or-all setting.

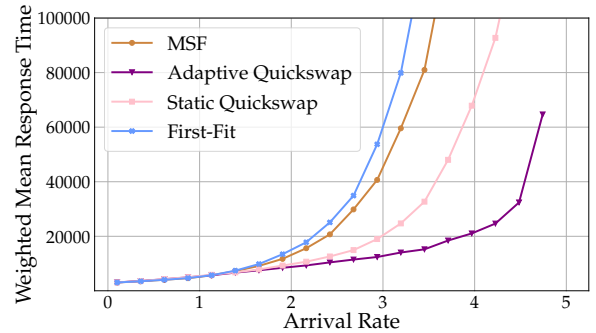


Figure 2: Mean response times as a function of arrival rates for MSFQ and competitor policies using Google Borg traces.

Theorem 2. The mean response time under an MSFQ policy, $\mathbb{E}[T]$, depends on the first and second moments of the durations of each phase and the mean number of jobs at the beginning of each phase. For a given MSFQ policy, we can compute all of these quantities.

Proof. We show that $\mathbb{E}[T]$ can be written in terms of phase durations by conditioning on which phase a job arrives during, and whether a job is a class-1 or a class- k job.

We then derive the Laplace transforms of the phase durations and the expected number of jobs at the beginning of each phase. We use these transforms to compute the overall mean response time. Although the transforms we derive are recursively defined, we provide a calculator¹ that symbolically differentiates these transforms to compute the mean response time. $\square$

4. EVALUATIONS

We evaluate the accuracy of our mean response time analysis and compare MSFQ with the MSF, nMSR and First-Fit policies via simulations of the one-or-all setting. Figure 1 shows that our analysis is accurate and that MSFQ outperforms all competitor policies at all arrival rates.

We also develop variants of MSFQ called Adaptive Quickswap and Static Quickswap that work outside of the simplified one-or-all setting. For this evaluation, we use simulations driven by traces from the Google Borg cluster scheduler. Here, 86% of the system load is contributed by the 0.34% of jobs with the largest server needs. To ensure that our scheduling policy does not ignore these important jobs, we evaluate this trace using a weighted mean response time metric that weights class- i jobs by the amount of load they contribute to the system. We show that our policies perform favorably compared to all competitors on this weighted mean response time metric (see Figure 2).

5. REFERENCES

- [1] Chen, Z., Anggraito, A., Oliaro, D., Marin, A., Marsan, M. A., Berg, B., and Grosof, I. Improving nonpreemptive multiserver job scheduling with quickswap. *Performance Evaluation* (2025).
- [2] Chen, Z., Grosof, I., and Berg, B. Improving multisource job scheduling with markovian service rate policies. *POMACS 9*, 2 (2025), 1–36.
- [3] Speitkamp, B., and Bichler, M. A Mathematical Programming Approach for Server Consolidation Problems in Virtualized Data Centers. *IEEE Transactions Services Computing* 3, 4 (2010), 266–278.
- [4] Tirmazi, M., Barker, A., Deng, N., Haque, M. E., Qin, Z. G., Hand, S., Harchol-Balter, M., and Wilkes, J. Borg: the next generation. In *EuroSys* (2020), pp. 1–14.

¹<https://github.com/jcpwfloi/msfq-calculator>

Optimizing Resource Allocation for Geographically-Distributed Inference by Large Language Models

Tingyang Sun¹, Ting He^{1*}, Bo Ji², Parimal Parag³

¹Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA

²Department of Computer Science, Virginia Tech, Blacksburg, VA, USA

³Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore, India

1. INTRODUCTION

Large language models (LLMs) have produced exceptional performance in many artificial intelligence tasks [2]. Modern LLMs pre-trained on large datasets have demonstrated promising utility for diverse applications [2]. However, to achieve state-of-the-art accuracy, such models often need to contain more than 50 billion parameters, which makes it expensive to work with these models due to the requirement of expensive hardware such as high-end GPUs.

To expand access to LLMs, several research groups have open-sourced their pre-trained LLMs [5]. However, the sheer size of these models makes it difficult to adopt them, even if no training is needed. Prior studies have shown that the bottleneck of running LLMs is not in the computation speed, but in the GPU memory. A 100B-parameter model will require 200 GB of GPU memory to load the model parameters at the standard half precision, and many LLMs are even larger. Providing this much GPU memory is financially challenging for many users.

Recently, a model-parallel approach has been proposed to address the above challenge through resource pooling across distributed devices. This approach distributes a model to multiple devices at the granularity of transformer blocks (a.k.a. pipeline parallelism [3]) or neurons (a.k.a. tensor parallelism) to run large models on devices with small GPUs. In particular, [1] has shown that pipeline parallelism can be used to run LLM inference tasks on geographically distributed servers, each with only a few GB of GPU memory, at a much faster rate than local parameter offloading. This is achieved by letting each server host a subset of consecutive blocks, and each inference request is routed through a chain of servers that collectively host the entire model.

In this work, we consider LLM inference over geographically distributed servers using pipeline parallelism and client-centric communication, as illustrated in Fig. 1. Although the feasibility of such systems has been validated in [1], there is a lack of fundamental understanding of how to optimally manage their performance. The current solution in [1] based

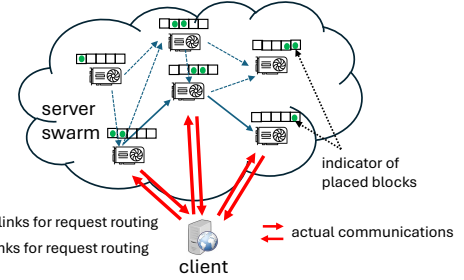


Figure 1: System architecture for pipeline-parallel LLM inference using client-centric communication.

on heuristics for resource allocation. It remains open how to optimize the performance of such systems, while taking into account the unique characteristics of GPUs and LLM inference tasks. This work aims to fill this gap by rigorously formulating and addressing the block placement and request routing problem, using PETALS [1] as a concrete example.

2. MODEL

We consider the inference process of a pre-trained LLM with a decoder-only architecture [4], which is commonly adopted by state-of-the-art LLMs (e.g., GPT and its variants [2]). The input/output layers only contain a small percentage of the model parameters (e.g. < 3% for BLOOM-176B [5]) and can therefore be hosted by the client. The model blocks contain most of the parameters and are thus delegated to the servers [1].

Given a pre-trained LLM with L blocks, we consider a distributed system with client-side and server-side caching, which uses this LLM to serve autoregressive sequence generation requests via a hub-spoke communication pattern as shown in Fig. 1. In this work, we focus on supporting short-prompt, long-response queries, where users ask concise questions to obtain detailed and comprehensive answers. The system contains a set of servers V_s and a set of clients V_c , communicating through TCP connections. Each client represents an ingress point for inference requests, which can be a host owned by end users or a proxy server that submits requests on behalf of end users.

Define $|R|$ as the total number of requests. For each inference request, the client initiates an inference session. We first consider the offline case with a given set of requests to understand the structure of the problem, and then address the online case with sequentially arriving requests.

To formulate the offline case, each request has up to $l_{\max}^I$ input tokens and generates up to $l_{\max}^O$ output tokens. For

*This work was partly supported by the National Science Foundation under award CNS-2106294. Parimal was supported by Qualcomm Inc. under Qualcomm University Relations 6G India, Anusandhan National Research Foundation (ANRF) under Grant CRG/2023/008854, and Office of International Relations at IISc under IISc-PSU collaboration research grant. Corresponding author: tinghe@psu.edu

short-prompt long-response queries, we have $l_{\max}^I \ll l_{\max}$. As for the online case, we need to additionally model the state of each server as detailed in the methods section.

Each server $j \in V_s$ has a per-block processing time of up to $\tau_j^I(l_{\max}^I)$ during the prefill phase and a per-token-per-block processing time of τ_j during the decoding phase. The connection between each client c and each server j has a per-input round trip time (RTT) of up to $t_{cj}^I(l_{\max}^I)$ and a per-token RTT of t_{cj} .

If a request from client c with input length $l_{\max}^I$ and output length $l_{\max}$ is processed by a chain of servers p with each server $j \in p$ processing k_j blocks, it will incur a total inference time of

$$\sum_{j \in p} \left(t_{cj}^I(l_{\max}^I) + k_j \tau_j^I(l_{\max}^I) \right) + (l_{\max} - 1) \sum_{j \in p} (t_{cj} + k_j \tau_j). \quad (1)$$

Each server $j \in V_s$ has a GPU memory of M_j , which denotes the effective memory capacity for storing model parameters and attention caches. The size of each block is s_m bytes. The size of each attention cache is $s_c := 2d_{\text{model}} \cdot (l_{\max}^I + l_{\max}) \cdot \text{dtype_bytes}$ bytes, where d_{model} is the embedding dimension and $\text{dtype_bytes} = 2$.

A server j storing m_j blocks and processing k_j^r blocks for each request r has a total memory consumption of

$$s_m m_j + s_c \sum_{r \in \mathcal{R}} k_j^r. \quad (2)$$

where s_m is the memory size per block, and s_c is the memory size per attention cache.

3. METHODS

We formulate a joint optimization of the placement of model blocks at servers (called block placement BP) and the selection of server chains for inference requests (called request routing RR) based on performance models experimentally validated on a real distributed LLM inference system. Our formulation features a compact representation of the decision variables that allows the problem to be formulated as a mixed integer linear programming (MILP) problem with a polynomial number of variables/constraints.

The NP-hardness of BPRR implies the need of efficient suboptimal algorithms. We develop a three-step algorithm called CG-BPRR by decomposing the joint optimization into subproblems optimizing the number of blocks per server m , the starting block per server a , and the request routing f , achieving polynomial complexity and a guaranteed average inference time per token under feasible block placements. It proceeds as follows:

Step 1: Conservatively compute the number of blocks m_j for each server j as $\min(\lfloor M_j / (s_m + s_c |R|) \rfloor, L)$.

Step 2: Sort servers by increasing amortized inference time defined as $\tilde{t}_j := \frac{1}{m_j} \max_{c \in V_c} (t_{cj} + \tau_j m_j)$ (fastest first), then greedily assign consecutive blocks to each server in this order, prioritizing ranges that improve performance for the most under-served blocks or balance load across well-served ones.

Step 3: For each client c , build a routing graph with S-client and D-client nodes, edges between servers hosting adjacent block ranges, and costs t_{ij}^c , then route requests via shortest path from S-client to D-client.

We then consider the online setting with dynamically arriving requests, for which we show that the offline algorithm

can be adapted into a two-time-scale solution that solves the block placement problem via robust optimization and the online request routing problem via a variation of shortest-path routing, with link costs $t_{ij}^W(t) + l_{\max} t_{ij}^c$ designed to approximate the total inference time used at each server. The online algorithm inherits the same performance guarantee as the offline algorithm as long as the number of concurrent requests is within a design limit, which can be tuned to trade off the waiting time and the per-token inference time after waiting.

4. RESULTS AND EVALUATION

Evaluations employ PETALS-based experiments on A100 GPUs and MIGs, alongside a validated MATLAB simulator capable of predicting GPU-based performance using CPU resources alone. This simulator replicates block placement and request routing decisions, enabling assessments of large-scale deployments beyond the constraints of available GPU hardware.

Clustered scenarios: Configured with three heterogeneous clusters (Cluster0 for remote clients, Cluster1 with high performance A100 servers, Cluster2 with lower performance MIGs), the proposed algorithm achieves 60–80% reductions in average per-token inference times compared to PETALS heuristics. Improvements arise from optimized memory allocation, reducing first-token times by minimizing out-of-memory errors.

Scattered scenarios: Tested on topologies like AboveNet (23 nodes), BellCanada (48 nodes), and GTS-CE (149 nodes), gains are confirmed, larger in resource-constrained settings. Varying parameters like number of servers C , request rate λ , and $l_{\max}$ in the simulator shows enhancements more pronounced under higher loads or fewer resources.

Ablation studies: Variations of PETALS (optimized ordering, fixed number of blocks, enhanced routing decision) highlight synergistic benefits of the integrated approach.

Algorithm running times remain negligible (under 0.05 seconds) compared to inference durations. The simulator’s results align with experiments in times and logic, facilitating scalable evaluations and future research on distributed LLM inference without extensive GPU access.

5. REFERENCES

- [1] A. Borzunov, M. Ryabinin, A. Chumachenko, et al. Distributed inference and fine-tuning of large language models over the internet. In *Advances in Neural Information Processing Systems*, volume 36, pages 12312–12331, 2023.
- [2] T. B. Brown, B. Mann, N. Ryder, et al. Language models are few-shot learners. In *International Conference on Neural Information Processing Systems*, 2020.
- [3] Y. Huang, Y. Cheng, A. Bapna, et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. In *Advances in Neural Information Processing Systems*, volume 32, 2019.
- [4] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. 2018.
- [5] T. L. Scao, A. Fan, C. Akiki, et al. BLOOM: A 176B-parameter open-access multilingual language model, 2023.

LLMEmu: A Lightweight Performance Emulator for High-Fidelity Distributed LLM Training

Siyuan Yang¹, Enda Yu¹, Pingjing Lu¹, Dezun Dong^{2*}

^{1 2} College of Computer Science and Technology, National University of Defense Technology, China

1. ABSTRACT

The prohibitive cost of training trillion-parameter large language models (LLMs) necessitates low-cost emulation tools for distributed system optimization. In modern large-scale clusters, communication often becomes the primary bottleneck to scalability. However, existing emulators, such as vTrain and ASTRA-Sim, overlook dynamic network factors that significantly impact performance at scale, resulting in limited emulation accuracy. This work offers an efficient and reliable tool for training system optimization and parallel strategy exploration, considerably lowering the barrier to large-scale AI research. We present LLMEmu, a distributed training emulator that combines real kernel profiling and actual communication execution. First, computation is profiled through real CUDA kernel traces on GPU nodes to construct an operator-level latency lookup table, enabling GPU-like execution on CPU clusters. Second, inter-node communication is executed using communication library primitives (e.g., AllReduce, Send/Recv), triggered by communication anchors embedded in the execution graph, and implemented using a pluggable communication backend.

LLMEmu can seamlessly model hybrid parallelism strategies and supports multiple collective algorithms. Its lightweight design incorporates gradient bucketing with latency reuse to minimize overhead while maintaining extensibility to various network interconnects. The effectiveness of LLMEmu is validated through its performance results, demonstrating an average prediction error of only 2.17% on 24-GPU clusters, which outperforms vTrain by 21.09%, and confirming its scalability in modeling training cost distributions across 128-node CPU emulations under varying network conditions.

2. DESIGN

We introduce LLMEmu, a real-execution-based LLM distributed training emulator (Fig. 1). The workflow comprises five key components: system configuration input, computation graph construction, real-world compute capability calibration, empirical communication measurement, and iterative time emulation. LLMEmu takes as input a set of system-level configurations, including model architecture, hardware specifications (e.g., GPU type and count), cluster settings (e.g., bandwidth, NIC type), and parallel strategies (DP, TP, PP) (Step 1). Based on this input, a pre-

*Corresponding author: Dezun Dong, dong@nudt.edu.cn

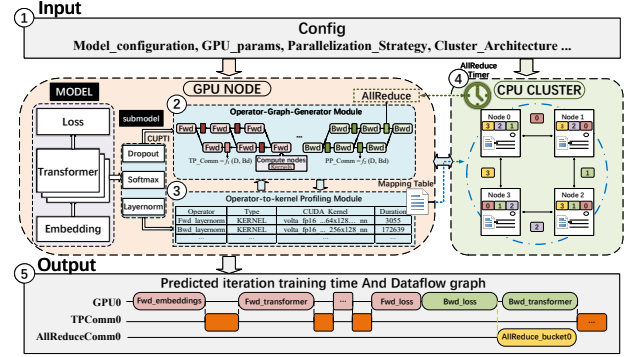


Figure 1: LLMEmu Overview.

processing stage constructs a Computation-Communication Graph (CCG) using offline profiling. Each model operator is mapped to a compute node, and communication tasks are inserted to reflect inter-node and intra-node data exchange (Step 2). To model computation, we profile each high-level operator once on a reference GPU node, recording its execution latency and associated CUDA kernels. These measurements form a compute template (Step 3), which is distributed across a CPU cluster configured to emulate a GPU cluster of equivalent scale under the assumption of homogeneous compute behavior, enabling accurate GPU emulation using only CPUs. For communication, LLMEmu departs from static modeling by executing actual Gloo-based collectives. Intra-node transfers, such as tensor-parallel (TP) and pipeline-parallel (PP) interactions, are emulated using the logP model to reflect different interconnects (e.g., NVLink, PCIe). Cross-node operations, such as gradient synchronization, trigger real AllReduce calls, and their latency is recorded by the rank-0 node (Step 4). We follow PyTorch’s gradient bucketing mechanism to avoid overestimating performance for small messages, issuing AllReduce only when the accumulated gradient exceeds 25MB [1]. Overall, LLMEmu integrates real communication behavior, fine-grained operator timing, and flexible parallel strategy modeling to provide accurate iteration time prediction for large-scale distributed training (Step 5). Given that each iteration in pretraining follows an identical computation graph, data flow, and communication pattern, the execution time of one iteration can be regarded as the minimal repeating unit of the whole training process. By accurately measuring the cost of a single iteration, LLMEmu captures the core

performance characteristics of the training system.

3. EVALUATION

To evaluate the emulation accuracy of LLMemu, we conduct experiments on a 6-node cluster using models of different scales: GPT-3 1.3B and 0.76B, and BERT-0.336B and 0.11B, with DeepSpeed. Multiple parallel configurations are tested, combining data, tensor, and pipeline parallelism (DP, TP, PP). For each configuration, we measure the actual iteration time during real training and compare it with the predictions generated by LLMemu (Fig. 2).

Across all tested models, LLMemu demonstrates high prediction fidelity. For GPT-3 1.3B and 0.76B, the average relative errors are 2.17% and 2.41%, respectively, with prediction deviations remaining within 0.20%–5.00%. For smaller BERT models, accuracy remains consistently high, achieving mean errors of 1.84% (BERT-0.336B) and 1.56% (BERT-0.11B). These results indicate that LLMemu can accurately emulate both computation and communication behaviors in distributed LLM training on real hardware. Due to space constraints, we only present the accuracy results on the 6-node configuration.

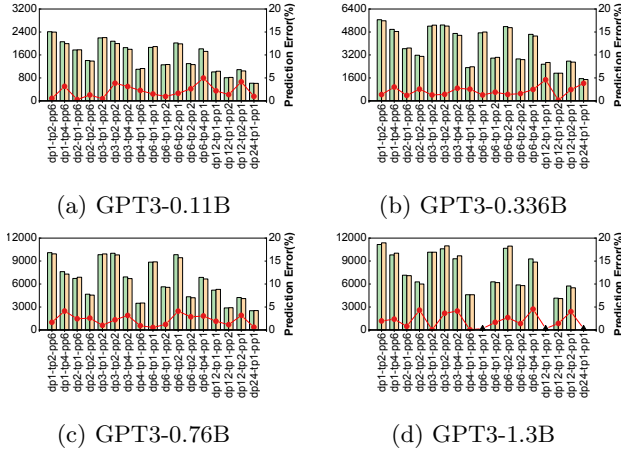


Figure 2: Comparison between LLMemu-predicted and actual iteration time on clusters with 6 servers (4 V100 GPUs per server). The x-axis denotes different distributed training strategies in the format: DP(x)-TP(y)-PP(z), total GPUs $N = x \times y \times z$. ▲ in the first row denotes OOM(CUDA out-of-memory).

We also conduct large-scale simulations on a CPU cluster (64-core EPYC 7452, 256 GB RAM) equipped with 1 Gbps Ethernet and 56 Gbps InfiniBand. The target GPU system has two 32 GB V100s per node. AllReduce is modeled via ring-chunked algorithm with a 0.1 scaling factor to emulate NCCL. We evaluate per-iteration overhead for GPT-3 (2.7B and 30B) under various cluster sizes; parallel settings are listed in Table 1.

Fig. 3 demonstrates that our emulator accurately predicts iteration time across different cluster sizes by separating total latency into computation and communication components. For GPT-3 30B under 56 Gbps interconnects, computation time decreases by 65.20% as GPUs increases. The reduction in compute time mainly comes from increased PP, which allows finer model partitioning, smaller per-GPU operators, and lower overall workload. By contrast, when PP is fixed, adding more nodes yields only minor gains, due to im-

Table 1: Training setup for GPT-3 model training.

Training Configuration	$N = 16$	$N = 32$	$N = 64$	$N = 128$
TP/DP/PP	4/4/4	4/8/4	4/8/8	4/16/8
Batch Size	128	256	512	1024

TP, DP, and PP specify the degree of each parallelism type in the 3D configuration. The computation graph is generated based on 4 V100 GPUs per node. Batch size set to 128 for 16-node runs to avoid OOM(CUDA Out Of Memory).

proved scheduling efficiency. This minor gain arises not from reduced workload but improved scheduling efficiency: with more GPUs, pipeline stages are more evenly distributed, and compute streams experience fewer synchronization stalls, as also noted in prior work [2].

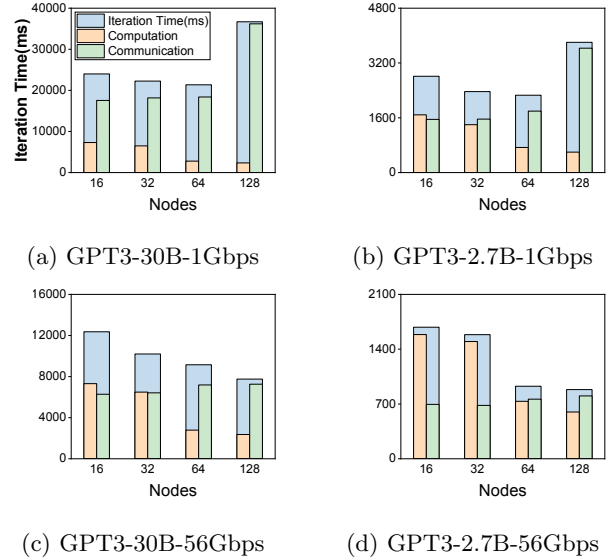


Figure 3: Breakdown of LLMemu-predicted iteration time, including computation and AllReduce communication, across various node scales, network bandwidths, and model sizes.

4. ACKNOWLEDGEMENTS

This work is supported by the National Key Research and Development Program of China under Grant No.2022YFB4501702, by the National Natural Science Foundation of China under Grant No.62421002 and No.U24B20151.

5. REFERENCES

- [1] S. Li, Y. Zhao, R. Varma, O. Salpekar, P. Noordhuis, T. Li, A. Paszke, J. Smith, B. Vaughan, P. Damania, et al., PyTorch Distributed: Experiences on Accelerating Data Parallel Training, Proc. VLDB Endow. 13 (12) (2020) 3005–3018.
- [2] D. Narayanan, A. Harlap, A. Phanishayee, V. Seshadri, N. R. Devanur, G. R. Ganger, P. B. Gibbons, M. Zaharia, PipeDream: generalized pipeline parallelism for DNN training, in: SOSP, 2019, pp. 1–15.

ScaleGS: Scalable Distributed Framework for Large-Scale 3D Gaussian Splatting with Edge Communication

Yong Kou^a, Jinlong He^a, Xia Yuan^{a,b}, Dening Luo^b, Yanci Zhang^{a,*}

^aCollege of Computer Science, Sichuan University, Chengdu 610065, China

^bChengdu University of Information Technology, Chengdu 610225, China

E-mails: kouyong@stu.scu.edu.cn, jinlonghe_dorcy@foxmail.com, yuanxia@cuit.edu.cn, dening@cuit.edu.cn, yczhang@scu.edu.cn

ABSTRACT

3D Gaussian Splatting (3DGS) has recently demonstrated outstanding performance in 3D reconstruction and real-time rendering. However, its scalability to large scenes remains limited by single-GPU memory constraints. We propose ScaleGS, a scalable distributed training framework for large-scale 3DGS with lightweight edge-aware communication. (1) We present a spatial median-guided binary partitioning algorithm that divides the point cloud into balanced, non-overlapping, and spatially contiguous cuboid regions for efficient multi-GPU management. To ensure global view consistency, each GPU independently grows and updates only its local Gaussians, while cross-GPU Gaussians are accessed only for rendering and loss computation. (2) We design a lightweight edge communication strategy to significantly reduce cross-GPU communication overhead. A greedy GPU-Tile remapping algorithm leverages the spatial concentration of Gaussians to confine cross-GPU communication to edge regions, effectively decoupling communication complexity from GPU count, with per-GPU complexity remaining $\mathcal{O}(1)$. An optimized all-to-all communication scheme is also introduced to eliminate redundant transmissions. (3) Our framework introduces an adaptive edge-refined load balancing mechanism that periodically monitors GPU workloads and selectively migrates Gaussians between neighboring GPUs to maintain balance and spatial continuity with negligible cost. Evaluations on large-scale 4K scenes show that ScaleGS consistently outperforms state-of-the-art methods, achieving up to 20% faster training and approximately 20% model size reduction on 8 Tesla P40 GPUs without compromising reconstruction quality. Project page: <https://aicodeclub.github.io/ScaleGS>.

1. FRAMEWORK

We propose ScaleGS (Figure 1), a distributed framework that systematically addresses challenges in visual consistency, communication scalability, and load balancing.

Median-Guided Partitioning. We introduce a spatial median-guided binary partitioning algorithm that recursively identifies the region with the highest point density and bisects it along its longest axis, resulting in a set of non-overlapping, balanced, and spatially continuous cuboidal subregions that form a regular 3D decomposition well suited for distributed training.

Autonomous Partition Growth. In distributed 3DGS, merging partitions without special handling often produces boundary artifacts due to inconsistent local views. We propose an autonomous partition growth strategy that ensures global Gaussian uniqueness and synchronizes essential Gaussian information, enabling artifact-free merging without post-processing.

Pixel-Tile Parallelism. To reduce single-GPU training time and memory usage, we adopt a pixel-tile parallelism by dividing each image into disjoint tiles and assigning each tile to a GPU via GPU-Tile mapping. Each GPU computes the loss for its assigned tile in parallel, reducing the single-GPU computation time to $1/M$, where M is the number of GPUs.

Lightweight Edge Communication. To reduce cross-GPU communication overhead, we develop a greedy GPU-Tile remapping strategy that exploits the spatial locality of Gaussian primitives to restrict cross-GPU Gaussian access to a minimal set of edge regions between adjacent GPUs (Figure 2). We optimize the all-to-all communication to eliminate unnecessary cross-GPU synchronization.

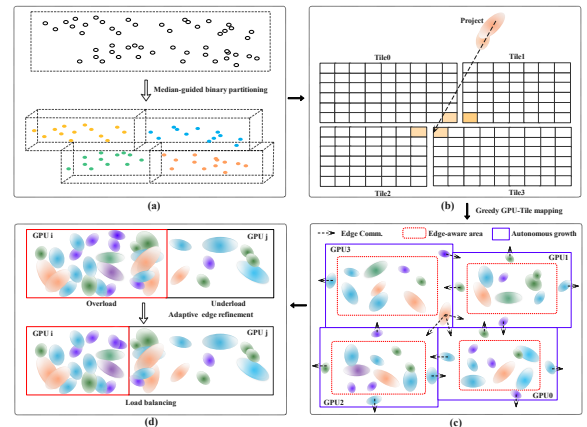


Figure 1: Overview of ScaleGS. (a) The initial point cloud is partitioned into non-overlapping, balanced, and spatially contiguous cuboidal subregions. (b) Pixel tiles are dynamically assigned to GPUs via one-to-one mapping. (c) The greedy remapping strategy limits Gaussian access to region edges, allowing each GPU to update its Gaussians autonomously. (d) The system periodically monitors load and redistributes a small number of Gaussians and optimizer states between imbalanced neighbors.

Table 1: Throughput (TP, iterations/sec), Training time (TT, sec), and image quality for Grendel-GS and ScaleGS (Ours) on Rubble, MatrixCity, and Garden. PS: PSNR, SS: SSIM, LP: LPIPS.

Method	Scene	1×GPU/BS1					2×GPU/BS2					4×GPU/BS4					8×GPU/BS2					8×GPU/BS4					8×GPU/BS8				
		TP↑	TT↓	PS↑	SS↑	LP↓	TP↑	TT↓	PS↑	SS↑	LP↓	TP↑	TT↓	PS↑	SS↑	LP↓	TP↑	TT↓	PS↑	SS↑	LP↓	TP↑	TT↓	PS↑	SS↑	LP↓	TP↑	TT↓	PS↑	SS↑	LP↓
Grendel-GS	Rubble	3.50	8613	26.17	0.806	0.216	3.70	8453	26.97	0.823	0.206	3.80	8021	27.35	0.832	0.202	4.10	7510	27.69	0.836	0.192	4.50	6791	27.56	0.836	0.202	4.90	6220	27.56	0.836	0.199
	MatrixCity	9.50	3240	25.09	0.780	0.231	11.30	2698	25.38	0.788	0.225	13.20	2290	25.48	0.791	0.224	15.10	2016	25.57	0.793	0.226	19.18	1632	26.21	0.807	0.215	23.90	1317	26.13	0.805	0.216
	Garden	9.45	3257	30.35	0.898	0.166	13.50	2351	31.41	0.921	0.153	20.50	1516	31.68	0.924	0.159	31.47	967	32.10	0.936	0.145	34.50	883	31.70	0.925	0.149	36.20	843	31.61	0.920	0.155
Ours	Rubble	3.46	8713	25.94	0.801	0.219	4.01	7886	26.98	0.824	0.206	4.30	7192	27.48	0.835	0.200	4.48	6809	27.67	0.839	0.198	5.60	5420	27.69	0.839	0.198	5.90	5297	27.59	0.837	0.196
	MatrixCity	9.20	3264	25.11	0.782	0.229	12.67	2383	25.35	0.788	0.226	14.98	2084	25.53	0.792	0.224	17.36	1792	25.62	0.794	0.223	23.20	1396	26.48	0.813	0.212	28.60	1195	26.21	0.807	0.218
	Garden	9.01	3202	29.92	0.881	0.171	16.10	2129	31.20	0.916	0.156	24.43	1278	31.70	0.927	0.150	39.40	792	32.17	0.938	0.144	41.09	760	31.67	0.927	0.152	41.92	738	31.59	0.925	0.151

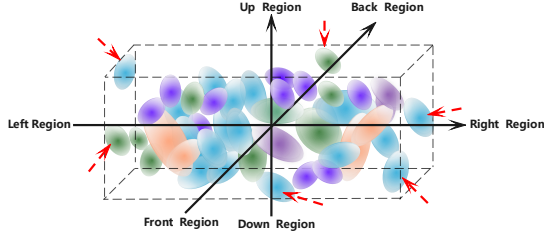


Figure 2: In 3D space, a cuboid has at most six face-adjacent neighbors. Exploiting this property, we introduce a greedy GPU-Tile mapping algorithm that restricts inter-GPU exchanges to the boundary Gaussians of at most six adjacent regions for local loss computation and gradient updates. Therefore, ScaleGS limits each GPU’s communication to at most six peers.

Adaptive Edge Refinement. The system periodically monitors the Gaussian load on each GPU and selectively adjusts partition edges between imbalanced GPU pairs. These localized updates maintain spatial continuity while achieving fast and stable load redistribution.

2. EXPERIMENT

All experiments are conducted on a single-node server equipped with 8 NVIDIA Tesla P40 GPUs.

2.1 Scalability and Training Efficiency

Table 1 shows that as the number of GPUs increases, ScaleGS consistently outperforms Grendel-GS in throughput and training speed, achieving up to 20% improvement due to our proposed algorithm and load-balancing strategy. As shown in Table 2, compared to VastGaussian and CityGaussian, ScaleGS achieves up to 30% better scalability and communication efficiency.

2.2 Rendering Quality

Figure 3 and Table 2 show that ScaleGS maintains high reconstruction quality in all scenes, mainly due to the spatial concentration of Gaussians. Overlap-based methods like CityGaussian and VastGaussian often produce artifacts. Supplemental results on MatrixCity confirm our method eliminates boundary artifacts effectively.

3. CONCLUSION

We propose ScaleGS, a scalable distributed training framework for large-scale 3D Gaussian Splatting. Through non-overlapping spatial partitioning and autonomous partition

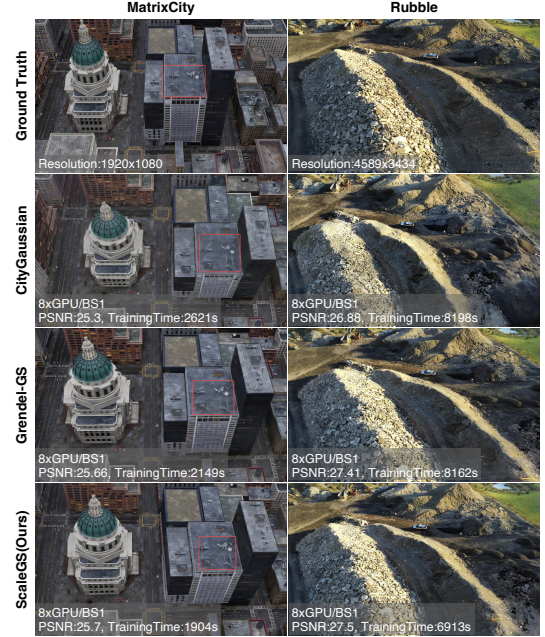


Figure 3: On Rubble, ScaleGS achieves a 20% reduction in training time compared to SOTA methods, while preserving rendering quality.

Table 2: Performance comparisons on the Garden scene. TP and TT are defined in Table 1.

Method	2×GPU/BS1				4×GPU/BS1				8×GPU/BS1			
	TP↑	TT↓	PSNR↑		TP↑	TT↓	PSNR↑		TP↑	TT↓	PSNR↑	
VastGaussian	12.49	2425	30.39		18.98	1610	31.20		26.47	1230	31.36	
CityGaussian	10.43	3082	30.10		16.44	1892	31.41		27.32	1289	31.43	
Grendel-GS	12.47	2390	30.56		19.80	1594	31.36		29.93	1139	32.10	
Ours	15.24	2210	30.98		21.45	1410	31.32		33.42	912	31.92	

growth, edge-aware cross-GPU communication with decoupled complexity, and lightweight adaptive load balancing, ScaleGS overcomes the GPU-scaling bottlenecks of existing approaches. **In particular, our framework achieves a per-GPU communication complexity of $\mathcal{O}(1)$, thereby establishing a theoretically grounded foundation for scalable distributed training in 3D Gaussian Splatting.** Extensive experiments demonstrate that ScaleGS significantly outperforms current SOTA methods in training speed, model size, and scalability, while maintaining rendering quality, underscoring its potential in large-scale 3D reconstruction.

TiFSN: A Wavelet-EC-TCN Model for Quadrotor UAV Trajectory Prediction Based on Time-Frequency-Spatial Feature Fusion

Huan Zhao¹, Yong Kou², Yuxin Xue³, Shuang Wang⁴, Zhaojun Gu^{4*}

¹College of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China

²College of Computer Science, Sichuan University, Chengdu 610065, China

³College of Safety Science and Engineering, Civil Aviation University of China, Tianjin 300300, China

⁴Institute of Science, Technology and Innovation, Civil Aviation University of China, Tianjin 300300, China

ABSTRACT

Flight trajectory prediction (FTP) with high precision is the core technology for the autonomous flight of quadrotor unmanned aerial vehicles (UAVs) in environments with limited navigation signals. In response to the problem that most existing methods focus on the features of a single domain and ignore the cross-domain feature correlation, making it challenging to maintain high accuracy in FTP, a model based on **Time-Frequency-Spatial Network** named TiFSN is proposed. Firstly, based on wavelet transform (WT), the velocity signal is extended to time-frequency joint features. Furthermore, a fusion mechanism is established to construct a multi-domain feature set. Finally, an extended channels-based temporal convolutional network (EC-TCN) is designed, which achieves high-precision FTP. Experiments were conducted on real flight datasets, and the results show that the model significantly improved the evaluation metrics compared to baseline methods.

1. METHODOLOGY

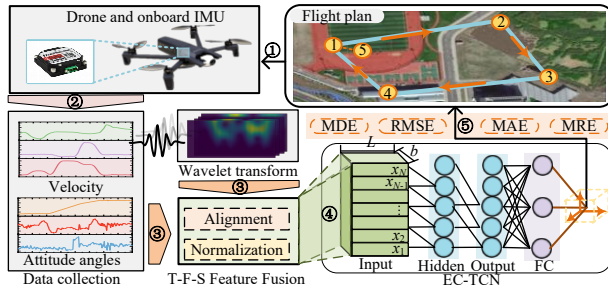


Figure 1: Framework of TiFSN.

Figure 1 shows the working and evaluation process of the TiFSN, which mainly includes the following parts:

- 1) **Flight trajectory planning module:** An autonomous flight plan program is developed, which can convert the three-dimensional spatial coordinate sequence into a set of instructions for the UAV.

*Corresponding author: Zhaojun Gu, zjgu@cauc.edu.cn

- 2) **Data acquisition module:** The flight data, containing velocity components and attitude angles, are obtained by inertial measurement units (IMUs) [1].
- 3) **Time-frequency-spacial feature fusion module:** Firstly, WT is used to perform time-frequency decomposition on the velocity signal [2]. Subsequently, the attitude angles are aligned and standardized, and finally connected with the former to form a set of joint features, providing standardized input.
- 4) **Extended channels-based TCN:** The feature matrix is fed into EC-TCN, and a structure based on extended causal convolution is used to perform long-range dependency modeling [3]. The output is mapped to spatial trajectories through a fully connected layer.
- 5) **Performance evaluation module:** The results of TiFSN are compared with the real coordinates, and quantitative evaluation is implemented based on indicators such as mean deviation error (MDE) [4], mean absolute error (MAE) and so on.

2. EXPERIMENT

This study use the Adam optimizer [5] to train the model parameters, with an initial learning rate of 0.001 and a decay rate of 0.5 every 10 epochs. The total training epochs are set as 30, with an early stop tolerance of 10 epochs. The loss function is MSE [6], and the time window length is 20. baseline models are as follows: **M1** RNN, **M2** GRU, **M3** CNN, **M4** LSTM, **M5** Transformer, **M6** BiLSTM, **M7** (EC-)TCN, without WT.

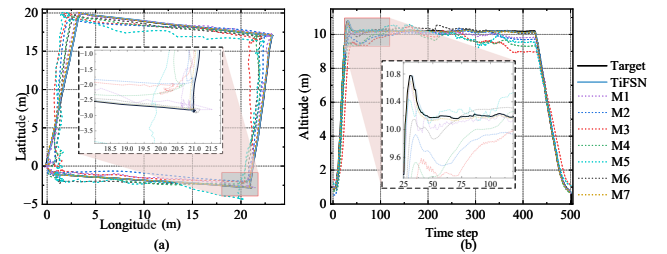


Figure 2: Visualization of flight trajectory: (a) Trajectory in the $x - y$ plane, (b) Variation of altitude.

Fig 2 presents a comparison of different methods during stages of takeoff, cruise, heading adjustment, and landing.

Copyright is held by author/owner(s).

Table 1: Comparison of testing performance.

Metrics	M1	M2	M3	M4	M5	M6	M7	TiFSN
Size(bit)	8k	376k	56k	496k	1.7M	1.3M	92k	92k
GPU	TC(ms)	0.276(± 1.837)	0.109(± 0.468)	0.182(± 3.341)	0.099(± 0.011)	0.247(± 0.336)	0.122(± 0.028)	0.287(± 0.307)
	MDE(m)	1.118(± 0.607)	2.830(± 1.827)	3.313(± 1.552)	2.248(± 1.232)	4.962(± 2.799)	2.778(± 1.279)	0.307(± 0.209)
CPU	TC(ms)	0.415(± 1.264)	0.351(± 0.232)	0.168(± 0.561)	0.596(± 1.220)	1.602(± 2.051)	0.928(± 1.033)	0.709(± 1.057)
	MDE(m)	1.359(± 0.685)	2.526(± 1.105)	3.469(± 2.076)	3.578(± 1.219)	6.185(± 1.603)	1.756(± 0.903)	0.452(± 0.216)

Overall, the TiFSN model significantly outperforms the comparison methods in various indicators, verifying the effectiveness of TiFSN. Thanks to the effective fusion of cross-domain features, the MDE of TiFSN is 0.196, which is 18.7% lower than the suboptimal model M7. However, the MDE under the CNN (M3) model reaches a maximum value of 1.997m, indicating that convolution processing relying only on local spatiotemporal features cannot effectively model long-range trajectory correlations.

The model with the minimum validation loss during the training is retained as the optimal model. Table 1 shows the performance comparison of TiFSN and 7 baseline models, tested 30 times on GPU and CPU, respectively, covering three core metrics: model file size, average time cost (TC), and MDE. All data are presented in the form of *mean* $\pm$ *standard deviation*.

To further validate the generalization of TiFSN, three sets of generalization test datasets were tested using the optimal models saved. Furthermore, we conducted tests by connecting with ANAFI drones using the onboard Ambarella H22 processor, which integrates an ARM Cortex-A9 dual-core CPU and a dedicated image signal processor (ISP), capable of processing computations for small neural networks.

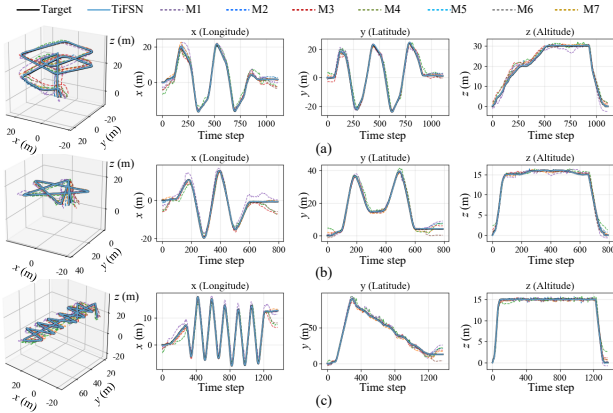


Figure 3: Prediction performance under different trajectories: (a) T1: Rectangular ascend, (b) T2: star shaped, (c) T3: zigzag shaped.

3. CONCLUSION

This study proposes a model named TiFSN to address the high-precision FTP problem of quadrotor UAV. The study firstly extends the data features based on WT. Subsequently, a feature set is constructed by combining the spatial angles with time-frequency features of velocity. Finally, EC-TCN is designed for training and testing. The experimental results show that the TiFSN achieves an average MDE of 0.196 m,

effectively improving the accuracy of FTP tasks. Besides, the indicators such as RMSE and MAE are significantly better than baseline models. The generalization experiment verified the adaptability to different complex paths. The ablation experiment proved the influence of wavelet decomposition layers and extension channels on the performance of the model. In future research, we will focus on the application of WT technology in more scenarios, such as multi-machine collaborative trajectory prediction.

4. ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China under Grant U2333201 and the Fundamental Research Funds for the Central Universities of Civil Aviation University of China under No. 3122025054.

5. REFERENCES

- [1] Y. Liu, H. Wang, J. Fan, J. Wu, T. Wu, Control-oriented UAV highly feasible trajectory planning: A deep learning method, *Aerospace Science and Technology* 110 (2021) 106435. [doi:10.1016/j.ast.2020.106435](https://doi.org/10.1016/j.ast.2020.106435)
- [2] M. Vidal, A. M. Aguilera, Wavelet thresholding on independent subspace factorizations of spatially indexed wide functional data for robust estimation of cortical activity, *Mathematics and Computers in Simulation* 232 (2025) 346–361. [doi:10.1016/j.matcom.2025.01.012](https://doi.org/10.1016/j.matcom.2025.01.012)
- [3] A. Liu, K. Zhu, C. Song, L.-h. Sun, F. Cai, Natural phenolic compounds against trichothecenes: From protective mechanisms to future perspectives, *Journal of Agricultural and Food Chemistry* 73 (15) (2025) 8740–8755, publisher: American Chemical Society. [doi:10.1021/acs.jafc.4c12790](https://doi.org/10.1021/acs.jafc.4c12790)
- [4] J. Luo, L. Tang, Q. Chen, Z. Zhang, Trajectory Design and Bandwidth Allocation Considering Power-Consumption Outage for UAV Communication: A Machine Learning Approach, *IEEE Transactions on Industrial Informatics* 20 (2) (2024) 2519–2528. [doi:10.1109/TII.2023.3292522](https://doi.org/10.1109/TII.2023.3292522)
- [5] D. P. Kingma, Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980* (2014).
- [6] X. Liu, Y. Liu, Y. Chen, L. Hanzo, Trajectory design and power control for multi-UAV assisted wireless networks: A machine learning approach, *IEEE Transactions on Vehicular Technology* 68 (8) (2019) 7957–7969. [doi:10.1109/TVT.2019.2920284](https://doi.org/10.1109/TVT.2019.2920284)

Switching Constrained OCO with Predictions and Feedback Delays

Weici Pan, Zhenhua Liu
Stony Brook University

weici.pan@stonybrook.edu, zhenhua.liu@stonybrook.edu

ABSTRACT

We examine Online Convex Optimization (OCO) problems with feedback delay and a strict limit on decision switching, which exists in applications such as smart grid and learning. Existing algorithms developed for traditional OCO struggle in this setting, often violating switching constraints or incurring high regrets, as evidenced by simulations. In this paper, we establish a new algorithm, Follow-the-Maximally-Coupled-Latest-Leader (FMCLL), achieving a near-optimal regret of $O(T/S)$ for such problems with delayed feedbacks and a bound of $O(T/S - \tau)$ for problems with predictions of τ rounds, even though the player is only allowed to move at most S times in expectation across T rounds. FMCLL meets performance bounds in scenarios with delays and predictions by using maximal coupling sampling to inform algorithm design for switching-constrained problems. It is also extended to a bandit feedback setting. Simulations demonstrate FMCLL's superiority over traditional Gradient Descent or Follow-the-Leader algorithms, excelling under adversarial or stochastic losses and reducing constraint violations.

1. INTRODUCTION

Given $\mathcal{A}$ as an algorithm for OCO chosen from the policy family, that maps a certain game history to a decision in the decision set: $x_t^{\mathcal{A}} = \mathcal{A}(f_1, \dots, f_{t-1}) \in \mathcal{D}$. The regret of $\mathcal{A}$ post T rounds is the difference between the total incurred cost and that of the best fixed decision in hindsight:

$$\text{Regret}_T(\mathcal{A}) = \sup_{\{f_1 \dots f_t\} \subset \mathcal{F}} \left(\sum_{t=1}^T f_t(x_t^{\mathcal{A}}) - \min_{x \in \mathcal{D}} \sum_{t=1}^T f_t(x) \right).$$

The online player's goal is to have the lowest possible regret relative to $\mathcal{A}$.

However, when it involves feedback delays and predictions under the switching constraint, things diverge from the traditional settings. First is the feedback delay. In practice, decision makers often collect the feedback information about the environment with delays [4]. It is also possible in many applications that the online player can make predictions of future losses [3, 2]. In previous research of switching constrained optimization without any feedback delays, a minimax regret of $\Theta(T/\sqrt{S})$ has been demonstrated [1], while another study [5] presents a regret bound of $O(T/S)$ for gen-

eral convex losses and $O(T/S^2)$ for strongly convex losses. These findings hold significance because, in many real-world applications, there is a preference for online algorithms that exhibit greater continuity in their actions across multiple rounds.

1.1 Contributions

To the best of our knowledge, there has been no prior work addressing the topic of Lazy OCO with delayed or advanced information. We would like to introduce our main contributions as follows:

(1) We propose a novel problem setting where OCO problems meet both switching constraints and feedback delay or predictions. They cannot be trivially tailored to serve the new setting.

(2) We develop the first efficient lazy algorithm, Follow-the-Maximally-Coupled-Latest-Leader, specifically designed for Lazy OCO with feedback delay. The $O(T/S)$ regret upper bound of this algorithm has been proven to be comparable to the lower bound of no-delay algorithms in convex loss and oblivious adversary cases.

(3) We also extend the algorithm and results to bandit feedback, which is applicable to real practices.

(4) Our algorithm is general. It does not require the delay to be fixed, and is capable of handling Lazy OCO problems with predictions, matching the regret bound for online algorithms.

2. FORMULATION

Unlike the standard OCO setting, the player does not necessarily observe f_t immediately after choosing x_t . Instead, feedback may arrive with delay or be partially revealed in advance. Let k_t denote the smallest index such that $f_1, \dots, f_{k_t-1}$ have been revealed before the player selects x_t , but f_{k_t} and all later functions are not. We call f_{k_t} the *latest observed loss function* at round t . The value of k_t partitions the feedback into three categories: - If $k_t = t$, the player observes f_t immediately after choosing x_t (full-information feedback); - If $k_t < t$, the player receives feedback about f_t only after x_t is chosen (delayed feedback); - If $k_t > t$, the player observes f_t before choosing x_t (predicted feedback).

Moreover, a key distinction from classical OCO is the presence of a switching constraint. Let the switch count be defined as $\mathcal{S}_T = \sum_{t=1}^{T-1} \mathbf{1}\{x_{t+1} \neq x_t\}$. We require the player's strategy to satisfy $\mathbb{E}[\mathcal{S}_T] \leq S$ for some given switching budget $S \in [0, T]$, where the expectation is over the internal randomness of the algorithm (and over the loss sequence if

stochastic).

3. ALGORITHMS

Due to page limit we omit technical details here. Our method Follow-the-Maximally-Coupled-Latest-Leader (FM-CLL), extends FTRL/FTPL-style approaches to the switching-constrained setting. At each round, the learner constructs a perturbed cumulative loss function using the most recent available feedback, augmented with a regularizer and a barrier function to guarantee stability. The corresponding minimizer defines a leader distribution. The next decision is then obtained through a maximal-coupling mechanism, which probabilistically reuses the previous action when consecutive distributions overlap, and otherwise switches to a new sample from the updated distribution. This design ensures responsiveness to new feedback while keeping the expected number of switches within the prescribed budget.

To realize maximal coupling, we employ a procedure referred to as LazySample. Given two consecutive leader distributions, the subroutine maximizes the probability of reusing the previous decision, thereby reducing unnecessary switches. Formally, the probability of switching equals the total variation distance between the two distributions. This coupling-based mechanism provides an explicit and principled control over switching frequency, a property essential for regret guarantees under strict switching budgets.

4. RESULTS

Our framework accommodates both delayed and predictive feedback, as well as their combination. In the delayed case, losses are revealed only after several rounds, whereas in the predictive case, the learner has access to future losses in advance. The FMCLL algorithm applies uniformly to both scenarios, and the analysis shows that the regret guarantees smoothly interpolate between them. Moreover, in mixed settings where some rounds involve delays and others predictions, FMCLL can allocate the switching budget adaptively and maintain near-optimal performance.

THEOREM 4.1. *Let $\mathcal{D} \subset \mathbb{R}^d$ be convex with diameter D , and assume that each f_t is G -Lipschitz, β -smooth, and convex over $\mathcal{D}$. Suppose the loss feedback follows a revealing sequence $\{k_1, \dots, k_T\}$ with $k_t \leq t$. Assume the barrier B satisfies $0 \leq B(x) \leq 1$ for all x in the shrunk domain $\{x_0 + (1 - \frac{1}{T})(x - x_0) : x \in \mathcal{D}\}$ for arbitrary $x_0 \in \mathcal{D}$. Then running FMCLL with fixed noise scale $\sigma_t = \sigma$ and regularizer $R(x) = \frac{1}{2\eta}\|x - x_0\|^2$, we have:*

$$\mathbb{E}[R_T] \leq 2\eta G^2(\alpha + 1)T + \frac{D^2}{2\eta} + \sigma\sqrt{d}D + GD + 1, \quad (1)$$

$$\mathbb{E}[S_T] \leq \eta\beta dT + \frac{\sqrt{d}GT}{\sigma}, \quad (2)$$

where $\alpha := \frac{1}{T} \sum_{t=1}^T \mathbb{E}[t - k_t]$ is the average delay.

$$\sigma = \frac{2\sqrt{d}GT}{S}, \quad \eta = \min \left\{ \frac{D}{2G\sqrt{(\alpha + 1)T}}, \frac{S}{\beta dT} \right\} \quad (3)$$

$$\Rightarrow \mathbb{E}[R_T] = O \left(\sqrt{(\alpha + 1)T} + \frac{dT}{S} \right), \quad \mathbb{E}[S_T] \leq S. \quad (4)$$

THEOREM 4.2. *Let $\mathcal{D} \subset \mathbb{R}^d$ be convex with diameter D , and assume each f_t is convex, G -Lipschitz, and β -smooth*

over $\mathcal{D}$. Suppose that feedback is revealed τ rounds in advance, i.e., $k_t = t + \tau$. Assume the barrier B satisfies $0 \leq B(x) \leq 1$ for all x in the shrunk domain $\{x_0 + (1 - \frac{1}{T})(x - x_0) : x \in \mathcal{D}\}$ for arbitrary $x_0 \in \mathcal{D}$. Then running FMCLL with fixed $\sigma_t = \sigma$ and $R(x) = \frac{1}{2\eta}\|x - x_0\|^2$ yields

$$\mathbb{E}[R_T] \leq \frac{D^2}{2\eta} + \sigma\sqrt{d}D + GD + 1, \quad (5)$$

$$\mathbb{E}[S_T] \leq \eta\beta d(T - S\tau) + \frac{\sqrt{d}G(T - S\tau)}{\sigma}. \quad (6)$$

$$\sigma = \frac{2\sqrt{d}G(T - S\tau)}{S}, \quad \eta = \frac{S}{2\beta d(T - S\tau)} \quad (7)$$

$$\Rightarrow \mathbb{E}[R_T] = O \left(d \left(\frac{T}{S} - \tau \right) \right), \quad \mathbb{E}[S_T] \leq S. \quad (8)$$

5. BANDIT EXTENSION

Beyond delayed and predictive feedback, an important extension is the bandit feedback setting, where the learner observes only a scalar loss at the played point rather than the full loss function.

THEOREM 5.1. *Assume each f_t is convex and G -Lipschitz on $\mathcal{D} \subset \mathbb{R}^d$, with $f_t(x) \in [0, 1]$. Let $D = \text{diam}(\mathcal{D})$. Run a modified version of the algorithm for bandit feedback with*

$$\delta = (d^2 T / S)^{1/4}, \quad \eta = \frac{D\delta}{d\sqrt{T}}. \quad (9)$$

Then the expected regret and switch count satisfy

$$\mathbb{E} \left[\sum_{t=1}^T f_t(x_t) \right] - \min_{x \in \mathcal{D}} \sum_{t=1}^T f_t(x) = O \left(DGT^{3/4}S^{-1/4} \right), \quad (10)$$

$$\sum_{t=2}^T \Pr[x_t \neq x_{t-1}] \leq S. \quad (11)$$

Acknowledgement: This work is supported by the US National Science Foundation under grant numbers CNS-2106027, CNS-2214980, CNS-2146909, and CCF-2046444.

6. REFERENCES

- [1] L. Chen, Q. Yu, H. Lawrence, and A. Karbasi. Minimax regret of switching-constrained online convex optimization: No phase transition. *Advances in Neural Information Processing Systems*, 33:3477–3486, 2020.
- [2] N. Chen, J. Comden, Z. Liu, A. Gandhi, and A. Wierman. Using predictions in online optimization: Looking forward with an eye on the past. *ACM SIGMETRICS Performance Evaluation Review*, 44(1):193–206, 2016.
- [3] Y. Lin, G. Goel, and A. Wierman. Online optimization with predictions and non-convex losses. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(1):1–32, 2020.
- [4] A. Mitra, H. Hassani, and G. J. Pappas. Online federated learning. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 4083–4090. IEEE, 2021.
- [5] U. Sherman and T. Koren. Lazy OCO: Online convex optimization on a switching budget. In *Conference on Learning Theory*, pages 3972–3988. PMLR, 2021.

Bayesian Optimization for Dynamic Pricing and Learning

Anush Anand
IIIT Hyderabad

anush.anand@students.iiit.ac.in

Pranav Agrawal
IIIT Hyderabad

pranv.agrawal@students.iiit.ac.in

Tejas Bodas
IIIT Hyderabad

tejas.bodas@iiit.ac.in

ABSTRACT

We study the dynamic pricing problem in which a seller, facing an unknown stochastic demand function, must adaptively choose prices to maximize cumulative revenue. The seller observes only noisy demand observations after posting prices, leading to the need for simultaneous learning of the demand model and the optimization of pricing decisions. Existing methods typically rely on parametric assumptions (e.g., polynomial or logit demand) and optimize using Iterative Least Squares or reinforcement learning (RL) algorithms. These approaches are vulnerable to model misspecification, slow convergence, and high variance under noise.

We propose a nonparametric framework based on *Bayesian Optimization* (BO), where the revenue function is modeled using Gaussian Processes (GPs) and optimized through acquisition functions. We develop BO-based algorithms for both the **infinite inventory** and **finite inventory** settings. For the infinite inventory case, we design the *BO-Inf* algorithm, a GP-UCB variant that achieves $O(\sqrt{T} \log^2 T)$ regret, and a computationally efficient *Lightweight BO-Inf* variant based on bucketed pricing. For the finite inventory case, we introduce two approaches: (i) *GP-Fin-Model-Based*, which integrates GP posterior demand estimation with dynamic programming, and (ii) *BO-Fin-Heuristic*, a model-free algorithm that directly applies BO to finite-horizon decision-making. We establish theoretical guarantees for our algorithms and demonstrate, through extensive experiments, that they outperform classical bandit, a Controlled Variance pricing (CVP), and deep RL baselines in terms of convergence speed, robustness to noise, and resilience to model misspecification.

Keywords

Bayesian Optimization, Gaussian Processes, Dynamic Pricing, Bandits, Reinforcement Learning

1 Introduction

Dynamic pricing has long been studied in revenue management, with applications in airlines, hotels, ride-hailing, and e-commerce. A central difficulty is the lack of knowledge of the demand curve: the seller sets prices but only observes realized demand corrupted by noise. This requires algorithms that both *learn* the demand structure and *optimize* prices simultaneously.

The operations research literature traditionally adopts parametric assumptions on demand, such as linear, polynomial, or logit models, which allow tractable estimation via Iterative Least Squares (ILS) or Maximum Likelihood. Algo-

rithms such as Greedy ILS and Constrained ILS [1] achieve asymptotic optimality under correct specification. However, these approaches degrade substantially when assumptions are violated.

More recently, reinforcement learning (RL) methods, such as policy-gradient (PPO) or actor-critic (SAC, DDPG), have been proposed. While flexible, these methods converge slowly, require large amounts of exploration, and are difficult to deploy when sellers face constraints on the frequency of price changes.

In this work, we adopt a *Bayesian nonparametric* approach, modeling the unknown revenue function as a Gaussian Process (GP). Bayesian Optimization (BO) provides a natural framework for selecting prices by balancing exploration and exploitation using acquisition functions. Unlike parametric methods, BO makes minimal assumptions about functional form; unlike RL, it provides explicit uncertainty quantification and faster convergence with fewer samples. We design BO-based algorithms for both infinite and finite inventory regimes and provide theoretical regret guarantees.

2 Problem Formulation

We consider a seller who interacts with the market over T periods. At each time t , the seller posts a price $p_t \in [p_l, p_h]$ and observes a noisy demand realization $d_t \sim D(p_t)$. The revenue is $r_t = p_t \cdot d_t$, where the expected revenue function is $\bar{R}(p) = p \cdot \mathbb{E}[D|p]$. The goal is to minimize *regret* relative to the optimal price $p^* = \arg \max_p \bar{R}(p)$:

$$\mathcal{R}_T = \sum_{t=1}^T (\bar{R}(p^*) - \bar{R}(p_t)).$$

In the finite inventory case, the seller starts with C units of stock and faces horizon T and N selling seasons. At time t , the state is (s, t) with s units remaining. A price p yields random demand q , and the next state is $(s - q, t + 1)$. The objective is to maximize expected total revenue in season n , with regret defined relative to the optimal dynamic policy.

3 Algorithms

We now present our proposed algorithms. Throughout, we denote:

- t – the current time step, with horizon T ,
- n – the current season ($n \leq N$)
- $p_t \in [p_l, p_h]$ – the price chosen at time t , where $[p_l, p_h]$ is the feasible price range,
- d_t – the random demand observed at time t ,
- $r_t = p_t d_t$ – the realized revenue,

- $r(p) = p \cdot \mathbb{E}[d|p]$ – the expected revenue at price p ,
- $\mathcal{D}_{t-1} = \{(p_i, r_i)\}_{i=1}^{t-1}$ – the dataset of past price-revenue pairs,
- $\mu_{t-1}(p)$ – the GP posterior mean given $\mathcal{D}_{t-1}$,
- $\sigma_{t-1}(p)$ – the GP posterior standard deviation,
- κ – the exploration parameter used in GP-UCB acquisition.

3.1 BO-Inf

The *BO-Inf* algorithm applies the GP-UCB acquisition rule. At each round t , the next price is chosen by

$$p_t = \arg \max_{p \in [p_l, p_h]} \mu_{t-1}(p) + \kappa \sigma_{t-1}(p).$$

Here, $\mu_{t-1}(p)$ captures the algorithm’s current belief about expected revenue at price p , while $\sigma_{t-1}(p)$ captures the uncertainty. The parameter κ_t balances exploration (trying uncertain prices) with exploitation (choosing prices with high mean). We show that BO-Inf achieves regret $O(\sqrt{T} \log^2 T)$ under standard assumptions on the GP kernel and bounded noise.

3.2 Lightweight BO-Inf

The complexity of GP regression scales as $O(N^3)$ where N is the number of observations. To address scalability, *Lightweight BO-Inf* partitions the price space into B buckets. For each bucket, observed revenues are aggregated into a representative value. GP regression is then performed on these B points, reducing complexity to $O(B^3)$. Let $\tilde{\mu}_{t-1}(p)$ and $\tilde{\sigma}_{t-1}(p)$ denote the posterior mean and variance computed from the bucketed data. The decision rule is analogous:

$$p_t = \arg \max_{p \in [p_l, p_h]} \tilde{\mu}_{t-1}(p) + \kappa \tilde{\sigma}_{t-1}(p).$$

3.3 GP-Fin-Model-Based

In the finite-inventory setting, the state is (s, t) where s is the remaining stock and $t \in \{1, \dots, T\}$ is the current period. At season n , the GP posterior gives a Gaussian model for demand at price p with mean $\mu_n(p)$ and variance $\sigma_n^2(p)$. The algorithm constructs a discrete transition kernel by binning the continuous GP posterior:

$$\hat{P}_n(q | s, p) = \int_{q-0.5}^{q+0.5} \mathcal{N}(x; \mu_n(p), \sigma_n^2(p)) dx \quad (q = 0, \dots, s),$$

with any tail mass placed on the endpoints as needed. The value function is computed under this GP-derived model via backward dynamic programming:

$$\hat{V}_n(s, t) = \max_{p \in [p_l, p_h]} \sum_{q=0}^s \hat{P}_n(q | s, p) [p \cdot q + \hat{V}_n(s - q, t + 1)],$$

with $\hat{V}_n(\cdot, T + 1) = 0$. The policy $\psi_n(s, t)$ is the argmax above; the GP is re-fitted only at season boundaries using past data, and the resulting policy is executed during the season.

3.4 BO-Fin-Heuristic

The *BO-Fin-Heuristic* avoids dynamic programming. Instead, it directly uses the GP posterior mean $\mu_{n-1}(p)$ and variance $\sigma_{n-1}(p)$ to estimate cumulative returns under finite inventory. Prices are chosen using a BO-style acquisition criterion:

$$p_t = \arg \max_{p \in [p_l, p_h]} p \cdot \min \left(s_t, \mu_{t-1}(p) \cdot (T - t + 1) \right) + \kappa e^{-\lambda t} \sigma_{t-1}(p).$$

While lacking formal regret guarantees, the algorithm converges quickly and is computationally efficient.

4 Theoretical Analysis

For BO-Inf, we derive a high-probability regret bound of order $O(\sqrt{T} \log^2 T)$, adapting the analysis of [2]. The regret for Lightweight BO-Inf is bounded by the sum of the BO-Inf regret and a discretization term of order $O(1/B)$. For the finite inventory case, we establish that GP-Fin-Model-Based achieves near-optimality under assumptions of Hölder continuity of the revenue function and bounded noise. While BO-Fin-Heuristic lacks formal guarantees, empirical evidence demonstrates robust convergence.

5 Experiments

We conduct extensive experiments on synthetic and semi-realistic demand models.

Synthetic polynomial demand BO-Inf outperforms ILS, CVP, and Thompson Sampling in terms of convergence speed and regret, especially under noise. With observation noise at 50% of demand magnitude, BO methods maintain low regret, while parametric methods diverge.

Model misspecification For exponential demand functions, parametric baselines fail to converge, whereas BO remains effective.

Finite inventory GP-Fin-Model-Based achieves higher cumulative revenue and faster convergence than Endogenous Learning, PPO, SAC, and DDPG. BO-Fin-Heuristic achieves strong performance at significantly reduced computational cost.

Scalability Lightweight BO-Inf scales to horizons of thousands of rounds while maintaining near-optimal regret.

Our results suggest that BO methods offer a promising alternative to RL in dynamic pricing. They require fewer price changes, adapt to nonparametric demand, and provide explicit uncertainty estimates. The tradeoff between GP-Fin-Model-Based and BO-Fin-Heuristic highlights the tension between theoretical guarantees and practical efficiency.

6 Conclusion

We introduced Bayesian Optimization algorithms for dynamic pricing under infinite and finite inventory regimes. BO-Inf achieves sublinear regret, Lightweight BO-Inf provides computational scalability, GP-Fin-Model-Based integrates GP posteriors with dynamic programming, and BO-Fin-Heuristic offers a lightweight alternative. Across experiments, BO methods outperform classical and deep RL baselines, demonstrating robustness to noise and model misspecification. Beyond dynamic pricing, our work highlights the potential of GP-bandits for a broader class of revenue management problems.

Acknowledgments

This work was supported by the ANRF MATRICS project grant number MTR/2023/000042 from the Anusandhan National Research Foundation (ANRF).

7 References

- [1] O. Besbes and A. Zeevi. Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research*, 57(6):1407–1420, 2009.
- [2] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *ICML*, 2012.

Designing asymptotically optimal policies for continuous-time weakly coupled MDPs*

M. Perbal^{a,b}, B.J. Prabhu^{a,b}, I.M. Verloop^{a,b}

^a LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France,

^b IRT, Université de Toulouse, CNRS, Toulouse INP, Toulouse, France

{matthieu.perbal,balakrishna.prabhu}@laas.fr, ina-maria.verloop@cnrs.fr

ABSTRACT

We study the continuous-time Weakly Coupled Markov Decision Process (WCMDP), a class of decision problems involving multiple interacting Markov processes (or “arms”) subject to shared resource constraints. We present a general framework for policy design using a combination of an underlying Markov process and a sequence of mappings. Our main theoretical result establishes sufficient conditions for asymptotic optimality as the number of arms grows.

We construct both deterministic and randomized policies based on a solution to a linear program (LP). These policies initially assign actions to arms—either proportionally (deterministic) or randomly—based on conditional measures derived from the LP. As this initial allocation may violate feasibility constraints, we introduce a mapping to enforce the resource constraints are satisfied.

1. INTRODUCTION

Weakly coupled MDPs (WCMDPs) are a subclass of MDPs composed of multiple interacting subsystems (also called arms) which are coupled only through common constraints on the joint action space. It arises naturally in a broad range of applications, including resource allocation in communication networks, multi-agent systems, large-scale scheduling, and inventory management. Due to the high dimensionality of some of these applications, practical solutions design low-complexity policies. The majority of previous work in this direction has focused on the subclass of multi-armed restless bandit problems, while a few recent works have sought to generalize the methodology to the weakly-coupled class. All these works, with the notable exception of [Hong et al.(2023), Verloop(2016), Xiong et al.(2022)], assume a discrete-time setting for which they propose deterministic or randomized policies that are asymptotically optimal in the sense that when the number of arms becomes large, the (suitably scaled) performance of these policies tends to that of the optimal one. For the continuous time setting, both [Xiong et al.(2022)] and [Hong et al.(2023)] propose a randomized policy that they show to be asymptotically optimal for a multi-armed restless bandit prob-

*Corresponding author: Matthieu Perbal. Extended abstract of [Perbal et al.(2025)]. All authors were partially supported by the French “Agence Nationale de la Recherche (ANR)” through the project ANR-22-CE25-0013-02.

lem, however, to the best of our knowledge, no such optimality result exists for the continuous-time WCMDPs. In [Xiong et al.(2022)], a randomized policy was proposed and analysed based on the LP-based methodology developed in [Verloop(2016)] for deterministic policies. The LP is formulated in the discrete-time setting leading to believe that they uniformize the continuous-time process and use results for the discrete-time setting. However, care needs to be taken when uniformizing processes induced by randomized policies. While deterministic policies induce a continuous-time Markov chain on the original state space, the dynamics of processes under randomized policies are no longer Markovian but rather semi-Markovian (see [Puterman(1994)] Chap 11). This process can be uniformized as well but only under certain conditions [Beutler and Ross(1987)]. This step does not seem to appear in [Xiong et al.(2022)]. We shall follow another approach: instead of uniformizing the semi-Markov process, we make the dynamics Markovian by including the action in the state description.

2. MODEL DESCRIPTION

We consider a multi-class WCMDP in continuous time with K classes of arms. There are Nx_k class- k arms, where x_k is a strictly positive integer and $N \in \mathbb{N}$ is some scaling parameter. At any moment in time, a class- k arm is in a certain state $i \in \mathcal{I}_k := 1, 2, \dots, I_k$, with $I_k < \infty$. A controller decides at each moment in time which action to give to each arm from a finite set $\mathcal{A}$. When action a is performed on a class- k arm in state i , it makes a transition to state j after an exponentially distributed amount of time with rate $q_k(j|i, a)$. For a given policy π , $X_{k,i}^{\pi,N}(t)$ represents the number of class- k arms in state $i \in \mathcal{I}_k$ and $Z_{k,i,a}^{\pi,N}(t)$ the number of class- k arms that are in state i and receive action a .

The arms are weakly coupled through shared resource constraints that must be satisfied at every point in time. Specifically, let $b_{k,r}(i, a)$ denote the consumption of resource r when a class- k arm is in state (i, a) . A feasible policy π should satisfy the following R resource constraints:

$$\sum_{k=1}^K \sum_{i,a} b_{k,r}(i, a) Z_{k,i,a}^{\pi,N}(t) \leq \beta_r N, \quad \forall r \in \{1, \dots, R\} \quad \forall t > 0.$$

We assume that $b_{k,r}(i, 0) = 0$, for all k, r, i , so that keeping an arm passive ($a = 0$) is always a feasible action.

Let $C_k(i, a)$ be the holding cost per unit of time for having a class- k arm in state i under action a . Our objective is to find a policy π^* that is asymptotically optimal (see [Perbal et al.(2025)] for its definition).

3. MAIN RESULT

DEFINITION 1. A policy π^1 is defined through a mapping G^N on some state space E_N and a Markov Process $\tilde{Z}^N(\cdot)$ living on E_N , if

$$Z^{\pi,N}(\cdot) = G^N(\tilde{Z}^N(\cdot)).$$

The sets E_N are such that for any N it holds that $E_N \subset \{N\tilde{z} : \tilde{z} \in E\}$, for some compact space E .

The motivation for introducing this broader class of policies is to also accommodate randomized policies. For example, consider a randomized policy that assigns virtual actions according to a certain probability distribution, and let the process $\tilde{Z}(\cdot) = (\tilde{Z}_{k,i,a}(\cdot))$ represent the numbers of arms in a given state that are assigned the virtual action a . Note that this virtual process might not represent a feasible allocation. The mapping $G^N(\tilde{Z})$ then decides which arms are indeed activated at their virtual actions, making sure that the so-obtained allocation is indeed feasible.

The following proposition gives sufficient conditions for a policy defined through G^N and $\tilde{Z}^N$ to be asymptotically optimal. Its proof can be found in [Perbal et al.(2025)].

PROPOSITION 1. Assume a policy π^* is defined through $G^N(\cdot)$ and $\tilde{Z}^N(\cdot)$ living on some compact space set E . Assume it holds that:

- The process $\frac{\tilde{Z}^N(\cdot)}{N}$ is unichain.
- There exist a continuous function G on E such that :

$$\sup_{\tilde{z} \in E} \left| \frac{1}{N} G^N(\tilde{z}N) - G(\tilde{z}) \right| \xrightarrow{N \rightarrow \infty} 0.$$

- There exists a process $\tilde{Z}(\cdot)$ on E such that

$$\lim_{N \rightarrow \infty} \mathbb{P} \left(\sup_{t \leq T} \left| \frac{\tilde{Z}^N(t)}{N} - \tilde{Z}(t) \right| \geq \varepsilon \right) = 0, \forall T > 0, \forall \varepsilon > 0.$$

and $\tilde{Z}(\cdot)$ has a global attractor denoted by $\tilde{z}^* \in E$, that is, regardless of the initial condition, it holds that $\lim_{t \rightarrow \infty} \tilde{Z}(t) = \tilde{z}^*$.

- For any policy π , we have the following asymptotic lower bound,

$$C(G(\tilde{z}^*)) \leq \lim_{N \rightarrow \infty} \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T C\left(\frac{Z^{\pi,N}(t)}{N}\right) dt, \quad a.s..$$

Then, the policy π^* is asymptotically optimal.

4. LP-BASED POLICIES

A class of asymptotically optimal policies can be obtained by solving a linear program² (LP) associated with the model, where the usual constraint is relaxed to hold only in the time-average.

Let $z^* = (z_{k,i,a}^*)_{k,i,a}$ be a solution of this LP. Under an LP-based policy π , if the process $Z^{\pi,N}(\cdot)$ admits a unique invariant measure $\mu^N = (\mu_{k,i,a}^N)_{k,i,a}$, the policy is designed so that as N grows, the long-run rescaled number of arms in each tuple (k, i, a) converges to $z_{k,i,a}^*$.

¹We write π instead of π^N to keep the notation light, but it should be kept in mind that the policy does depend on N .

²See [Verloop(2016)] for a complete definition of the LP.

We propose two types of LP-based policies: randomized and deterministic. As mentioned in the introduction, under any randomized policy, the process $Z^{\pi,N}(\cdot)$ is semi-Markov. However, we construct a randomized policy that induces a Markov process $\tilde{Z}^{\pi,N}(\cdot)$ on another space, which can then be mapped via a suitable function G^N to define $Z^{\pi,N}(\cdot)$ as in Definition 1.

For simplicity, we illustrate our results with deterministic policies (for randomized policies, we refer to [Perbal et al.(2025)]). Define

$$\chi_{k,i}^*(a) := \frac{z_{k,i,a}^*}{\sum_a z_{k,i,a}^*}, \quad \forall a \in \mathcal{A}.$$

The function $G^N(\cdot)$, is defined such that a proportion $\chi_{k,i}^*(a)$ of the class- k arms in state i receives action a , if feasible. That is, with $(x_{k,i})_{(k,i)}$ the proportion of arms in state (k, i) , we set $G_{k,i,a}^N(Nx) =$

$$\begin{cases} \left[\min \left(Nx_{k,i}, \min_{r \leq R} d_{N,r}(k, i, x) \right) \chi_{k,i}^*(a) \right], & a \neq 0; \\ Nx_{k,i} - \sum_{a=1}^A G_{k,i,a}^N(Nx), & a = 0, \end{cases}$$

where $d_{N,r}(k, i, x) =$

$$\frac{\left(N\beta_r - \sum_{(k',i') \in A_{k,i}} \sum_{a'} b_{k',r}(i', a') G_{k',i',a'}^N(Nx) \right)_+}{\sum_{a=0}^A b_{k,r}(i, a) \chi_{k,i}^*(a)}.$$

Finally, under the unichain and global attractor assumptions, all the conditions of Proposition 1 hold for LP-based policies, ensuring asymptotic optimality. We acknowledge that the global attractor assumption is strong and should be relaxed as has been done in the discrete-time setting [Goldsztajn and Avrachenkov(2024), Zhang et al.(2025)]. It will be interesting to see whether their results can be extended to continuous time.

5. REFERENCES

- F.J. Beutler and K.W. Ross. 1987. Uniformization for semi-Markov decision processes under stationary policies. *Journal of Applied Prob.* 24, 3 (1987), 644–656.
- D. Goldsztajn and K. Avrachenkov. 2024. Asymptotically Optimal Policies for Weakly Coupled Markov Decision Processes. arXiv:2406.04751.
- Y. Hong, Q. Xie, Y. Chen, and W. Wang. 2023. Restless bandits with average reward: Breaking the uniform global attractor assumption. In *Proc. of NeurIPS*.
- M. Perbal, B.J. Prabhu, and I.M. Verloop. 2025. Designing asymptotically optimal policies for continuous-time weakly coupled MDPs. To appear in *Performance* 2025.
- M.L. Puterman. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley.
- I.M. Verloop. 2016. Asymptotic optimal control of multi-class restless bandits. *Annals of Applied Prob.* 26 (4), 1947-1995 (2016).
- G. Xiong, S. Wang, and J. Li. 2022. Learning infinite-horizon average-reward restless multi-action bandits via index awareness. In *Proc. of NeurIPS*.
- X. Zhang, Y. Hong, and W. Wang. 2025. ID policy (with reassignment) is asymptotically optimal for heterogeneous weakly-coupled MDPs. arXiv:2502.06072.

Strategic Pricing and Ranking in Recommendation Systems with Seller Competition

Tushar Shankar Walunj
IEOR, IIT Bombay
tusharwalunj@iitb.ac.in

Jayakrishnan Nair
EE, IIT Bombay
jayakrishnan.nair@ee.iitb.ac.in

Veeraruna Kavitha
IEOR, IIT Bombay
vkavitha@iitb.ac.in

Priyank Agarwal
IEOR, IIT Bombay

22n0453@iitb.ac.in

ABSTRACT

We study recommendation systems where sellers compete for visibility by offering commissions to a platform that curates a personalized ranked menu and sets item prices. Customers interact with the menu under a cascade click model (e.g., as in [1, 3]), scanning the items sequentially from top to bottom. The seller–platform interaction is formulated as a Stackelberg game with sellers as leaders and the platform as a follower. We consider two variants: in the *Order and Pricing (OP)* game, the platform chooses both ranking and prices, while in the *Order at Set-Prices (OS)* game, sellers pre-fix prices and the platform decides only the ranking.

The platform’s problem is modeled as an MDP and solved via backward induction. In the low exploration regime (small γ), we derive closed-form optimizers using greedy index policies, which substantially reduce computational complexity. In this regime, the OP and OS games coincide, allowing us to analyse a unified seller game. With symmetric sellers (in their strength), pure Nash equilibria exist, and sellers operate at marginal cost (similar to the Bertrand duopoly equilibrium). With asymmetric sellers, due to discontinuities from rank jumps, pure NE does not exist, but the game dynamics exhibit oscillatory behaviour. We formalize this with the μ -connected equilibrium cycle (μ -EC), a set-valued equilibrium concept that captures such oscillatory behaviour. Numerical experiments in the full paper demonstrate both the approximation accuracy and the cyclic best-response dynamics [5][Figures 2-3].

1. INTRODUCTION

Platforms such as Amazon and Walmart rely on recommendation systems that display ranked menus of items to customers to increase sales. The position of a seller in these menus directly affects visibility and revenue, so sellers compete for better placement by offering commissions to the platform. This leads to a two-sided tension: the platform seeks to maximize revenue through strategic ranking and pricing, while sellers adjust their commission offers to influence their placement and improve their chances of appearing in a higher position.

In the literature, ranking, pricing, and assortment prob-

lems are usually studied separately [3, 1], whereas we study them together, making seller strategies an explicit part of the system. This integrated approach uncovers more complex dynamics: platform policies not only affect customer choices but also reshape the competitive environment among sellers. Our key aims are (i) to provide tractable and near-optimal platform policies, (ii) to characterize and study the equilibrium behaviour of the induced seller game, and (iii) to propose an equilibrium concept suited for games with discontinuous payoffs when classical pure NE does not exist.

Our analysis shows how this interaction leads to equilibrium behaviour that differs from the well-known classical outcomes because of discontinuities caused by rank jumps. To capture this, we propose the μ -EC, a set-valued solution concept: we extend the recently introduced concept *equilibrium cycle* ([6]) to include stability against measure-zero violations and avoid topological difficulties, thereby proposing μ -EC. Similar cyclical behaviour has also been observed empirically in auctions and advertising systems (e.g., [7]), where sellers repeatedly adjust bids without convergence. It also resembles Edgeworth cycles [2, 4] in markets like airlines and gasoline, where firms engage in recurring price wars. The μ -EC framework formalizes these oscillatory yet stable outcomes within recommendation systems.

2. PROBLEM FORMULATION

Each seller a offers a product with commission $\delta_a \in [0, 1]$. The platform constructs a menu of size M from N sellers. Customers interact with the menu sequentially from top to bottom: while exploring item a , they purchase with probability $\beta_a(p_a)$, continue browsing with probability $(1 - \beta_a(p_a))\gamma$, or exit with the remaining probability. The platform’s expected revenue depends jointly on the commissions $\{\delta_a\}$, the prices $\{p_a\}$, and the ordering of the items (or the menu).

The platform’s optimization problem can be formulated as an MDP with states given by the remaining items and actions corresponding to the choice of the next item (and price, in OP). Solving the MDP exactly is computationally intensive, as the state space grows exponentially with N . However, in the low-exploration regime (small γ), using backward induction, we derive a simple structure: the optimal ranking reduces to a greedy index rule, where each item is assigned a per-item score combining commission and price sensitivity. Ordering by this index yields closed-form policies that are computationally efficient. Moreover, in this regime, the OP and OS games coincide, so the seller game can be studied

within a unified game.

3. MAIN THEORETICAL INSIGHTS

Our analysis yields the following main insights:

- **Platform optimization:** For small γ , the platform's MDP simplifies to a greedy index policy. This reduces computational complexity from exponential (brute-force DP) to nearly linear in menu size, while maintaining high accuracy.
- **Unified seller game:** In the small γ regime, the OP and OS games coincide. Hence, a single unified seller game suffices to capture the seller competition in an important practical regime (where customers typically explore few items).
- **Equilibrium structure:**
 1. With symmetric sellers, pure-strategy NE exist, where sellers operate at marginal cost (analogous to equilibrium in the Bertrand competition).
 2. With asymmetric sellers, due to rank-induced discontinuities, pure NE does not exist. We propose the μ -connected equilibrium cycle, a set-valued solution concept defined by stability against external deviations, absence of external chains, internal instability, and minimality, while permitting measure-zero exceptions.

The μ -EC explains persistent oscillations observed in best-response dynamics, formalizing the cyclical adjustments in commissions that arise when sellers differ in strength.

4. NUMERICAL RESULTS

Our simulations highlight both the validity of the small- γ approximation and the game dynamics between sellers.

Approximation accuracy: Seller utilities under the exact method using dynamic programming equations for the OP game were compared with the index policy in the small γ regime. For $\gamma = 0.2$, the curves align almost perfectly; even at $\gamma = 0.4$, the deviation is only about 2.5%, and at $\gamma = 0.7$ it remains below 6%. These results confirm that the approximation provides accurate and computationally efficient predictions across a broad range of exploration rates.

Cyclic best-response dynamics: In asymmetric settings, simulations show that commissions do not converge to a fixed point but instead oscillate persistently. Figure 1 illustrates a representative instance: small commission changes trigger rank jumps, causing opponent sellers to readjust. Instead of settling at a fixed point, strategies of sellers remain confined to a connected region, exactly as described by the μ -EC.

Implications: Firstly, the small- γ index rules provide simple, computationally efficient heuristics for menu construction that closely approximate the DP solution. Next, the threat role of the $(M + 1)$ -th seller is important: a seller just outside the displayed menu acts as a constraint on those inside, forcing top sellers to offer higher commissions. Finally, the new solution concept generalizes classical equilibrium notions among pure strategies and explains seller behaviour that cycles through certain stable regions of the strategy space.

5. CONCLUDING REMARKS

We develop a unified framework combining the cascade click model, the platform optimization problem using MDPs, and seller competition through commissions. A key contribution is formulating the platform optimization problem as an MDP, enabling recursive computation of optimal pricing

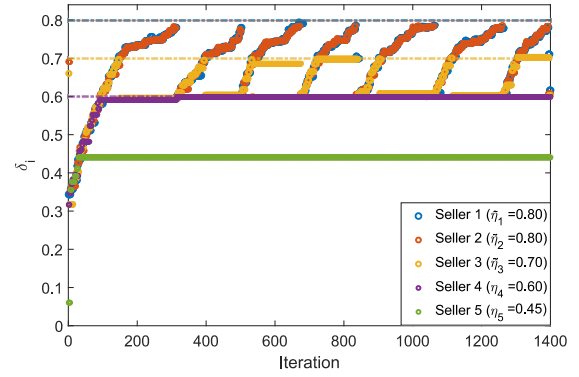


Figure 1: ε -best-response dynamics in an asymmetric setting. Commissions oscillate persistently rather than converging, consistent with the μ -EC framework. Horizontal dashed lines — theoretical thresholds.

and ranking. The MDP admits tractable solutions, and the small- γ approximation yields closed-form indices that unify the OP and OS games. With symmetric sellers, pure-strategy equilibria exist; with asymmetric sellers, the μ -EC captures the resulting oscillatory outcomes.

We extend the recently introduced solution concept *equilibrium cycle* [6] to include stability against measure-zero violations and avoid some topological difficulties, and thereby propose μ -EC. This novel solution concept offers a solution for discontinuous games where classical pure NE does not exist. Overall, our work provides a unified and tractable framework to analyse recommendation systems with strategic sellers, bridging optimization, game theory, and customer behaviour modelling. It opens up directions for further research, including the design of fair or regulated platform policies, dynamic learning in response to customer feedback.

6. REFERENCES

- [1] R. Dhangra, H. S. Verma, A. Reiffers-Masson, and V. Kavitha. Multi-page menu recommendation in cascade model with externalities. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 6614–6619. IEEE, 2021.
- [2] D. Escobari, N. G. Rupp, and J. Meskey. An analysis of dynamic price discrimination in airlines. *Southern Economic Journal*, 85(3):639–662, 2019.
- [3] D. Kempe and M. Mahdian. A cascade model for externalities in sponsored search. In *International Workshop on Internet and Network Economics*, pages 585–596. Springer, 2008.
- [4] M. D. Noel. Edgeworth price cycles. In *The New Palgrave Dictionary of Economics*. Springer, 2011.
- [5] T. S. Walunj, V. Kavitha, J. Nair, and P. Agarwal. Strategic pricing and ranking in recommendation systems with seller competition, 2025.
- [6] T. S. Walunj, S. Singhal, V. Kavitha, and J. Nair. Equilibrium cycle: A “dynamic” equilibrium. *arXiv preprint arXiv:2411.08471*, 2024.
- [7] X. Zhang and J. Feng. Cyclical bid adjustments in search-engine advertising. *Management Science*, 57(9):1703–1719, 2011.

Near-Optimal PCM Wear Leveling Under Adversarial Attacks

Tomer Lange Joseph (Seffi) Naor Gala Yadgar
Computer Science Department, Technion, Haifa, Israel
{tlange, naor, gala}@cs.technion.ac.il

ABSTRACT

Phase-change memory (PCM) is a promising memory technology known for its speed, high density, and durability. However, despite these advantages, a key challenge limiting PCM's widespread adoption is its finite endurance: each cell supports only a limited number of writes before becoming unreliable, and the failure of a single cell can compromise the entire device. This vulnerability makes PCM particularly susceptible to adversarial attacks that induce excessive writes to accelerate device failure.

Wear-leveling mechanisms, which aim to distribute write operations evenly across PCM cells, are among the most widely used techniques to mitigate the limited endurance of PCM. However, while PCM wear leveling has been extensively studied in the systems and architecture communities, this problem remains largely unexplored from a theoretical perspective. We bridge this gap by presenting a novel algorithm that leverages hardware feedback to optimize PCM performance. We prove that our algorithm achieves near-optimal worst-case guarantees and outperforms state-of-the-art practical solutions both theoretically and empirically, providing an efficient approach to prolonging PCM lifespan.

1. BACKGROUND

PCM stores data in physical *cells*. Each cell holds a single logical *line*, which serves as the basic unit of read and write operations. To write new data to a cell, the cell must first be *erased*, which increases its *wear*. Once the wear of any cell surpasses a fixed threshold, referred to as the *erase limit*, the entire device becomes unusable.

As wear might accumulate unevenly across cells, wear leveling strives to balance wear by *relocating* frequently written lines to cells with low wear, necessitating an efficient mechanism for tracking the physical location of each line. This task is handled by the *PCM controller*, which has a non-PCM *auxiliary memory* dedicated to storing mapping information and supporting wear leveling. However, this memory has a *severely limited capacity*, typically several orders of magnitude smaller than the PCM device itself. Therefore, the challenge lies in designing a mapping mechanism that not only enables efficient wear leveling, but also has low memory overhead.

Practical wear-leveling strategies use simplified mapping schemes that reduce memory overhead at the cost of reloca-

tion flexibility. For example, the well-known *Start-Gap* algorithm [3] periodically shifts all lines while preserving their original relative ordering, enabling to determine line locations by storing only the initial ordering and a single global offset. However, this rigid relocation pattern limits effectiveness under unbalanced workloads, where finer-grained adjustments are needed to balance wear. To address this, Seznec [4] introduced a more flexible approach known as *subset-based mapping*, which utilizes auxiliary memory to partition lines into subsets and maintain a separate offset per subset. However, while this allows independent rotations and more localized adjustments, current implementations manage subsets inefficiently, leading to suboptimal wear leveling and highlighting the need for further improvements.

The only theoretical study of PCM wear-leveling algorithms is by Devanny et al. [1], who introduced the *cache-augmented Start-Gap* algorithm. Unlike Seznec's approach, this algorithm addresses the limitation of Start-Gap by leveraging auxiliary memory to cache frequently written lines. However, its effectiveness relies on a large cache, making it impractical in systems with limited auxiliary memory.

Hardware feedback. As repeated writes degrade PCM cells, write counts are often used as a proxy for wear. However, the bit error rate (BER) provides a more direct signal of cell health, capturing cumulative damage beyond write counts, such as variability or retention loss. Even though BER is routinely monitored during read/write verification, most existing wear-leveling algorithms remain *wear oblivious*, relocating lines based only on logical access patterns. This might result in moving frequently written data into already degraded cells, limiting wear-leveling effectiveness and motivating the design of algorithms that explicitly rely on hardware feedback to inform relocation decisions.

2. CONTRIBUTION

To bridge the gap between practical wear-leveling mechanisms and theoretical analysis, we introduce a rigorous model that captures the constraints of real-world PCM systems, including limited auxiliary memory. Specifically, we consider a PCM device consisting of N physical cells, each storing a logical line. Each cell can endure at most B erasures before becoming permanently unusable. At each time step t , a line is updated. As a result, the cell storing that line is erased and its wear increases by one. The device fails when any cell reaches the erase limit B . To extend the device's lifespan, a wear-leveling algorithm may relocate lines across cells. However, before a line can be placed in a new

cell, that cell must be erased, further contributing to its wear.

We assume access to a non-PCM auxiliary memory with unlimited endurance. This memory consists of M variables, each storing $\log N$ bits. It must store a logical-to-physical mapping that supports retrieving the location of any logical line in $O(1)$ time for efficient lookups. However, this memory can also be used for caching or for storing additional data to guide relocation decisions. In particular, it may record wear information, which we assume can be accurately estimated based on hardware feedback.

In the *online PCM utilization problem*, we are given an arbitrary sequence of write requests σ , the device parameters N and B , and the size M of the auxiliary memory. The objective is to maximize the number of requests served before any cell reaches the erase limit B . Similarly to prior studies [1, 2], we evaluate algorithms by comparing performance to the best achievable outcome. In particular, note that any algorithm, even an offline algorithm, can write at most $T^* = N \cdot B$ times to the PCM device. Therefore, we define the normalized *utilization* measure as $U = T/T^*$, where T denotes the number of write requests successfully served before any cell reaches the erase limit.

Our results. The main contribution of this paper is *Min-Max*, the first wear-leveling algorithm that exploits hardware feedback to improve PCM endurance. Building on Seznec’s subset-based mapping approach [4], Min-Max groups lines into logical subsets, and partitions cells into physical subsets called *segments*. Each subset is always stored within a single segment. Wear leveling is achieved by occasionally swapping subsets while preserving their cyclic ordering.

To further balance wear, the auxiliary memory is used not only for mapping, but also for recording lightweight wear information (inferred from hardware feedback) that guides relocation decisions. Inspired by the approach of [2], this information is used to simultaneously minimize differences in wear between cells (1) from different segments, and (2) within the same segment. We analyze the worst-case utilization of Min-Max as a function of the auxiliary memory size M , and obtain the following result.

THEOREM 1. *The Min-Max algorithm guarantees utilization of $1 - O\left(\sqrt{\frac{N \ln N}{MB}}\right)$ on any input w.h.p.*

Comparison to previous work. The analysis of [1] implies that cache-augmented Start-Gap achieves utilization of $1 - O\left(\sqrt{\frac{N^2 \ln N}{M^2 B}}\right)$ on any input with high probability. Similarly to Min-Max, this bound improves as the auxiliary memory size increases. However, compared to Min-Max, this algorithm requires significantly larger memory. In particular, while cache-augmented Start-Gap needs $M \gg N \cdot \sqrt{\frac{\ln N}{B}}$ to achieve near-optimal utilization of $1 - o(1)$ on any input, Min-Max achieves the same guarantee already for $M \gg N \cdot \frac{\ln N}{B}$. This difference is substantial for practical PCM parameters, where cache-augmented Start-Gap could require auxiliary memory that is nearly three orders of magnitude larger than that needed by Min-Max to achieve similar performance.

To compare Min-Max against state-of-the-art practical wear-leveling strategies, we evaluate Seznec’s algorithm [4], which employs a structured randomized approach and is

known for achieving high utilization even under unbalanced workloads. While being commonly used as a benchmark in experimental studies, the worst-case performance of Seznec’s algorithm has never been theoretically analyzed. We bridge this gap by establishing the following bound.

THEOREM 2. *If Seznec’s algorithm guarantees utilization of $1 - \varepsilon$ w.h.p. on any input, then $\varepsilon = \Omega\left(\sqrt[3]{\frac{N \ln N}{MB}}\right)$.*

Notice that as auxiliary memory is typically much smaller than main PCM storage, our bound on Seznec’s algorithm may improve upon the results in [1]. However, Min-Max still outperforms Seznec’s algorithm: while the utilization of both algorithms depends on the ratio $\frac{N \ln N}{MB}$, Min-Max approaches optimal utilization asymptotically faster. This improvement is especially relevant given the many follow-up works proposing practical enhancements to Seznec’s method, where Min-Max offers an alternative design with stronger theoretical foundation.

Experimental evaluation. In addition to our theoretical analysis, we conduct an extensive experimental evaluation to assess the practical effectiveness of Min-Max. To account for real-world constraints, including inaccuracies in hardware feedback, we also introduce a practical variant that categorizes wear levels into a limited number of distinguishable ranges. Our evaluation benchmarks Min-Max and its variant against Seznec’s algorithm and cache-augmented Start-Gap across diverse unbalanced workloads while varying key parameters, including the erase limit, device capacity, and auxiliary memory size.

The results demonstrate that both versions of Min-Max exhibit high utilization, even under adversarial workloads. Moreover, Min-Max requires significantly less auxiliary memory than previous solutions and extends the device’s lifespan by up to 66% under practical PCM parameters. These findings underscore the potential of Min-Max for real-world deployment, providing a robust and efficient solution for PCM wear leveling with strong theoretical guarantees.

3. REFERENCES

- [1] W. E. Devanny, M. T. Goodrich, and S. Irani. A competitive analysis for the start-gap algorithm for online memory wear leveling. *Information Processing Letters*, 166:106042, 2021.
- [2] T. Lange, J. Naor, and G. Yadgar. SSD wear leveling with optimal guarantees. In *2024 Symposium on Simplicity in Algorithms (SOSA)*, pages 306–320. SIAM, 2024.
- [3] M. K. Qureshi, J. Karidis, M. Franceschini, V. Srinivasan, L. Lastras, and B. Abali. Enhancing lifetime and security of PCM-based main memory with start-gap wear leveling. In *Proceedings of the 42nd annual IEEE/ACM international symposium on microarchitecture*, pages 14–23, 2009.
- [4] A. Seznec. A phase change memory as a secure main memory. *IEEE Computer Architecture Letters*, 9(1):5–8, 2010.

CommonSense: Efficient Set Intersection (SetX) Protocol Based on Compressed Sensing

Jingfan Meng
Georgia Institute of
Technology
Atlanta, GA, USA
jeffmeng@live.com

Tianji Yang
Georgia Institute of
Technology
Atlanta, GA, USA
tyang425@gatech.edu

Jun Xu
Georgia Institute of
Technology
Atlanta, GA, USA
jx9@gatech.edu

ABSTRACT

In the set reconciliation (SetR) problem, two parties Alice and Bob, holding sets A and B respectively, communicate to learn the symmetric difference $A \Delta B$. In this work, we study a related but under-explored problem: set intersection (SetX) [7], where both parties learn $A \cap B$ instead. Existing solutions typically reuse SetR protocols due to the absence of dedicated SetX protocols and the misconception that SetR and SetX have comparable costs. Observing that SetX is fundamentally cheaper than SetR, we have developed a multi-round SetX protocol that outperforms the information-theoretic lower bound of SetR problem. In our SetX protocol, Alice sends Bob a compressed sensing (CS) sketch of A to help Bob identify his unique elements (those in $B \setminus A$). This solves the SetX problem, if $A \subseteq B$. Otherwise, Bob sends a CS sketch of the residue (a set of elements he cannot decode) back to Alice for her to decode her unique elements (those in $A \setminus B$). As such, Alice and Bob communicate back and forth until both parties agrees on $A \cap B$. On real world datasets, experiments show that our SetX protocol reduces the communication cost by 8 to 10 times compared to the IBLT-based SetR protocol.

1. INTRODUCTION

Set reconciliation (SetR) – reconciling two large sets A and B held by hosts Alice and Bob respectively – has been widely studied for networking, systems, and database applications. Well-established prior SetR solutions include PinSketch, D.Digest, and Graphene [6, 3, 4]. In this paper, we focus on a related but under-explored set intersection problem (SetX), where Alice and Bob want to know the intersection $|A \cap B|$ instead of the union. Although SetX appears in practice, most prior work solves it by running a full SetR protocol and discarding parts of the output. For example, Graphene uses a SetR subroutine even when only the intersection is needed [7]. Under the standard sparse-difference assumption $|A \Delta B| \ll |A| \leq |B| \ll 2^u = |U|$ (universe size), the information-theoretic lower bound for SetX is substantially lower than that for SetR [6], suggesting that a substantially lower communication cost may be achievable in practice. Building on this insight, we introduce CommonSense, a compressed sensing (CS) based protocol for minimizing the communication cost. Its core idea, in uni-directional case of $A \subseteq B$ is simple: Alice sends a compact CS sketch $M1_A$ of her set indicator vector; Bob subtracts his CS sketch $M1_B$

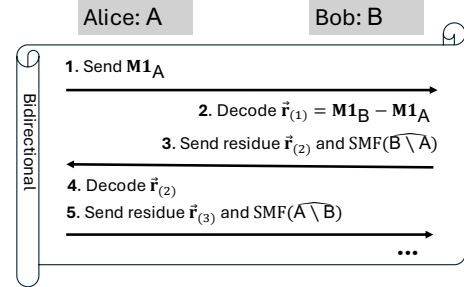


Figure 1: Algorithm Overview of SetX

from $M1_A$ and recovers Alice’s missing elements from the difference. We have shown that Alice’s sketch $M1_A$ only needs to be of size $O(|B \setminus A|)$, which is much smaller than encoding A itself in a naive solution. Entropy compression (BCH, ANS) can further reduce the size of $M1_A$.

Unlike a prior CS work that used a real-valued sparse matrix (which inflates the transmission cost) to be “compliant” with RIP (restricted isometry property) in the L_2 space, our design encodes nonzero measurements as small integers and proves recovery guarantees using RIP-1 (L_1 space) theory instead. We also implement and release the data structures and code [5] used to accelerate SSMP/MP, the state-of-the-art CS decoding algorithm for RIP-1.

Our contributions can be summarized as follows.

- We show a large gap between SetX’s and SetR’s information theoretic lower bounds, debunking a common belief [6].
- We present CommonSense, the first exact, efficient SetX protocol (for both the unidirectional and the bidirectional cases) that beats the informational-theoretic lower bound of SetR, by applying CS to this problem [7].
- We identify and analyze a binary sparse CS matrix that is both application-friendly (fast updates, small measurements) and theory-compliant (RIP-1), and validate our approach with experiments [1].

2. COMMONSENSE PROTOCOL

The protocol, in general, runs multiple communication rounds (Figure 1). We first describe a simple special case in Section 2.1 and then extend it to the general case in Section 2.2.

2.1 Unidirectional Protocol

This simplified protocol is for the special case of $A \subseteq B$, which only needs one message from Alice to Bob.

Protocol. Alice sends $\mathbf{M1}_A$, the CS sketch of (the indicator vector of) her set A to Bob, as shown in the first step of Figure 1. Here, $\mathbf{M}$ is the CS matrix that is generated (once and then fixed) at the beginning, and $\mathbf{M}$ is known to both Alice and Bob. As explained earlier, the CS sketch $\mathbf{M1}_A$ is small in size and can be further compressed using entropy encoding.

Decoding Algorithm. After Bob receives $\mathbf{M1}_A$, he tries to recover elements in $B \setminus A$ from the CS sketch difference $\mathbf{M1}_B - \mathbf{M1}_A$ using an MP (matching pursuit) algorithm. In theory, we need to use an MP algorithm for the L_1 space such as SSMP (sequential sparse matching pursuit), since our CS matrix $\mathbf{M}$ satisfies RIP-1 instead. In practice, however, the vanilla MP (for L_2), which is faster than SSMP, works for this use case empirically. We denote Bob’s estimation of (the ground truth) $B \setminus A$ as $\hat{B} \setminus A$.

2.2 Bidirectional Protocol

In this general case, the protocol performs multiple rounds of communications as shown in Figure 1. The decoding algorithm, after each communication step, is the same as that described in 2.1 for the unidirectional case. In the general case, it needs to solve an additional problem called common hallucination.

Common Hallucination. It can be shown that, when only the CS sketch of the “residue” is sent back and forth between Alice and Bob and when $(\hat{B} \setminus A) \cap (A \setminus \hat{B}) \neq \emptyset$, an element in $A \cap B$ can be erroneously considered by both Alice and Bob to be one of their unique elements. For example, when Bob “hallucinates” that he “sees” the element in the CS sketch difference (residue) due to noise in the residue, he subtracts it from the residue (and considers it one of his unique elements); this false subtraction leaves its “negative” in the residue that causes Alice to do the same. We call this problem *common hallucination*. It can be solved using SMFs (set membership filters) as shown in the paper.

3. EVALUATION

We compare CommonSense with Graphene and IBLT, the state of art SetR protocols, using the total amount of communication as the performance metric. In evaluating bidirectional SetX (in Figure 3), $|A \setminus B|$ is fixed to 10,000. The performance results are shown in Figures 2 and 3.

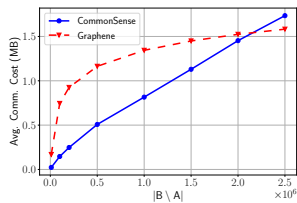


Figure 2: Unidirectional CommonSense results as $|B \setminus A|$ increases from 10,000 to 2,500,000.

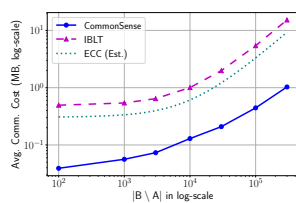


Figure 3: Bidirectional SetX results as $|B \setminus A|$ increases from 100 to 300,000. Both axes are log-scale.

We also measure the real-world performance of CommonSense by computing the SetX problem on different Ethereum world states [2]. The dataset characteristics are summarized in Table 4. Evaluations show that our algorithm reduces communication costs by 8 to 10 times compared to IBLT (Table 5).

S =	Block ID	S	S \setminus A	A \setminus S
A	22399992	292,222,740	—	—
B	22392874	291,992,904	340,292	570,128
C	22020359	280,973,256	5,636,348	16,885,832

Table 4: Summary of the Ethereum dataset.

	Comm. Cost	IBLT	CommonSense
SetX(A, B)		48.59 MB	5.866 MB
SetX(A, C)		1.192 GB	118.1 MB

Table 5: Communication costs for SetX, averaged from 5 runs.

4. ACKNOWLEDGMENT

We thank our shepherd Prof. Hans van den Berg, and anonymous reviewers, for their insightful comments that have helped to improve the quality and the presentation of this work. This work was supported in part by the National Science Foundation under Grant No. CNS-2007006 and by a seed gift from Dolby Laboratories.

5. REFERENCES

- [1] R. Berinde, A. C. Gilbert, P. Indyk, H. Karloff, and M. J. Strauss. Combining geometry and combinatorics: A unified approach to sparse signal recovery. In *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, pages 798–805, 2008.
- [2] V. Buterin et al. Ethereum white paper: a next generation smart contract & decentralized application platform. *First version*, 53(99):1, 2014.
- [3] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1):97–139, 2008.
- [4] D. Eppstein, M. T. Goodrich, F. Uyeda, and G. Varghese. What’s the difference? efficient set reconciliation without prior context. In *Proceedings of the ACM Special Interest Group on Data Communication*, pages 218–229, 2011.
- [5] J. Meng, T. Yang, and J. Xu. Source codes and datasets of CommonSense.
- [6] Y. Minsky, A. Trachtenberg, and R. Zippel. Set reconciliation with nearly optimal communication complexity. *IEEE Transactions on Information Theory*, 49(9):2213–2218, Sept. 2003.
- [7] A. P. Ozisik, G. Andresen, B. N. Levine, D. Tapp, G. Bissias, and S. Katkuri. Graphene: Efficient interactive set reconciliation applied to blockchain propagation. In *Proceedings of the ACM Special Interest Group on Data Communication*, pages 303–317, 2019.

Content and Access Networks Synergies: Tradeoffs in Public and Private Investments by Content Providers

Pranay Agarwal
BITS Pilani Hyderabad Campus
agarwalpranay09@gmail.com

D. Manjunath
IIT Bombay
dmanju@ee.iitb.ac.in

ABSTRACT

The ever-increasing demand for better Internet experience has necessitated an upgrade in the capacity of access networks. Internet service providers (ISPs) have been demanding that the content providers (CPs) share the cost of upgrading network infrastructure. A CP can make a *public investment* towards improving the ISP infrastructure to boost its revenue. A CP can also make a *private investment* in its infrastructure and boost its profits. In this work, we consider multiple CPs and a neutral ISP. We consider different models of interaction among CPs—centralized allocation, non-cooperative game, and bargaining game—and study the trade-off between a public and private investment of a CP for each model. Via numerical results, we evaluate the impact of different incentive structures on the utility of the CPs. We see that the bargaining game can result in higher public investment than the non-cooperative and centralized models. However, this benefit gets reduced if the CPs are incentivized to invest in private infrastructure.

1. INTRODUCTION

Internet service is provided to users by two complementary networks—Internet service provider networks (ISPs) and content provider (CP) networks. Despite their complementarity, the share of ISPs in the Internet chain pie has been steadily decreasing. On the other hand, the technological advances in content creation and rendering have made steady advances. The burden of delivering these advances is with the ISPs, which in turn have to make heavy investments, while the CPs are expected to reap a larger share of the economic surplus that will result from this investment. Because of this, ISPs have been demanding CPs to share the investments in the access networks. Indeed, some CPs have invested to improve public infrastructure. However, CPs can also choose to invest in improving its own infrastructure, for example, building a content distribution network with significant caching capabilities or peering with ISPs [3].

A closely related work is [2], where the CPs make a voluntary payment to a neutral ISP which the ISP can use to upgrade its infrastructure. The total contribution of the CPs is analyzed under different types of games. Similarly, [1] proposes a profit-sharing model between the ISP and CPs. The existing works have not considered the impact of CPs' private investments, and this is the motivation for this work.

In this work, we consider multiple CPs and a neutral ISP. Each CP can make a *public investment* towards improving

ISP infrastructure. Since the ISP is neutral, the upgraded infrastructure benefits the users of all CPs. A CP also make a *private investment* to improve its infrastructure benefiting only its users. In this scenario, our objective is to study the effects of the incentive structures and the interaction models among the CPs on the (i) investments in the access networks, and on the (ii) social utility.

2. SYSTEM MODEL

In this work, we consider that N CPs and one neutral ISP together provide Internet services to a fixed user population. There is a basic infrastructure that is present at the ISP and the CPs, and the interest here is in new investments that need to be made to improve the quality of service for the users. Let q_n and p_n denote the public and private investment of the CP n , respectively, for all $n \in \mathcal{N}$, where $\mathcal{N} := \{1, 2, \dots, N\}$. Let $Q := \sum_{n=1}^N q_n$ be the total public investment and the $P := \sum_{n=1}^N p_n$ be the total private investment. Since the ISP is neutral, Q and p_n determine the profit of a CP n . Denoting the revenue per unit traffic for CP n to be r_n , the surplus, or the utility function for CP n , denoted by U_n , is

$$U_n(q_n, p_n) = r_n a_n \log(1 + Q + b_n \sqrt{p_n}) - (p_n + q_n). \quad (1)$$

3. INTERACTION MODELS

We first consider **Centralized Allocation**, wherein both the Q and the p_n are determined centrally as a solution to the following optimization problem.

$$\begin{aligned} \max_{Q, \{p_n\}} \quad & \sum_{n=1}^N (U_n(p_n, q_n) - p_n) - Q, \\ \text{s. t. } \quad & p_n \geq 0 \quad \forall n \in \mathcal{N} \text{ and } Q \geq 0. \end{aligned} \quad (2)$$

The optimum value of p_n for a given value of Q that solves (2), denoted by $p_n^*(Q)$, is given by

$$p_n^*(Q) = \left(\frac{\sqrt{(1+Q)^2 + 2b_n^2 r_n a_n} - (1+Q)}{2b_n} \right)^2. \quad (3)$$

Similarly, the optimum value of Q that solves (2), denoted by Q_C^* , is obtained by solving

$$\sum_{n=1}^N \frac{\sqrt{(1+Q)^2 + 2b_n^2 r_n a_n}}{b_n^2} = (1+Q) \left(\sum_{n=1}^N \frac{1}{b_n^2} \right) + 1. \quad (4)$$

Copyright is held by author/owner(s).

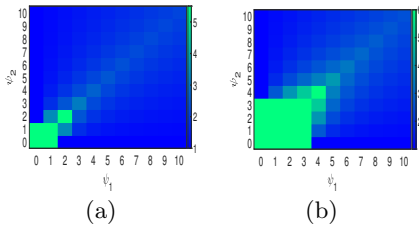


Figure 1: Variation of η against different values of $\psi_n (= r_n a_n)$ for $N = 2$ for $\mathbf{b} = [1, 1]$ in (a) and $\mathbf{b} = [2, 2]$ in (b).

The public-private trade-off under *centralized allocation*, denoted by γ_C , is obtained as

$$\gamma_C = \frac{Q_C^*}{\sum_{n \in \mathcal{N}} p_n^*(Q_C^*)} = \frac{2Q_C^*}{\sum_{n=1}^N r_n a_n - (1 + Q_C^*)}. \quad (5)$$

We now consider **Non-Cooperative Game** wherein the CPs choose the p_n and q_n strategically, i.e., they do not coordinate and choose these quantities to selfishly maximize their utility. Using (3), we can rewrite U_n as

$$U_n = r_n a_n f_n(Q) - p_n^*(Q) - q_n \quad (6)$$

where $f_n(Q) = \log \left(\frac{\sqrt{(1+Q)^2 + 2b_n^2 r_n a_n} + (1+Q)}{2} \right)$. Let q_n^* denote the equilibrium public investment maximizing the utility of CP n . Then, a *pure strategy Nash equilibrium* exists if and only if

$$U_n(q_n^*, Q_{-n}^*) \geq U_n(q_n, Q_{-n}^*) \quad (7)$$

holds for $\{q_n\}_{n \in \mathcal{N}}$ where $Q_{-n}^* = \sum_{m \in \mathcal{N}, m \neq n} q_m^*$. Let Q_N^* denote the net public investment at the Nash Equilibrium, and is given by $Q_N^* := \sum_{n \in \mathcal{N}} q_n^*$. Then, a **key result** in *Non-Cooperative Game* is that only the CP which has the maximum value of $\left(r_n a_n - \frac{b_n^2}{2}\right)$ contributes to the public investment and the remaining CPs free-ride on this. If more than one CP has the maximum value of $\left(r_n a_n - \frac{b_n^2}{2}\right)$, i.e., some symmetry exists between the CPs, $q_n > 0$ for all such CPs. All the CPs though invest in their private infrastructure and the optimum value of p_n can be obtained by substituting Q by Q_N^* in (3). Let η denote the price of anarchy defined as $\eta := \frac{Q_C^*}{Q_N^*}$. Further, $\Gamma := \frac{U_C^*}{U_N^*}$, where U_C^* and U_N^* denote the optimum total utility in *Centralized Allocation* and *Non-Cooperative Game*, respectively. Then, another **key result** in *Non-Cooperative Game* is that $\eta > 1$. Further, if $\max_{n \in \mathcal{N}} \left(r_n a_n - \frac{b_n^2}{2}\right) > 1$, $\Gamma > 1$. Thus, the CPs reduce their net utility in non-cooperative interaction by playing selfishly.

We finally consider the **Bargaining Game**, wherein the CPs bargain with each other. In the case of disagreement, CPs do not make any public investment. Therefore, $Q = 0$ is the point of disagreement. Let $U_n^{*,D}$ denote the optimum utility of the CP n in case of disagreement and is obtained after optimizing (1) over p_n for $Q = 0$. The set of feasible utility functions of the CPs, denoted by $\mathcal{U}$, is defined as

$$\mathcal{U} = \{(U_1(Q), \dots, U_N(Q)) \mid Q \geq 0 \text{ and } U_n(Q) \geq U_n^{*,D} \forall n \in \mathcal{N}\}.$$

A Nash bargaining solution $\mathbf{U}^B = (U_1^B, \dots, U_N^B)$ is defined to be the solution of the problem formulated as

$$\max \prod_{n \in \mathcal{N}} (U_n - U_n^{*,D}) \text{ s. t. } (U_1, \dots, U_N) \in \mathcal{U}. \quad (8)$$

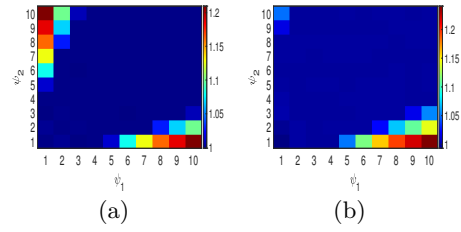


Figure 2: Variation of β against different values of $\psi_n (= r_n a_n)$ for $N = 2$ for $\mathbf{b} = [1, 1]$ in (a) and $\mathbf{b} = [1, 2]$ in (b).

Let Q_B^* denote the optimum total public investment in the bargaining game. Let β denote the benefit of bargaining which is defined as $\beta := \frac{Q_B^*}{Q_C^*}$. Then, the **key results** in the *Bargaining game* are as follows. A unique non-zero solution of (8) exists if $\sum_{n \in \mathcal{N}} \frac{\sqrt{1+2b_n^2 r_n a_n} - 1}{b_n^2} > 1$. Further, $\beta \geq 1$ under certain conditions implying that the bargaining game can result in higher public investment than the centralized allocation.

4. NUMERICAL RESULTS

We observe from Fig. 1(a) that η is high for the region ($\psi_1 \leq 1.5, \psi_2 \leq 1.5$). This happens as $\max_{n \in \mathcal{N}} \left(r_n a_n - \frac{b_n^2}{2}\right) \leq 1$ in this region. Same holds for the region ($\psi_1 \leq 3, \psi_2 \leq 3$) in Fig. 1(b) as $\mathbf{b} = [2, 2]$. In the remaining region, we observe $\eta > 1$. When $\psi_1 \approx \psi_2$, both CPs invest in public infrastructure leading to a higher value of Q_C^* , and hence, η . However, when there is a large asymmetry between the CPs, only the dominating CP makes a major investment in public infrastructure, leading to a lower value of η (See Fig. 1(a)). Similar observations can be made from Fig. 1(b). Also, we observe that η is slightly higher for $\mathbf{b} = [2, 2]$ than $\mathbf{b} = [1, 1]$. This happens as both the CPs are incentivized to invest more in private than public for $\mathbf{b} = [2, 2]$ than $\mathbf{b} = [1, 1]$, leading to lower value of net public investment.

We observe from Fig. 2(a) and 2(b) that $\beta = 1$ when the CPs are comparable, i.e., when $\psi_1 \approx \psi_2$. However, when CPs are non-comparable, i.e., either $\psi_1 \gg \psi_2$ or $\psi_2 \gg \psi_1$, the bargaining game results in higher public investment, and hence, $\beta > 1$. Further, we observe from Fig. 2(a) that β is symmetrical for $\psi_1 \gg \psi_2$ and $\psi_2 \gg \psi_1$ for $\mathbf{b} = [1, 1]$. However, in the case of $\mathbf{b} = [1, 2]$, β is higher for $\psi_1 \gg \psi_2$ than $\psi_2 \gg \psi_1$ (Refer Fig. 2(b)). Although CP 2 dominates the total public investment as $\psi_2 \gg \psi_1$, it has an incentive for private investment since $\mathbf{b} = [1, 2]$. Thus, both Q_B^* and Q_C^* reduces and in turn β reduces.

5. REFERENCES

- [1] B. Jullien and M. Bouvard. Fair cost sharing: Big tech vs telcos. Technical report, Toulouse School of Economics, October 2022.
- [2] A. Kalvit, S. Pinjani, G. Kasbekar, D. Manjunath, and J. Nair. Capacity expansion of neutral ISPs via content peering charges: The bargaining edge. *Performance Evaluation*, 133:43–56, 2019.
- [3] S. Patchala, S. Lee, C. Joo, and D. Manjunath. On the economics of network interconnections and its impact on net neutrality. *IEEE Transactions on Network and Service Management*, 18(4):4789–4800, 2021.

A Control-Theoretic Perspective on BBR/CUBIC Congestion-Control Competition

Simon Scherrer*
ETH Zürich

Adrian Perrig
ETH Zürich

Stefan Schmid
TU Berlin

ABSTRACT

To understand the fairness properties of the BBR congestion-control algorithm (CCA), previous research has analyzed BBR behavior with a variety of models. However, previous model-based work suffers from a trade-off between accuracy and interpretability: While dynamic fluid models generate highly accurate predictions through simulation, the causes of their predictions cannot be easily understood. In contrast, steady-state models predict CCA behavior in a manner that is intuitively understandable, but often less accurate. This trade-off is especially consequential when analyzing the competition between BBR and traditional loss-based CCAs, as this competition often suffers from instability, i.e., sending-rate oscillation. Steady-state models cannot predict this instability at all, and fluid-model simulation cannot yield analytical results regarding preconditions and severity of the oscillation. As a result, we lack an analytical understanding of when oscillation occurs, how severe it is, and how it impacts fairness.

To overcome this trade-off, we extend the recent dynamic fluid model of BBR by means of control theory. Based on this control-theoretic analysis, we derive quantitative conditions for BBR/CUBIC oscillation, identify network settings that are susceptible to instability, and find that these conditions are frequently satisfied by practical networks. Our analysis illuminates the fairness implications of BBR/CUBIC oscillation, by deriving and experimentally validating fairness bounds that reflect the extreme rate distributions during oscillation. In summary, our analysis shows that BBR/CUBIC oscillation is frequent and harms BBR fairness, but can be remedied with our control-theoretic framework.

1. OVERVIEW

During the global deployment of the BBR congestion-control algorithm (CCA) [1], researchers have documented several cases of unexpected BBR behavior, all with significant implications on performance and fairness [6, 4, 5, 2]. To improve the safety of CCA deployment in the future, recent research has aimed at a more fundamental understanding of BBR based on *modeling*, i.e., mathematical descriptions that enable insights valid for a wide range of network settings.

This model-based work has taken two different approaches.

*Corresponding author: simon.scherrer.ch@gmail.com

First, the *steady-state models* by Ware et al. [7] and Mishra et al. [4] identify a steady state of the congestion-control dynamics, and predict BBR behavior with closed-form expressions for network metrics in this steady state. Second, the *dynamic fluid model* from our previous work [5] describes the transient behavior of BBR by means of ordinary differential equations (ODEs), and predicts BBR properties by fluid-model simulation, i.e., numerical ODE solving.

These two approaches suffer from a trade-off between accuracy and *interpretability*, i.e., transparency regarding the *causes* of model predictions. As we demonstrate in our work, the dynamic fluid model is more accurate than the steady-state models, mainly because the fluid-model can represent behavior outside the steady state. However, the dynamic fluid model has so far mostly been used for simulation, which arrives at predictions through lengthy numerical computation. As such, the predictions of dynamic fluid models are less explainable than the predictions of steady-state models, because steady-state models only consist of a handful of closed-form expressions that can be intuitively understood.

In our work, we bridge this gap between accuracy and interpretability of models. In particular, we demonstrate that a *control-theoretic analysis* of the dynamic fluid model allows insights that are accurate, interpretable, and rigorously provable. We illustrate the value of this stability analysis by focusing on the competition between BBR and CUBIC [3], the most prevalent CCAs in the Internet. BBR/CUBIC competition frequently suffers from instability (*rate oscillation*), in which flows using different CCAs regularly obtain highly uneven bandwidth shares. While congestion-control oscillation (e.g., sawtooth in Reno) is sometimes necessary to maximize utilization or achieve per-flow fairness, BBR/CUBIC oscillation is unnecessary, unintentional and even harmful to fairness: Unfairness between BBR and CUBIC flows is especially high when oscillation occurs.

Crucially, this instability is invisible in steady-state models, as these models *assume* stability. In contrast, fluid-model simulation reveals this oscillation, but must be conducted for specific network settings to indicate the occurrence and the severity of the oscillation. More general insights into the oscillation are possible thanks to our control-theoretic analysis of the fluid model: This analysis yields (i) mathematical conditions on network parameters, indicating when oscillation provably occurs, (ii) analytical bounds on the severity of the oscillation, describing the extreme rate distributions of BBR/CUBIC competition, and (iii) intuitive explanations of stabilizing adaptations of BBR.

In summary, we present the following contributions:

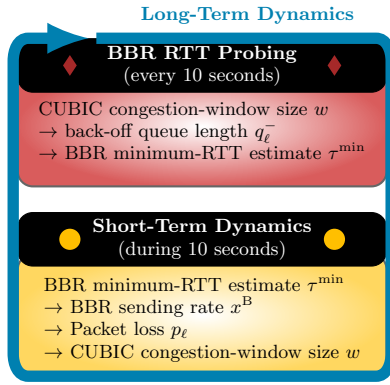


Figure 1: Mechanism of BBR/CUBIC oscillation.

Documentation of instability. We show that steady-state models predict BBR fairness towards CUBIC less accurately than dynamic models, due to their assumption that BBR/CUBIC competition converges. In reality, convergence is prevented by persistent sending-rate oscillation, which we systematically explain for the first time (§2). While the oscillation is predicted accurately by fluid-model simulation, such simulation cannot provide a formal understanding of preconditions and severity of the oscillation.

Stability analysis of BBR/CUBIC fluid model. We apply a control-theoretic analysis to a joint BBR/CUBIC fluid model, thereby describing the BBR/CUBIC oscillation analytically and accurately. For this analysis, we first derive the equilibrium of the BBR/CUBIC fluid model. Then, we derive an instability condition on network parameters, i.e., a network condition that is provably conducive to oscillation. This condition is derived using Lyapunov theory, center-manifold theory, and fixed-point iteration.

Insights into temporal fairness. Besides identifying the preconditions for oscillation, the stability analysis also illuminates fairness implications of BBR/CUBIC oscillation, yielding worst-case fairness bounds and approximate tight fairness bounds that we experimentally validate.

Countermeasure evaluation. We evaluate multiple adaptations of BBR for suppressing oscillation, and explain the results by means of our control-theoretic framework. These redesigned versions include several ad-hoc modifications of BBR, which we design and implement, and the official releases of BBRv2 and BBRv3.

Together, our results provide both a rigorous theoretical foundation and practical guidance for improving the stability and fairness of future congestion control.

2. OSCILLATION MECHANISM

As visualized in Fig. 1, the oscillation between BBR and CUBIC arises from the interaction of BBR’s periodic RTT-probing step and the short-term dynamics that follow until the next probe. During RTT probing, BBR sharply reduces its sending rate to estimate the propagation delay, but if CUBIC’s congestion window is large, it continues filling the buffer, preventing full clearance. This leads BBR to overestimate the minimum RTT, which inflates its congestion window, triggers aggressive bandwidth probing (a high probing strength), and substantial rate growth. The resulting high BBR rate causes CUBIC’s congestion window to shrink due to packet loss. Since both flows keep the buffer

occupied, BBR’s inflated RTT estimate cannot be corrected until it times out, sustaining the imbalance for roughly 10 seconds. At the next probe, CUBIC’s congestion-window size is smaller, and therefore BBR may recover an accurate RTT estimate and temporarily reduce its rate, allowing CUBIC to grow again. This mechanism sets up a repeating cycle of inflated RTT estimates, aggressive BBR growth, CUBIC backoff, and eventual recovery.

3. ANALYSIS OVERVIEW

We distinguish short-term and long-term dynamics.

The *short-term dynamics* describe the continuous BBR/CUBIC competition *between* the RTT probing steps of BBR. Between these steps, the BBR probing strength α is fixed, and determines the equilibria and the convergence behavior of the short-term dynamics. We show that unique and asymptotically stable short-term equilibria always exist.

Building on these short-term dynamics, we derive the *long-term dynamics*, describing the BBR/CUBIC competition *across* RTT-probing steps. In each RTT-probing step, the CUBIC congestion-window size w determines the probing strength α and thus the short-term dynamics for the next 10 seconds. These short-term dynamics then determine the congestion-window size w at the next RTT-probing step. Hence, the long-term dynamics form a discrete process, which has a unique equilibrium, but does not converge to it under mild conditions.

4. REFERENCES

- [1] Neal Cardwell, Yuchung Cheng, C Stephen Gunn, Soheil Hassas Yeganeh, and Van Jacobson. BBR: congestion-based congestion control. *Communications of the ACM*, 60(2):58–66, 2017.
- [2] Jose Gomez, Elie Kfoury, Jorge Crichigno, Elias Bou-Harb, and Gautam Srivastava. A performance evaluation of TCP BBRv2 alpha. In *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*, pages 309–312. IEEE, 2020.
- [3] Sangtae Ha, Injong Rhee, and Lisong Xu. CUBIC: a new TCP-friendly high-speed TCP variant. *ACM SIGOPS operating systems review*, 42(5):64–74, 2008.
- [4] Ayush Mishra, Wee Han Tiu, and Ben Leong. Are we heading towards a BBR-dominant Internet? In *Proceedings of the 22nd ACM Internet Measurement Conference (IMC)*, pages 538–550, 2022.
- [5] Simon Scherrer, Markus Legner, Adrian Perrig, and Stefan Schmid. Model-based insights on the performance, fairness, and stability of BBR. In *Proceedings of the 22nd ACM Internet Measurement Conference (IMC)*, pages 519–537, 2022.
- [6] Dominik Scholz, Benedikt Jaeger, Lukas Schwaighofer, Daniel Raumer, Fabien Geyer, and Georg Carle. Towards a deeper understanding of TCP BBR congestion control. In *Proceedings of the IFIP Networking Conference (IFIP Networking) and Workshops*, pages 1–9. IEEE, 2018.
- [7] Ranysha Ware, Matthew K Mukerjee, Srinivasan Seshan, and Justine Sherry. Modeling BBR’s interactions with loss-based congestion control. In *Proceedings of the Internet Measurement Conference*, pages 137–143, 2019.

On Efficient Solutions of General Structured Markov Processes in Quantum Computational Environments

Vasileios Kalantzis, Mark S. Squillante, Shashanka Ubaru
Mathematics of Computation, IBM Research
Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA
vkal@ibm.com, mss@us.ibm.com, shashanka.ubaru@ibm.com

1. INTRODUCTION

Consider an irreducible, ergodic discrete-time structured Markov process $\{X(n); n \in \mathbb{Z}_+\}$ on the two-dimensional state space $\hat{\Omega} = \{(i, j) : i \in \mathbb{Z}_+, j \in [M]\}$ with transition probability matrix having the block Toeplitz-like structure

$$\mathbf{P} = \begin{bmatrix} \mathbf{B}_0 & \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3 & \cdots \\ \mathbf{A}_{-1} & \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 & \cdots \\ \mathbf{0} & \mathbf{A}_{-1} & \mathbf{A}_0 & \mathbf{A}_1 & \cdots \\ \vdots & & \ddots & \ddots & \ddots \end{bmatrix}, \quad (1)$$

where $\mathbf{B}_{i,i \in \mathbb{Z}_+}$, and $\mathbf{A}_{i,i \in \mathbb{Z}_{\geq -1}}$, are nonnegative matrices in $\mathbb{R}^{M \times M}$ s.t. $\mathbf{A} = \sum_{i=-1}^{\infty} \mathbf{A}_i$ and $\mathbf{B} = \sum_{i=0}^{\infty} \mathbf{B}_i$ are stochastic, and M can be finite or infinite. Define $\boldsymbol{\pi} := (\pi_0, \pi_1, \pi_2, \dots)$, $\pi_i := (\hat{\pi}(i, 1), \dots, \hat{\pi}(i, M))$, and $\hat{\pi}(i, j) := \lim_{n \rightarrow \infty} \mathbb{P}[X(n) = (i, j)]$, $\forall (i, j) \in \hat{\Omega}$. Since the Markov process $\{X(n); n \in \mathbb{Z}_+\}$ is irreducible and ergodic, the limiting probability vector $\boldsymbol{\pi}$ is the stationary distribution of this process s.t. its invariant probability vector $\boldsymbol{\pi}$ exists and is uniquely determined as the solution of $\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}$ and $\boldsymbol{\pi}\mathbf{1} = 1$, where $\mathbf{1} = (1, \dots, 1)^\top$.

The most-efficient algorithms for computing the stationary distribution $\boldsymbol{\pi}$ of such M/G/1-type processes on digital computers consist of cyclic reduction (CR) methods. Despite the computational benefits of CR, the time to compute $\boldsymbol{\pi}$ can still be prohibitive for large stochastic models. Quantum computers offer the potential of achieving significant performance advantages for certain computational problems. Our focus is therefore on devising, from a mathematical perspective, the first quantum algorithms for efficiently computing $\boldsymbol{\pi}$ for structured Markov processes. We derive a mathematical analysis that establishes theoretical properties of our quantum algorithms, including the potential for significant performance improvements over the most-efficient CR algorithms in settings of both theoretical and practical importance. Although motivated by general structured Markov processes, our quantum algorithms can be exploited to address a much larger class of computational problems, as well as a subroutine in solving larger problems involving the stationary distribution on a quantum computer. This short paper briefly summarizes our full paper (*arXiv:2404.17959*, 2025) to which we refer the reader for all details and references.

2. STRUCTURED MARKOV PROCESSES

We can easily establish that the stationary distribution $\boldsymbol{\pi}$ of ergodic M/G/1-type Markov processes can be obtained in

terms of the matrix $\mathbf{G}_{\min}$, the minimal nonnegative solution of the nonlinear matrix equation $\mathbf{X} = \sum_{i=-1}^{\infty} \mathbf{A}_i \mathbf{X}^{i+1}$. Hence, our ultimate goal concerns determining $\mathbf{G}_{\min}$, for which the most-efficient classical methods are based on variants of CR.

The general CR approach basically consists of transforming the nonlinear equation for $\mathbf{X}$ into a semi-infinite linear system that leads to a linear system with respect to a matrix $\mathbf{H}$, and then applying the CR method in functional form. A single iteration of CR comprises applying an even-odd permutation to the block rows and to the block columns of the matrix $\mathbf{H}$, followed by one iteration of block Gaussian elimination. Since this transformation maintains the unchanged structure of the linear system, CR is repeatedly applied recursively which yields a sequence of block Hessenberg, block Toeplitz-like infinite systems that provably converges quadratically to a limit system whose solution can be explicitly evaluated. This involves the matrix $\mathbf{H}^{(n)}$ of the infinite system at each iteration n that is fully characterized by its first and second block rows whose respective block entries define in functional form two matrix power series by means of corresponding recursions. The main computational task of CR then consists of computing the first two block rows of the Schur complement matrix $\mathbf{H}^{(n+1)} = \mathbf{I} - \mathbf{U}_{22}^{(n)} - \mathbf{U}_{21}^{(n)}(\mathbf{I} - \mathbf{U}_{11}^{(n)})^{-1}\mathbf{U}_{12}^{(n)}$. This includes computations of infinite block triangular Toeplitz matrices, such as $\mathbf{T}_1 = (\mathbf{I} - \mathbf{U}_{11}^{(n)})$, $\mathbf{T}_2 = -\mathbf{U}_{21}^{(n)}$, $\mathbf{T}_3 = \mathbf{U}_{12}^{(n)}$ and $\mathbf{T}_4 = (\mathbf{I} - \mathbf{U}_{22}^{(n)})$, which we approximate with infinite banded Toeplitz matrices defined by parameters chosen to achieve the desired level of accuracy ϵ .

3. QUANTUM ALGORITHMS & ANALYSIS

We now derive our quantum CR **Alg. Q.1** and **Alg. Q.3** to address the computational bottlenecks of classical CR methods, with a particular interest on the computation phase. We first observe that each CR iteration involves various linear algebraic operations on Toeplitz matrices, specifically solving Toeplitz linear systems and Toeplitz matrix products. The computation phase of our quantum approach is based in part on the concept of approximating Toeplitz matrices by associated circulant matrices. Let f denote a generating function of a sequence of $N \times N$ Toeplitz matrices $\mathbf{T}^{[N]}$, where $N = d \cdot M$ and d is the numerical degrees of the matrix power series generated by iteration n of our quantum CR. It can be shown that the circulant matrices associated with the sequence of Toeplitz matrices converge super linearly whenever f is a positive function in the Wiener algebra class $\mathcal{W}$. Since the $\mathbf{P}$ matrix in (1) is nonnegative and stochastic, and thus the Toeplitz entries have finite sum, the associated circulant matrix is a very accurate approximation to the corresponding

Toeplitz matrix for sufficiently large N . We then exploit the resulting circulant matrices and operations thereon within the context of quantum computing environments to realize a theoretical potential of exponential speedups for the computation phase of these matrix operations.

After an initialization step performed on a digital computer, we operate on the Toeplitz matrices $\mathbf{T}_1$ and $\mathbf{T}_2$ with all subsequent steps executed on a quantum computer. Let f_1, f_2 be the generating functions of these two Toeplitz matrices, respectively. Upon loading the N columns of the matrix $\mathbf{T}_3$ onto the quantum computer as an initial state $|\psi_0\rangle$, we operate on each of the N columns of $\mathbf{T}_3$ independently and in quantum parallel by considering N sets of $\log(N)$ -qubits that independently encode the N columns of $\mathbf{U}_{12}^{(n)}$. Then, we approximately compute $\mathbf{T}_2 \mathbf{T}_1^{-1} |\psi_0\rangle = \Phi^\dagger \Lambda(f_2) \Lambda^{-1}(f_1) \Phi |\psi_0\rangle$, where Φ is the Fourier matrix, and $\Lambda(f_1), \Lambda(f_2)$ are diagonal matrices respectively corresponding to the eigenvalues of the associated circulant matrices of $\mathbf{T}_1, \mathbf{T}_2$. With oracle access to the functions f_1, f_2 by exploiting appropriate oracle quantum circuit methods, and given the Fourier coefficients, we compute these generating functions in a straightforward manner. Moreover, since these functions are periodic, we can design efficient quantum circuits to estimate the functions using the corresponding Fourier expansions.

Next, we apply QFT to the columns of $\mathbf{T}_3$ to obtain $|\psi'_0\rangle = \Phi |\psi_0\rangle$. To encode the eigenvalues of $\mathbf{T}_1$, we exploit the oracle access to the values of f_1 to map $|\psi'_0\rangle = \sum_{j=0}^N b_j |j\rangle \rightarrow \sum_{j=0}^N b_j |j\rangle |f_1(2\pi j/N)\rangle$. To invert the eigenvalues, we employ aspects of the quantum algorithmic HHL approach by adding a qubit, using controlled-rotations to invert the phases (eigenvalues), and using uncompute (restore state of corresponding qubits for reuse) to obtain an approximation to the state $|\psi_1\rangle = \Lambda^{-1}(f_1) |\psi'_0\rangle$. We now encode the eigenvalues of $\mathbf{T}_2$ exploiting the oracle access to the values of f_2 and map $|\psi_1\rangle = \sum_{j=0}^N \tilde{b}_j |j\rangle \rightarrow \sum_{j=0}^N \tilde{b}_j |j\rangle |f_2(2\pi j/N)\rangle$. Finally, we apply iQFT to obtain a state proportional to $|\psi_2\rangle = \Phi^\dagger \Lambda(f_2) \Lambda^{-1}(f_1) \Phi |\psi_0\rangle$. Upon computing the state $|\psi^*\rangle$, **Alg. Q.1, Alg. Q.3** check the quality of the approximate solution by computing the quantity $\|\mathbf{1} - (\mathbf{A}_{-1} + \hat{\mathbf{Z}}_0) \mathbf{1}\|_\infty$. Let $|a_{-1}\rangle = \text{vec}(\mathbf{A}_{-1})$ and $\text{mask}(|\psi^*\rangle) = \text{vec}(\hat{\mathbf{Z}}_0)$, where $\text{mask}(|\psi^*\rangle)$ considers only the amplitudes corresponding to $\hat{\mathbf{Z}}_0$ in $|\psi^*\rangle$. We load the columns of $\mathbf{A}_{-1}$ to form the state $|a_{-1} z_0\rangle = |a_{-1}\rangle + \text{mask}(|\psi^*\rangle)$, and compute $\|\mathbf{1} - (\mathbf{A}_{-1} + \hat{\mathbf{Z}}_0) \mathbf{1}\|_\infty$ from $|a_{-1} z_0\rangle$. If $\|\mathbf{1} - (\mathbf{A}_{-1} + \hat{\mathbf{Z}}_0) \mathbf{1}\|_\infty > \epsilon$, we repeat the inner loop; else, we proceed to output $\mathbf{J} = (\mathbf{I} - \hat{\mathbf{Z}}_0)^{-1} \mathbf{A}_{-1}$ by loading $|a_{-1}\rangle = \text{vec}(\mathbf{A}_{-1})$, and then using the above Toeplitz solution approach to form the state $|\tilde{j}\rangle = \text{vec}(\mathbf{J})$.

We end this section by turning to a mathematical analysis of the theoretical properties of our quantum algorithms. After establishing our main result of first-order bounds on sources of algorithmic errors, we establish our main results of upper bounds on the cumulative error up to iteration n of our **CR Alg. Q.1, Alg. Q.3** to respectively be exponential and linear in n . Lastly, we establish our main theoretical result on the asymptotic computational complexity of our quantum CR algorithms in comparison with the corresponding classical CR algorithms **Alg. C.1, Alg. C.3**. Let $\mu = f_{1,\max}/f_{1,\min}$ for the generating function f_1 associated with **Alg. Q.1, Alg. Q.3**, τ_{readout} the time complexity to readout the results from the quantum computer, and $d_{\max}^Q$ and $d_{\max}^C$ the maximum numerical degrees of the matrix power series generated by our **Alg. Q.1, Alg. Q.3** and the corresponding

Alg. C.1, Alg. C.3, respectively, with $N^Q = d_{\max}^Q \cdot M$.

THEOREM 3.1. *The overall computational complexity of the quantum CR **Alg. Q.1** and **Alg. Q.3** is given by*

$$O(\mu(\tau_{\text{load}} + \log^2 N^Q + \tau_{\text{oracle}}) + \tau_{\text{readout}}), \quad d_{\max}^Q = O(d_{\max}^C), \quad (2)$$

and requiring $N^Q(2 \log N^Q + 1)$ qubits. Supposing $\mu, \tau_{\text{oracle}}$ to be $O(\text{poly log } N^Q)$, we then have that the computational complexity $O(\mu(\log^2 N^Q + \tau_{\text{oracle}}))$ of **Alg. Q.1, Alg. Q.3** represents a theoretical potential for exponential speedup of the computation phase over the computational complexity of

$$O(M^3 d_{\max}^C + M^2 d_{\max}^C \log d_{\max}^C) \quad (3)$$

for **Alg. C.1, Alg. C.3**. Further supposing τ_{load} to be $O(\text{poly log } N^Q)$ or subexponential in $(\log N^Q)$, we then additionally have that the computational complexity $O(\mu(\tau_{\text{load}} + \log^2 N^Q + \tau_{\text{oracle}}))$ of **Alg. Q.1, Alg. Q.3** represents a theoretical potential for polynomial-to-exponential speedup over the computational complexity of [3] for **Alg. C.1, Alg. C.3**. Moreover, further supposing τ_{readout} to be $O(\text{poly log } N^Q)$ or subexponential in $(\log N^Q)$, we then additionally have that the overall computational complexity [2] of **Alg. Q.1, Alg. Q.3** represents a theoretical potential for polynomial-to-exponential speedup over the overall computational complexity of [3] for **Alg. C.1, Alg. C.3**.

Given $\mu = f_{1,\max}/f_{1,\min}$ for the generating function f_1 of **Alg. Q.1, Alg. Q.3** and given Toeplitz matrix properties together with usual ratios $f_{1,\max}/f_{1,\min}$ found in practice for M/G/1-type processes, we typically have μ to be $O(\text{poly log } N^Q)$ for sufficiently large N^Q . By exploiting oracle quantum circuit methods together with Toeplitz matrix properties and the corresponding generating functions f_1 and f_2 , we can have the oracle time τ_{oracle} to be $O(\text{poly log } N^Q)$ for sufficiently large N^Q . In such cases, Theorem 3.1 establishes that our quantum algorithms provide a theoretical potential for an exponential speedup of the computation phase in the decision-space on a quantum computer over that in [3] for **Alg. C.1, Alg. C.3**, which is of particular interest from a mathematical perspective. Next, by exploiting appropriate block encoding quantum circuit methods together with Toeplitz matrix properties to support efficient data loading, we can additionally have the load time τ_{load} to be $O(\text{poly log } N^Q)$ or subexponential in $(\log N^Q)$ for sufficiently large N^Q . In such cases, Theorem 3.1 establishes that our quantum algorithms for both input and computation phases, ignoring readout time, provide a theoretical potential for a polynomial-to-exponential speedup over that in [3] for **Alg. C.1, Alg. C.3**. Lastly, by exploiting properties of the result matrix $\mathbf{G}_{\min}$ such as instances where these matrices are sufficiently sparse to support efficient sampling-based readout, we can additionally have the readout time τ_{readout} to be $O(\text{poly log } N^Q)$ or subexponential in $(\log N^Q)$ for sufficiently large N^Q . In such cases, Theorem 3.1 establishes that our quantum algorithms for the input, computation and readout phases provide a theoretical potential for an overall polynomial-to-exponential speedup over that in [3] for **Alg. C.1, Alg. C.3**. We further note that, when the computation phase of our **Alg. Q.1, Alg. Q.3** represents a component of a larger computational problem on the quantum computer, then readout of the $\mathbf{G}_{\min}$ matrix is not needed and the overall computational complexity does not include the readout phase; the same can be true for the input phase.

Comparing approximations in the ASIP tandem queue

Wesley Geelen
Eindhoven University of
Technology
geelenwlc@gmail.com

Maria Vlasiou
University of Twente
m.vlasiou@utwente.nl

Yaron Yeger
Tel Aviv University
yaronyeg@mail.tau.ac.il

1. INTRODUCTION

Tandem stochastic systems (TSSs) are networks of simple stochastic components through which particles—customers, jobs, molecules—move sequentially. Despite the apparent simplicity of individual nodes, their aggregate dynamics are often highly complex. TSSs are widely used in communication networks [2], operations research [3], and physics [7].

The Asymmetric Inclusion Process (ASIP) [5] differs by allowing clustering: multiple particles may occupy the same site forming inseparable clusters that move together. It thus models unidirectional transport with irreversible aggregation, bridging TJNs and ASEP. ASIP captures clustering phenomena observed in road traffic [9], maritime logistics [6], and intracellular transport [4]. However, unlike TJNs, ASIP lacks product-form solutions; its state-space dependencies make exact stationary distributions intractable beyond very small systems. Existing attempts [10] show that the number of equations grows exponentially with sites, and only asymptotic approximations are available. This motivates the search for tractable approximations.

2. APPROXIMATION METHODS

This study develops two complementary methods, RMF and PSA, to approximate general ASIP systems for site-occupancy distributions and related performance measures. By comparing them against explicit formulas and simulations, we aim to evaluate their respective strengths and highlight how these methods make ASIP analytically and computationally tractable at scale.

Replica Mean-Field (RMF). The Replica Mean-Field (RMF) limit [1] simplifies dynamics by replicating the system infinitely and randomising interactions. Thus, in the limit, site arrivals become asymptotically Poisson, dependencies vanish, and effective independent-site dynamics emerge. As a result, RMF decouples sites, preserves average interactions, and provides analytic but approximate insight into ASIP.

Power Series Algorithm (PSA). PSA is a numerical method that assumes steady-state probabilities can be expressed as power series of the load, which are then computed via expansions. Coefficients are determined recursively from balance equations, yielding accurate approximations. While purely numerical, PSA is flexible, computationally fast, and systematically improvable. PSA has been widely applied in queueing models [8] and can aid optimisation.

3. THE RMF APPROXIMATION SCHEME

Consider R replicas of an n -site ASIP tandem queue. Each replica receives arrivals at site 1 according to an independent Poisson process with rate λ . Each site j has a gate that opens after exponentially distributed times with rate μ_j . When gate j of replica r opens, all particles in that site move instantaneously to site $j+1$ of a randomly chosen replica s , with equal probability $1/R$. At the last site, particles exit the system upon gate opening. Thus, the replicas form a randomised network of coupled tandem queues.

Theorem. *An explicit compact product-form approximation for the site occupancy PGF is given by*

$$G(z_1, z_2, \dots, z_n) = \prod_{j=1}^n \frac{1 + \sum_{i=1}^{j-1} \frac{\lambda(1-z_j)}{\mu_i}}{1 + \sum_{i=1}^j \frac{\lambda(1-z_j)}{\mu_i}}.$$

The probabilities that k ($k = 1, 2, 3, \dots$) particles are present at site j ($j = 1, 2, \dots, n$) are:

$$p_0 = \frac{1 + \lambda \sum_{i=1}^{j-1} \frac{1}{\mu_i}}{1 + \lambda \sum_{i=1}^j \frac{1}{\mu_i}}, \quad p_k = \frac{\lambda^k (\sum_{i=1}^j \frac{1}{\mu_i})^{k-1}}{\mu_j (1 + \lambda \sum_{i=1}^j \frac{1}{\mu_i})^{k+1}}.$$

The first and second moments are given by:

$$E[X_j] = \frac{\lambda}{\mu_j}, \quad E[X_j^2] = 2 \frac{\lambda}{\mu_j} \sum_{i=1}^j \frac{\lambda}{\mu_i} + \frac{\lambda}{\mu_j}.$$

The approximate PGF of the overall load is given by

$$E[z^{\sum_{j=1}^n X_j}] = \prod_{j=1}^n \frac{1 + \sum_{i=1}^{j-1} \frac{\lambda(1-z)}{\mu_i}}{1 + \sum_{i=1}^j \frac{\lambda(1-z)}{\mu_i}}.$$

4. THE PSA APPROXIMATION SCHEME

For the ASIP model, PSA expands the occupancy probabilities as a truncated power series in the traffic intensity $\rho = \lambda/\mu_1$. The coefficients of this series are recursively determined from the system's global balance equations. Each iteration of the scheme adds a new polynomial term.

Let $\vec{X} = (X_1, \dots, X_n)$ denote the occupancy vector and $p(\vec{x}) = \mathbb{P}(\vec{X} = \vec{x})$ denote the steady-state occupancy probability. In PSA, we approximate

$$p(\vec{x}) \approx \sum_{k=0}^N b_k(\vec{x}) \rho^{|\vec{x}|+k},$$

where the coefficients $b_k(\vec{x})$ are computed iteratively. The recursion arises from substituting the series into the global

balance and normalization equations. Each term requires using the previously determined recursive set of equations.

$$\begin{aligned}
\sum_{j=1}^n d_j(\vec{x}) b_k(\vec{x}) = & \\
& - a_1(\vec{x}) b_{k-1} \mathbb{1}_{\{k>0\}} + a_1(\vec{x} - \hat{e}_1) \mathbb{1}_{\{x_1>0\}} b_k(\vec{x} - \hat{e}_1) \\
& + \mathbb{1}_{\{x_n=0\}} \sum_{i=0}^{k-1} \sum_{m=1}^k d_n(\vec{x} + m\hat{e}_n) b_i(\vec{x} + m\hat{e}_n) \mathbb{1}_{\{k=i+m\}} \\
& + \sum_{i=2}^n \sum_{m=1}^{x_i} d_{i-1}(\vec{x} + m(\hat{e}_{i-1} - \hat{e}_i)) \mathbb{1}_{\{x_{i-1}=0\}} \mathbb{1}_{\{x_i>0\}} \cdot \\
& \cdot b_k(\vec{x} + m(\hat{e}_{i-1} - \hat{e}_i)). \quad (1)
\end{aligned}$$

$$b_k(\vec{0}) = - \sum_{0 < |\vec{x}| \leq k} b_{k-|\vec{x}|}(\vec{x}). \quad (2)$$

The scheme is then given by:

1. Choose $N \in \mathbb{N}$ and set $l := 1$.
2. For all $(k, \vec{x}) \in \mathbb{N}^{n+1}$ with $\vec{x} \neq \vec{0}$ and with $k + |\vec{x}| = l$, compute $b_k(\vec{x})$ recursively using equation (1).
3. Compute $b_l(\vec{0})$ using equation (2).
4. Set $l := l + 1$; if $l \leq N$, return to step (2), otherwise stop.

Occupancy distributions and expected quantities (such as moments or marginal probabilities) can be evaluated within the same iterative framework.

5. APPROXIMATIONS EVALUATION

For homogeneous ASIP systems, we consider a 10-site system with 10 particles, varying the arrival rate λ from 0.05 to 0.85 and setting all gate opening rates to $\mu = 1$. Site-occupancy distributions, along with means and variances, are examined. For site-occupancy distributions, the difference between the methods is insignificant for the first few sites. As we move downstream and as λ increases, we see that, compared to simulation, RMF slightly overestimates the probability of an empty site and underestimates the probability of one or more particles being present at that site. For all sites, the PSA approximation closely matches the simulation's results. Both approximation methods, when compared with simulations produce similar and relatively accurate results with small absolute errors. Total variation and L2 distances between the approximations and simulation are provided. RMF produces the exact mean, while the absolute relative error of PSA as a function of the arrival rate remains below 10^{-14} . For variances, in the first site the differences between the approximations and the exact values are minimal. The approximated RMF variance loses its accuracy as the site index increases, while the PSA approximation differs insignificantly from the exact variance.

Heterogeneous ASIP systems are considered for an 8-site system with arrival rate $\lambda = 0.45$. Five heterogeneity scenarios are considered: monotone decreasing and increasing rates, bimodal variants, and a randomly perturbed set. PSA and RMF approximations are compared with simulations. Heterogeneity introduces significant variability in

site-occupancy distributions. For example, in a system draining quickly (high downstream rates), RMF and PSA perform differently at downstream sites, with PSA generally providing higher accuracy: In cases where earlier gates have longer average times between openings, particle accumulation amplifies discrepancies in RMF estimates. PSA consistently matches simulation results closely, achieving runtimes 1.5–12 times faster than full Monte Carlo simulations, highlighting its computational advantage over simulations. The total variation distance and L2 norm between the approximations and simulations remain below 0.06 in the worst case, in the numerical experiments examined.

6. CONCLUSION

RMF and PSA offer complementary tools for ASIP tandem queues: RMF delivers simple, analytic approximations that scale efficiently and provide reasonable estimates for small to moderate occupancy indices. PSA, while computationally more intensive than RMF, achieves near-exact accuracy and accelerates performance evaluation relative to simulation, especially in heterogeneous systems. Together, they provide efficient alternatives to heavy simulations for performance evaluation of stochastic clustering networks.

7. REFERENCES

- [1] F. Baccelli, M. Davydov, T.O. Taillefumier (2022). Replica-mean-field limits of fragmentation-interaction- aggregation processes, *J. Appl. Probab.* **59**, 38–59.
- [2] D.P. Bertsekas, R.G. Gallager, Data Networks, 2nd ed., Prentice Hall, Hoboken, NJ, 1992.
- [3] H. Chen, D.D. Yao, Fundamentals of Queueing Networks, Springer, Berlin/Heidelberg, 2001.
- [4] J.S. Bonifacino and B.S. Glick (2004). The mechanisms of vesicle budding and fusion. *Cell* **116**, 153–166.
- [5] O.L. Bonomo, S. Reuveni, 2019. Occupancy correlations in the asymmetric simple inclusion process, *Phys. Rev. E* **100**, 042109.
- [6] C. P. Brum, R. G. de Faria Corrêa, D. A. de J. Pacheco, T. C. C. Nepomuceno, A. L. R. de Oliveira, and K. B. Marsola (2023). Efficiency analysis of inland waterway locks in maritime transportation systems: practical, economic and policy implications, *Maritime Policy & Management*, **51**(6), 1013–1036.
- [7] V.Y. Chernyak, M. Chertkov, D.A. Goldberg, K.J. Turitsyn (2010). Non-equilibrium statistical physics of currents in queuing networks, *J. Stat. Phys.*, **140**, 819–845.
- [8] G. Koole (1994). On the power series algorithm, *Report, Department of Operations Research, Statistics and System Theory, CWI*, 1–16.
- [9] K. C. Lan, C. M. Chou, and C. Y. Li (2014). Effects of driving behaviour on vehicle cluster formation and its application, *IET Intelligent Transport Systems*, **8**(5), 453–463.
- [10] S. Reuveni, I. Eliazar, U. Yechiali (2011). Asymmetric Inclusion Process, *Phys. Rev. E*, **84**, 041101.

On Mixed-Precision Iterative Methods and Analysis for Nearly Completely Decomposable Markov Processes

Vasileios Kalantzis, Mark S. Squillante, Chai Wah Wu
 Mathematics of Computation, IBM Research
 Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA
 vkal@ibm.com, mss@us.ibm.com, cwwu@us.ibm.com

1. INTRODUCTION

Consider an irreducible, ergodic discrete-time Markov process $\{X(s); s \in \mathbb{Z}_+\}$ defined on the state space $[n] := \{1, \dots, n\}$ with transition probability matrix $\mathbf{P}$ taking the *nearly completely decomposable* (NCD) block structural form

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \cdots & \mathbf{P}_{1m} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \cdots & \mathbf{P}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{m1} & \mathbf{P}_{m2} & \cdots & \mathbf{P}_{mm} \end{pmatrix}, \quad (1)$$

where $\mathbf{P}_{ij} \in \mathbb{R}^{n_i \times n_j}$, $i, j \in [m]$, are nonnegative block matrices such that $\sum_{j \in [m]} \mathbf{P}_{ij}$ are stochastic (i.e., rows sum to 1 with each element in $[0, 1]$), $\mathbf{P} \in \mathbb{R}^{n \times n}$ is a stochastic matrix, $n = \sum_{i \in [m]} n_i$, and the elements of $\mathbf{P}_{ij}$ are tiny relative to those of $\mathbf{P}_{ii}$, $i, j \in [m]$, $i \neq j$; namely, $\|\mathbf{P}_{ii}\| = O(1)$ and $\|\mathbf{P}_{ij}\| = O(\epsilon)$, with $\|\cdot\|$ the spectral norm of a matrix and $\epsilon > 0$ a tiny constant. Define $\boldsymbol{\pi} := (\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_m)$, $\boldsymbol{\pi}_i := (\pi_{i1}, \dots, \pi_{in_i})$, $\pi_{ik} := \lim_{s \rightarrow \infty} \mathbb{P}[X(s) = (\sigma_i + k)]$, $k \in [n_i]$, where $\sigma_i := \sum_{\ell=1}^{i-1} n_\ell$, $i \in [m]$. Since the Markov process $\{X(s); s \in \mathbb{Z}_+\}$ is irreducible and ergodic, the limiting probability vector $\boldsymbol{\pi}$ is the stationary distribution of this process such that its invariant probability vector $\boldsymbol{\pi}$ exists and is uniquely determined as the solution of $\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}$ and $\boldsymbol{\pi}\mathbf{1} = \|\boldsymbol{\pi}\|_1 = 1$, where $\mathbf{1} = (1, \dots, 1)^\top$ is the column vector containing all ones, and $\|\cdot\|_1$ is the l_1 norm of a vector.

Beyond their broad applications, NCD structural properties represent a fundamental area in the theory of Markov processes. The most efficient approaches for computing the stationary distribution $\boldsymbol{\pi}$ of such NCD Markov processes are based on similar forms of aggregation-disaggregation. Despite significant performance benefits, these methods can still be prohibitively expensive for large-scale processes and/or real-time settings. Our focus is therefore on general NCD Markov processes and on the design and analysis of a general mathematical framework for computing the $\boldsymbol{\pi}$ of NCD processes that addresses the performance bottlenecks of existing methods. Our framework combines: (1) advances in computer architectures related to mixed-precision computation – to reduce computation times at the expense of inaccuracies; (2) advances in iterative approximate computing methods – to mitigate inaccuracies, further reduce computation times, and guarantee convergence. We derive a mathematical analysis that establishes theoretical properties of our general framework, including convergence rates, and then present

numerical experiments that demonstrate orders of magnitude improvements in computation times over existing methods. We note that our general mathematical framework can be analogously exploited within the context of numerical methods for general Markov processes beyond those having NCD structural properties. This short paper briefly summarizes our full paper [1] to which we refer the reader for all details.

2. ALGORITHMIC SOLUTIONS

The goal of our general algorithmic framework is to provide significant computational and theoretical improvements over the most efficient existing methods. While our general mathematical framework can be applied to any of these methods and beyond, we focus here on applying our algorithmic framework to the method of Koury, McAllister, Stewart (KMS). We consider two such applications of our general algorithmic framework that primarily differ in the degree of aggressiveness with which they exploit mixed-precision computation and iterative approximate computing methods.

The performance bottlenecks of the KMS algorithm concern the steps of computing the solution of $m + 1$ systems of linear equations in each (outer-loop) iteration, namely a series of $n_i \times n_i$ linear systems in one step, $i \in [m]$, and an $m \times m$ linear system in another step. To address these bottlenecks, the first application of our general algorithmic framework **Alg. 3.1** consists of replacing every invocation of a full-precision linear system solver within each outer-loop iteration by a mixed-precision linear system solver based on the iterative refinement (IR) method. Once the level of reduced precision has been appropriately determined, the IR method is used to solve each of these linear systems where the iterative process is repeated until the residual falls below a relative tolerance for double-precision arithmetic. The second application of our general algorithmic framework **Alg. 3.2** takes a more aggressive approach by replacing every invocation of a full-precision linear system solver in each outer-loop iteration t with a fixed number of steps k_t of the mixed-precision Richardson iteration (RI) method. The sequence of k_t inner-loop RI steps is monotonically increased with the progression of the outer-loop iterations t , starting with fewer RI steps to reduce computation at the expense of not realizing full-precision results, which is then addressed through appropriate levels of preconditioning and iteration.

3. MATHEMATICAL ANALYSIS

We now turn to derive a mathematical analysis of our general algorithmic framework, summarizing our main theoretical results for both of its applications which concern,

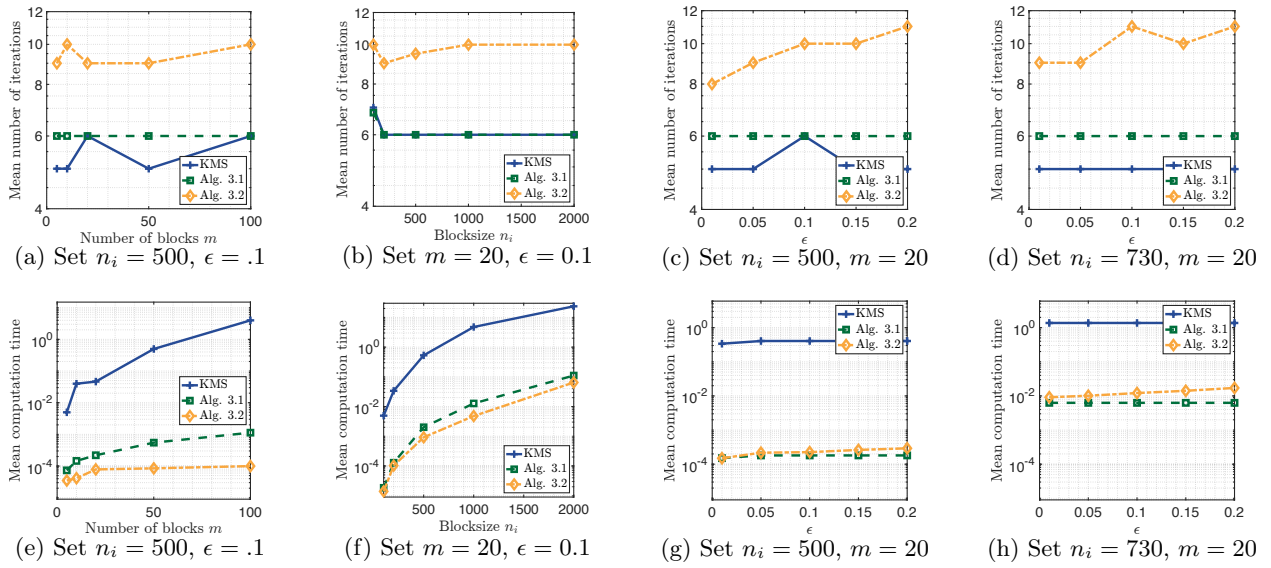


Figure 1: Performance of Algorithm 3.1, Algorithm 3.2 and KMS baseline for diagonal matrices $\mathbf{P}_{ii}$ with dimension n_i , number of blocks m , and off-diagonal matrices $\|\mathbf{P}_{ij}\| = O(\epsilon)$, averaged over 10 independent trial runs (with tiny standard deviations).

on the one hand, significant reductions in execution times afforded by mixed-precision computation at the expense of inaccuracies in the results and, on the other hand, the ability to mitigate inaccurate computations, further reduce execution times, and guarantee convergence afforded by iterative approximate computing methods. Our main theoretical results establish that **Alg. 3.1** is guaranteed to converge with an approximation error $\|\pi^{(t)} - \pi\|$ decreasing by a factor of $O(\epsilon)$ at each iteration t ; and that the inner-loop use of the IR method is guaranteed to converge linearly with a decreasing factor which is typically much faster than the decreasing factor of $O(\epsilon)$ for the outer-loop convergence.

THEOREM 3.1. ***Alg. 3.1** converges with an approximation error in the solution $\pi^{(t)}$ at each iteration t that decreases by a factor of $O(\epsilon)$. The mixed-precision IR method for solving every linear system $\mathbf{Ax} = \mathbf{b}$ in each outer-loop iteration t converges linearly with an approximation error that decreases by a factor of $O(\kappa(\mathbf{A}))$ in each inner-loop iteration.*

Our main theoretical results also establish that **Alg. 3.2** is guaranteed to converge with an error $\|\pi^{(t)} - \pi\|$ decreasing in regard to a tradeoff between an increased number of outer-loop iterations t with fewer RI steps k_t and a decreased computational complexity of the RI method for smaller k_t . We derive numerical linear algebraic methods to address such well-known tradeoffs. By starting with an inexact application of the RI method and increasing the number of RI steps k_t with the number of outer-loop iterations t , we avoid over-solving in the earlier stages when the estimate $\pi^{(t)}$ is far from π using a geometric criterion for k_t such as γ^t or $\frac{\rho}{(t+1)^\theta}$.

THEOREM 3.2. ***Alg. 3.2** converges with an approximation error in the solution $\pi^{(t)}$ at each iteration $t > t^*$ that decreases by a factor of $O(\epsilon)$. The mixed-precision RI method with k_t steps converges linearly with an approximation error that decreases by a spectral radius factor, which for sufficiently large k_{t^*} in the increasing sequence $k_1 < \dots < k_{t^*} < k_t$ of the outer-loop iterations $t > t^*$ is dominated by this outer-loop.*

4. NUMERICAL EXPERIMENTS

In this section we present a representative sample of numerical experiments for solving NCD Markov processes that support our theoretical results and empirically evaluate our general algorithmic framework, demonstrating that our framework exhibits relatively little or no increase in the number of outer-loop iterations and orders of magnitude improvements in the computation time over the most efficient KMS baseline. Fig. 1 presents the mean number of iterations and the mean time (in seconds) for computing the stationary distribution of NCD Markov processes with **Alg. 3.1**, **Alg. 3.2**, and the KMS baseline, averaged over ten independent experimental trial runs. Each vertical pair of plots fixes two of the three NCD parameters m , n_i and ϵ (as noted), while varying the remaining parameter. The first three vertical pairs of plots are based on randomly generated matrices under the specified parameters, whereas the last vertical pair of plots is based on matrices selected from a library of real-world linear systems. We first observe that the mean number of iterations under **Alg. 3.1** are essentially identical to that of the baseline KMS, with the mean number of iterations under **Alg. 3.2** only requiring a few additional iterations (though of lower computational complexity). The mean number of iterations as a function of the varying parameter remains relatively flat for all algorithms, except for **Alg. 3.2** requiring slightly more outer-loop iterations as ϵ increases, all as expected due to the algorithmic properties. We further observe that both applications of our algorithmic framework provide orders of magnitude reduction in the mean computation time, linearly increasing with m and n_i and remaining consistent with increasing ϵ . **Alg. 3.2** often provides the lowest mean computation times, though typically having relatively small differences with **Alg. 3.1** due to their algorithmic properties.

5. REFERENCES

- [1] V. Kalantzis, M.S. Squillante, C.W. Wu. On mixed-precision iterative methods and analysis for nearly completely decomposable Markov processes. arXiv:2504.06378, 2025.

Size-Aware Dispatching to Fluid Queues

Runhan Xie
University of California,
Berkeley
runhan_xie@berkeley.edu

Esa Hyytiä
University of Iceland
esa@hi.is

Rhonda Righter
University of California,
Berkeley
rrighter@berkeley.edu

ABSTRACT

We develop a fluid-flow model for routing problems, where fluid consists of different size particles and the task is to route the incoming fluid to n parallel servers using the size information in order to minimize the mean latency.

The problem corresponds to the dispatching problem of (discrete) jobs arriving according to a stochastic process. In the fluid model the problem reduces to finding an optimal path to empty the system in n -dimensional space. We use the calculus of variation to characterize the structure of optimal policies. Numerical examples shed further light on the fluid routing problem and the optimal control of large distributed service systems.

1. INTRODUCTION

The problem of routing jobs to parallel FCFS (first-come first-served) servers based on job sizes and server workloads in order to minimize, e.g., mean latency is important for many applications [1, 2, 3]. Though SRPT scheduling for each server is optimal in this context, FCFS is the dominant scheduling policy in practice; see, e.g., [4], [5].

Even under Markov assumptions with homogeneous FCFS servers, there is no simple characterization of the optimal dispatching policy when both the size of an arriving job and the workloads at all servers are observed [6]. We therefore consider stable dispatching systems at the fluid limit, so instead of individual jobs arriving according to a point process in time, we model the incoming jobs as a continuum, while keeping the notion of job sizes as a control parameter of the fluid flow process.

The value function, or total latency, of the fluid system approximates the value function of the underlying stochastic system, but is more computationally tractable for determining the optimal policy and evaluating heuristics.

The main contributions of this paper are as follows:

1. We formulate the fluid control problem that captures the dynamics of the original dispatching system when the number of jobs in the system is large. To our knowledge, we are the first to study size-aware flow control for fluid models of queueing systems.
2. We show that shorter jobs should be routed to shorter queues, which reduces the problem to an optimal path

problem, and that the optimal path is invariant under appropriate scaling, which reduces the number of dimensions required to determine the path. We also observe that unbalancing the workloads is beneficial, especially under heavy load.

3. For some paths, and for the optimal path in heavy traffic, the fluid value function matches the value function for the original stochastic model.
4. We show, for two servers, that latency decreases as job sizes become more variable in convex ordering.
5. We develop new heuristics for the original model based on the optimal fluid policy when workloads are large, but are adjusted to avoid idling for small workloads.

The fluid control problem reduces to the problem of determining an optimal path to empty the system. We give integral expressions for costs of arbitrary paths. Fixed control actions correspond to straight line segments, for which closed-form expressions are available.

2. RELATED WORK

The dispatching problem has been extensively studied in the queueing theory literature, and many policies have been proposed and shown to be optimal under different assumptions on the available information (server states and job sizes). If the system is homogeneous and state-aware only, then Join-the-Shortest-Queue (JSQ) [7, 8] or Least-Work-Left (LWL) [9, 10, 11, 12] have been shown to be optimal. If the system is size-aware only, then Size-Interval-Task-Assignment (SITA) [13] is shown to be optimal [14]. With the routing history as the only state information, Round-Robin (RR) is optimal, and combining RR and SITA can out-perform RR or SITA alone [15, 16, 17].

The problem is much harder when the dispatching policy is both size- and state-aware. Sequential dispatching heuristics, that use both state and job-size information, and that route short jobs to short queues, were introduced in [18], and a particular one, called DICE, was shown numerically to have excellent performance. Another dispatching heuristic, CARD, was proposed in [19] and proven to be delay optimal in heavy traffic, and was shown to perform well in simulations. Our work provides theoretical support for heuristics like CARD and DICE that unbalance the loads and send shorter jobs to shorter queues.

Fluid models have been applied to study staffing problems e.g. [20, 21, 22], scheduling problems e.g. [21, 23, 24, 25], and dispatching problems e.g. [26, 27, 28, 29, 30]. These

models and their controls are often state aware, i.e., they use remaining fluid in the system for decision making, but are not job-size aware. We take a first step in studying size-aware fluid dispatching problems.

3. FLUID DISPATCHING MODEL

Each of n FCFS servers has service rate $1/n$, and work comes in as a fluid flow at rate $\lambda < 1$ comprising different size particles with known density $f(x)$, CDF $F(x)$, and CCDF $\bar{F}(x) = 1 - F(x)$, and with mean $E[X] = 1$, so that $\rho = \lambda E[X] = \lambda < 1$. The incoming flow is to be split among servers as a function of job sizes (or size of particles).

The state information, assumed known, is denoted by $\mathbf{u} = (u_1, \dots, u_n)$, where u_i defines the backlog, or remaining workload, in queue i . Thus, with no arrivals, queue i would empty at time $u_i n$. A control policy (job dispatcher) π splits the incoming flow of jobs λ into n sub-flows, based on job sizes and the current backlog. That is, the policy chooses arbitrary subsets of job sizes from non-negative reals, $A_i(\mathbf{u})$, and routes jobs of size $x \in A_i(\mathbf{u})$ to server i . We will show that for an optimal policy, each subset comprises a single non-overlapping interval, and moreover, shorter jobs should be routed to servers with smaller backlog.

The instantaneous rate of jobs, $\lambda_i(\mathbf{u})$, and the work flow, $\rho_i(\mathbf{u})$, routed to server i is then

$$\lambda_i(\mathbf{u}) := \lambda \int_{A_i(\mathbf{u})} f(x) dx; \rho_i(\mathbf{u}) := \lambda \int_{A_i(\mathbf{u})} x f(x) dx$$

where $\sum \lambda_i(\mathbf{u}) = \lambda = \rho = \sum \rho_i(\mathbf{u})$. For ease of notation, we often omit “ $(\mathbf{u})$ ” and write λ_i and ρ_i , but we remind the reader that these rates are constantly changing according to the control policy.

Given our objective of minimizing latency, each arriving job assigned to server i incurs cost $u_i n$, so server i incurs waiting time cost at rate $\lambda_i u_i n$, and the total cost rate is

$$c(\mathbf{u}) := n \sum_i \lambda_i u_i. \quad (1)$$

For all meaningful policies $\mathbf{u} = 0$ is an absorbing state where no further costs are incurred, but there may be others. Let n_0 denote the number of empty servers, $n_0 := |\{i : u_i = 0\}| = \sum_i \mathbf{1}(u_i = 0)$. Whenever $n_0 \cdot (1/n) \geq \rho$, the empty servers can process all incoming jobs without increasing their backlogs, incurring no further costs, so we call any state with $n_0 \geq n\rho$ an absorbing state. We can assume, without loss of generality, that the system stops (is absorbed) in such a state. The question is how to move from the current state $\mathbf{u}$ to an absorbing state at minimal cost. It is easy to show that before absorption, incoming fluid should be routed to servers with zero fluid at rate $\rho_i = 1/n$, and all servers should work at rate $1/n$.

When fewer than n^* servers are empty, we have

$$\dot{u}_i = \rho_i - \frac{1}{n} =: -\nu_i,$$

where ν_i is queue i 's instantaneous drainage rate, with $\rho_i = 1/n$ if $u_i = 0$. The cost (value function) of policy $\pi = (A_1(\mathbf{u}), \dots, A_n(\mathbf{u}))$, with $\tau_\pi(\mathbf{u})$ denoting the time to absorption, is

$$v_\pi(\mathbf{u}) := \int_0^\infty c(\mathbf{u}_\pi(t)) dt = \int_0^{\tau_\pi(\mathbf{u})} c(\mathbf{u}_\pi(t)) dt \forall \mathbf{u}.$$

Remark 1 (Random Split) *With load-balancing random split (RND), the total cost incurred is*

$$C_i = \int_0^{\frac{n}{1-\rho} u_i} c_i(t) dt = n \int_0^{\frac{n}{1-\rho} u_i} \frac{1-\rho}{n} t dt = \frac{\rho n u_i^2}{2(1-\rho)},$$

$$v_{RND}(\mathbf{u}) = \sum_{i=1}^n C_i = \frac{\rho n}{2(1-\rho)} (u_1^2 + \dots + u_n^2),$$

which, for $\rho < 1$ is finite for any job size distribution, regardless of $E[X^2]$. Moreover, this matches the value function of the stochastic system; for each $M/G/1$ -FCFS queue, $v(u) = \rho u^2 / (2(1-\rho))$ [31, 32].

4. ANALYSIS WITH n SERVERS

We first show that the optimal policy is a dynamic SITA-type policy, which, for a given $\mathbf{u}$, splits job sizes into n non-overlapping intervals, sending shorter jobs to servers with less work. That is, with $u_1 \geq u_2 \geq \dots \geq u_n$, the optimal policy is determined by size thresholds $0 = h_0(\mathbf{u}) \leq \dots \leq h_n(\mathbf{u}) = \infty$, so that $A_{n+1-i}(\mathbf{u}) = [h_{i-1}(\mathbf{u}), h_i(\mathbf{u}))$.

Proposition 1 *The optimal policy assigns the shortest jobs to the server with the smallest backlog, the next size interval to the server with the second smallest backlog, and so on.*

PROOF. We prove this by contradiction. Suppose $u_i > u_j$ and the optimal policy routes jobs of size $(x, x + \delta)$ to server i and jobs of size $(y, y + \delta')$ to server j , where $x + \delta < y$. We choose δ and δ' so that

$$E[X \cdot \mathcal{I}(x \leq X \leq x + \delta)] = E[X \cdot \mathcal{I}(y \leq X \leq y + \delta')],$$

i.e., the loads contributed by the two streams of jobs are equal. Let $\lambda_x = \lambda P(x \leq X \leq x + \delta)$ and $\lambda_y = \lambda P(y \leq X \leq y + \delta')$, so $\eta := \lambda_x - \lambda_y > 0$. The cost rate due to these subflows of jobs is

$$c_{ij} = n(\lambda_x u_i + \lambda_y u_j).$$

Interchanging the corresponding job flows reduces the cost rate:

$$c_{ji} = n(\lambda_x u_i + \lambda_y u_j) = n(\lambda_y u_i + \lambda_x u_j + \eta(u_i - u_j)) < c_{ij},$$

which contradicts the optimality of the proposed policy. $\square$

Corollary 1 (Structure of optimal policies) *The optimal dispatching policy can be characterized by $n-1$ appropriately chosen state-dependent thresholds $h_i(\mathbf{u})$. Therefore, the optimal policy from any initial point $\mathbf{u}$ is determined by a path from that point to an absorbing state.*

4.1 The Value Function in Terms of Paths

Note that there is a one-to-one correspondence between a policy π starting in some state $\mathbf{u}$ and a path $\mathbf{r}(s)$ from $\mathbf{u}$ to $(0, \dots, 0)$, where s is the curve parameter. By time reversal, we can also consider paths from the origin to $\mathbf{u}$. We shall consider only paths $\mathbf{r}(s) = (r_1(s), \dots, r_n(s))$ that are admissible by an appropriately chosen control policy. Then $\mathbf{r}(0) = (0, \dots, 0)$, $\mathbf{r}(s_1) = \mathbf{u}$, and $\mathbf{r}'(s)$ defines the direction of the path at point s , which must coincide with the direction of the drainage rates $\boldsymbol{\nu} = (\nu_1, \dots, \nu_n)$, $\boldsymbol{\nu}(s) = A(s) \mathbf{r}'(s)$, where the scalar function $A(s)$ adjusts the “speed” at curve point s to the actual drift. Before the path reaches an absorbing state (while $n_0 < n^*$), the total drift is $\nu_1 + \dots + \nu_n =$

$1 - \rho$, giving $A(s) = (1 - \rho)/(r'_1(s) + \dots + r'_n(s))$, where $r'_1(s) + \dots + r'_n(s) > 0$ as long as $\rho < 1$. Once the path reaches an absorbing state, $r'_i(s) = 0$ if $u_i = 0$, and $r'_i(s) = 1/n$ if $u_i > 0$. Then,

$$\boldsymbol{\nu}(s) = \frac{1 - \rho}{r'_1(s) + \dots + r'_n(s)} (r'_1(s), \dots, r'_n(s)).$$

Thus, the (Euclidean) length of the drainage vector $\|\boldsymbol{\nu}(s)\|$ along the path $\mathbf{r}(s)$ is

$$\|\boldsymbol{\nu}(s)\| = \frac{1 - \rho}{r'_1(s) + \dots + r'_n(s)} \|\mathbf{r}'(s)\|.$$

Note that $\nu_i(s) < 0$ if server i is receiving more work than it can process (at the given time).

Lemma 1 (Value function) *The total accumulated cost along a work-conserving path $\mathbf{r}(s)$ is*

$$v_{\mathbf{r}}(\mathbf{u}) = \frac{1}{1 - \rho} \int_0^{s_1} c(s)(r'_1(s) + \dots + r'_n(s)) ds, \quad (2)$$

where $c(s)$ is the cost rate (per unit time) that depends on the respective control actions according to (1).

As $\mathbf{r}'(s)$ defines λ_i and ρ_i , the cost rate $c(s)$ depends solely on $\mathbf{r}(s)$ and $\mathbf{r}'(s)$, and therefore (2) can be evaluated (at least numerically if not in closed form) for any given path $\mathbf{r}(s)$.

4.2 Scale-free paths

Let us consider a family of “scale-free” paths that are obtained by *scaling* a reference path $\mathbf{r}_0(s)$,

$$\mathbf{r}(s) = \alpha \mathbf{r}_0(s),$$

where $\mathbf{r}_0(s)$ is some fixed path to $\mathbf{u} = \mathbf{r}_0(s_1)$ and $\alpha > 0$ is a free scaling parameter. Substituting $\mathbf{r}(s)$ into (2) reveals a quadratic relationship between the two value functions:

Lemma 2 *The value function of $\mathbf{r}(s) = \alpha \mathbf{r}_0(s)$ is*

$$v_{\mathbf{r}}(\alpha \mathbf{u}) = \alpha^2 v_{\mathbf{r}_0}(\mathbf{u}).$$

Lemma 2 gives the following two important structural properties of the optimal policy.

Corollary 2 *Scale-free paths are optimal.*

PROOF. By contradiction, suppose $\mathbf{r}(s)$ and $\tilde{\mathbf{r}}(s)$ are optimal paths for two states $\mathbf{u}$ and $\tilde{\mathbf{u}} = \alpha \mathbf{u}$ such that $v(\mathbf{u}) \neq \alpha^2 v(\mathbf{u}_0)$, so $\tilde{\mathbf{r}}(s) \neq \alpha \mathbf{r}(s)$. Then either $\mathbf{r}(s)$ or $\tilde{\mathbf{r}}(s)$ cannot be optimal. First, if $v(\alpha \mathbf{u}) > \alpha^2 v(\mathbf{u})$, then $\tilde{\mathbf{r}}(s) = \alpha \mathbf{r}(s)$ has lower cost than $\mathbf{r}(s)$. The other case is similar. $\square$

Corollary 3 *The value function for any scale-free path scales quadratically in the Euclidean distance $|\mathbf{u}|$,*

$$v_{\mathbf{r}}(\mathbf{u}) = |\mathbf{u}|^2 w(\theta),$$

where $w(\theta)$ is the value function at unit distance in the direction θ in n -dimensional space.

Due to the scaling property, we need only determine $w(\theta)$ in an $n - 1$ dimensional surface to obtain the optimal paths for every state in the n -dimensional space.

5. OPTIMAL PATHS WITH TWO SERVERS

For $n = 2$ servers, both with service rate $1/2$, we assume that $u_1 \geq u_2$, so that u_1 corresponds to the horizontal axis. Now it is also convenient to consider an alternate state representation, (x, y) , where $x = u_1 + u_2$ is the total backlog and $y = u_1 - u_2$ is the imbalance. From the scale-free property (Corollary 2) the optimal path depends only on the relative imbalance, either captured by θ , the angle for the point (u_1, u_2) , or the relative queue difference, $\hat{y} = y/x$. This means that if, for some point (x_0, y_0) with relative imbalance $\hat{y}_0$, the optimal $\hat{y}'_0 = 0$, i.e., the relative imbalance should not change, then this will be true until the system empties. We call such a path a straight to the origin (STO) path. Because our paths are continuous, the path will visit a point with a given $\hat{y}$ either exactly once, or will maintain $\hat{y}$ on an STO path. Indeed, we have the following corollary.

Corollary 4 (Monotonicity of optimal paths) *For two servers, the optimal fluid path is monotonic in $\hat{y}$. That is, from any initial point (x, y) , the optimal path will always increase or decrease the relative imbalance until the system empties or until some $\hat{y}_0$, after which it will follow an STO path.*

A consequence is that if the optimal policy empties queue 2 before queue 1, queue 2 will remain empty, even if $\rho > 1/2$. Similarly, if the optimal path moves to perfectly balanced queues, they will remain balanced.

For each point (x, y) any path is characterized by $y' = dy/dx$, which in turn, for an optimal path, is controlled by a single job-size threshold $h = h(x, y) = h(u_1, u_2)$: jobs shorter than h are assigned to queue 2 and the rest to queue 1. Let $g(h)$ denote the load due to particles (jobs) smaller than h , and recalling that $\lambda = \rho$, we have

$$g(h) := \rho \int_0^h z f(z) dz = \rho P\{X \leq h\} \cdot E[X|X \leq h].$$

Then, while $u_2 > 0$ ($y < x$), u_2 changes at rate $g(h) - 1/2$, u_1 changes at rate $\rho - g(h) - 1/2$, x changes at rate $\rho - 1$, and

$$y' = \frac{(\rho - g(h) - 1/2) - (g(h) - 1/2)}{\rho - 1} = \frac{2g(h) - \rho}{1 - \rho},$$

so $g(h) = (\rho + (1 - \rho)y')/2$. Because $0 \leq g(h) \leq \rho$, we have, for $y < x$, that $|y'| = |y'(x)| \leq \rho/(1 - \rho)$. That is, as the load increases there is more room to maneuver in terms of changing the path direction, including having the ability to increase (temporarily) the load at one server if $\rho > 1/2$.

To capture total costs in terms of the control y' , let $\Phi(y')$ denote the fraction of jobs forwarded to queue 2,

$$\Phi(y') := F \left[g^{-1} \left(\frac{\rho + (1 - \rho)y'}{2} \right) \right].$$

The cost rate is then

$$2\rho(u_2 \Phi(y') + u_1 (1 - \Phi(y'))) = \rho(x + (1 - 2\Phi(y'))y),$$

and the total cost for an arbitrary path $y' = y'(x)$ (and $y(x)$) from initial state (x_1, y_1) is

$$v(x_1, y_1) = \frac{\rho}{1 - \rho} \left[x_1^2/2 + \int_0^{x_1} (1 - 2\Phi(y'))y dx \right].$$

Because x_1^2 is independent of the path, an optimal policy minimizes

$$T = \int_0^{x_1} (1 - 2\Phi(y')y) dx.$$

As observed in Corollary 4, the optimal $\hat{y}$ will be monotone until some $\hat{y}_0 = y_0/x_0$, after which it will be constant until $(x, y) = (0, 0)$. For such an STO path, $y' = \hat{y}_0$ and

$$T_{\text{STO}}(x_0, y_0) = x_0^2 \hat{y}_0 (1 - 2\Phi(\hat{y}_0)).$$

When $\hat{y} = 1$ ($u_2 = 0$), $\Phi(\hat{y}) = \Phi(1) = F(g^{-1}(1/2))$, and $T_{\text{STO}}(x_0, x_0) = x_1^2 (1 - 2\Phi(1))$. If $u_2 = 0$ and $\rho < 1/2$, then $\Phi(1) = 1$ and $T_{\text{STO}}(x_0, x_0) = 0$.

Because F and g are nondecreasing, so is Φ , and $\Phi(\hat{y}) \geq \Phi(0) > 1/2$. Taking the derivative we have the following.

Proposition 2 *The cost of an STO path, from any point (x_0, y_0) directly to the origin, is decreasing in $\hat{y}_0 = y_0/x_0$, for any job-size distribution.*

We conjecture that the optimal $\hat{y}' \geq 0$ (the relative imbalance should increase for the optimal path). This is supported numerically as well as by the proposition above.

Let us now consider the optimal policy when $\rho \rightarrow 1$. Define the MWL (most work left) policy as the policy that routes all jobs (fluid) to the long queue until the short queue is empty; then short jobs are routed to the short queue such that the rate of fluid to the short queue is $\max\{\rho, 1/2\}$. It is not hard to derive the following, where $\bar{\Phi}(x) = 1 - \Phi(x)$.

Proposition 3 *The MWL value function, $v_{\text{MWL}}(\mathbf{u})$, is*

$$2\rho[2u_1u_2 + (2\rho - 1)u_2^2 + \frac{\rho\bar{\Phi}(1)}{1 - \rho}(u_1 + u_2(2\rho - 1))^2]$$

and MWL is asymptotically optimal, with

$$(1 - \rho)v_{\text{MWL}}(\mathbf{u}) \rightarrow 2\rho\bar{\Phi}(1)(u_1 + u_2)^2 = 2\rho\bar{\Phi}(1)x^2.$$

The first two terms in the MWL value function correspond to the cost to empty the short queue, and the last term (the dominating term in heavy traffic) is the cost for the STO path that then empties the long queue once the short queue is empty.

For the original stochastic dispatching system with identical servers, the mean waiting time is $E[W] = E[v(X, 0)]$ [33], i.e., $(1 - \rho)E[W] \rightarrow 2\rho(1 - \Phi(1))E[X^2]$. This agrees with the exact analysis of the asymptotically optimal policy for the original stochastic system [19], suggesting that the far more tractable fluid value function may be a good approximation for the original value function even when $\rho < 1$.

We now consider the effect of job size variability on the dispatching cost. Intuitively, since the policy uses job-size information, we expect more variability to decrease costs, which is in fact the case. Recall that for two random variables X and Y , X is more variable than Y in the convex sense, $X \geq_{cx} Y$, if $Ef(X) \geq Ef(Y)$ for all convex functions f . The proof of the following is in the appendix.

Proposition 4 *$X \geq_{cx} Y \Rightarrow$ the cost is lower for any path when X is a random job size rather than Y , so the optimal cost is also lower.*

Now let us consider the extreme case, where all jobs have the same size, $X \equiv 1$. This is equivalent to allowing an

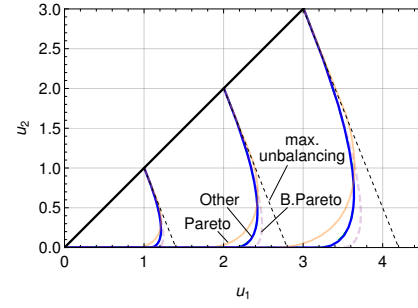


Figure 1: Optimal paths with exponential, uniform, and two Pareto distributions when $\rho = 0.7$.

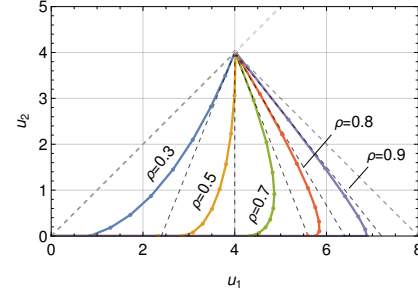


Figure 2: Optimal paths depending on ρ for exponential job sizes.

arbitrary job-size distribution but job sizes are not observed or are not used in the dispatching algorithm. In this case, we can show that any work-conserving dispatching policy is optimal: the proof is in the appendix.

Proposition 5 *All size-unaware work-conserving dispatching policies are equally good.*

6. NUMERICAL EXAMPLES

6.1 Optimal fluid paths

Let us first study how the shape of the job size distribution affects the optimal trajectories. Figure 1 shows the optimal paths for exponential, uniform, and both bounded and normal Pareto distributions. The bounded Pareto distribution with $\alpha = 1$ is truncated to $[1/66, 6]$, so that the mean is approximately one, and the variance is approximately 2. The variance for the Pareto is infinite.

The scaling property is clearly visible, and the difference in job-size distributions matters more when the relative imbalance is larger. It also seems advantageous to empty one queue at maximal speed when backlogs are identical regardless of the job-size distribution. (The feasibility conditions are indicated by dashed lines on the graph.) The trajectories are consistent with our conjecture that the relative imbalance always increases along any optimal trajectory to the origin (or until one queue empties).

Figure 2 shows the optimal paths with exponentially distributed job sizes at different loads. As ρ increases, the optimal strategy unbalances the queues more, consistent with our heavy-traffic result.

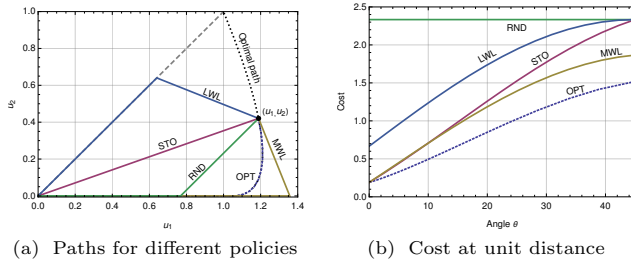


Figure 3: Heuristic and optimal paths and costs with exponential job sizes and $\rho=0.7$.

6.2 Heuristic policies

Next, we compare some heuristic routing policies with the optimal dispatching policy for fluid flow with exponential job sizes and $\rho = 0.7$.

Figure 3(a) depicts the paths for different heuristics and the optimal policy. The OPT path unbalances the workloads strongly, but not as much as MWL, which does it maximally. All of the chosen heuristics and the optimal policy have the scaling property, so the costs for the whole (u_1, u_2) -plane can be characterized by following an arc at unit distance. The resulting costs (value functions) are shown in Figure 3(b) as a function of the angle θ for points at unit distance, $\|\mathbf{u}\| = 1$. The costs for all policies are increasing in θ (decreasing in the imbalance). When the fluid system starts with a large imbalance (small θ), RND and LWL have significantly larger costs than STO or MWL.

Let us next consider heuristics for the original stochastic dispatching system, using the value function of the fluid model as an approximation for the value function of the original system, and again assuming exponential job sizes.

We propose two heuristics that use the optimal fluid policy when backlogs are large, but adjusts to avoid idling when the short queue length is small. The F-BLB (Fluid with a buffer lower bound) policy uses the fluid optimal policy when the backlog of the short queue is greater than some threshold, or buffer lower bound, $u_2 > u_B$, and uses the LWL policy otherwise. Numerically we observe that $u_B = 3$ gives good results. The F-BLBH policy uses the fluid optimal policy when $u_2 > u_B$ and the arriving job size exceeds a threshold h_S . Jobs of size less than h_S are always sent to the short queue, and LWL is followed if $u_2 \leq u_B$. In the numerical examples, we use $(u_B, h_S) = (2, 1.5)$.

We also consider two sequential heuristics, DICE [18] and CARD [19]. DICE routes a job of size x to the queue with the least backlog if $u_2 + x < \tau$, where we use $\tau = 6$, i.e., the virtual buffer can accommodate six average-sized jobs. CARD uses two job-size thresholds to define “small,” “medium,” and “large” jobs, and it always routes small jobs to the short queue and large jobs to the long queue, and it routes medium jobs to the short (long) queue if the short queue length is below (above) some threshold. DICE generally tends to perform better than the sequential CARD heuristic [19], while requiring a single tuning parameter that works well for all loads, rather than CARD’s three parameters that all need to be tuned for each load.

Finally, we consider the Short-to-Short and Long-to-Long (SSLL) heuristic. Jobs shorter than a threshold h_S are always routed to the shorter queue, and the rest go to the longer queue. We use the load balancing threshold, $h \approx$

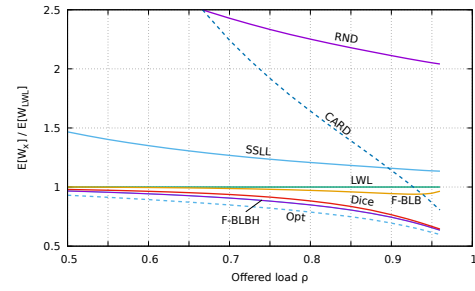


Figure 4: Performance with different heuristic policies relative to LWL as a function of ρ .

1.678 (that is specific to job-size distribution). This is also known as (static) SITA with switch, which is an improvement on SITA, where short jobs are always sent to a given fixed queue, regardless of the relative backlogs.

Figure 4 shows the numerical results as a function of ρ for the heuristics and the optimal policy. The y -axis is scaled by the waiting time with LWL, as a base-line policy, i.e. the relative performance metric for policy π is the ratio,

$$\frac{E[W_\pi]}{E[W_{LWL}]}$$

The numerical results are based on average waiting time observed during relatively long simulation runs of 100-200 million arrivals. We can observe that F-BLB does a better job than LWL as load increases (LWL is optimal when the system is lightly loaded). CARD, while provably optimal as the load goes to 1, does not perform well at lower loads. On the other hand, the rather simple DICE and F-BLBH heuristics have near-optimal performance across all loads. We also note that if the initial loads are balanced, LWL is the individually optimal policy for strategic jobs that choose a queue upon arrival, so we can view the curve for the optimal policy in Figure 4 as a measure of the “cost of anarchy.”

7. FUTURE WORK

Many of our fluid results hold for two servers. We would like to extend our results and heuristics to more than two servers. One possibility for a heuristic approach, along the lines of the CARD heuristic, is to carefully control the fluid to the short queue, while dividing the remaining fluid to balance the loads of the other queues.

Heuristics based on the fluid model work well when there are a large number of jobs in all queues. We expect that they will work well for a model in which there are occasional large bursts of traffic, and queues have finite buffers, so that overflows are forced into shorter queues. We plan to explore the impact on the optimal fluid paths when queues have finite buffers, as well as the effect of bursty arrivals.

8. APPENDIX

8.1 Proof of Proposition 4

Consider two job-size distributions, $X \sim F_X$ and $Y \sim F_Y$, with similar subscripts for T , g , and g^{-1} . Because F , g , g^{-1} ,

and $z = z(y')$ are non-decreasing, we have

$$\begin{aligned} g_X(h) &\leq g_Y(h) \quad \forall h \geq 0 \\ \Rightarrow g_X^{-1}(z) &\geq g_Y^{-1}(z) \quad \forall z \in [0, \rho] \\ \Rightarrow \Phi_X(y') &\geq \Phi_Y(y') \quad \forall y' \in [-\rho/(1-\rho), \rho/(1-\rho)] \\ \Rightarrow T_X &\leq T_Y. \end{aligned}$$

Also,

$$\begin{aligned} g_X(h) &= \rho \int_0^h x f_X(x) dx = \rho E[X|X \leq h] P\{X \leq h\} \\ &= \rho[1 - E[X|X > h] P\{X > h\}] \\ &= \rho[1 - E[(X - h)^+]]. \end{aligned}$$

Because $\rho[1 - (x - h)^+]$ is convex in x , the result follows.

8.2 Proof of Proposition 5

Here our admissible dispatching policies are the fraction of jobs, p , to be routed to server 2, depending on the backlogs, (u_1, u_2) , and by work conserving, we assume that when $u_2 = 0$, the fraction of jobs routed to server 2 will be $p = 1/(2\lambda) = 1/(2\rho)$ (the maximal proportion to incur 0 costs, by making the load $= 1/2$). Given the fraction of jobs routed to server 2 is p ,

$$y' = \frac{(2p-1)\rho}{1-\rho} \quad \Rightarrow \quad p = \frac{(1-\rho)y' + \rho}{2\rho},$$

and the cost rate is given by

$$c = 2\rho(pu_2 + (1-p)u_1) = 2\rho\left(\frac{x+y}{2} - py\right) = \rho x - (1-\rho)y'y$$

which is similar to our earlier cost rate, with p replacing Φ . The total cost rate is

$$v_{det}(x_1, y_1) = \frac{1}{1-\rho} \left[\rho x_1^2/2 - \int_0^{x_1} (1-\rho)y'y dx \right].$$

Thus, ignoring constants and x_1^2 , which does not depend on the path, we obtain the following equivalent variational problem for the optimal path:

$$T = \int_0^{x_1} y'y dx = \max.$$

Integrating T by parts gives

$$\int_0^{x_1} y'y dx = \int_0^{y_1} y dy = \frac{y_1^2}{2},$$

i.e. any allowed (work-conserving) path is optimal.

9. REFERENCES

- [1] W. Winston, "Optimality of the shortest line discipline," *Journal of applied probability*, vol. 14, no. 1, pp. 181–189, 1977.
- [2] P. K. Johri, "Optimality of the shortest line discipline with state-dependent service rates," *European Journal of Operational Research*, vol. 41, no. 2, pp. 157–161, 1989.
- [3] A. Ephremides, P. Varaiya, and J. Walrand, "A simple dynamic routing problem," *IEEE transactions on Automatic Control*, vol. 25, no. 4, pp. 690–693, 2003.
- [4] S. Madni, M. Abd Latiff, M. Abdullahi, S. Abdulhamid, and M. Usman, "Performance comparison of heuristic algorithms for task scheduling in iaas cloud computing environment," *PLoS One*, vol. 12, no. 5, 2017.
- [5] M. Tirmazi, A. Barker, N. Deng, M. Haque, Z. Qin, S. Hand, M. Harchol-Balter, and J. Wilkes, "Borg: the next generation," *Proceedings of the Fifteenth European Conference on Computer Systems, ACM*, 2020.
- [6] E. Hyytiä, P. Jacko, and R. Righter, "Routing with too much information?" *Queueing Systems*, vol. 100, no. 3, pp. 441–443, 2022.
- [7] R. R. Weber, "On the optimal assignment of customers to parallel servers," *Journal of Applied Probability*, vol. 15, no. 2, pp. 406–413, 1978.
- [8] W. Winston, "Optimality of the shortest line discipline," *Journal of applied probability*, vol. 14, no. 1, pp. 181–189, 1977.
- [9] D. Daley, "Certain optimality properties of the first-come first-served discipline for G/G/s queues," *Stochastic Processes and their Applications*, vol. 25, pp. 301–308, 1987.
- [10] S. G. Foss, "Approximation of multichannel queueing systems," *Siberian Mathematical Journal*, vol. 21, no. 6, pp. 851–857, 1980.
- [11] G. M. Koole, *On the optimality of FCFS for networks of multi-server queues*. Centre for Mathematics and Computer Science, 1992.
- [12] O. T. Akgun, R. Righter, and R. Wolff, "Partial flexibility in routeing and scheduling," *Advances in Applied Probability*, vol. 45, no. 3, pp. 673–691, 2013.
- [13] M. Harchol-Balter, M. Crovella, and C. Murta, "Task assignment in a distributed system: Improving performance by load unbalancing," in *Proceedings of SIGMETRICS*, vol. 98, 1998.
- [14] H. Feng, V. Misra, and D. Rubenstein, "Optimal state-free, size-aware dispatching for heterogeneous M/G/-type systems," *Performance evaluation*, vol. 62, no. 1-4, pp. 475–492, 2005.
- [15] A. Ephremides, P. Varaiya, and J. Walrand, "A simple dynamic routing problem," *IEEE transactions on Automatic Control*, vol. 25, no. 4, pp. 690–693, 1980.
- [16] Z. Liu and D. Towsley, "Optimality of the round-robin routing policy," *Journal of applied probability*, vol. 31, no. 2, pp. 466–475, 1994.
- [17] Z. Liu and R. Righter, "Optimal load balancing on distributed homogeneous unreliable processors," *Operations Research*, vol. 46, no. 4, pp. 563–573, 1998.
- [18] E. Hyytiä and R. Righter, "On sequential dispatching policies," in *32nd International Telecommunication Networks and Application Conference (ITNAC'22)*, Wellington, New Zealand, Nov. 2022, pp. 1–6.
- [19] R. Xie, I. Grosf, and Z. Scully, "Heavy-traffic optimal size-and state-aware dispatching," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 8, no. 1, pp. 1–36, 2024.
- [20] J. M. Harrison and A. Zeevi, "A method for staffing large call centers based on stochastic fluid models," *Manufacturing & Service Operations Management*, vol. 7, no. 1, pp. 20–36, 2005.

- [21] W. Whitt, "Fluid models for multiserver queues with abandonments," *Operations research*, vol. 54, no. 1, pp. 37–54, 2006.
- [22] —, "Staffing a call center with uncertain arrival rate and absenteeism," *Production and operations management*, vol. 15, no. 1, pp. 88–102, 2006.
- [23] R. Atar, C. Giat, and N. Shimkin, "The $c\mu/\theta$ rule for many-server queues with abandonment," *Operations Research*, vol. 58, no. 5, pp. 1427–1439, 2010.
- [24] Y. Chen and J. Dong, "Scheduling with service-time information: The power of two priority classes," *arXiv preprint arXiv:2105.10499*, 2021.
- [25] N. Zychlinski, C. W. Chan, and J. Dong, "Managing queues with different resource requirements," *Operations Research*, vol. 71, no. 4, pp. 1387–1413, 2023.
- [26] R. Talreja and W. Whitt, "Fluid models for overloaded multiclass many-server queueing systems with first-come, first-served routing," *Management Science*, vol. 54, no. 8, pp. 1513–1527, 2008.
- [27] O. Perry and W. Whitt, "Responding to unexpected overloads in large-scale service systems," *Management Science*, vol. 55, no. 8, pp. 1353–1367, 2009.
- [28] R. Stanojevic and R. Shorten, "Distributed dynamic speed scaling," in *2010 Proceedings IEEE INFOCOM*. IEEE, 2010, pp. 1–5.
- [29] J. Huang, Y. Liu, R. Li, K. Li, J. An, Y. Bai, F. Yang, and G. Xie, "Optimal power allocation and load balancing for non-dedicated heterogeneous distributed embedded computing systems," *Journal of Parallel and Distributed Computing*, vol. 130, pp. 24–36, 2019.
- [30] S. Hou, W. Ni, S. Chen, S. Zhao, B. Cheng, and J. Chen, "Real-time optimization of dynamic speed scaling for distributed data centers," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 3, pp. 2090–2103, 2020.
- [31] E. Hytiä, A. Penttinen, and S. Aalto, "Size-and state-aware dispatching problem with queue-specific job sizes," *European Journal of Operational Research*, vol. 217, no. 2, pp. 357–370, 2012.
- [32] E. Hytiä, R. Richter, and S. Aalto, "Task assignment in a heterogeneous server farm with switching delays and general energy-aware cost structure," *Performance Evaluation*, vol. 75, pp. 17–35, 2014.
- [33] E. Hytiä and R. Richter, "Towards the optimal dynamic size-aware dispatching," *Performance Evaluation*, vol. 164, p. 102396, 2024.

Large-scale analysis of load-balancing policies for free-floating car-sharing models

Christine Fricker
INRIA Paris, DI-ENS
christine.fricker@inria.fr

Hanene Mohamed
MODAL'X, Université Paris
Nanterre
hmohamed@parisnanterre.fr

Alessia Rigonat
INRIA Paris, DI-ENS
alessia.rigonat@inria.fr

ABSTRACT

In *free-floating* (FF) car-sharing systems, FF cars share parking spaces in public space with much more numerous private cars. The system is modelled as a closed queuing network with two classes of customers, FF and private cars, with different orders of magnitude. They move between N nodes, each with a large capacity CN , $C > 0$.

Mean-field techniques are adapted to the presence of private cars, which act as a fast-varying random environment. This allows us to study the impact of load-balancing policies on the distribution of the number of FF cars per zone when the system gets large. For two load-balancing policies, respectively modifying the routing probabilities and the pick-up rates, we obtain an explicit distribution, a binomial negative distribution driven by a fixed point equation. The influence of parameters on performance is derived.

1. INTRODUCTION

Context. Free-floating (FF) car-sharing systems are becoming increasingly popular as an alternative means of transport that is both environmentally friendly and cost-effective. These car-sharing systems do not require physical stations. FF cars share public space with the more numerous private cars. Few studies take into account this sharing of capacity between private and FF cars. Indeed, by dividing the service area into small zones, from 0.25 km^2 to 1 km^2 , acting as stations [5], transport research often assimilates FF car sharing to a classical model based on stations with arbitrarily fixed capacities.

Model. The stochastic model presented here is specific to FF. It involves a closed system of M_N FF cars and L_N private cars moving according to two different Markov dynamics between N zones. Each zone has a large capacity CN , $C > 0$. Private cars are assumed to be more numerous than FF cars, so that M_N is of the order of N while L_N is of the order of N^2 . For further discussion on the orders of magnitude, see Appendix A. Travel times for both types of cars follow exponential distributions, while routing is different: uniform for private cars and weighted for FF cars. This modeling is flexible enough to take into account balancing policies based on the number of FF cars available in each zone.

Aim of the paper. This paper studies the impact of load-balancing policies on the distribution of the number of FF cars per zone when the system is large. For the application, the objective is to have one FF car per zone. In particular, two incentive policies are considered, aimed at better distributing FF cars and improving their availability. The first policy has repulsive routing (RR) rates, which discourage users from parking their cars in zones close to saturation. The second presents adaptive pick-up (AP), which adapts the pick-up rates to the number of FF cars parked locally.

Results. The large-scale analysis of our model with load-balancing policies gives that the lack of parking spaces is independent of the presence of free-floating cars. The limiting invariant distribution of the number of FF cars per zone is derived in Theorem 3.1. The analysis uses mean-field techniques, adapted to the presence of private cars which act as a random environment for FF cars. For the RR and AP policies, this limiting distribution is explicit. It is a negative binomial distribution with a parameter driven by a fixed point equation. This is a generalization of the geometric distribution of the zero-incentive case.

The impact of each of the two policies on system performance is studied. We prove that the performance is a decreasing function of the fleet size parameter. This allows us to quantify the minimum fleet size required to attain a given performance threshold, in terms of the strength of the load-balancing policies. Qualitatively, when the strength of repulsion in the RR policy or that of adaptivity in the AP policy increases, the fleet size that guarantees a certain performance becomes smaller. Moreover, both policies exhibit a significant performance improvement when transitioning from zero-incentives to low-incentives dynamics, after which the gains become progressively smaller.

Related work. From a queuing point of view, the model proposed here is a network with two classes of customers with two different orders of magnitude. This model is suited to FF and close to the one introduced in [3] and analyzed in [2] where FF car routing is uniform among the N zones and private cars arrive and leave every zone independently, without moving between zones. Thus, the asymptotic behavior of the system is that of $N M/M/CN/CN$ queues of class-2 customers (private cars), disrupted by the presence of a small number of class-1 customers (FF cars) moving between N nodes (zones) as a closed system. From the point of view of FF cars, competition with private cars to share the capacity of each zone creates a random environment that varies rapidly due to the numerical imbalance between FF and private cars. In [2], it is shown that the operator has

no control over the lack of parking spaces, but only over the availability of FF cars. Dynamic pricing can be used to better distribute FF cars in the service area. In [7], it is shown that even small differences in pricing by a simple adaptive algorithm can affect demand or routing. Dynamic pricing therefore seems a promising way for the operator for rebalancing based on incentive policies.

Beyond shared mobility, load-balancing is a main topic in stochastic networks with a huge literature. However, most papers concern parallel queues rather than closed systems. It consists in schemes like d-choice from [9], Power-of-2 from [6], Join-the-shortest-queue (JSQ), Join-the-idle-queue (JIQ), etc. Our load-balancing policies are closer to the dynamics introduced by Spitzer in [8], in the context of interacting particle systems, where the jump rates depend only on the state of the departure site (zero range).

2. MODEL DESCRIPTION

The model considered in this paper is a system with two classes of customers that move between N nodes, each with capacity CN , for $C \in \mathbb{N}^*$. The total numbers of class-1 customers (FF cars) and class-2 customers (private cars) are, respectively, M_N and L_N . We assume

$$\lim_{N \rightarrow \infty} \frac{M_N}{N} = s > 0 \quad \text{and} \quad \lim_{N \rightarrow \infty} \frac{L_N}{N^2} = w > 0,$$

where the parameter s is called the fleet size parameter. In the following $w_N \stackrel{\text{def}}{=} L_N/N^2$. Therefore, the two classes of customers have different orders of magnitude with respect to the scaling parameter N . This reflects the fact that in the real world, private cars outnumber FF cars.

The dynamics of the model are described as follows.

- There is a single server in each node for class-1 customers. Service times at node i have exponential distribution with parameter $\lambda(v_i)$, where $\lambda : \mathbb{N} \mapsto \mathbb{R}^+$ is a bounded increasing function, and v_i is the number of class-1 customers at node i . At service completion, the class-1 customer leaves the node and starts a switch-over time. Service times represent user inter-arrival times in the zone.
- The switch-over time has exponential distribution with parameter μ . At the end of the switch-over, the customer of class-1 chooses the next node, say node j , according to probability $P_j(v) = p(v_j)/\sum_{l=1}^N p(v_l)$, where $p : \mathbb{N} \mapsto \mathbb{R}_+$ is a decreasing function and

$$\exists C > 0, \quad 1/C \leq \frac{\min_{k \geq 0} p(k)}{\max_{k \geq 0} p(k)} \leq C. \quad (1)$$

If the customer does not find an available slot at node j , they start a new switch-over time with exponential distribution with the same parameter μ . Then they choose a node according to probability $P(v)$, and repeat the process until they find an available slot.

- Once they join a node, class-2 customers are immediately served. The service times have an exponential distribution with parameter β .
- The switch-over time of class-2 customers has exponential distribution with parameter α . At the end of the switch-over, the customer of class 2 chooses the next node to visit with uniform probability $1/N$. For

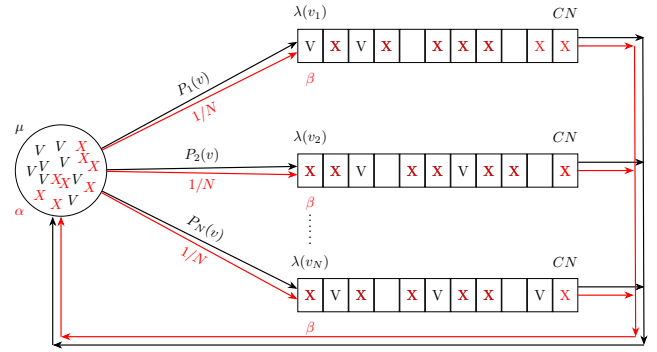


Figure 1: The system with N nodes of capacity CN connected to the $(N+1)$ -th central node. Two classes of customers: FF cars V and private cars X .

class-2 customers, saturation is managed in the same way as for class-1 customers.

- All inter-arrival times, service times and switch-over times are independent.

The system can be seen as a generalization of a closed Jackson network with $N+1$ nodes and two classes of customers. The $(N+1)$ -th node has infinite capacity, it represents the route containing all cars in circulation (see Figure 1).

3. MARKOV PROCESS

For $i = 1, \dots, N$, let us denote by $m_i^N(t)$ the number of available slots, $V_i^N(t)$ the number of class-1 customers and $X_i^N(t)$ the number of class-2 customers, at node i , at time t . Note that, for $i = 1 \dots N$ and $t \geq 0$, the relation

$$m_i^N(t) + V_i^N(t) + X_i^N(t) = CN$$

holds. Thus $X^N(t)$ can be omitted and the state of the system is described by the process

$$(m^N(t), V^N(t)) = ((m_i^N(t), V_i^N(t))_{1 \leq i \leq N}),$$

which is a Markov process on the state space $\mathcal{S}^N$ defined as the set of $(m, v) \in \mathbb{N}^{2N}$ with the following constraints

$$m_i + v_i \leq CN, \sum_{k=1}^N v_k \leq M_N, \sum_{k=1}^N (CN - m_k - v_k) \leq L_N.$$

The transitions of this Markov process modify only the state of one node at a time. The transition rates for node i are, from state $(m, v) = (m_i, v_i)_{1 \leq i \leq N}$ to

$$(m, v) + \begin{cases} -e_i + f_i & \mathbb{I}_{\{m_i > 0\}} \mu (M_N - \sum_{l=1}^N v_l) P_i(v) \\ +e_i - f_i & \lambda(v_i) \mathbb{I}_{\{v_i > 0\}} \\ +e_i & \beta (CN - m_i - v_i) \\ -e_i & \mathbb{I}_{\{m_i > 0\}} \alpha (L_N - \sum_{k=1}^N (CN - m_k - v_k)) / N, \end{cases}$$

where $(e_i, f_i)_{1 \leq i \leq N}$ is the canonical basis of $\mathbb{R}^{2N}$.

3.1 Preliminary results

The following lemma gives a coupling between the process of available slots and a birth-and-death process, which will be useful to prove convergence results. For this, we assume that λ is a bounded function on $\mathbb{N}$.

LEMMA 3.1 (COUPLING). Assume $w > C$. For any $\beta_0 > \beta$, any w_0 such that $C < w_0 < w$ and N large enough, for $i = 1, \dots, N$, there exists a coupling between process $(m_i^N(t))$ and $(L_m^i(Nt))$, where $(L_m^i(t))$ is the number of customers in a $M/M/1$ queue with arrival rate $\beta_0 C$ and service rate $\alpha(w_0 - C)$, such that

$$L_m^i(0) = m_i^N(0), \quad m_i^N(t) \leq L_m^i(Nt), \quad t \geq 0. \quad (2)$$

See Appendix for the proof. The following proposition gives that, under some condition on the parameters, $(m_i^N(t))$ at scale Nt is negligible compared to N for every starting state of the system. Its limit will be specified by Theorem 3.1.

PROPOSITION 3.1 (MAGNITUDE OF AVAILABLE SLOTS). Assume $\beta C < \alpha(w - C)$ and let $1 \leq i \leq N$ be fixed. The process $(m_i^N(Nt)/N)$ converges in distribution to (0).

PROOF. It is sufficient to construct the previous coupling of Lemma 3.1 and to apply Lemma A.1. $\square$

3.2 Mean-field analysis

Let us denote by $(\Lambda^N(t))$ the empirical measure process of the state process $(m^N(t), V^N(t))$, i.e.

$$\Lambda^N(t) = \frac{1}{N} \sum_{i=1}^N \delta_{(m_i^N(t), V_i^N(t))}. \quad (3)$$

The objective is to analyse the behavior of the state process through the evolution equation (21) of the empirical measure process, written in the Appendix. From the transition rates, using assumption (1), which implies easily that $P_i(v)$ is of order $1/N$, it holds that, for any i , process $(m_i^N(t))$ is fast with respect to the scaling parameter N compared to the process $(V_i^N(t))$. Therefore, intuitively, for large values of N , the indicator function $\mathbb{I}_{\{m_i^N(t) > 0\}}$ behaves like the stationary probability that the number of available slots is non-zero. This reflects the stochastic averaging principle. Based on this limiting probability, independent of i by homogeneity, two regimes can be identified:

- If this limiting probability is equal to one, the system is said to be in the underloaded regime, and there is no saturation in the nodes.
- Otherwise, it is in the overloaded regime.

In the following, we study the overloaded regime, more interesting from a theoretical and a practical point of view.

DEFINITION 3.1 (OCCUPATION MEASURE). For a non-negative Borelian function g with compact support on $\mathbb{R}_+ \times \mathbb{N}^2$, the occupation measure μ_N is defined by

$$\langle \mu_N, g \rangle = \frac{1}{N} \sum_{i=1}^N \int_{\mathbb{R}_+} g(u, m_i^N(Nu), V_i^N(Nu)) du. \quad (4)$$

The next theorem is the main result of the paper. It describes the long-time behavior of a large system in the overloaded regime.

THEOREM 3.1. Assume $\beta C < \alpha(w - C)$. The sequence of occupation measures (μ_N) on state space $\mathbb{R}_+ \times \mathbb{N}^2$ defined by equation (4) is tight. Moreover, for the weak convergence, any limiting point μ_∞ satisfies

$$\langle \mu_\infty, g \rangle = \int_{\mathbb{R}_+} \mathbb{E}[g(u, X_1, X_2)] du \quad (5)$$

for any function g with finite support on $\mathbb{R}_+ \times \mathbb{N}^2$, where X_1 and X_2 are independent random variables such that

- X_1 has a geometric distribution on $\mathbb{N}$, with parameter $\beta C / \alpha(w - C)$,
- the distribution π_2 of X_2 is related to the choice of the load-balancing policy and is solution to the fixed point equation,

$$\pi_2(k) = \pi_2(0) \left(\frac{\mu \beta C}{\alpha(w - C)} \frac{s - \int_{\mathbb{N}} v \pi_2(dv)}{\int_{\mathbb{N}} p(v) \pi_2(dv)} \right)^k \prod_{i=1}^k \frac{p(i-1)}{\lambda(i)}, \quad (6)$$

for $k \in \mathbb{N}_+$.

To prove Theorem 3.1 we adapt the proof of [2, Theorem 2] to our framework. See Appendix for the key arguments.

REMARK 1. The limiting number of available slots per zone X_1 does not depend on the FF cars parameters. As in [2] without incentives, this is due to the negligible order of magnitude of FF cars compared to private cars.

4. LOAD-BALANCING POLICIES

In this section, we focus on two examples of policies where the limiting distribution π_2 (see Theorem 3.1) of the number of FF cars per zone can be computed explicitly. The two policies act at two times: at car drop-off and at car pick-up.

- **Repulsive routing (RR) policy:** it discourages users to park their FF car in zones with many FF cars;
- **Adaptive pick-up (AP) policy:** it adapts the pick-up rate in a zone to the number of FF cars parked there.

For $r \in \mathbb{N}^+$ and $\rho \in (0, 1)$, let us denote by $\pi_{r,\rho}$ the Negative Binomial distribution with parameters r and ρ , given, for $k \in \mathbb{N}$, by

$$\pi_{r,\rho}(k) = (1 - \rho)^r \rho^k \binom{k + r - 1}{k},$$

and let us introduce the following notation

$$A = \frac{\mu \beta C}{\lambda \alpha(w - C)}. \quad (7)$$

4.1 Repulsive routing (RR) policy

The following proposition gives the limiting distribution π_2 for a particular choice of routing function $p : \mathbb{N} \mapsto \mathbb{R}_+$, while the departure rate $\lambda(k) = \lambda > 0$ is fixed.

PROPOSITION 4.1. Let us assume $\beta C < \alpha(w - C)$ and $\lambda(k) = \lambda > 0$ for all $k \in \mathbb{N}$. For some parameter $r \in \mathbb{N}^+$, consider a routing mechanism with

$$p_r(k) = 1 + \frac{r-1}{k+1}, \quad k \in \mathbb{N}. \quad (8)$$

Then the distribution π_2 of the random variable X_2 in (5) is the negative binomial distribution with parameters r and $\rho(r)$, unique solution of equation

$$(1 - \rho)^{r+1} + \rho(1 - As - Ar) = 1 - As. \quad (9)$$

PROOF. Replacing $p(k) = p_r(k)$ and $\lambda(k) = \lambda$ in (6), the equilibrium distribution π_2 of the the number of FF cars is such that, for $k \in \mathbb{N}$,

$$\pi_2(k) \propto \rho^k \frac{(k+r-1)!}{k!}$$

with

$$\rho \stackrel{\text{def}}{=} A \frac{s - \int_{\mathbb{N}} v \pi_2(dv)}{\int_{\mathbb{N}} (1 + \frac{r-1}{v+1}) \pi_2(dv)}, \quad (10)$$

thus π_2 is a negative binomial distribution with parameters r and ρ . With straightforward computations of its right-hand side as a function of ρ , equation (10) can be rewritten

$$1 = A \frac{s - r \frac{\rho}{1-\rho}}{1 - (1-\rho)^r}, \quad (11)$$

if $\rho \in (0, 1]$, which gives (9).

Let us denote by f , the function defined for every $\rho \in (0, 1]$, by

$$f(\rho) = (1-\rho)^{r+1} + \rho(1 - As - Ar) + As - 1.$$

Since f is continuous in $[0, 1]$, $f(0) = As > 0$ and $f(1) = -Ar < 0$, there exists at least one solution of the fixed-point equation (9). For $\rho \in (0, 1]$,

$$f''(\rho) = r(r+1)(1-\rho)^{r-1} > 0,$$

thus the solution of (9) is unique. $\square$

Impact of parameter r . In the routing probability (8), the term $(r-1)/(k+1)$ reflects some discouragement for users to join a zone, inversely proportional to the occupation of this zone by FF cars. The parameter r expresses the level of discouragement that the operator applies to users. When $r = 1$ there is no discouragement, the choice is uniform among the nodes. The level of discouragement increases with r . More precisely, when r increases, the distribution obtained by renormalizing $p_r(k)$ at all stations becomes more and more concentrated in zones with few or no FF cars. The following proposition states that, when r becomes large, the limiting distribution $\pi_{r,\rho(r)}$ converges to a Poisson distribution. This shows that the mass of the limiting invariant distribution $\pi_{r,\rho(r)}$ becomes increasingly concentrated around the mean (see Figure 2). Recall that the Negative Binomial distribution with parameters r and ρ is denoted by $\pi_{r,\rho}$.

PROPOSITION 4.2. *For $r = 1$, the routing is uniform and $\pi_{1,\rho(1)}$ is a geometric distribution with parameter*

$$\rho(1) = \frac{1}{2} \left(1 + As + A - \sqrt{(1 + As + A)^2 - 4As} \right). \quad (12)$$

For $r \rightarrow \infty$, $\pi_{r,\rho(r)} \xrightarrow{d} \text{Poisson}(\gamma)$, where $\gamma > 0$ is the unique solution to the fixed-point equation

$$\gamma = s - \frac{1 - e^{-\gamma}}{A}. \quad (13)$$

PROOF. For $r = 1$, the system is a Jackson network with $N + 1$ nodes. In this case, one gets a geometric distribution $\pi_{1,\rho(1)} = (1 - \rho(1))\rho(1)^k$ with explicit parameter $\rho(1)$ given by (12). In order to study the convergence of the limiting distribution $\pi_{r,\rho(r)}$ when r becomes large, let us consider the generating function of a random variable $X_{r,\rho(r)}$ with a

negative binomial distribution $\pi_{r,\rho(r)}$. It is given by

$$G_{X_{r,\rho(r)}}(u) = \left(\frac{1 - \rho(r)}{1 - u\rho(r)} \right)^r, \quad |u| < 1.$$

Note that $\lim_{r \rightarrow \infty} \rho(r) = 0$. Moreover and that $\lim_{r \rightarrow \infty} r\rho(r) = \gamma$ where γ is the unique solution to the fixed point equation (13), see Appendix for details. Then

$$\lim_{r \rightarrow \infty} G_{X_{r,\rho(r)}}(u) = e^{\gamma(u-1)}, \quad |u| < 1,$$

which is the generating function of a random variable with a Poisson distribution with parameter γ . It follows that $X_{r,\rho(r)}$ converges in distribution, when r becomes large, to a random variable X with a Poisson distribution with parameter γ (see Figure 2). $\square$

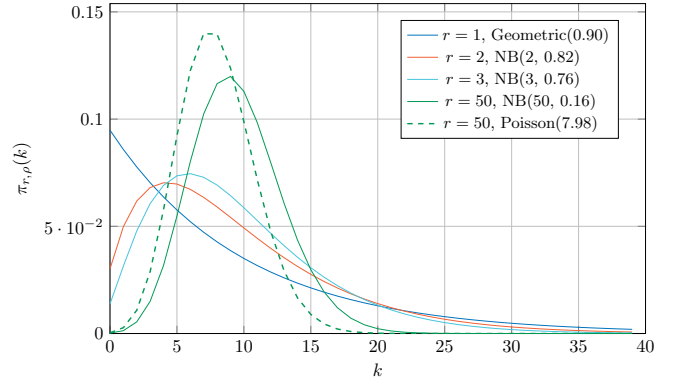


Figure 2: Plot, for the RR policy, of the Negative Binomial distribution with parameters r and $\rho(r)$. For large values of r , approximation by a $\text{Poisson}(r\rho(r))$ distribution. $A = 2$ and $s = 10$.

4.2 Adaptive pick-up (AP) policy

In this section, we study the limiting distribution π_2 for a particular choice of the pick-up rate function λ , while the routing function is $p(k) = 1$ for all $k \in \mathbb{N}$. The parameter ρ is again derived from a fixed point equation and has an explicit expression.

PROPOSITION 4.3. *Let us assume $\beta C < \alpha(w - C)$, $p(k) = 1$ for all $k \in \mathbb{N}$, and consider the following pick-up rates for class-1 customers. For $k \in \mathbb{N}$,*

$$\lambda_r(k) = \lambda \left(1 - \frac{r-1}{k+r-1} \right),$$

for some parameter $r \in \mathbb{N}^+$. Then the distribution π_2 of the random variable X_2 in (5) is a negative binomial distribution with parameters r and ρ where ρ is given by

$$\rho = \frac{1}{2} \left(As + Ar + 1 - \sqrt{(As + Ar + 1)^2 - 4As} \right). \quad (14)$$

PROOF. Replacing $p(k) = 1$ and $\lambda(k) = \lambda_r(k)$ in (6), the limiting distribution π_2 is given by

$$\pi_2(k) = \pi_2(0) A^k \left(s - \int_{\mathbb{N}} v \pi_2(dv) \right)^k \frac{(k+r-1)!}{k!(r-1)!}.$$

This is a negative binomial distribution with parameters r

and ρ with

$$\rho \stackrel{\text{def}}{=} A \left(s - \int_{\mathbb{N}} v \pi_2(dv) \right). \quad (15)$$

Writing the right-hand side of (15) in terms of ρ , the following fixed point equation is derived for $\rho \in (0, 1]$

$$\rho = A \left(s - r \frac{\rho}{1 - \rho} \right), \quad (16)$$

rewritten as a second-degree polynomial equation

$$\rho^2 - \rho(As + Ar + 1) + As = 0,$$

which has two distinct solutions $0 < \rho_- < 1 < \rho_+$. Therefore, there exists a unique value of $\rho \in (0, 1]$, with explicit expression (14), suitable to be the parameter of the negative binomial distribution π_2 . $\square$

REMARK 2. For $r = 1$, the transition rates are the same for both policies. Thus the limiting distribution $\pi_{1,\rho}$ is again a geometric distribution with explicit parameter $\rho = (1 + As + A - \sqrt{(1 + As + A)^2 - 4As})/2$. For the AP policy, studying the convergence of the distribution $\pi_{r,\rho(r)}$ when r becomes large makes no sense. In fact $\lim_{r \rightarrow \infty} \lambda_r(k) = 0$ for every $k \in \mathbb{N}$, thus the limiting pick-up rate of cars is 0.

5. PERFORMANCE EVALUATION

Users of FF car-sharing systems face the problem of finding available cars and parking spaces throughout the service area. While in the underloaded regime, the limiting proportion of zones without parking spaces is 0, in the overloaded regime the limiting proportion of zones without available parking spaces is $\beta C / \alpha(w - C)$ by Theorem 3.1, and depends only on the parameters of the environment (private cars and characteristics of the zone). Therefore, the car-sharing operator can only act on the limiting probability of finding available FF cars. The following corollary gives straightforwardly an explicit expression for this probability under the RR and the AP policies.

COROLLARY 5.1 (PERFORMANCE). *The limiting probability of failure, defined as the stationary probability of absence of FF cars at any zone, is given, for fixed r , by*

$$\pi_{r,\rho(r)}(0) = (1 - \rho(r))^r,$$

where $\rho(r) \in (0, 1]$ is the solution to the fixed point equation (9) for the RR policy, and (16) for the AP policy.

As a by-product, the following proposition states that the limiting probability of failure decreases with the fleet size.

PROPOSITION 5.1 (INFLUENCE OF FLEET SIZE). *The limiting probability of failure is a decreasing function of s .*

See Appendix for the proof. As a consequence, given a threshold, say ε , for the limiting probability of failure, one can compute the minimum fleet size s_ε that assures

$$\pi_{r,\rho}(0) \leq \varepsilon.$$

By simple calculations, the minimum fleet size s_ε is explicitly given by

$$s_\varepsilon(r) = \begin{cases} \frac{1-\varepsilon}{A} + r \frac{1-\varepsilon^{1/r}}{\varepsilon^{1/r}} & \text{for the RR policy} \\ \frac{1-\varepsilon^{1/r}}{A} + r \frac{1-\varepsilon^{1/r}}{\varepsilon^{1/r}} & \text{for the AP policy.} \end{cases}$$

Note that the difference between the two expressions of the minimum fleet size s_ε is of the order of $\varepsilon^{1/r}$, which is negligible, for very small thresholds ε in practice. The parameter r represents the strength of repulsion in the RR policy and that of the adaptivity in the AP policy. For both policies, for a fixed value of ε , the minimum fleet size s_ε is a decreasing function in r . See Figure 3. Figure 3 shows that both policies exhibit a significant performance improvement when increasing the parameter from $r = 1$ to $r = 2$, after which the gains become progressively smaller.

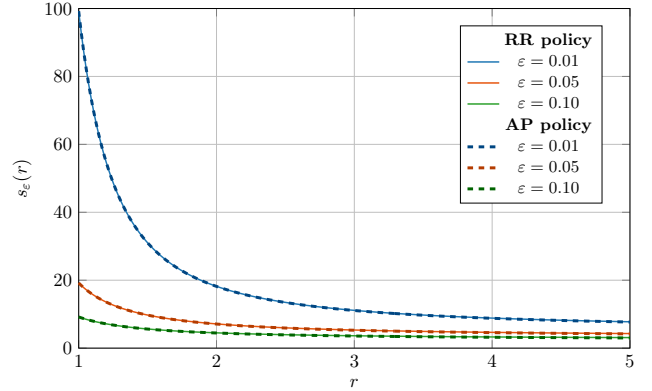


Figure 3: Plot of the minimum fleet size $s_\varepsilon = s_\varepsilon(r)$ that guarantees the failure probability to be smaller than ε , for different ε . Increasing the parameter r from $r = 1$ to $r = 2$, $s_{0.01}$ decreases from 100 to less than 20 for both the RR and the AP policies. $A = 5$. The two policies cannot be compared, since the total demand from users is not the same.

6. CONCLUSION AND FUTURE WORK

In this paper, we model some load-balancing policies for FF car-sharing systems assuming that the operator can act on the drop-off and pick-up rates of cars. We show that private cars behave as a fast random environment for FF cars. Thus private cars, in addition to the urban characteristics of the service area, determine if the system is in underloaded or overloaded regime.

Adapting mean-field techniques to this framework, we derive the limiting invariant distribution of the number of available spots and of FF cars in any zone of the service area. Explicit expressions for this distribution are obtained for two load-balancing policies: Repulsive Routing (RR) and Adaptive Pick-up (AP). In particular, for both policies, the number of FF cars in a given zone has a negative binomial distribution. We define the failure probability of a FF car-sharing system as the limiting proportion of zones without available FF cars. We show that the probability of failure is a decreasing function of the fleet size parameter s . Finally, we compute the minimum fleet size that guarantees that the probability of failure is less than some threshold, as a function of the strength of the incentive policy.

In future work, we will study the influence of incentive policies when the system is not homogeneous. In practice, a natural example is a two-cluster system where the service area is divided into two regions: a high-demand area and a low-demand area. Incentives need to be adapted to the inhomogeneous demand of FF cars.

ACKNOWLEDGEMENTS

This research benefited from the support of the FMJH Program Gaspard Monge for optimization and operations research and their interactions with data science.

7. REFERENCES

- [1] S. N. Ethier and T. G. Kurtz. *Markov processes: characterization and convergence*. John Wiley & Sons, 2009.
- [2] C. Fricker, H. Mohamed, and A. Rigonat. Stochastic averaging and mean-field for a large system with fast varying environment with applications to free-floating car-sharing. Accepted to *Queueing Systems*, 2025.
- [3] C. Fricker, H. Mohamed, A. Rigonat, and M. Trépanier. A new stochastic model for carsharing suited to free-floating. *Transportation Research Procedia*, 82:2395–2409, 2025.
- [4] T. G. Kurtz. Averaging for martingale problems and stochastic approximation. In *Applied Stochastic Analysis: Proceedings of a US-French Workshop, Rutgers University, New Brunswick, NJ, April 29–May 2, 1991*, pages 186–209. Springer, 2005.
- [5] K. Lippoldt, T. Niels, and K. Bogenberger. Effectiveness of different incentive models in free-floating carsharing systems: A case study in milan. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1179–1185. IEEE, 2018.
- [6] M. Mitzenmacher. The power of two choices in randomized load balancing. *IEEE transactions on parallel and distributed systems*, 12(10):1094–1104, 2002.
- [7] N. Neijmeijer, F. Schulte, K. Tierney, H. Polinder, and R. R. Negenborn. Dynamic pricing for user-based rebalancing in free-floating vehicle sharing: A real-world case. In *Computational Logistics: 11th International Conference, ICCL 2020, Enschede, The Netherlands, September 28–30, 2020, Proceedings 11*, pages 443–456. Springer, 2020.
- [8] F. Spitzer. Interaction of markov processes. In *Random Walks, Brownian Motion, and Interacting Particle Systems: A Festschrift in Honor of Frank Spitzer*, pages 66–110. Springer, 1991.
- [9] N. D. Vvedenskaya, R. L. Dobrushin, and F. I. Karpelevich. Queueing system with selection of the shortest of two queues: An asymptotic approach. *Problemy Peredachi Informatsii*, 32(1):20–34, 1996.

APPENDIX

A. PRELIMINARIES

Discussion on the orders of magnitude. Since the objective for the system is to have one free-floating car per zone, the scaling has been designed so that the fleet size per zone, which is the quantity of interest, is of unit order. We therefore choose the same order of magnitude for the total number of free-floating cars and the number of zones. The number of private cars per zone is proportional to the number of parking spaces per zone, this models well what happens in reality. The choice of having the number of parking spaces per zone (and, therefore, of private cars per zone) of order N is to be in line with Kelly’s scaling for loss networks. One could choose other orders of magnitude for this ratio.

LEMMA A.1 (STABLE $M/M/1$ QUEUE). *If $(L(t))$ is the number of customers in a $M/M/1$ queue with arrival and service rates respectively η and ν with $\eta < \nu$ then, if $L(0) = 0$, for all $\delta \geq 0$, the process*

$$\left(\frac{L(N^\delta t)}{\log^2 N} \right)$$

converges in distribution to (0). In particular, for the convergence in distribution, process

$$\left(\frac{L(Nt)}{N} \right)$$

converges to (0).

The proof is omitted. See [2].

B. PROOFS OF SECTION 3

PROOF OF LEMMA 3.1. We can construct a coupling $(m_i^N(t), L_m^i(Nt))$ where $(L_m^i(t))$ is the number of customers in a $M/M/1$ queue with arrival rate η and service rate ν and $L_m^i(0) = m_i^N(0)$, where η and ν are properly chosen such that $\eta < \nu$. If the system is in state (m, v) , the jumps and the respective rates of process $(m_i^N(t))$ are

$$\begin{cases} -1 & \left(\alpha N (w_N - C + \sum_{l=1}^N (m_l + v_l)/N) \right. \\ & \left. + \mu (M_N - \sum_{l=1}^N v_l) P_i(v) \right) \mathbb{1}_{m_i > 0}, \\ +1 & \beta (CN - m_i - v_i) + \lambda(v_i) \mathbb{1}_{v_i > 0}. \end{cases}$$

The rate for jump -1 is larger than $\alpha N(w_N - C) \mathbb{1}_{m_i > 0}$ and the rate for jump $+1$ is smaller than $\beta CN + \lambda(v_i)$. Recall that the function λ is bounded, thus for any fixed $\beta_0 > \beta$, $w_0 < w$ and N large enough, $\beta CN + \lambda(v_i) < \beta_0 CN$ and $\alpha N(w_N - C) \mathbb{1}_{m_i > 0} > \alpha N(w_0 - C) \mathbb{1}_{m_i > 0}$. Choosing therefore $\eta = \beta_0 C$ and $\nu = \alpha(w_0 - C)$, $(L_m^i(t))$ is the birth and death process with transitions

$$k \rightarrow k + \begin{cases} -1 & \alpha(w_0 - C) \mathbb{1}_{\{k > 0\}} \\ +1 & \beta_0 C \end{cases}$$

thus equation (2) holds for N large enough. $\square$

The following lemma is part of the proof of Theorem 3.1.

LEMMA B.1. Assume $\beta C < \alpha(w - C)$. The sequence of occupation measures (μ_N) on state space $\mathbb{R}_+ \times \mathbb{N}^2$ is tight and, for the weak convergence of measures, any limiting point μ_∞ can be expressed as

$$\langle \mu_\infty, g \rangle = \int_{\mathbb{R}_+ \times \mathbb{N}^2} g(u, m, v) \pi_u(dm, dv) du \quad (17)$$

for any non-negative Borelian function g on $\mathbb{R}_+ \times \mathbb{N}^2$, where π_u is a probability measure on $\mathbb{N}^2$.

PROOF OF LEMMA B.1. First, let us prove that for fixed $\varepsilon > 0$, there exists $K_0 > 0$ such that for $K \geq K_0$, $K \in \mathbb{N}$, for every $N \in \mathbb{N}$ and $t \geq 0$,

$$\mathbb{E}(\mu_N([0, t] \times \mathbb{N} \times [K, +\infty])) < \varepsilon t / 2. \quad (18)$$

Using Markov inequality, by definition (4) of μ_N , it holds that

$$\begin{aligned} \mathbb{E}[\mu_N([0, t] \times \mathbb{N} \times [K, +\infty])] &= \frac{1}{N} \sum_{i=1}^N \int_0^t \mathbb{P}(V_i^N(Nu) \geq K) du \\ &\leq \frac{1}{N} \sum_{i=1}^N \int_0^t \mathbb{E} \left(\frac{V_i^N(Nu)}{K} \right) du \\ &\leq \frac{1}{N} \int_0^t \frac{M_N}{K} du = \frac{tM_N}{NK}. \end{aligned}$$

Using that $\lim_{N \rightarrow \infty} M_N/N = s$, we get

$$\limsup_{N \rightarrow \infty} \mathbb{E}[\mu_N([0, t] \times \mathbb{N} \times [K, +\infty])] \leq \frac{ts}{K}.$$

Thus, there exists K_1 and N_0 such that, for $N \geq N_0$ and $K \geq K_1$, equation (18) holds. One can then choose K_2 such that, for $N < N_0$ and $K \geq K_2$, equation (18) holds. Therefore equation (18) holds for $K \geq K_0 = \sup(K_1, K_2)$ and uniformly in N .

Second, let us prove that for fixed $\varepsilon > 0$, there exists $\tilde{K}_0$ such that for any $K \geq \tilde{K}_0$, $K \in \mathbb{N}$, for any $N \in \mathbb{N}$ and $t \geq 0$,

$$\mathbb{E}(\mu_N([0, t] \times [K, +\infty] \times \mathbb{N})) < \varepsilon t / 2. \quad (19)$$

Using the coupling $(m_i(t), L_m^i(Nt))$ of Lemma 3.1, with $L_m^i(Nt)$ the number of customers in a $M/M/1$ with arrival rate $\beta_0 C$ and service rate $\alpha(w_0 - C)$ for all $\beta_0 > \beta$ and $w_0 < w$, relation $m_i^N(Nt) \leq L_m^i(N^2 t)$ holds for all $t \geq 0$ and N large enough. This yields

$$\begin{aligned} \mathbb{E}[\mu_N([0, t] \times [K, +\infty] \times \mathbb{N})] &= \frac{1}{N} \sum_{i=1}^N \int_0^t \mathbb{P}(m_i^N(Nu) \geq K) du \\ &\leq \frac{1}{N} \sum_{i=1}^N \int_0^t \mathbb{P}(L_m^i(N^2 u) \geq K) du \\ &\leq \frac{1}{N} \sum_{i=1}^N \frac{1}{N^2} \int_0^{N^2 t} \mathbb{P}(L_m^i(u) \geq K) du \end{aligned}$$

and, by ergodic theorem,

$$\limsup_{N \rightarrow \infty} \mathbb{E}[\mu_N([0, t] \times [K, +\infty] \times \mathbb{N})] \leq \left(\frac{\beta_0 C}{\alpha(w_0 - C)} \right)^K t$$

for all $\beta_0 > \beta$ and $w_0 < w$. Choosing β_0 and w_0 such that $\beta_0 C < \alpha(w_0 - C)$, it is thus sufficient to choose K large enough to obtain (19) for N sufficiently large. It is then easy to check that (19) holds uniformly in N , choosing K sufficiently large.

From equations (18) and (19), for $t \geq 0$ and $\varepsilon > 0$ fixed, there exist $\bar{K}_0$ such that for any $K \geq \bar{K}_0$, $K \in \mathbb{N}$,

$$\inf_N \mathbb{E}(\mu_N([0, t] \times [0, K]^2)) \geq (1 - \varepsilon)t.$$

Thus, since it is tight, the sequence (μ_N) is relatively compact, due to [4, Lemma 1.3], which is a consequence of Prohorov's theorem (see also [1, Theorem 2.2]). Moreover [4, Lemma 1.4] gives that any limiting point μ_∞ of this sequence can be represented as in equation (17). $\square$

PROOF OF THEOREM 3.1. We just give a sketch of the proof. See analogous details in the proof of [2, Theorem 2]. The main idea is that an averaging principle holds with a fast process and a slow process. The general method for this is described in Kurtz [4].

From the definition of the empirical measure (3), for any function f with finite support on $\mathbb{N}^2$ and $t \geq 0$,

$$\Lambda^N(t)(f) = \frac{1}{N} \sum_{i=1}^N f(m_i^N(t), V_i^N(t)),$$

used in the following in this form

$$\Lambda^N(t)(f) = \int_{\mathbb{N}^2} f(m, v) \Lambda^N(t)(dm, dv).$$

Let us introduce, for $(m, v) \in \mathbb{N}^{2N}$ and $(i, j) \in \mathbb{N}^2$, the following notation

$$\Delta_{i,j}(f)(m, v) = f(m + i, v + j) - f(m, v). \quad (20)$$

The functional evolution equation for the empirical measure process is

$$\Lambda^N(t)(f) = \Lambda^N(0)(f) \quad (21)$$

$$\begin{aligned} &+ \int_0^t \int_{\mathbb{N}^2} \Delta_{(1,-1)}(f)(m, v) \lambda(v) \mathbb{1}_{\{v>0\}} \Lambda^N(s)(dm, dv) ds \\ &+ \mu \int_0^t \frac{\frac{M_N}{N} - \int_{\mathbb{N}^2} v \Lambda^N(s)(dm, dv)}{\int_{\mathbb{N}^2} p(v) \Lambda^N(s)(dm, dv)} \\ &\quad \int_{\mathbb{N}^2} p(v) \Delta_{(-1,1)}(f)(m, v) \mathbb{1}_{\{m>0\}} \Lambda^N(s)(dm, dv) ds \\ &+ \beta C N \int_0^t \int_{\mathbb{N}^2} \Delta_{(1,0)}(f)(m, v) \Lambda^N(s)(dm, dv) ds \\ &- \beta \int_0^t \int_{\mathbb{N}^2} (m + v) \Delta_{(1,0)}(f)(m, v) \Lambda^N(s)(dm, dv) ds \\ &+ \alpha N (w_N - C) \\ &\quad \int_0^t \int_{\mathbb{N}^2} \Delta_{(-1,0)}(f)(m, v) \mathbb{1}_{\{m>0\}} \Lambda^N(s)(dm, dv) ds \\ &+ \alpha \int_{\mathbb{N}^2} (m + v) \Lambda^N(s)(dm, dv) \\ &\quad \int_0^t \int_{\mathbb{N}^2} \Delta_{(1,0)}(f)(m, v) \Lambda^N(s)(dm, dv) ds \\ &+ \frac{1}{N} \sum_{i=1}^N M_{i,f}^N(t) \end{aligned}$$

where $(M_{i,f}^N(t))$ are square integrable martingales.

Let us choose $f(m, v) = h(m)k(v)$, replace t by Nt and divide by N^2 in equation (21). It is easy to check that the

martingale term

$$\left(\frac{M_f^N(Nt)}{N^2} \right) = \left(\frac{1}{N^3} \sum_{i=1}^N M_{i,f}^N(Nt) \right)$$

converges to (0) as $N \rightarrow \infty$, together with some of the integral terms. For the remaining terms, Lemma B.1 gives that, when $N \rightarrow \infty$,

$$0 = \int_0^t \int_{\mathbb{N}^2} k(v) \left(\beta C \Delta_1(h)(m) + \alpha(w - C) \Delta_{-1}(h)(m) \mathbb{1}_{m>0} \right) \pi_u(dm, dv) du,$$

for all $t \geq 0$. Note that assumption (1) is used to prove that the third term in the right-hand side of (21) converges to 0. Let $\pi_t(dm|v)$ be the conditional distribution of m given v , and $\pi_{2,t}(dv)$ the marginal distribution of v . For almost all $t \geq 0$,

$$\int_{\mathbb{N}} (\beta C \Delta_1(h)(m) + \alpha(w - C) \Delta_{-1}(h)(m) \mathbb{1}_{m>0}) \pi_t(dm|v) = 0,$$

$\pi_{2,t}(dv)$ almost surely for all functions h with finite support on $\mathbb{N}$. We recognize the integral of the infinitesimal generator of a $M/M/1$ queue with arrival rate βC and service rate $\alpha(w - C)$ with $\beta C < \alpha(w - C)$, therefore $\pi_t(dm|v)$ is a geometric distribution with parameter $\beta C / \alpha(w - C)$ for all $v \geq 0$. In particular $\pi_t(dm|v)$ is independent of v , thus $\pi_t(dm|v) = \pi_{1,t}(dm)$ and

$$\int_{\mathbb{N}} \mathbb{1}_{m>0} \pi_t(dm|dv) = 1 - \pi_{1,t}(0) = \frac{\beta C}{\alpha(w - C)}. \quad (22)$$

Choosing then $f(m, v) = k(v)$, replacing t by Nt in (21), and dividing it by N , it holds that

$$\begin{aligned} \frac{\Lambda^N(Nt)(f)}{N} &= \frac{\Lambda^N(0)(f)}{N} \\ &+ \int_0^t \int_{\mathbb{N}^2} \Delta_{-1}(k)(v) \mathbb{1}_{\{v>0\}} \lambda(v) \Lambda^N(Nu)(dm, dv) du \\ &+ \mu \int_0^t \frac{\frac{M_N}{N} - \int_{\mathbb{N}^2} v \Lambda^N(u)(dm, dv)}{\int_{\mathbb{N}^2} p(v) \Lambda^N(u)(dm, dv)} \\ &\quad \int_{\mathbb{N}^2} \Delta_1(k)(v) \mathbb{1}_{\{m>0\}} p(v) \Lambda^N(Nu)(dm, dv) du \\ &+ \frac{1}{N^2} \sum_{i=1}^N M_{i,k}^N(Nt). \end{aligned} \quad (23)$$

Letting $N \rightarrow \infty$, the martingale term and the left-hand side of equation (23) converge to 0. Then Lemma B.1 gives that

$$0 = \int_0^t \int_{\mathbb{N}^2} \left(\lambda(v) \Delta_{-1}(k)(v) \mathbb{1}_{v>0} + \mu \frac{s - \int_{\mathbb{N}} v \pi_2(dv)}{\int_{\mathbb{N}} p(v) \pi_2(dv)} \Delta_1(k)(v) \mathbb{1}_{m>0} p(v) \right) \pi_u(dm, dv) du.$$

Conditioning by m and using (22), we get

$$0 = \int_0^t \int_{\mathbb{N}} \left(\lambda(v) \Delta_{-1}(k)(v) \mathbb{1}_{v>0} + \mu \frac{\beta C}{\alpha(w - C)} \frac{s - \int_{\mathbb{N}} v \pi_2(dv)}{\int_{\mathbb{N}} p(v) \pi_2(dv)} \Delta_1(k)(v) p(v) \right) \pi_{2,u}(dv) du.$$

Therefore, for almost all $t \geq 0$,

$$0 = \int_{\mathbb{N}} \left(\lambda(v) \Delta_{-1}(k)(v) \mathbb{1}_{v>0} + \frac{\mu \beta C}{\alpha(w - C)} \frac{s - \int_{\mathbb{N}} v \pi_2(dv)}{\int_{\mathbb{N}} p(v) \pi_2(dv)} \Delta_1(k)(v) p(v) \right) \pi_2(dv). \quad (24)$$

In the right-hand side of equation (24) is the generator of a birth-and-death process that, when in state $n \in \mathbb{N}$, has birth rate

$$\mu \frac{\beta C}{\alpha(w - C)} \frac{s - \int_{\mathbb{N}} v \pi_2(dv)}{\int_{\mathbb{N}} p(v) \pi_2(dv)} p(n),$$

and death rate $\lambda(n)$.

The marginal distribution π_2 has therefore the following expression. For $k \in \mathbb{N}^+$

$$\pi_2(k) = \pi_2(0) \prod_{i=1}^k \left(\mu \frac{\beta C}{\alpha(w - C)} \frac{s - \int_{\mathbb{N}} v \pi_2(dv)}{\int_{\mathbb{N}} p(v) \pi_2(dv)} \frac{p(i-1)}{\lambda(i)} \right)$$

which gives straightforwardly (6). The existence of such π_2 is guaranteed by lemma B.1.

□

C. PROOF OF SECTION 4

PROOF OF PROPOSITION 4.2. Let us detail in this proof that $\lim_{r \rightarrow \infty} r\rho(r) = \gamma$. Using (11), we obtain straightforwardly that $r\rho(r) < s$. Thus, for any sequence $(r_n)_n$ converging to infinity, $(r_n\rho(r_n))_n$ is bounded and therefore admits at least one limiting point. Then, rewriting the second term of the left-hand side of (9) as $r\rho(1/r - As/r - A)$, any limiting point satisfies (13). And $\gamma \mapsto 1 - \exp(-\gamma) - A\gamma$ is continuous and non-decreasing from 0 to $+\infty$. Therefore, for fixed $s > 0$, there is a unique γ solution of (13). □

D. PROOF OF SECTION 5

PROOF OF PROPOSITION 5.1. Let us denote by $\varphi(\rho) = \pi_{r,\rho(r)}(0) = (1 - \rho)^r$ the limiting probability of failure as a function of ρ . Then $\pi_{r,\rho(r)}(0)$ as a function of s is given by $\psi = \varphi \circ s^{-1}$ and its derivative by $\psi' = (\varphi' \circ s^{-1}) / (s' \circ s^{-1})$. Differentiating function φ with respect to ρ gives $\varphi'(\rho) = -r(1 - \rho)^{r-1} < 0$. For the RR policy, for every $\rho \in [0, 1]$, the fleet size as a function of ρ is given by

$$s(\rho) = \frac{1}{A(1 - \rho)} (1 - (1 - \rho)^{r+1} - \rho(1 - Ar)),$$

and its derivative by

$$s'(\rho) = \frac{r(A + (1 - \rho)^{r+1})}{A(1 - \rho)^2} > 0.$$

For the AP policy, for every $\rho \in [0, 1]$, $s(\rho)$ is given by

$$s(\rho) = \frac{\rho}{A} + r \frac{\rho}{1 - \rho}$$

and

$$s'(\rho) = \frac{1}{A} + \frac{r}{(1 - \rho)^2} > 0.$$

Therefore, for both policies, $\psi' < 0$ and we can conclude that $\pi_{r,\rho(r)}(0)$ is a decreasing function of s . □

On the geometry of the stability regions of randomly modulated queueing systems

N. Soprano-Loto^a, U. Ayesta^{b,c,d,e} and I. M. Verloop^{b,c}

^aINRIA, Centre Inria de Paris, 48 rue Barrault, 75013 Paris, France

^bIRIT-CNRS, 2 rue C. Camichel, 31071 Toulouse, France

^cUniversité de Toulouse, INP, 31071 Toulouse, France

^dIKERBASQUE - Basque Foundation for Science, 48011 Bilbao, Spain

^eUPV/EHU, Univ. of the Basque Country, 20018 Donostia, Spain

ABSTRACT

We study the maximum stability region (MSR) of a scheduling problem involving multiple queues, a single server, and randomly modulated dynamics. In the case where the modulation process is autonomous, takes values in a finite set, and is in stationary regime, we characterise the stability region as a Minkowski sum of deGua simplices, structures known as *cephoids* in the convex geometry literature. Beyond endowing the stability region with a rich mathematical structure, this apparently novel connection enables an explicit description of the MSR in the 2-queue case, and provides a simple iterative scheme to obtain its minimal H-description in the general case.

Keywords

Queueing systems, scheduling, random modulations, stability, Minkowski sum, deGua simplices, cepheids

1. INTRODUCTION

Multi-queue scheduling systems in which service capacities fluctuate according to an underlying random environment arise naturally in applications ranging from wireless networks to manufacturing lines. In such systems, characterizing the set of arrival-rate vectors that can be sustained (the stability region) is a fundamental problem in performance evaluation. Existing approaches primarily focus on “throughput-optimal” policies, but they often fall short of providing an explicit characterisation of the maximal stability region (MSR) in terms of the input parameters.

In this work, we unveil a novel connection between queueing systems and convex geometry. Specifically, when the environment process has a finite state-space and is in stationary regime, the MSR can be shown to coincide with the Minkowski sum of certain simplices (called deGua simplices) whose parameters encode the average service rates in each environment. These Minkowski sums, known as *cephoids* in the convex geometry literature, inherit a rich combinatorial and geometric structure that we exploit to derive: (i) an explicit description of the MSR as the solution set of a finite system of non-redundant linear inequalities – the so-

called minimal H-description – in 2-queue systems; and (ii) a simple iterative scheme to derive the same type of minimal description in higher dimensions (which in particular provides a universal and explicit formula for the number of facets of the MSR in terms of the number of queues and environments).

1.1 Bibliographic review

We review the literature spanning the two fields that motivate this paper: queueing systems and convex geometry.

1.1.1 Stability of queueing systems

The seminal paper [1] considers a slotted system in which the service rate alternates between 0 and a positive value (the so-called ON/OFF setting), establishing necessary and sufficient conditions for stability, and showing that a Max Weight policy is throughput optimal. For an equivalent continuous-time system, a Loynes’ construction was employed in [2], and, more recently, the effect of decision epochs on decision making has also been studied in [3]. More general environments are considered in [4, 5], where throughput-optimal policies are analyzed and the MSR is characterized in terms of so-called static policies. We finally refer to [6], which investigates the H-description of the stability region in multi-queue, multi-server queueing systems.

1.1.2 Cepheids

Despite many existing theoretical developments, a general description of sums of polytopes is incomplete due to its complexity. For a comprehensive overview of the existing literature, which includes also combinatorial and algorithmic perspectives, we refer to [7] and the references therein. When the involved polytopes are deGua simplices, the resulting Minkowski sums are called *cephoids*¹. A deeper understanding is available in this more restrictive case, much of which is presented in Rosenmüller’s recent monograph [8], which builds on a series of papers coauthored with Palaschke.

2. THE BASIC MODEL

We introduce the model that serves as a basis for our analysis. Consider a system of K queues, with queue indices in

¹Originally, the term cepheid referred to a ‘cephalopodic’ structure, but the name was later adapted for convenience; see the introduction of [8] for a related discussion.

the set $\mathcal{K} = \{1, \dots, K\}$. Each queue $k \in \mathcal{K}$ has its specific arrival rate $\lambda_k \in [0, \infty)$. By simplicity, we assume that the arrival process at queue k is ruled by an autonomous i.i.d. process $(A_k(t))_{t \in \mathbb{N}_0}$ consisting of $\mathbb{N}_0$ -valued random variables satisfying $\mathbb{E}[A_k(t)] = \lambda_k$ and $\text{Var}[A_k(t)] < \infty$. The system is influenced by a stochastic environmental process $(Y(t))_{t \in \mathbb{N}_0}$ taking values in the finite set $\mathcal{M} = \{1, \dots, M\}$. We assume that this process evolves (also autonomously) as an irreducible discrete-time Markov chain with invariant distribution ν . Each environment $m \in \mathcal{M}$ defines a vector $\mu^m \in [0, \infty)^K$, where, for each $k \in \mathcal{K}$, μ_k^m denotes the potential service rate in queue k under environment m . For each k , we construct a potential service process $(D_k(t))_{t \in \mathbb{N}_0}$, consisting of $\mathbb{N}_0$ -valued random variables, as a function of $Y(t)$ and an exogenous source of randomness. This process satisfies $\mathbb{E}[D_k(t) \mid Y(t) = m] = \mu_k^m$, and its variance is assumed to be uniformly bounded. We further assume that all the involved processes are mutually independent for different queue indexes.

In each time-slot, a single server chooses one queue to serve according to a policy, i.e., a rule specifying how the decision is made. All policies discussed herein are Markovian, so we will refer to them simply as policies. The formal definition, which encompasses random policies, follows.

DEFINITION 1 (POLICY). A (Markovian) policy is a kernel $\alpha : \mathcal{Q} \times \mathcal{M} \rightarrow \mathcal{K}$. In other words, for every (q, m) , $\alpha_{q,m}$ is a probability on $\mathcal{K}$.

The ‘queue-lengths’ process $(Q(t))_{t \in \mathbb{N}_0}$, which takes values in the state space $\mathcal{Q} = \mathbb{N}_0^K$, will be iteratively constructed as a function of the objects already introduced. Define $Q(0)$ arbitrarily. Assume that $Q(t)$ has been defined. Let $Z(t)$ be the queue index selected by the policy, precisely defined as a $\mathcal{K}$ -valued random variable drawn with conditional law $\alpha_{Q(t), Y(t)}$ independently of everything else. For each k , the iterative queue-lengths construction follows:

$$Q_k(t+1) = [Q_k(t) + A_k(t) - D_k(t) \mathbb{1}\{Z(t) = k\}]^+.$$

3. STABILITY NOTIONS

The process $((Q(t), Y(t)))_{t \in \mathbb{N}_0}$ is a Markov chain, which takes values in the countable state-space $\mathcal{Q} \times \mathcal{M}$.

DEFINITION 2 (STABILITY). A Markov chain is said to be stable if it is irreducible and aperiodic, and the return time to any state has finite mean. The stability region of a policy α is defined as

$$\mathcal{S}_\alpha = \left\{ \lambda \in [0, \infty)^K : \begin{array}{l} \text{the Markov chain associated} \\ \text{to } (\lambda, \alpha) \text{ is stable} \end{array} \right\}.$$

The maximum stability region (MSR) is defined as the set of arrival rates for which there exists a stable policy. Precisely,

$$\text{MSR} = \bigcup_{\alpha \text{ policy}} \mathcal{S}_\alpha.$$

We will see in Theorem 4 below, which is a restatement of [5, Theorem 1], that the notion of MSR is closely related to the following sub-family of policies.

DEFINITION 3 (STATIC POLICIES). A policy α is said to be static if it depends only on the environmental state and not on the queue lengths, that is, $\alpha_{q,m} = \alpha_{q',m}$ for all $m \in \mathcal{M}$ and $q, q' \in \mathcal{Q}$. We will write $\phi_m = \alpha_{q,m}$ to emphasize such independence.

THEOREM 4 ([4, THEOREM 3.1]). The following characterisations for the MSR holds:

$$\begin{aligned} \overline{\text{MSR}} &= \overline{\bigcup_{\phi \text{ static}} \mathcal{S}_\phi} \\ &= \left\{ \lambda \in [0, \infty)^K : \lambda_k \leq \sum_{m \in \mathcal{M}} \nu(m) \phi_m(k) \mu_k^m \right\}. \end{aligned} \quad (1)$$

In the previous statement, the bars denote topological closures, which avoids complications from critical cases. The first identity shows that static policies suffice to generate the MSR. Since $\sum_{m \in \mathcal{M}} \nu(m) \phi_m(k) \mu_k^m$ is the average service rate of queue k under policy ϕ , the condition in (1) provides a per-queue arrival-service rate characterisation. Static policies are theoretically relevant, as they allow the abstract description of the MSR given in the previous theorem, but they lack practical applicability. For optimal policies, see [4, 5].

4. DEGUA SIMPLICES AND CEPHOIDS

The notion of cepheid stands on the notions of Minkowski sum and deGua simplex, defined as follows:

DEFINITION 5 (MINKOWSKI SUM). The Minkowski sum of two subsets $A, B \subseteq \mathbb{R}^K$ is defined as

$$A + B = \{x + y : x \in A, y \in B\}.$$

This operation is associative and commutative, hence its definition extends to the sum of a finite number of subsets.

DEFINITION 6 (DEGUA SIMPLEX). A K -dimensional deGua simplex is characterised by a vector $u \in [0, \infty)^K$. It is defined as

$$\Gamma_u = \text{conv}\{0, u_1 e^1, \dots, u_K e^K\},$$

where $\{e^1, \dots, e^K\}$ is the canonical basis of $\mathbb{R}^K$ and conv stands for the convex hull.

A useful characterisation of a deGua simplex is the following one: $\lambda \in \Gamma_u$ if and only if $\lambda \leq u \odot h$ for some K -dimensional stochastic vector h . Here the inequality is coordinate-wise, the symbol $\odot$ denotes the Hadamard (or coordinate-wise) product, and a vector is called stochastic if its entries are non-negative and sum to 1.

Once the notions of Minkowski sum and deGua simplex have been introduced, the notion of cepheid is easily defined.

DEFINITION 7 (CEPHOID). A cepheid is the Minkowski sum of finitely many deGua simplices. The cepheid formed from the deGua simplices $(\Gamma_{u^m})_{m \in \mathcal{M}}$ will be denoted by $\sum_{m \in \mathcal{M}} \Gamma_{u^m}$.

The aforementioned characterisation for deGua simplices immediately results into the following one for cepheids: $\lambda \in \sum_{m \in \mathcal{M}} \Gamma_{u^m}$ if and only if there exist K -dimensional stochastic vectors $(h^m)_{m \in \mathcal{M}}$ such that

$$\lambda \leq \sum_{m \in \mathcal{M}} u^m \odot h^m. \quad (2)$$

To avoid trivialities, we will assume in the remainder that all the involved cepheids satisfy the following condition:

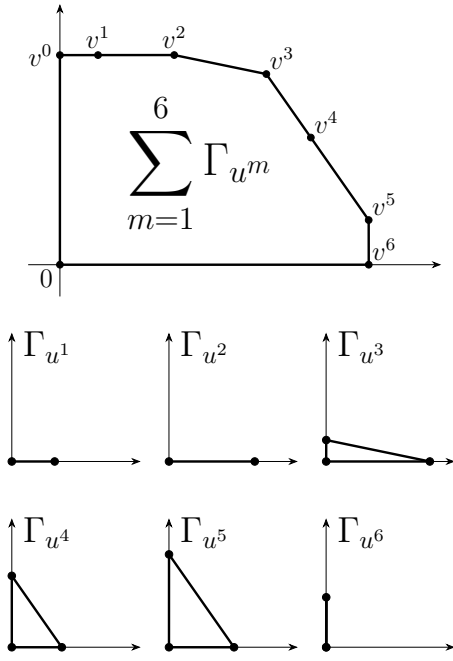


Figure 1: 2-dimensional cepheid formed by 6 deGua simplices.

$u^m \neq 0$ for every $m \in \mathcal{M}$, and, for every $k \in \mathcal{K}$, there is at least one $m \in \mathcal{M}$ for which $u_k^m > 0$.

Before stating Theorem 8, our main result, we illustrate some low-dimensional examples of cepheids. The general structure of cepheids in higher dimensions is deferred to Section 6.

2-dimensional cepheids. We present a procedure for summing finitely many 2-dimensional deGua simplices $(\Gamma_{u^m})_{m \in \mathcal{M}}$. For further details, we refer the reader to [8, Chapter 1]. The slope of the m -th deGua simplex is defined as $s_m = -\frac{u_2^m}{u_1^m} \in [-\infty, 0]$. Without loss of generality, we assume that the elements of $\mathcal{M}$ are ordered in such a way that $s_m \geq s_{m+1}$ for every $1 \leq m < M$. We define the set of points $v^0, v^1, \dots, v^M$ as follows:

$$\begin{aligned} v_1^0 &= 0, v_2^0 = \sum_{m \in \mathcal{M}} u_2^m \\ v^m &= v^{m-1} + (u_1^m, -u_2^m), \quad m \in \mathcal{M}. \end{aligned}$$

The polygonal contour $v^0 - v^1 - \dots - v^M - 0 - v^0$ determines the border of the cepheid. See Figure 1 for an example of a 2-dimensional cepheid formed by 6 deGua simplices.

3-dimensional cepheids. In [9], some analytical expressions have been derived for the 3-dimensional case that allow rapid implementation, resulting in the outcome shown in Figure 2.

5. THE KEY CONNECTION: STABILITY REGIONS ARE CEPHOIDS

The next result states that the MSR of queueing mod-

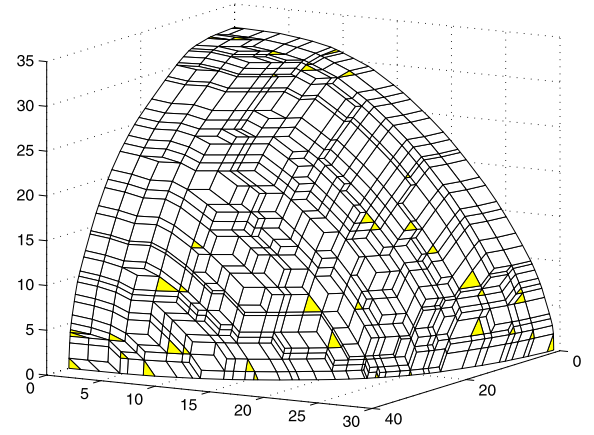


Figure 2: 3-dimensional cepheid formed from 40 deGua simplices, using the MATLAB implementation developed in [9].

els under random modulations, as described in Section 2, admits a cepheid description.

THEOREM 8 (MSRs ARE CEPHOIDS). *The topological closure of the MSR of the model described in Section 2, is a cepheid with associated deGua simplices explicitly defined in terms of the statistical properties of the modulation process. Precisely,*

$$\overline{\text{MSR}} = \sum_{m \in \mathcal{M}} \Gamma_{\nu(m)\mu^m}. \quad (3)$$

In the previous statement, the vectors $(u^m)_{m \in \mathcal{M}}$ used as bases for cepheids are identified with $(\nu(m)\mu^m)_{m \in \mathcal{M}}$, thereby translating the definitions of cepheids into the framework of queueing systems.

PROOF OF THEOREM 8. We saw that λ belongs to the r.h.s. of (3) if and only if inequality (2) holds for a family of stochastic vectors $(h^m)_{m \in \mathcal{M}}$ (taking, as mentioned, $u^m = \nu(m)\mu^m$, $m \in \mathcal{M}$). Under identification $\phi_m = h^m$, $m \in \mathcal{M}$, inequality (2) becomes a restatement of the inequalities in (1), and the result follows by Theorem 4. $\square$

6. STABILITY IMPLICATIONS OF THE CEPHOID IDENTIFICATION

Identifying the mathematical structure of the MSR allows us to obtain the following three results: a readily verifiable necessary condition for stability, the explicit characterisation of the MSR in the 2-dimensional case, and an iterative scheme for describing the MSR in higher dimensions.

6.1 Necessary condition

The proof of the following necessary condition will appear in the full version of the paper.

COROLLARY 9. *For λ to be in $\overline{\text{MSR}}$ it is necessary that*

$$\langle \lambda, w \rangle \leq \sum_{m \in \mathcal{M}} \nu(m) \max_{k \in \mathcal{K}} \mu_k^m w_k \quad \forall w \in \mathbb{R}^K.$$

In particular, if μ^m has all its entries strictly positive for every $m \in \mathcal{M}$, $\lambda \in \overline{\text{MSR}}$ implies

$$\sum_{k \in \mathcal{K}} \frac{\lambda_k}{\mu_k^m} \leq \sum_{m' \in \mathcal{M}} \nu(m') \max_{k \in \mathcal{K}} \frac{\mu_k^{m'}}{\mu_k^m}, \quad m \in \mathcal{M}. \quad (4)$$

Observe that the inequalities in (4) can be interpreted as per-environment bounds on the loads.

6.2 2-dimensional MSR description

As already mentioned, the H-description of a polytope refers to its representation as the solution set of a finite system of linear inequalities. The cepheid identification provides the following H-description of the MSR for $K = 2$. Its proof, which stands on the description of 2-dimensional cepheids given in Section 4, will be available in the full version of the paper.

COROLLARY 10. *Consider the case $K = 2$. For each $m \in \mathcal{M}$, let n^m be the unit external normal vector defined as*

$$n^m = \begin{cases} \left| \left(\frac{1}{\mu_1^m}, \frac{1}{\mu_2^m} \right) \right|^{-1} \left(\frac{1}{\mu_1^m}, \frac{1}{\mu_2^m} \right) & \text{if } \mu_1^m, \mu_2^m > 0 \\ e^1 & \text{if } \mu_2^m = 0 \\ e^2 & \text{if } \mu_1^m = 0 \end{cases}.$$

Then $\overline{\text{MSR}}$ contains precisely the vectors $\lambda \in [0, \infty)^2$ satisfying the following inequalities:

$$\langle \lambda, n^m \rangle \leq \sum_{m' \in \mathcal{M}} \nu(m') \max \left\{ \mu_1^{m'} n_1^m, \mu_2^{m'} n_2^m \right\}, \quad m \in \mathcal{M}.$$

Observe that, when μ^m has strictly positive entries for every $m \in \mathcal{M}$, the conditions given in this result are precisely (4). In other words, (4) is necessary and sufficient in this case. We will see that this is not at all the case when $K > 2$.

6.3 MSR description in higher dimensions

We introduce the following classifications of H-descriptions which prevent redundant or unnecessary information.

DEFINITION 11. *An H-description is called minimal if no inequality in the system can be removed without altering the polytope, and it is said to be weakly minimal if every inequality defines a facet.*

A lack of minimality occurs, for example, in Corollary 10 when two different environments $m, m' \in \mathcal{M}$ satisfy $\mu_1^m \mu_2^{m'} = \mu_1^{m'} \mu_2^m$, making the corresponding constraints redundant. The weak minimality requirement allows repeated inequalities provided that their associated hyperplanes intersect the polytope in facets, but excludes those intersecting the polytope in lower dimensional faces, or not intersecting it at all. For instance, the H-description given in [6] does not satisfy weak minimality.

There is a notion of *non-degeneracy* for cepheids (of any dimension), introduced in [8], which excludes homotheties among the generating subsimplices. For $K = 2$, this condition simply prevents two deGua simplices from having the same slope, avoiding in this way the redundancy mentioned in the previous paragraph. Under this non-degeneracy condition, the H-description provided in Corollary 10 (as well as the one in Corollary 12 below) is minimal, and it is weakly minimal in the absence of non-degeneracy.

The *Pareto surface* of a cepheid is the set of points λ on its topological boundary such that, for every $\varepsilon > 0$, the point $\lambda + (\varepsilon, \dots, \varepsilon)$ does not belong to the cepheid. The *Pareto faces* are defined as the facets contained in the Pareto surface. In the 2-dimensional case treated in Section 4, the Pareto surface is simply the polyline $v^0 - v^1 - \dots - v^M$.

We will explain how existing results for cepheids allow the description of the Pareto surface in any dimension – which of course provides a full description of the cepheid. This description, together with Theorem 8, will lead to Corollary 12.

There are two main differences between the 2- and the K -dimensional H-descriptions: (i) in the general case, there are

$$L_{K,M} = \binom{K+M-2}{K-1}$$

inequalities instead of just M , and (ii) the hyper-spaces associated with these inequalities are no longer given explicitly, but are instead derived through a recursive procedure. In higher dimensions, $L_{K,M}$ gives the number of Pareto faces in the non-degenerate case, and serves as an upper bound in the degenerate one. For instance, in the ON-OFF case (see [1, 2]), there are $M - 1 = 2^K - 1$ inequalities (one per non-empty subset of queues, as the empty set yields a null service rate vector), which is strictly fewer than $L_{K,M-1}$ for $K > 2$. This reflects the high degeneracy of the ON-OFF structure, which produces several redundant inequalities.

6.3.1 Some preliminaries on cepheids

It is convenient to identify the system of generating vectors $(u^m)_{m \in \mathcal{M}}$ with the matrix $U \in [0, \infty)^{M \times K}$ whose rows are given by $\text{row}_m(U) = u^m$, $m \in \mathcal{M}$. We can associate to the matrix U a family $\mathcal{J}_U$ consisting of $L_{K,M}$ Boolean matrices. This family is called the *set of reference systems*, and provides a precise description of the Pareto surface. Specifically, $\mathcal{J}_U$ is a family

$$\mathcal{J}_U = \{J_l \in \{0, 1\}^{M \times L} : l \in [L_{K,M}]\}$$

that allows us to express the Pareto surface as $\bigcup_{l \in [L_{K,M}]} F_l$, where

$$F_l = \sum_{m \in \mathcal{M}} \text{conv} \left\{ u_k^m e^k : J_l(m, k) = 1 \right\}, \quad l \in [L_{K,M}].$$

In the non-degenerate case, F_l is a Pareto face for every l . In the degenerate case, the dimension of F_l is $K - 1$ for every l , the dimension of $F_l \cap F_{l'}$ is at most $K - 2$ if $l \neq l'$, and a Pareto face can be the union of different F_l 's. For each m , the term $\text{conv} \{ u_k^m e^k : J_l(m, k) = 1 \}$ is a standard sub-simplex of the m -th deGua simplex, and the m -th row of J_l is indicating the indices which are being considered in such a sub-simplex.

It is instructive to describe the reference systems in the case $K = 2$. As we did in Section 4, assume that the environments are ordered according to their slopes. In this case, $L_{K,M} = M$, and the l -th matrix J_l is defined as

$$\text{row}_m(J_l) = \begin{cases} (1, 0) & \text{if } m < l \\ (1, 1) & \text{if } m = l \\ (0, 1) & \text{if } m > l \end{cases}.$$

Once the reference systems have been found, the (weakly minimal) H-representation of the cepheid can be obtained as follows. For each $l \in [L_{K,M}]$, the unit exterior normal n^l to F_l is obtained by solving the linear system requiring orthogonality to every standard sub-simplex $\text{conv} \{ u_k^m e^k : J_l(m, k) = 1 \}$, $m \in \mathcal{M}$. A point in F_l is then easily constructed as the sum of one point from each such sub-simplex. By projecting this point onto n^l , we obtain the

distance a_l between the origin and the hyper-space generated by each F_l . Hence we get the following representation for the cepheid:

$$\sum_{m \in \mathcal{M}} \Gamma_{u^m} = \left\{ \lambda \in [0, \infty)^K : \langle \lambda, n^l \rangle \leq a_l \ \forall l \in [L_{K,M}] \right\}.$$

The *dual cepheid* is the one formed by the columns of U . In other words, the matrix representing the dual cepheid is U^T . Interestingly, the reference systems of the dual cepheid are simply the transposes of those of the original cepheid: $\mathcal{J}_{U^T} = \{J^T : J \in \mathcal{J}_U\}$. This immediately provides a characterisation of the dual cepheid in terms of the characterisation of the original one. Duality will be necessary for the iterative scheme.

6.3.2 Iterative scheme

We now describe an iterative procedure for determining the Pareto faces, adapted from [8, Chapter 6]. We do not discuss its computational complexity and refer the reader to this book and references therein for further details.

We first treat the case $M \geq K$. Suppose inductively that, for every $m \in \mathcal{M}$, the set of reference systems associated with $(u^{m'})_{m' \in \mathcal{M} \setminus \{m\}}$ is known. This is equivalent to saying that we know $\mathcal{J}_{U_{-m}}$, where U_{-m} is the matrix obtained from U after removing the m -th row. We will define a mapping $\zeta_m : \mathcal{J}_{U_{-m}} \rightarrow \{0, 1\}^{M \times K}$ as follows. For $J \in \mathcal{J}_{U_{-m}}$, compute its associated normal vector n^J as in Section 6.3.1, let $k \in \mathcal{K}$ be such that

$$n_k^J u_k^m = \max_{k' \in \mathcal{K}} n_{k'}^J u_{k'}^m,$$

and define $\zeta_m(J)$ by adding to J the row $u_k^m e^k$ in the m -th position. Precisely,

$$\text{row}_{m'}(\zeta_m(J)) = \begin{cases} \text{row}_{m'}(J) & \text{if } m' \neq m \\ u_k^m e^k & \text{if } m' = m \end{cases}.$$

The reference system $\mathcal{J}_U$ is obtained by running this procedure over all $m \in \mathcal{M}$ (with possible repetitions), that is, $\mathcal{J}_U = \bigcup_{m \in \mathcal{M}} \zeta_m(\mathcal{J}_{U_{-m}})$. Note that, since we are dealing with Boolean matrices, it is straightforward to discard repetitions in this union.

If $M < K$, simply apply duality and the previous case. Finally observe that this procedure finishes because the set of reference system is known for $K = 2$.

COROLLARY 12. *For general K , we have*

$$\overline{\text{MSR}} = \{\lambda \in [0, \infty)^K : \langle \lambda, n^l \rangle \leq a_l \ \forall l \in [L_{K,M}]\},$$

where the families of non-null vectors $(n_l)_{l \in [L_{K,M}]}$ and strictly positive numbers $(a_l)_{l \in [L_{K,M}]}$ are obtained by the iterative procedure described in Sections 6.3.2 (combined with Section 6.3.1 for the normals and the distances to the hyper-spaces) applied to the vectors $(u^m)_{m \in \mathcal{M}}$ defined as $u^m = \nu(m)\mu^m$, $m \in \mathcal{M}$. This representations is weakly minimal, and it is minimal in the non-degenerate case.

We highlight that this characterisation is fundamentally different from the one given in (1), as it is direct rather than formulated as an existence condition. To the best of our knowledge, this type of characterisation exists only for ON/OFF-type environments.

7. CONCLUSION

Determining minimal H-descriptions of sums of polytopes is a challenging problem. In general, no closed-form expression is known for the minimal H-description of the sum of two general polytopes in terms of their respective minimal H-descriptions. The cepheid identification introduced in this work provides a structural framework that makes such a description possible for the MSR of randomly modulated multi-queue single-server systems, in arbitrary dimensions. Without this identification, obtaining the minimal H-description seems not be feasible.

8. ACKNOWLEDGMENTS

The authors acknowledge Matthieu Jonckheere for insightful discussions. This research is partially supported by the French “Agence Nationale de la Recherche (ANR)” through the project ANR-22-CE25-0013-02 (ANR EPLER).

9. REFERENCES

- [1] L. Tassioulas and A. Ephremides, “Dynamic server allocation to parallel queues with randomly varying connectivity,” *IEEE Transactions on Information Theory*, vol. 39, no. 2, pp. 466–478, 1993.
- [2] Nicholas Bambos and George Michailidis, “Queueing networks of random link topology: Stationary dynamics of maximal throughput schedules,” *Queueing Systems*, vol. 50, no. 1, pp. 5–52, May 2005.
- [3] Nahuel Soprano-Loto, Urtzi Ayesta, Matthieu Jonckheere, and Ina Maria Verloop, “Decision-epoch matters: Unveiling its impact on the stability of scheduling with randomly varying connectivity,” arXiv:2403.18686, Mar. 2024.
- [4] S. Shakkottai and A. L. Stolyar, “Scheduling for multiple flows sharing a time-varying channel: The exponential rule,” *Translations of the American Mathematical Society-Series 2*, vol. 207, pp. 185–202, 2002.
- [5] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, “Scheduling in a queueing system with asynchronously varying service rates,” *Probability in the Engineering and Informational Sciences*, vol. 18, no. 2, pp. 191–217, 2004.
- [6] Hassan Halabian, Ioannis Lambadaris, and Chung-Horng Lung, “Explicit characterization of stability region for stationary multi-queue multi-server systems,” *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 355–370, 2014.
- [7] Rolf Schneider, *Convex Bodies: The Brunn–Minkowski Theory*, Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2nd edition, 2013.
- [8] Joachim Rosenmüller, “Cepheids: Minkowski sums of deGua Simplices: Theory and applications,” Center for Mathematical Economics Working Papers 629, Bielefeld University, Center for Mathematical Economics (IMW), Bielefeld, 2019.
- [9] Jean-Paul Bourrières, “The minkowski sum of simplices in 3-dimensional space: An analytical description,” *Computational Geometry*, vol. 47, no. 8, pp. 856–867, 2014.

Optimal service and switching strategies for polling systems

Purva Joshi
Eindhoven University of
Technology
p.c.joshi@tue.nl

Marko A.A. Boon
Eindhoven University of
Technology
m.a.a.boon@tue.nl

Erik M.M. Winands
University of Amsterdam
Rabobank
e.m.m.winands@uva.nl

ABSTRACT

In this paper, we investigate how polling systems with intelligent policies perform compared to the optimal service policy. To this end, we introduce two new policies: an intelligent server idling policy incorporating information about short-term future arrivals, and an informed switching policy that uses queue lengths to switch to the next non-empty queue. For benchmarking purposes, the optimal service schedule is obtained by formulating the problem as a Mixed Integer Program, with full arrival information. Our results show that with these intelligent policies, polling systems significantly improve in performance, thereby narrowing the gap between polling policies and optimal schedules.

Categories and Subject Descriptors

G.3 [Mathematics of Computing]: Probability and Statistics—*Queueing Theory*

General Terms

Theory, Performance

Keywords

Polling model, optimisation, switching rules, idling

1. INTRODUCTION

A polling system consists of multiple queues served by a single server in a fixed sequence. In a polling system, setup times are required for the server to move from one queue to another. For analytical purposes, it is typically assumed that these setup times are incurred even when the next queue is empty. As a result, the server continues switching between queues even when the entire system is empty, implying that the server never idles. This behaviour is not only obviously suboptimal, but it also leads to the anomaly that increasing setup times can actually improve system performance (see, e.g., [5]).

Despite having received much scientific attention, the following question remains largely unanswered: *How does the quality of service achieved by a polling system compare to*

that of an optimal policy? The objective of the present study is to make a step in answering this question. Therefore, we explore polling systems with a novel smart server idling policy and introduce the concept of informed switching. The latter policy allows the server to move directly to the next non-empty queue while bypassing any empty queues. As these switching policies seem to be difficult to analyse from a queueing perspective, we perform simulation studies under the exhaustive, k -limited and gated service policies. In our simulation study, we examine a situation in which the server has perfect foresight regarding future arrivals. This assumption enables us to derive the truly optimal service policy by formulating the scheduling problem as a Mixed Integer Program. The optimal policy serves as a benchmark for evaluating the effectiveness of the idling and informed switching policies. Perfect foresight is also a realistic, practical assumption, since the polling system is frequently part of a multi-stage framework implying that full information is available on upstream departures (see, e.g., [15]).

Our study shows that informed switching significantly improves system performance in many situations. We can also conclude that the combined idling and informed switching policy most closely approaches the performance of the optimal strategy for moderate traffic situations. However, in highly asymmetric cost cases, a considerable performance gap remains. Finally, our results show that in heavy traffic, idling is not necessarily advantageous, as customers are (almost) always waiting in one of the queues. The main conclusion of our research is that intelligent idling and switching rules bring performance levels in polling systems closer to the optimal performance benchmark. From a practical point of view, this is an important conclusion as computing the optimal solution is an NP-hard problem requiring substantial computational resources. In contrast, the intelligent polling strategies can be implemented in an operationally efficient way.

The structure of this paper is as follows. In Section 2, we introduce the class of polling models and formulate the objective function. Section 3 discusses the characteristics of several idling and switching policies together with the optimal policy. Subsequently, in Section 4 we present the setup and the results of the simulation study. Finally, Section 5 states our main conclusions and outlines several promising directions for future research.

2. MODEL AND OBJECTIVE

This section describes the polling systems under consideration, followed by a formulation of the objective function.

We consider a system with a single server for $N \geq 1$ queues, in which there is an infinite buffer capacity for each queue. Customers arrive at queues according to (independent) Poisson processes with rates λ_i and $\lambda := \sum_{i=1}^N \lambda_i$. The service times at queue i are independent, identically distributed random variables with mean $\mathbb{E}[B_i]$ and finite second moment $\mathbb{E}[B_i^2]$. Let the arrival times and the beginning-of-service times of customers to queue i be denoted by $\{T_{i,j}^A\}_{j=1}^\infty$ and $\{T_{i,j}^B\}_{j=1}^\infty$, respectively. When the server starts service at queue i , a setup time is incurred, of which the first and second moments are denoted by $\mathbb{E}[S_i]$ and $\mathbb{E}[S_i^2]$, respectively. The mean total setup time $\mathbb{E}[S]$ in a cycle (i.e., the time between two successive visits to a queue) is given by $\mathbb{E}[S] = \sum_{i=1}^N \mathbb{E}[S_i]$, and we use $\text{Var}[S]$ to denote the variance of the total setup time S . Lastly, we define $\rho_i := \lambda_i \mathbb{E}[B_i]$ as the contribution of queue i to the total load $\rho = \sum_{i=1}^N \lambda_i \mathbb{E}[B_i]$.

We investigate systems with the following service disciplines:

- *Exhaustive*: when the server polls a queue, it serves its customers until that queue is empty;
- *Gated*: when the server polls a queue, it serves all, and only, customers found at the polling instant;
- *k-limited*: when the server polls a queue, it continues working at this queue until either a predefined number of k_i customers is served or until the queue becomes empty, whichever occurs first.

In the remainder of the paper, we assume that the polling system is stable (see [11, 8] for the conditions).

Our objective is to minimise the weighted sum of the mean waiting times $\mathbb{E}[W_i]$,

$$\min \sum_{i=1}^N c_i \mathbb{E}[W_i], \quad (1)$$

where c_i is the cost parameter of queue i . If c_i equals ρ_i for all queues, this cost function simplifies to the well-known pseudo-conservation law (see, e.g., [4]). Assuming identical mean service times across all queues, for the gated and exhaustive service policy the mean waiting time of an arbitrary customer $\mathbb{E}[W] = \sum_{i=1}^N \frac{\lambda_i}{\lambda} \mathbb{E}[W_i]$ directly follows from the pseudo-conservation law.

3. IDLING AND SWITCHING POLICIES

This section focuses on various intelligent idling and switching policies. Moreover, we introduce the optimal strategy.

3.1 Non-idling policy

This sub-section examines an anomaly in non-idling polling systems, providing additional motivation to incorporate idling policies into our study. Non-idling policies mean that as soon as the service-discipline-specific switching condition is met, the server switches to the next queue, in contrast to intelligent idling policies that will be discussed in the next section. In classical *non-idling* polling systems, setup times are incurred even when the next queue is empty. This also means that the server never idles and that it continues switching between queues even when the entire system is empty. About 35 years ago Sarkar and Zangwill [12] published a paper on non-idling polling systems, in which they numerically show that reduction of setup times can increase

the mean queue lengths. It can be seen from the pseudo-conservation law that an increase in mean setup times actually leads to *reduced* mean waiting times if

$$\frac{\text{Var}[S]}{\mathbb{E}[S]^2} > \frac{\rho \pm \sum_{i=1}^N \rho_i^2}{\rho(1-\rho)},$$

where the $\pm$ should be read as $-$ for exhaustive service and $+$ for gated service. In [6] it is demonstrated that server idling is an effective mechanism to prevent, to a large extent, queue lengths from increasing when setup times are reduced.

Motivated by this counter-intuitive effect of setup times, [7] proved that for exhaustive and gated non-idling polling systems, the queue length distribution is invariant with respect to deterministic shifts in setup times so long as the net shift is zero, *ceteris paribus* (see also [2]). This result has been extended to a more general setting, i.e., including k -limited service and all other policies where service is determined based on local queue information in [14], and we state this previously unpublished result below. Via standard probabilistic arguments, it is shown for a general class of policies that if setup times are decreased (increased) by a fixed total amount, it is immaterial which specific setup times are decreased (increased). The only restriction enforced on the policy is that it is allowed to use local information only. Consider the non-idling polling system as introduced in Section 2. Let $S_i^{b_i} = S_i + b_i$ denote the setup time which results when the original setup time S_i is shifted by $b_i \in [-b_i^{max}, \infty]$ time units where $b_i^{max} \in \mathbb{R}^+$ is the infimum of the support of S_i . The theorem below implies that the queue length distributions depend on the vector of shifts only via its aggregate sum. Without loss of generality, we focus on queue 1.

Theorem 1 *For a non-idling polling system with setup times $S_i^{b_i}$, the distribution of the queue length L_1 depends on the vector $(b_1, b_2, \dots, b_N)$ only via its sum b .*

Due to page restrictions, we omit the proof here and instead refer to [14, Theorem 4.3.3].

3.2 Intelligent policies

Motivated by the anomalous behaviour observed, we introduce a new class of intelligent idling policies that go beyond the rudimentary approach of extending setup times.

3.2.1 Idling policy

We introduce an intelligent server idling policy that has not been previously studied in the literature. Before the server is ready to switch over from the queue i currently being served, it checks if there will be any arrivals to queue i within the next x time units. If so, the server *waits* at queue i to serve the incoming arrival(s) as long as the constraints imposed by the service policy (e.g., maximum k services in one server visit in case of k -limited) are being satisfied. For exhaustive idling policies, the server idles at a queue if it detects incoming arrivals in the next x time units, and keeps serving that queue until it is empty again, and then again checks if idling is possible. Of course, idling and gated service are incompatible, since any arrivals during the server's idle period would be positioned behind the gate established at the beginning of the server's visit.

3.2.2 Informed switching policy

Under the informed switching policy, once the server has completed its visit to a queue and is ready to switch, it skips the setup phase at empty queues and starts setup only upon reaching the next non-empty queue. This means that at the end of a server visit to a queue, the server immediately switches over to the next non-empty queue. If the system is completely empty, the server goes to the queue where the next arrival will occur and starts setup at this queue immediately. State-dependent switching strategies have also been analysed in [1, 9, 13]. However, these models differ in various ways from our informed switching strategy.

3.3 Optimal policy

The optimal service schedule is obtained by solving the optimisation problem in Eq. (1) up to a time horizon T :

$$\begin{aligned} \min_{\substack{T_{i,j}^B \\ i=1,\dots,N, \\ j=1,\dots,L_i^T}} \quad & \sum_{i=1}^N \sum_{j=1}^{L_i^T} c_i W_{i,j} = \sum_{i=1}^N \sum_{j=1}^{L_i^T} c_i (T_{i,j}^B - T_{i,j}^A) \\ \text{subject to:} \quad & \begin{cases} T_{i,j}^B \geq T_{i,j}^A \\ T_{i,j}^B \geq T_{i,j-1}^B + B_i \\ \begin{cases} T_{i,j}^B \geq T_{k,l}^B + B_k + S_i & \text{if } T_{i,j}^B \geq T_{k,l}^B \\ T_{k,l}^B \geq T_{i,j}^B + B_i + S_k & \text{otherwise} \end{cases} \\ \text{for } k \neq i, k = 1, \dots, N, l = 1, \dots, L_k^T \\ \text{for } j = 1, \dots, L_i^T, i = 1, \dots, N. \end{cases} \end{aligned}$$

Here L_i^T is the number of customers at queue i who have received/ are receiving service, up to time T . We have set up the problem so that the constraints are as general as possible; this means that idling and/or informed switching is/are feasible with the constraints, but not necessary. Also note that the optimisation problem is solved with full information about arrival times $T_{i,j}^A$, for $j = 1, \dots, L_i^T$, and $i = 1, \dots, N$, thus the objective function effectively reduces to minimising $\sum_{i=1}^N \sum_{j=1}^{L_i^T} c_i T_{i,j}^B$, i.e., the cost-weighted sum of beginning-of-service times. This optimisation problem to obtain the optimal service policy is a Mixed Integer Program, which is known to be in the class of NP-hard problems.

4. SIMULATION STUDY

In this section we use a simulation study to evaluate the performance of the proposed intelligent policies.

4.1 Setup of the study

We evaluate the performance of polling models focusing on the cost-weighted mean waiting time. We test three service disciplines – Exhaustive (‘E’), Gated (‘G’) and k -limited (‘KL’) – under four policy variants as applicable: Classic non-idling and non-informed switching (‘C’), Idling (‘Id’), Informed Switching (‘InS’) and their combination (‘IdInS’). To examine the effects of straying from polling in cyclic order, we introduced two sub-variants for the ‘Idling + Informed Switching Exhaustive’ (‘IdInSE’) service policy: one where the next queue is chosen in *cyclic* order (‘IdInSEC’), and the other where the next queue is chosen based on the *longest waiting customer* (‘IdInSELW’).

By definition, the idling policies have access to information about arrivals in the next x time units, while informed switching policies have access to the queue lengths for the

Table 1: Parameter values for the symmetric case.

Parameter	Value (Unit, optional)
Number of queues	3
Service time	1 s
Setup time	2 s

Table 2: Loads, k_i and time horizon for the symmetric case.

Instance	ρ_i 's	Optimised k_i 's / Minimum req. k_i 's	T (s)	# Runs
Light	$[\frac{0.01}{3}, \frac{0.01}{3}, \frac{0.01}{3}]$	[1, 1, 1] / [1, 1, 1]	49999	100
Medium	$[\frac{0.5}{3}, \frac{0.5}{3}, \frac{0.5}{3}]$	[4, 4, 4] / [2, 2, 2]	120	17
Heavy	$[\frac{0.95}{3}, \frac{0.95}{3}, \frac{0.95}{3}]$	[40, 40, 40] / [38, 38, 38]	499995	100

Table 3: (Additional) parameter values for testing the effect of varying the idling parameter for the symmetric medium case.

Parameter	Value (Unit, optional)
Time horizon T	99999 s
Idling thresholds	[0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5]
# Runs	100

entire system and in case the system goes empty, the server knows which queue will have the next arrival. We have set the threshold for server idling, x to be equal to the setup times. For simplicity, we assume identical, deterministic service and setup times. For the k -limited service discipline, we optimised the service limits k_i based on the solution for the constrained optimisation presented in [3, Eq. (6.10)], by constraining $\sum_{i=1}^N k_i = K$ and accounting for stability.

The optimal service schedule is derived by solving the optimisation problem in Section 3.3, assuming full arrival information over the time horizon T . This problem is tractable for light traffic regimes (load tends to zero) over long time horizons. Under medium traffic regimes, we could only solve the optimisation for a small time horizon, otherwise the computations become extremely time- and memory-intensive. For heavy traffic (load tends to one), computing the optimal schedule becomes infeasible for all non-trivial time horizons for the same reason. We study symmetric cost cases (rendering cost values redundant) and cases with asymmetric costs. Under the chosen time horizons for the symmetric light and heavy cases, the system reaches a steady state.

Symmetric cost: We examine the waiting times in a symmetric three-queue polling model under light, medium, and heavy traffic. The parameters are shown in Tables 1 and 2. We also have conducted tests with asymmetric traffic loads in a symmetric costs situation, however for reasons of brevity, we omit presenting those tests here, since the insights to be gained from those tests were similar. For the symmetric case in medium traffic, we examine the effect of varying the idling threshold parameter in a steady state system. Idling is executed only if the server *knows* that there will be customers arriving within this idling time. Thus if the threshold parameter is set to be too low, the system is unable to exploit the full benefit of idling. If it is set too high, then the system suffers from unnecessary delays. The (additional) parameters for this test are shown in Table 3.

Table 4: Parameter values for the asymmetric case.

Parameter	Value (Unit, optional)
Number of queues	2
Arrival rates	$[0.7, 0.1]/[0.4, 0.2]$
Service time	1 s
Setup time	2 s
Waiting costs	$[1, 50]/[1, 5]$
Optimised k_i 's	$[16, 4]/[10, 10]$
Minimum req. k_i 's	$[14, 2]/[4, 2]$
Time horizon T	99
# Runs	90/72



Figure 1: Symmetric case in light traffic.

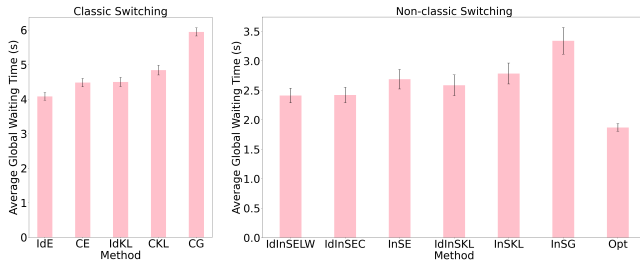


Figure 2: Symmetric case in medium traffic.

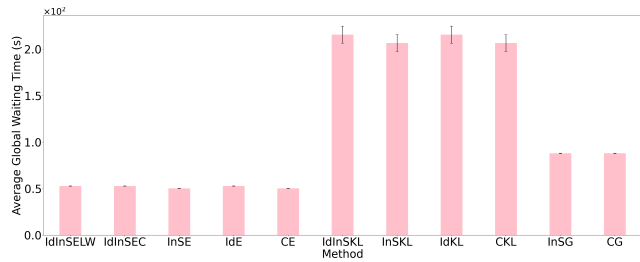


Figure 3: Symmetric case in heavy traffic.

Asymmetric costs: In order to study the effect of waiting costs, we examine a two-queue polling model with asymmetric load and cost structure: the queue with the higher load has lower costs, and vice versa. Out of the two cases we look at, this divide between queues in terms of the loads and costs is more extreme in one situation than the other. The parameters are presented in Table 4.

4.2 Discussion of the results

Figs. 1-3 present the global waiting times (averaged over several runs) for the above-described policies in the symmetric costs cases, Fig. 4 shows the effect of varying the idling threshold on the mean waiting times, and Figs. 5-6

show the asymmetric costs case. The error bars represent the respective standard errors of the mean. Note that we have performed more tests, all confirming our observations.

4.2.1 Symmetric cost

1. The informed switching variants have significantly lower mean waiting times than the classic variants across all service policies and traffic intensities, except in heavy traffic, where we observe similar performances. This demonstrates improved performance simply by adding queue lengths to the global state information.
2. The idling variants improve upon the performance exhibited by their respective classic service policies. However, the advantage diminishes in heavy traffic, when idling might not necessarily lower mean waiting times. Forced idling in heavy traffic leaves the server unoccupied during times when there are ‘almost always’ waiting customers that can be served immediately.
3. The combined idling and informed switching exhaustive policy most closely approximates the performance of the optimal benchmark for moderate traffic situations. This is quite significant since the optimisation problem is NP-hard, requiring heavy computational resources to compute the optimal solution, whereas a polling policy can generate a service schedule in a fraction of the time and computing resources.
4. While the optimal policy is not mandated to incorporate idling or informed switching, a closer examination of the optimal service schedule reveals that both strategies naturally emerge. Some instances where the optimal schedule outperforms the combined idling and informed switching exhaustive policy are when *the optimised server idles longer*. Below we conduct a more detailed analysis of how the performance of the strategies is influenced by the value of the idling parameter. Other instances where the optimal strategy outperforms the combined idling and informed switching exhaustive policy are when *it deviates from the visiting order* (cyclic or serving the longest waiting customer).

Some of the conclusions above are supported by findings in [10], where Liu et al. have also compared several service policies in polling systems. However, the policies considered in this paper are much more extensive, in particular those involving idling and informed switching.

Effect of the idling threshold: Fig. 4 shows that the value of the idling parameter has an effect on system performance in the symmetric medium case. Consistent with our intuition, there exists an optimal range, typically between 2 and 3 seconds for most policies, that minimises the mean waiting time. We have conducted a similar test for the symmetric heavy traffic case and noticed that the policy performance was determined only by the service discipline. Informed switching did not seem to affect the mean waiting time, which was also remarked upon earlier in the discussion of the results. The test also confirmed that mean waiting times are monotonically increasing with respect to the idling threshold parameter, thereby leading to the conclusion that idling is not helpful to improve performance in heavy traffic regimes.

4.2.2 Asymmetric cost

Medium asymmetric case: Fig. 5 shows that there is not

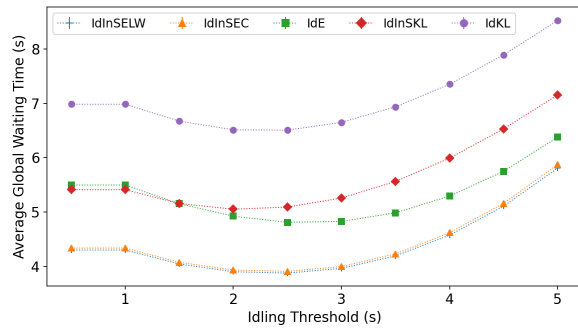


Figure 4: Effect of varying the idling threshold parameter for the symmetric case with medium traffic.

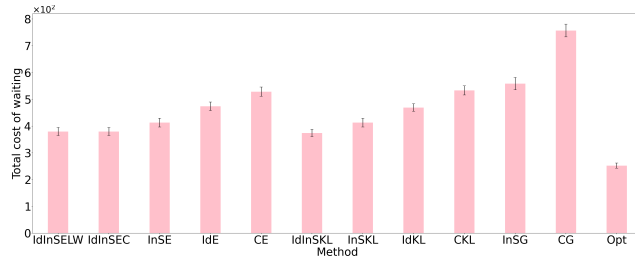


Figure 5: Case with medium asymmetry between queues.

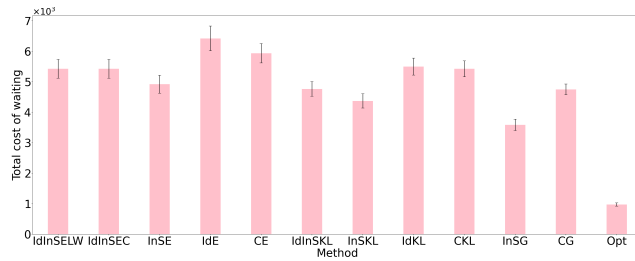


Figure 6: Case with high asymmetry between queues.

much difference in the performance of the exhaustive and k -limited policies. This is not only because of the moderate asymmetry, but also because the service limits k_i are set to be significantly higher than required for stability, which could skew the behaviour of the k -limited policies towards that of exhaustive. Both gated policies are amongst the poorer performing ones, which is notable in contrast to the highly asymmetric case (discussed below).

Highly asymmetric case: Due to the highly asymmetric cost and load structure, the k -limited and gated policies outperform the exhaustive policies as illustrated by Fig. 6. However, these policies are still far from the optimal benchmark. The k -limited and gated policies exhibit superior performance compared to the exhaustive variants, as they incorporate periodic checks for arrivals at queue 2 even when queue 1 is non-empty.

5. CONCLUDING REMARKS

In this paper we have shown that the quality of service levels in polling systems can be significantly improved by implementing intelligent idling and switching rules. Our results lead to many promising directions for future research, three

of which we outline here. Firstly, the assumption of perfect foresight regarding future arrivals is practically realistic given that polling systems often operate within multi-stage networks (see, e.g., [14, 15]). We aim to further investigate optimal strategies for such networks leveraging upon the introduced intelligent policies, potentially incorporating job size as a factor as well to optimise the service order *within* a queue. Secondly, we want to extend our research to polling models allowing for the simultaneous servicing of multiple queues. Thirdly, this paper focused on the minimisation of the weighted sum of *mean* waiting times. An interesting direction for future research is optimising *higher moments* or *tail probabilities*.

6. REFERENCES

- [1] K. Avrachenkov, E. Perel, and U. Yechiali. Finite-buffer polling systems with threshold-based switching policy. *TOP*, 24(3):541–571, 2016.
- [2] S. C. Borst and O. J. Boxma. Polling models with and without switchover times. *Operations Research*, 45(4):536–543, 1997.
- [3] S. C. Borst, O. J. Boxma, and H. Levy. The use of service limits for efficient operation of multistation single-medium communication systems. *IEEE/ACM Transactions on Networking*, 3(5):602–612, 1995.
- [4] O. J. Boxma and W. P. Groenendijk. Pseudo-conservation laws in cyclic-service systems. *Journal of Applied Probability*, 24(4):949–964, 1987.
- [5] R. B. Cooper, S. C. Niu, and M. M. Srinivasan. When does forced idle time improve performance in polling models? *Management Science*, 44:1079–1086, 1998.
- [6] J. P. Dorsman. Variance effects in polling systems: the limits of Japanese production theory?, 2009. BMI paper, VU Amsterdam.
- [7] A. Federgruen and Z. Katalan. The impact of setup times on the performance of multiclass service and production systems. *Operations Research*, 44:989–1001, 1996.
- [8] C. Fricker and M. R. Jaibi. Monotonicity and stability of periodic polling models. *Queueing Systems*, 15(1-4):211–238, 1994.
- [9] D. Gupta and M. M. Srinivasan. Polling systems with state-dependent setup times. *Queueing Systems*, 22:403–423, 1996.
- [10] J. Liu, P. Nain, and W. M. Townsley. On optimal polling policies. *Queueing Systems*, 11(3-4):295–317, 1992.
- [11] J. A. C. Resing. Polling systems and multitype branching processes. *Queueing Systems*, 13(4):409–426, 1993.
- [12] D. Sarkar and W. I. Zangwill. Variance effects in cyclic production systems. *Management Science*, 37:444–453, 1991.
- [13] M. P. Singh and M. M. Srinivasan. Exact analysis of the state-dependent polling model. *Queueing Systems*, 41(4):371–399, 2002.
- [14] E. M. M. Winands. *Polling, Production & Priorities*. PhD thesis, Eindhoven University of Technology, 2007.
- [15] E. M. M. Winands, I. J. B. F. Adan, and G. J. van Houtum. The stochastic economic lot scheduling problem: a survey. *European Journal of Operational Research*, 210(1):1–9, 2011.

Markov-modulated dependencies for a multiplicative Lindley-type recursion

Ioannis Dimitriou

Department of Mathematics, University of Ioannina, 45110, Ioannina, Greece

idimit@uoi.gr

ABSTRACT

In this paper, we study the stationary behaviour of a Markov-modulated multiplicative Lindley-type recursion $W_{n+1} = [V_n W_n + Y_n(V_n)]^+$. Given the state of the background Markov chain, we consider two cases: a) V_n equals either 1, or $a \in (0, 1)$, or it is negative with certain probabilities, and $Y_n(V_n) := Y_n = S_n - A_{n+1}$, where both A_n and S_n have a rational Laplace-Stieltjes transform (LST). b) V_n equals 1 or -1 according to certain probabilities, and $Y_n(V_n)$ follows a more general scheme, dependent on V_n . In both cases, we derive the LST of the stationary transform vector of $\{W_n\}_{n \in \mathbb{N}_0}$. In the second case, we also provide a recursive approach to obtain the steady-state moments and provide an asymptotic result. A simple numerical example illustrates the theoretical findings.

Keywords: Recursion; Laplace-Stieltjes transform; Markov-modulation

1. INTRODUCTION

In this work, we focus on the stationary analysis of a Markov-modulated multiplicative Lindley-type recursion

$$W_{n+1} = [V_n W_n + Y_n(V_n)]^+, \quad (1)$$

with $[x]^+ := \max\{x, 0\}$, and where $Y_n(V_n)$ can be written as a difference of two nonnegative random variables that may depend on V_n . Stochastic recursions have received much attention in the applied probability literature, and especially those of autoregressive type, due to their wide applicability across various scientific domains including biology, finance, and engineering, e.g., [8, 9, 16] and the references therein.

In this work, we focus on two models. In Model I, the V_n are either equal to one, or equal to a positive constant $a \in (0, 1)$, or negative, according to certain probabilities. We further assume that $Y_n(V_n) = S_n - A_{n+1}$ (i.e., independent of V_n), and we demand that both the positive and the negative parts of the Y_n have a rational LST. In Model II, the V_n are equal either to one, or to minus one, and $Y_n(V_n)$ have a more general form that is dependent on V_n . In both cases, $Y_n(V_n)$ are also dependent based on the state of an irreducible discrete time Markov chain with finite state space.

The vast majority of the related works focus on the investigation of the stability conditions, limit theorems and questions related to queueing applications, whereas our primary

focus lies on the derivation of results for the stationary distribution of the process under investigation. Furthermore, most of the related works refers to the scalar case. Our focus is on the investigation of the stationary behaviour of vector-valued stochastic recursions of the type in [1].

In [6], the authors investigated both the transient and the stationary behaviour of the (scalar) recursion $W_{n+1} = [aW_n + S_n - A_n]^+$, $a \in (0, 1)$. In such a case, W_n refers to the workload in a queueing system just before the n th arrival, which adds S_n work, and makes obsolete a fraction $1 - a$ of the already present work. The case where $a = 1$ describes the waiting time of the standard G/G/1 queue, while the case where $a = -1$ refers to the analysis of carousel systems [19]. In [4], the authors investigated the scalar recursion $W_{n+1} = [V_n W_n + S_n - A_n]^+$, where V_n can be positive or negative; see also [3] for a more general (scalar) case. In [5], the authors considered Lindley-type recursions with dependencies motivated by applications in queueing and insurance risk models. Recently, in [12] the authors generalized the work in [6] by considering non-trivial dependence structures among $\{S_n\}_{n \in \mathbb{N}}$, $\{A_n\}_{n \in \mathbb{N}}$; see also [14, 7].

In [11], the author investigated general Markov-modulated recursions of the form $W_{n+1} = [aW_n + S_n - A_{n+1}]^+$, $a \in (0, 1)$, thus, considered the Markov-dependent version of the process analysed in [6]. Note that the specific case of $a = 1$ in [11] corresponds to the waiting time in a single server queue with Markov-dependent interarrival and service times studied in [1]. The case where $a = -1$ was investigated in [20] in the context of carousel models.

1.1 Contribution

1. A major contribution of the present paper, regarding Model I, is that we derive and solve a system of Wiener-Hopf equations, which allows to study the stationary behavior of the $\{W_n\}_{n \in \mathbb{N}_0}$ process. To the best of the author's knowledge, in Model I we studied for the first time a recursion of the form $W_{n+1} = [V_n W_n + Y_n(V_n)]^+$, where $Y_n(V_n) = Y_n = S_n - A_{n+1}$, $P(V_n = 1) = p_1$, $P(V_n = a) = p_2$, where $a \in (0, 1)$, $P(V_n < 0) = p_3 = 1 - p_1 - p_2$, under a semi-Markov dependent framework (see Section 3). If $p_1 = 1$, the model is reduced to the one in [1], while if $p_2 = 1$ is reduced to the one in [11], and if $p_3 = 1$ is reduced to the one in [20]. We focus on the most interesting case where $0 < p_i < 1$, $i = 1, 2, 3$. The model in [11] is the most relevant to our model, however is focused on the case $P(V_n = a) = 1$, $a \in (0, 1)$. In this work, we allow V_n to take arbitrary distributions on $(-\infty, 0]$, thus we introduce to the model specific mathematical intricacies, but still, we are able to get explicit results in terms of LSTs. Moreover, contrary to the work in

[4] where the authors studied a scalar version, we introduce Markov-modulated dependencies, which lead to a vector-valued fixed-point functional equation that will be solved recursively, with some additional technical requirements in proving the convergence of the iteration scheme based on matrix norms (see [10, Th. 1]). In contrast to the scalar case analyzed in [4], the Markov-modulated framework we introduce results in a system of Wiener-Hopf equations (instead of a single equation in [4]) that we succeed to solve and prove the convergence of its solution. We also discuss the stability condition. Moreover, contrary to [13], where the authors studied a similar recursion, and derived the first two moments, in our work, we succeed to derive the stationary distribution of the process under study in terms of LSTs, thus we are able to derive all the moments, which is quite helpful in performance analysis. This is accomplished by applying the Wiener-Hopf method in a vector-valued functional equation, which seems to be a quite promising technique that can be used to solve a large number of interesting and unsolved problems in applied probability literature.

2. Our second contribution refers to the analysis of a Markov-modulated Lindley-type recursion in Model II, for which the monotonicity assumption does not hold. The scalar Lindley recursion refers to the class of discrete time Markov chains described by the recursion $W_{n+1} = F(W_n, X_n)$. An important assumption that is often made in the related literature is that the function $F(w, x)$ is non-decreasing in its main argument w [2]. The recursion we focus on does not satisfy that usual assumption, which in turn, does not allow us to exploit the known duality results between Markov processes and monotone increasing continuous processes; [2, Section IX.4]. We go one step further and study in detail the Markov-modulated version of that recursion that does not satisfy the monotonicity property, thus, filling an important gap in the related literature. Moreover, although we incorporate Markov-modulation to capture fluctuations in interarrival and service times, we are still able to solve explicitly the vector-valued functional equation in terms of LSTs. Contrary to the scalar case in [7], where the authors solved a Wiener-Hopf boundary value problem to derive the LST of the waiting time distribution, we solve a vector-valued functional equation by using simple arguments, without using the Wiener-Hopf method. This is quite promising since our framework, allows for solving complicated problems without increasing significantly the mathematical difficulties.

In Model II (Section 4), we study the Markov-dependent multiplicative Lindley-type recursion (1), where $P(V_n = 1) = p$, $P(V_n = -1) = q := 1 - p$, $p \in [0, 1]$. Moreover, contrary to the work in [7], where the authors studied a scalar version, in our work we consider Markov-modulated dependencies among $\{A_n\}_{n \in \mathbb{N}_0}$, $\{B_n\}_{n \in \mathbb{N}}$ (resp. $\{C_n\}_{n \in \mathbb{N}_0}$, $\{D_n\}_{n \in \mathbb{N}}$), and on top of that we further allow Y_n to depend on V_n : $Y_n(1) = S_n - A_{n+1}$, $Y_n(-1) = D_{n+1} - C_n$. Our main goal is to derive the LST of the steady-state distribution of $\{W_n\}_{n \in \mathbb{N}}$, when it exists. To the author's best knowledge, it is the first time that such a Markov-modulated multiplicative version of Lindley recursion is studied. We further provided a recursive approach to obtain the steady-state moments of the waiting time distribution, as well as to investigate the asymptotic behaviour of the steady-state waiting distribution for light-tailed service time distributions.

Despite our methodological contribution, our work has

also strong connections with respect to the performance analysis of service systems. In particular, this state-dependent queuing process, as discussed in detail below (see also [7]), is connected to LaPalice queues [15], in which customers are scheduled to be served such that the period between two consecutively scheduled customers is greater than or equal to the service time of the first customer. A LaPalice queue can be considered as a special queue in which each customer makes an appointment with the server but may arrive latter due to some unexpected delays that can be modeled as shifting delays. LaPalice queues are an alternative to flow control in high speed networks [15]. The Markov-modulated framework that we incorporate provides an even more realistic aspect of the model, since in many practical situations, customer arrivals and the service mechanism of a queueing system are influenced by some factors external or internal to the system [17]. In a more general framework, W_n may represent an inventory in time period n (e.g. cash), V_n may represent a multiplicative, possibly random, decay or growth factor between times n and $n + 1$ (e.g. interest rate) and $Y_n(V_n)$ may represent a quantity that is added or subtracted between times n and $n + 1$ (e.g. deposit minus withdrawal).

When $V_n = 1$ (i.e., $p = 1$) we have the MAP/G/1 queue [1], and when $V_n = -1$ (i.e., $p = 0$, equivalently $q = 1$) we have a modulated alternating service model with two service points that arise in the performance analysis of warehouse systems; see [18, 20]. Studying a recursion that contains the Markov-modulated version of both Lindley's classical recursion [1] and the recursion in [18, 19] as special cases it is of interest in its own right. As shown below, for $p \in (0, 1)$, our model describes a Markov-modulated FCFS queue, in which the service times and the interarrival times depend linearly and randomly on the waiting times, as well as on the value of V_n : Consider an extension of a modulated G/G/1 queue, which is specified by an ergodic sequence of non-negative random variables $\{(\bar{K}_n(V_n), \bar{L}_n(V_n), \hat{K}_n(V_n), \hat{L}_n(V_n), V_n)\}_{n \in \mathbb{N}_0}$. Given the state of an irreducible finite state Markov chain, the sequence $\{W_n\}_{n \in \mathbb{N}_0}$ is defined recursively by $W_{n+1} = [W_n + K_n(V_n) - L_n(V_n)]^+$, where $K_n(V_n) = \bar{K}_n(V_n) + \hat{K}_n(V_n)W_n$, $L_n(V_n) = \bar{L}_n(V_n) + \hat{L}_n(V_n)W_n$ with

$$\begin{aligned} \bar{K}_n(V_n) &= \begin{cases} S_n, & \text{when } V_n = 1, \\ D_{n+1}, & \text{when } V_n = -1, \end{cases} \\ \bar{L}_n(V_n) &= \begin{cases} A_{n+1}, & \text{when } V_n = 1, \\ C_n, & \text{when } V_n = -1. \end{cases} \end{aligned}$$

Then, $\bar{K}_n(V_n)$ can be considered the *nominal* service time of the n customer, while $\bar{L}_n(V_n)$ can be considered as the *nominal* interarrival time between customers n and $n+1$, because these would be the actual times if the state-dependent behavior were omitted. Moreover, they are both dependent on V_n , as well as on the state of the background Markov chain. Then, we have a modulated multiplicative Lindley recursion with $V_n = 1 + \bar{K}_n(V_n) - \bar{L}_n(V_n)$. When $V_n = 1$ (w.p. p), then $\bar{K}_n(1) = \bar{L}_n(1)$, while when $V_n = -1$ (w.p. $q = 1 - p$), $\bar{L}_n(-1) = 2 + \bar{K}_n(-1)$. See also [21, Sec. 5] for an application in scheduling arrivals in a production system.

2. MODEL DESCRIPTION

In this work, we focus on the stationary analysis of a Markov-dependent version of (1), where the distribution of the i.i.d random variables $\{Y_n(V_n)\}_{n \in \mathbb{N}_0}$ is regulated by an irreducible discrete-time Markov chain $\{Z_n\}_{n \in \mathbb{N}_0}$ with state space $E = \{1, 2, \dots, N\}$, one-step transition probability matrix $P := (p_{i,j})_{i,j \in E}$, and stationary distribution $\pi := (\pi_1, \dots, \pi_N)^T$.

Let $\Phi_{W,i}^n(s) := E(e^{-sW_n} 1_{\{Z_n=i\}})$, $i = 1, \dots, N$, $Re(s) \geq 0$, $n \geq 0$, and assuming the limit exists, let $\Phi_{W,i}(s) = \lim_{n \rightarrow \infty} \Phi_{W,i}^n(s)$, and $\tilde{\Phi}_W(s) := (\Phi_{W,1}(s), \dots, \Phi_{W,N}(s))^T$. We consider the following two model variants in [11]:
Model I: $P(V_n = 1) = p_1$, $P(V_n = a) = p_2$, $a \in (0, 1)$, $P(V_n < 0) = p_3 := 1 - p_1 - p_2$, $Y_n(V_n) := Y_n = S_n - A_{n+1}$. Let also for $n \geq 0$, $x, y \geq 0$, $i, j = 1, \dots, N$, $0 \leq r \leq n-1$:

$$\begin{aligned} P(A_{n+1} \leq x, S_n \leq y, Z_{n+1} = j | Z_n = i, (C_{r+1}, D_r, Z_r)) \\ = P(A_{n+1} \leq x, S_n \leq y, Z_{n+1} = j | Z_n = i) = p_{i,j} F_{S,i}(y) G_{A,j}(x), \end{aligned} \quad (2)$$

where $F_{S,i}(\cdot)$, $G_{A,j}(\cdot)$ denote the distribution functions of S_n , A_{n+1} , given $Z_n = i$, $Z_{n+1} = j$, respectively. Note that A_{n+1} , S_n , Z_{n+1} are independent of the past given Z_n , and A_{n+1} , S_n are conditionally independent given Z_n , Z_{n+1} . We assume that the sequences $\{A_n\}_{n \in \mathbb{N}}$ and $\{S_n\}_{n \in \mathbb{N}_0}$ are autocorrelated as well as cross-correlated.

Model II: $P(V_n = 1) = p$, $P(V_n = -1) = q := 1 - p$, and $Y_n(1) = S_n - A_{n+1}$, $Y_n(-1) = C_{n+1} - D_n$. Note that when $V_n = 1$, the dynamics among $\{A_n\}_{n \in \mathbb{N}}$, $\{S_n\}_{n \in \mathbb{N}_0}$ are governed by [2]. When $V_n = -1$, then, for $0 \leq r \leq n-1$

$$\begin{aligned} P(D_{n+1} \leq x, C_n \leq y, Z_{n+1} = j | Z_n = i, (D_{r+1}, C_r, Z_r)) \\ = P(D_{n+1} \leq x, C_n \leq y, Z_{n+1} = j | Z_n = i) = p_{i,j} F_{C,i}(y) G_{D,j}(x), \end{aligned}$$

where $F_{C,i}(\cdot)$, $G_{D,j}(\cdot)$ denote the distribution functions of C_n , D_{n+1} , given $Z_n = i$, $Z_{n+1} = j$, respectively. Note that D_{n+1} , C_n , Z_{n+1} are independent of the past given Z_n , and D_{n+1} , C_n are conditionally independent given Z_n , Z_{n+1} . We assume that the sequences $\{D_n\}_{n \in \mathbb{N}}$ and $\{C_n\}_{n \in \mathbb{N}_0}$ are autocorrelated as well as cross-correlated.

3. MODEL I

We consider the Markov-dependent mixed case, which is described by the recursion $W_{n+1} = [V_n W_n + S_n - A_{n+1}]^+$ under the Markov-dependent dynamics in [2], with $0 < p_i < 1$, $i = 1, 2, 3$, $p_1 + p_2 < 1$. Let us discuss the stability condition. We denote with X a generic random variable distributed as X_0 , and always assume that $E(|V|) < \infty$, $E(|Y|) < \infty$. For the scalar case, in [21] Theorem 1] it is shown that if $P(V < 0) > 0$ and $P(Y \leq 0) > 0$, then, W_n tends to a proper limit as $n \rightarrow \infty$. Note that for our case, V does not depend on the state of the background discrete-time Markov chain, although $P(V < 0) = p_3 > 0$. The other requirement is always satisfied since, in the simple case where $A|Z = j \sim \exp(\lambda_j)$, and $S|Z = j$ follow a general distribution with distribution function $B_j(\cdot)$, and LST $\beta_j^*(\cdot)$, $P(Y \leq 0) = \sum_{i=1}^N \sum_{j=1}^N \pi_i p_{i,j} \beta_j^*(\lambda_j) > 0$, thus, from the regenerative nature of W_n , the limit W is unique.

We further assume that $S|Z = i$ and $A|Z = j$ have a rational LST, with poles at $t_{1,i}, \dots, t_{m_i,i}$, $i = 1, \dots, N$, and $s_{1,j}, \dots, s_{l_j,j}$, $j = 1, \dots, N$, respectively, i.e.,

$$\Phi_{B,i}(s) := \frac{N_{B,i}(s)}{D_{B,i}(s)}, \quad \Phi_{A,j}(s) := \frac{N_{A,j}(s)}{D_{A,j}(s)},$$

with $Re(s_{r,j}) < 0$, $r = 1, \dots, l_j$, $j = 1, \dots, N$ and $Re(t_{k,i}) < 0$, $k = 1, \dots, m_i$, $i = 1, \dots, N$.

In such a setting, denote by $H_{i,j}(s) := E(e^{-sY_n} | Z_n = i, Z_{n+1} = j) = \frac{N_{Y,i,j}(s)}{D_{Y,i,j}(s)} = \frac{N_{B,i}(s) N_{A,j}(-s)}{D_{B,i}(s) D_{A,j}(-s)}$. Denote further the $N \times N$ matrix $H(s) := (H_{i,j}(s))_{i,j=1,\dots,N}$. Then, we have the following result.

THEOREM 1. For $s \in \Omega := \{s : Re(s) \geq 0, \det(I - p_1 F(s)) \neq 0\}$, the transform vector $\tilde{\Phi}_W(s)$ satisfies

$$\tilde{\Phi}_W(s) = R(s) \tilde{\Phi}_W(as) + \tilde{V}(s), \quad (3)$$

where $F(s) := P^T \circ H(s)$ and 'o' denotes the Hadamard product

$$\begin{aligned} R(s) &:= p_2(I - p_1 F(s))^{-1} F(s), \\ \tilde{V}(s) &:= (I - p_1 F(s))^{-1} \tilde{v}(s), \\ \tilde{v}(s) &:= \left(\frac{\sum_{\omega=0}^{l_1+\sum_{k=1}^N m_k} c_{\omega,1} s^\omega}{\prod_{k=1}^N D_Y^{(k,1)}(s)}, \dots, \frac{\sum_{\omega=0}^{l_N+\sum_{k=1}^N m_k} c_{\omega,N} s^\omega}{\prod_{k=1}^N D_Y^{(k,N)}(s)} \right)^T, \end{aligned}$$

where $c_{0,j} = \pi_j p_3 \prod_{k=1}^N D_Y^{(k,j)}(0)$. The solution to [3] is

$$\tilde{\Phi}_W(s) = \sum_{k=0}^{\infty} \prod_{m=0}^{k-1} R(a^m s) \tilde{V}(a^k s). \quad (4)$$

The remaining terms $c_{\omega,j}$, $\omega = 1, \dots, l_j + \sum_{k=1}^N m_k$, $j = 1, \dots, N$ are determined by solving the $\sum_{j=1}^N l_j + N \sum_{k=1}^N m_k$ linear equations given below:

$$\begin{aligned} p_2 \tilde{\zeta}_{r,j} F(\delta_{r,j}) \tilde{\Phi}_W(a \delta_{r,j}) &= -\tilde{\zeta} \tilde{v}(\delta_{r,j}), \\ p_3 \sum_{i=1}^N p_{i,j} N_Y^{(i,j)}(t_{f,k}) \prod_{\nu \neq i} D_{B,\nu}(t_{f,k}) \\ &\times \int_{-\infty}^0 \left[\sum_{v=0}^{\infty} \prod_{m=0}^{v-1} R(a^m t_{f,k} y) \tilde{V}(a^v t_{f,k} y) \right] \\ &\times P(V^- \in dy) = \sum_{\omega=0}^{l_j+\sum_{k=1}^N m_k} c_{\omega,j}(t_{f,k}) s^\omega, \end{aligned} \quad (5)$$

where $\delta_{r,j}$, $r = 1, \dots, l_j$, $j = 1, \dots, N$ are the $\sum_{j=1}^N l_j$ roots of $\det(I - p_1 F(s)) = 0$, such that $Re(\delta_{r,j}) > 0$, and $\tilde{\zeta}_{r,j}$, $r = 1, \dots, l_j$, $j = 1, \dots, N$, are nonzero row vectors satisfying $\tilde{\zeta}_{r,j}(I - p_1 F(\delta_{r,j})) = 0$.

PROOF. The detailed proof is given in [10]. $\square$

We can now obtain the steady-state moments. Let $m_i := \lim_{n \rightarrow \infty} E(W_1 | Z_n = i)$, $i = 1, \dots, N$, and $\tilde{m} = (m_1, \dots, m_N)^T$. Differentiate [3] with respect to s and let $s \rightarrow 0$. Then:

$$\tilde{m} = ((p_1 + ap_2)P^T - I)^{-1}(\Phi \tilde{\pi} + v'(0)),$$

where $\Phi = P^T \circ H'(0)$, where "r" denotes the first derivative.

4. MODEL II

We now focus on Model II, a Markov-modulated queue with service and interarrival times depending on waiting times as described in Section 2. We assume that $A_n | Z_n = j \sim \exp(\lambda_j)$, $D_n | Z_n = j \sim \exp(\mu_j)$, while $S_n | Z_n = j$, $C_n | Z_n = j$, are arbitrarily distributed with distribution functions $B_j(\cdot)$, and $C_j(\cdot)$, respectively. Denote also their LSTs with $\beta_j^*(s) = \int_0^\infty e^{-st} dB_j(t)$, $c_j^*(s) = \int_0^\infty e^{-st} dC_j(t)$, and let $\gamma_j = \int_0^\infty x dB_j(x)$, $\delta_j = \int_0^\infty x dC_j(x)$, $j = 1, \dots, N$.

In the following we discuss the stability condition. Following ideas from [21, 7], note that from [11], by replacing V_n with $\max\{0, V_n\}$, and $Y_n(V_n)$ with $\max\{0, Y_n(V_n)\}$, then, the resulting waiting times are least as large as those given in [11]. Then, exploiting ideas from [21], if W_n satisfies [11], then with probability 1, $W_n \leq Q_n$ for all n , where

$$Q_{n+1} = \max\{0, V_n\} Q_n + \max\{0, Y_n(V_n)\}, \quad n \geq 0, \quad (6)$$

and $Q_0 = W_0 \geq 0$. Thus, if W_n satisfies [11], Q_n satisfies [6], and Q_n converges to the proper limit Q , then $\{W_n\}_{n \in \mathbb{N}}$ is tight and $P(W > x) \leq P(Q > x)$ for all x , where W is the limit in distribution of any convergent subsequence of $\{W_n\}_{n \in \mathbb{N}}$. With that in mind, as well as [8] Theorem 1], implies that Q_n satisfying [6] converges to a proper limit if $P(\max\{0, V_n\} = 0) = P(V_n \leq 0) = q > 0$. Then, looking at [21] Theorem 1], the sequence $\{W_n\}_{n \in \mathbb{N}}$ is tight for all $\rho = \frac{E(\tilde{K}_0(V_0))}{E(L_0(V_0))} = \frac{\sum_{i=1}^N \pi_i (p\gamma_i + q\mu_i^{-1})}{\sum_{i=1}^N \pi_i (p\lambda_i^{-1} + q\delta_i)}$

and W_0 . If, in addition, $0 \leq p < 1$ and $\{(V_n, Y_n(V_n))\}_{n \in \mathbb{N}_0}$ given the state of the background discrete time Markov chain is a sequence of independent vectors with $P(V_0 \leq 0, Y_0(V_0)) > 0$, then the events $\{W_n = 0\}$ are regeneration points with finite mean time and $\{W_n\}_{n \in \mathbb{N}_0}$ converges in distribution to a proper limit W as $n \rightarrow \infty$ for all ρ and W_0 . Note that when $p = 1$, we need to have $\rho < 1$; [1]. Let $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_N)$, $M = \text{diag}(\mu_1, \dots, \mu_N)$, and $B^*(s) = \text{diag}(\beta_1^*(s), \dots, \beta_N^*(s))$, $H(s) = pB^*(s)P\Lambda$.

THEOREM 2. The transform row vector $\tilde{\Phi}_W^T(s)$ satisfies

$$\tilde{\Phi}_W^T(s)[H(s) + sI - \Lambda] = \tilde{v}(s), \quad (7)$$

where, $\tilde{\Phi}_W^T(0) = \tilde{\pi}^T$,

$$\begin{aligned} \tilde{v}(s) &= (s\tilde{k}_1 + s^2\tilde{k}_2 - q\tilde{\pi}\Lambda M)(M + sI)^{-1}, \\ \tilde{k}_l &= (k_{1,l}, \dots, k_{N,l}), l = 1, 2, \\ k_{j,1} &= q\pi_j\mu_j + \mu_j v_j^{(1)} - \lambda_j v_j^{(-1)}, j = 1, \dots, N, \\ k_{j,2} &= v_j^{(1)} + v_j^{(-1)}, j = 1, \dots, N, \\ v_j^{(1)} &= p \sum_{i=1}^N p_{i,j} \beta_i^*(\lambda_j) \Phi_{W,i}(\lambda_j), j = 1, \dots, N, \\ v_j^{(-1)} &= q[\pi_j - \sum_{i=1}^N p_{i,j} c_i^*(\mu_j) \Phi_{W,i}(\mu_j)], j = 1, \dots, N, \end{aligned}$$

PROOF. The detailed proof is given in [10]. $\square$

To fully specify $\tilde{\Phi}_W^T(s)$ we need to obtain $2N^2$ unknown terms, namely $\Phi_{W,i}(\lambda_j)$, $\Phi_{W,i}(\mu_j)$, $i, j = 1, \dots, N$, which are involved in the vectors $\tilde{k}_1, \tilde{k}_2$. This task is accomplished in the following steps. The first part of Theorem 3 (see [10] for the proof), gives the number and placement of the solutions of $\det(H(s) + sI - \Lambda) = 0$. That result is used in the second part to obtain N equations relating the elements of $\tilde{k}_1, \tilde{k}_2$.

THEOREM 3. 1. The equation $\det(H(s) + sI - \Lambda) = 0$, has exactly N zeros, say $s_1, \dots, s_N$, $\text{Re}(s_i) > 0$, $i = 1, \dots, N$. 2. Suppose that the N solutions s_i , $i = 1, 2, \dots, N$, with $\text{Re}(s_i) > 0$ stated above are distinct. Let $\tilde{a}_i = (a_{i,1}, \dots, a_{i,N})^T$ be a nonzero column vector satisfying $(H(s_i) + s_i I - \Lambda)\tilde{a}_i = 0$, $i = 1, \dots, N$, where $\tilde{0}$ is a row vector of zeros. Then, $\tilde{v}(s_i)\tilde{a}_i = 0$, or equivalently,

$$\sum_{k=1}^N v_k(s_i) a_{i,k} = 0, \quad (8)$$

where $\tilde{v}(s) = (v_1(s), \dots, v_N(s))$ with

$$v_j(s) = \frac{1}{\mu_j + s} [-q\pi_j \lambda_j \mu_j + s k_{j,1} + s^2 k_{j,2}], j = 1, \dots, N.$$

PROOF. The proof is given in [10]. $\square$

Expanding (7), and substituting $s = \mu_l$, $l = 1, \dots, N$, we obtain N^2 equations that relate $\Phi_{W,i}(\lambda_j)$, $\Phi_{W,i}(\mu_j)$, $i, j = 1, \dots, N$. More precisely, we have for $j, l = 1, \dots, N$:

$$\begin{aligned} &\lambda_j p \sum_{i=1}^N p_{i,j} \Phi_{W,i}(\mu_l) \beta_i^*(\mu_l) - (\lambda_j - \mu_l) \Phi_{W,j}(\mu_l) \\ &= \frac{1}{\mu_j + \mu_l} \left[\mu_l (\mu_l + \mu_j) p \sum_{i=1}^N p_{i,j} \Phi_{W,i}(\lambda_j) \beta_i^*(\lambda_j) \right. \\ &\quad \left. + \mu_l q (\lambda_j - \mu_l) \sum_{i=1}^N p_{i,j} \Phi_{W,i}(\mu_j) c_i^*(\mu_j) + (\mu_l + \mu_j) q \pi_j (\mu_l - \mu_j) \right]. \end{aligned} \quad (9)$$

The remaining $N(N-1)$ equations are obtained as follows: Expanding (7), and for each $j = 1, \dots, N$, substituting $s =$

λ_k , $k \neq j$, $k = 1, \dots, N$, we have:

$$\begin{aligned} &\lambda_j p \sum_{i=1}^N p_{i,j} \Phi_{W,i}(\lambda_k) \beta_i^*(\lambda_k) - (\lambda_j - \lambda_k) \Phi_{W,j}(\lambda_k) \\ &= \frac{1}{\mu_j + \lambda_k} \left[\lambda_k (\lambda_k + \mu_j) p \sum_{i=1}^N p_{i,j} \Phi_{W,i}(\lambda_j) \beta_i^*(\lambda_j) \right. \\ &\quad \left. + \lambda_k q (\lambda_k - \lambda_j) \left(\sum_{i=1}^N p_{i,j} \Phi_{W,i}(\mu_j) c_i^*(\mu_j) + (\lambda_k + \mu_j) \pi_j \right) \right]. \end{aligned} \quad (10)$$

Then, (8), (9), (10) constitute $2N^2$ linear equations for the $2N^2$ unknowns $\Phi_{W,i}(\lambda_j)$, $\Phi_{W,i}(\mu_j)$, $i, j = 1, \dots, N$. Once these terms are known, the vectors $\tilde{k}_1, \tilde{k}_2$ are known, and thus, $\tilde{v}(s)$ is known. This implies that $\tilde{\Phi}_W^T(s)$ is known. We are now ready to compute the moments of the steady-state waiting time distribution. For $i = 1, \dots, N$, denote:

$$\begin{aligned} m_{r,i} &= \lim_{n \rightarrow \infty} E(W_n^r 1_{\{Z_n=i\}}), \quad r = 0, 1, 2, \dots, \\ \tilde{m}_r &= (m_{r,1}, \dots, m_{r,N}), \quad r = 0, 1, 2, \dots, \\ \gamma_{r,i} &= \int_0^\infty x^r dB_i(x), \quad r = 0, 1, 2, \dots, \\ \Gamma_r &= \text{diag}(\gamma_{r,1}, \dots, \gamma_{r,N}), \quad r = 0, 1, 2, \dots \end{aligned}$$

Then, we have the following result (see [10] for the proof).

THEOREM 4. The moment-vectors $\tilde{m}_r$ satisfy the following recursive equations: $\tilde{m}_0 = \tilde{\pi}$,

$$\begin{aligned} \tilde{m}_1 &= [\tilde{m}_0 (p(P\Lambda - \Gamma_1 P\Lambda M) + M - \Lambda) - \tilde{k}_1] (\Lambda M)^{-1} (pP - I)^{-1}, \\ \tilde{m}_2 &= 2 [\tilde{k}_2 - \tilde{m}_1 (p(P\Lambda - \Gamma_1 P\Lambda M) + M - \Lambda) \\ &\quad - \tilde{m}_0 (p(\frac{1}{2}\Gamma_2 P\Lambda M - \Gamma_1 P\Lambda) + I)] (\Lambda M)^{-1} (pP - I)^{-1}, \end{aligned}$$

while for $r \geq 3$:

$$\begin{aligned} &\frac{(-1)^r}{r!} \tilde{m}_r (I - pP) \Lambda M = \frac{(-1)^{r-2}}{(r-2)!} \tilde{m}_{r-2} [p(\frac{1}{2}\Gamma_2 P\Lambda M - \Gamma_1 P\Lambda) + I] \\ &+ \frac{(-1)^{r-1}}{(r-1)!} \tilde{m}_{r-2} [p(P\Lambda - \Gamma_1 P\Lambda M) + M - \Lambda] \\ &+ p \sum_{j=0}^{r-3} \frac{(-1)^j}{j!} \tilde{m}_j \left[\frac{(-1)^{r-j-1}}{(r-j-1)!} \Gamma_{r-j-1} P\Lambda + \frac{(-1)^{r-j}}{(r-j)!} \Gamma_{r-j} P\Lambda M \right]. \end{aligned}$$

We now focus on an asymptotic result. Let $G(s) := H(s) + sI - \Lambda$, and $\text{Adj}(G(s))$ denotes its adjoint matrix. From Theorem 2 we know that

$$\tilde{\Phi}_W^T(s) = \tilde{v}(s) \frac{\text{Adj}(G(s))}{\det(G(s))}, \quad (11)$$

where $G(s)$, is a vector of analytic functions for $\text{Re}(s) > 0$. Assume that the LST $\beta_i^*(s)$ exists in a neighborhood of the origin. Then, due to (11), the functions $\Phi_{W,i}(s)$ are analytic for all s with $\text{Re}(s) > -R$, where $-R$ denotes the zero of $\det(G(s))$ with the largest real part in the negative halfplane. From the damping property of the Laplace transforms, we have $\mathcal{L}(e^{Rt} F_{W,i}(t)) = \Phi_{W,i}(s - R)$, $i = 1, \dots, N$, where $F_{W,i}(t)$ denotes the distribution of W when the background Markov chain is in state i , and $\mathcal{L}(g(t))$ denotes the LST of the function $g(t)$. Given that the limit exists, $\lim_{t \rightarrow \infty} e^{Rt} \tilde{F}_W^T(t) = \lim_{s \rightarrow 0} s \tilde{\Phi}_W^T(s - R) = \tilde{C}$, where $\tilde{F}_W^T(t) = (F_{w,1}(t), \dots, F_{w,N}(t))$. Let $s = -R$ be a simple pole of $\Phi_{W,i}(s)$, then we obtain, using de L'Hopital rule:

$$\tilde{C} = \frac{\tilde{v}(-R) \text{Adj}(G(-R))}{\frac{d}{ds} \det(G(s))|_{s=-R}}. \quad (12)$$

Thus, the waiting time distribution decays exponentially at rate R , and the corresponding constants are given in [12].

5. A SIMPLE NUMERICAL EXAMPLE

Due to space limitations, we just numerically illustrate the theoretical findings in Sections 3, 4. Let $N = 2$, and

$\Lambda = \text{diag}(2, 3)$, $M = \text{diag}(\mu_1 = 10/u, \mu_2 = 8/u)$, $u = 1, 2, 3$, $A|Z = j \sim \exp(\lambda_j)$, $j = 1, 2$.

Regarding Model I: Here we assume that $a = 0.2$, $S|Z = j \sim \exp(\mu_j)$. The parameter u is used to explore the effect of increasing the expected service duration on the mean workload. We compare the case where interarrival and service times are autocorrelated i.e., where $P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, and the case where there is no autocorrelation, i.e., where $P = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$. In such a scenario, we have positive cross-correlation between the interarrival and service times. Figure 1 shows that the mean workload is higher when there is autocorrelation, and increases as u increases.

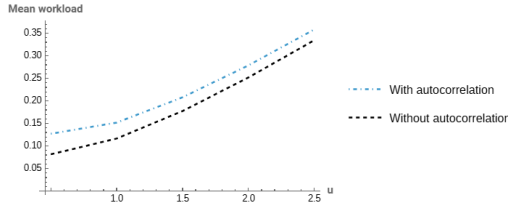


Figure 1: Mean workload vs u .

Regarding Model II: For convenience, we assume that $S|Z = j \sim \exp(\mu_j)$, and $C|Z = j \sim \exp(\theta_j)$, $j = 1, 2$, with $\theta_1 = 8$, $\theta_2 = 6$. $P = \begin{pmatrix} 0.2 & 0.8 \\ 0.6 & 0.4 \end{pmatrix}$. Figure 2 shows the expected waiting time as a function of p , for increasing values of u . Note that as u increases, the mean waiting time also increases, since the mean service time increases. That increase becomes more apparent as p increases, that is, when our model is more likely to operate as a regular MAP/G/1 queue rather than an alternating service model.

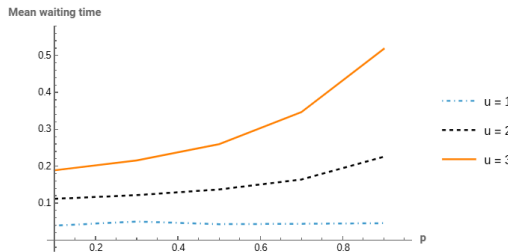


Figure 2: Mean waiting time vs p .

Acknowledgements

The author gratefully acknowledges the Empirikion Foundation, Athens, Greece (www.empirikion.gr) for the financial support of this work. This work is dedicated to the memory of Antonis Kontis.

6. REFERENCES

- [1] I. Adan and V. Kulkarni. Single-server queue with Markov-dependent inter-arrival and service times. *Queueing Systems*, 45:113–134, 2003.
- [2] S. Asmussen. *Applied probability and queues*. Springer, New York, 2003.
- [3] O. Boxma, A. Löpker, and M. Mandjes. On two classes of reflected autoregressive processes. *Journal of Applied Probability*, 57(2):657–678, 2020.
- [4] O. Boxma, A. Löpker, M. Mandjes, and Z. Palmowski. A multiplicative version of the Lindley recursion. *Queueing Systems*, 98:225–245, 2021.
- [5] O. Boxma and M. Mandjes. Queueing and risk models with dependencies. *Queueing Systems*, 102(1-2):69–86, 2022.
- [6] O. Boxma, M. Mandjes, and J. Reed. On a class of reflected AR(1) processes. *Journal of Applied Probability*, 53(3):818–832, 2016.
- [7] O. J. Boxma and M. Vasiou. On queues with service and interarrival times depending on waiting times. *Queueing Systems*, 56(3):121–132, 2007.
- [8] A. Brandt. The stochastic equation $Y_{n+1} = A_n Y_n + B_n$ with stationary coefficients. *Advances in Applied Probability*, 18(1):211–220, 1986.
- [9] D. Buraczewski, E. Damek, and T. Mikosch. *Stochastic models with power-law tails: The equation $X = AX + B$* . Springer, Cham, 2016.
- [10] I. Dimitriou. On a class of multiplicative Lindley-type recursions with Markov-modulated dependencies. *arXiv preprint arXiv:2508.20495 [math.PR]*, 2025.
- [11] I. Dimitriou. On Markov-dependent reflected autoregressive processes and related models. *Queueing Systems*, 109(3):1–39, 2025.
- [12] I. Dimitriou and D. Fiems. Some reflected autoregressive processes with dependencies. *Queueing Systems*, 106(1):67–127, 2024.
- [13] D. Fiems and E. Altman. Markov-modulated stochastic recursive equations with applications to delay-tolerant networks. *Performance Evaluation*, 70(11):965–980, 2013.
- [14] D. Huang. On a modified version of the Lindley recursion. *Queueing Systems*, 105(3):271–289, 2023.
- [15] P. Jacquet. Subexponential tail distribution in LaPalice queues. In *Proceedings of the 1992 ACM SIGMETRICS joint International Conference on Measurement and Modeling of Computer Systems*, pages 60–69, 1992.
- [16] J. Le Bihan and B. Kołodziejek. Perpetuities with light tails and the local dependence measure. *Stochastic Processes and their Applications*, page 104740, 2025.
- [17] A. Pacheco, L. C. Tang, and N. U. Prabhu. *Markov-Modulated Processes and Semiregenerative Phenomena*. WORLD SCIENTIFIC, 2008.
- [18] B. C. Park, J. Y. Park, and R. D. Foley. Carousel system performance. *Journal of Applied Probability*, 40(3):602–612, 2003.
- [19] M. Vasiou. *Lindley-type recursions*. PhD thesis, Technische Universiteit Eindhoven, 2006.
- [20] M. Vasiou, I. Adan, and O. Boxma. A two-station queue with dependent preparation and service times. *European Journal of Operational Research*, 195(1):104–116, 2009.
- [21] W. Whitt. Queues with service times and interarrival times depending linearly and randomly upon waiting times. *Queueing Systems*, 6:335–351, 1990.

In situ fine-tuning of in silico trained Optical Neural Networks

Gianluca Kosmella
Department of Electrical
Engineering
TU Eindhoven
g.k.kosmella@tue.nl

Ripalta Stabile
Department of Electrical
Engineering
TU Eindhoven
r.stabile@tue.nl

Jaron Sanders
Department of Mathematics &
Computer Science
TU Eindhoven
jaron.sanders@tue.nl

ABSTRACT

Optical Neural Networks (ONNs) offer ultrafast computation, high bandwidth, and low energy use by exploiting photonic hardware. Training, however, remains difficult since parameters learned on simplified *in silico* models often degrade when transferred to physical devices with noise and fabrication imperfections.

We study how noise misspecification during *in silico* training impacts performance and propose *Gradient-Informed Fine-Tuning* (GIFT). This is a lightweight *in situ* algorithm that adapts pretrained parameters. GIFT requires no full re-training or elaborate setups and comes with mathematical guarantees on when it improves performance.

Simulations on a five-layer ONN for MNIST classification show that GIFT yields up to 28% relative accuracy improvement under noise misspecification. Thus, GIFT offers a practical means to close the gap between digital models and physical ONNs.¹

1. INTRODUCTION

Empirically, Neural Network (NN) performance has been seen to follow a power law relationship that depends on the dataset size and model scale [13, 14]. Consequently, improving performance requires ever larger datasets and more complex models. These, in turn, demand significantly greater computational resources. However, advancements in compute performance are plateauing, driving a surge of research into alternative computing hardware solutions. Optical Neural Networks (ONNs) offer the promise of ultra-fast and energy-efficient computing due to the speed of light processing and the high bandwidth [35, 29, 25].

Despite these advantages, ONNs face several unique challenges that complicate their deployment. Noise, in particular, is a fundamental issue, arising from sources such as signal encoding, transmission, and processing, which can degrade network performance. Additionally, device-level imperfections, such as phase errors in Mach-Zehnder Interferometer (MZI) meshes, thermal crosstalk, and fabrication inconsistencies, introduce further variability that affects system performance [30, 18, 11]. Moreover, the high-precision and high-speed requirements of Digital-Analog Conversions

(DACs) and Analog-Digital Conversions (ADCs) add significant overhead in analog-digital architectures [15, 11].

A widely adopted training approach for ONNs is the *in silico* training, which involves training a digital model that simulates the physical hardware, allowing backpropagation to be performed computationally before mapping the optimized parameters to the actual optical hardware [37, 1, 42], where only inference can be executed. Effective *in silico* training requires accurate modeling of the ONN, including the activation functions (e.g. sigmoidal [28] or sinusoidal [31]), initialization schemes to avoid saturation, and compensation for quantization and noise effects in Photonic Integrated Circuits (PICs) [30, 16]. Nevertheless, *in silico* models are prone to model misspecification. Even small discrepancies between the simulated model and the physical hardware can accumulate and become amplified as they propagate through the network's layers [4].

Hybrid approaches have been proposed to mitigate misspecifications between model and physical ONN by performing forward passes directly on the ONN (*in situ*) while retaining backpropagation in a digital model [46, 42]. While this method reduces errors from model inaccuracies, it requires a complex interface to integrate feedback from physical hardware into the training process. Figure 1(left, middle) illustrate the *in silico* and hybrid training approaches, respectively.

While previous work has focused on adapting training algorithms to the unique properties of ONNs [11, 30], we propose a fine-tuning method that is largely agnostic to the specific model. This ensures reliable performance in a wide range of real-world applications.

Specifically, we give a novel *in situ* fine-tuning method to mitigate errors from model misspecification in *in silico*-trained ONNs, nicknamed GIFT. Unlike previous approaches, our method does not require incorporating hardware feedback into the digital model. Instead, we focus on directly fine-tuning the weights on the physical hardware to correct for discrepancies between digital model and physical hardware, as illustrated in Figure 1(right). This approach is motivated by the observation that parameters learned through *in silico* training are often close to the optimal parameters for the physical system, showing the need for a sophisticated retraining algorithm. Using our method, fair minimizers are first identified through standard *in silico* training, and then fine-tuned towards better minimizers directly on the ONN platform, *in situ*. To the best of our knowledge, this form of device-level fine-tuning is novel and offers a practical, computationally efficient solution for real-world ONN deployment.

¹The extended version of this paper, [20], is available at <https://arxiv.org/abs/2506.22122>.

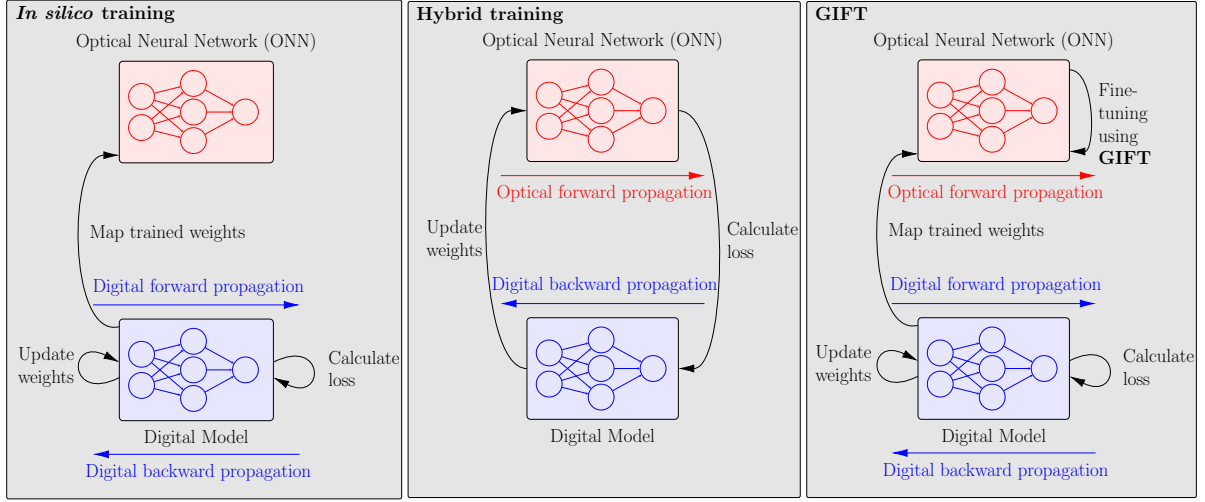


Figure 1: Comparison of training approaches. *Left:* Full *in silico* training, where both forward and backward passes occur in a digital model. *Middle:* Hybrid training, where optical forward passes are backpropagated through a digital model. *Right:* Gradient-Informed Fine-Tuning (GIFT), which fine-tunes an *in silico* trained model using optical forward passes, without requiring the complex experimental setup of hybrid training.

1.1 Related literature

Noise in ONNs arises from data encoding and signal processing (DACs/ADCs, laser fluctuations, modulator resolution), thermal crosstalk, cumulative optical communication noise, and nonlinear activation functions [5, 22, 4, 33, 37]. Device-level components such as microring weight banks, MZI meshes, and Semiconductor Optical Amplifier (SOA)-based networks introduce distinct noise, often modeled as Additive White Gaussian Noise (AWGN), as do fabrication errors in components like directional couplers.

To address these noise issues, strategies include quantization-aware methods [11, 41, 47, 15], noise-aware and hardware-in-the-loop training to improve robustness [45, 10, 38], loss-landscape smoothing via gradient penalties [17, 43, 40], and hardware/circuit-level approaches such as pruning, weight blocks, duplication, and design optimizations to reduce variation and crosstalk [12, 9, 19, 27].

Integrated photonics offers an ultra-low latency and energy-efficient platform for ONNs, through on-chip integration [36, 6, 39]. Key building blocks include MZI-meshes for linear transformations, while techniques such as Wavelength-Division Multiplexing (WDM) enhance parallelism and bandwidth [3, 8, 7, 24]. Optical nonlinearities have been demonstrated with photodetectors, saturable absorbers, and SOAs among others, supporting the implementation of diverse architectures from Multi-Layer Perceptrons (MLPs) to more advanced Convolutional Neural Networks (CNNs) and Spiking Neural Networks (SNNs) [1, 2, 26, 44]. The growing availability of standardized Process Design Kits (PDKs) paves the way for full-scale photonic integration [23].

2. SUMMARY OF RESULTS

A physical ONN is inherently noisy and thus, for given weights w and some input x , the ONN's output $\Phi^{\text{ONN}}(x, w)$ is a random variable with certain distribution. Ideally, a model M of the physical ONN incorporates a noise distribution $\mathcal{N}$ such that for all x and w and some realization $\mathbf{N} \sim \mathcal{N}$ of the modeled noise, its output $M(x, w, \mathbf{N})$ has the

exact same distribution as $\Phi^{\text{ONN}}(x, w)$. In this ideal scenario, an optimal w^* for the model M will also be optimal for the ONN Φ^{ONN} . However, a model is a mere approximation of reality and there is typically model misspecification.

In [20, Section 2.1], we construct a function M that can at the same time describe a noisy *in silico* NN and model an *in situ* ONN. There, the presumed noise distribution $\mathcal{N}_s$ includes a *noise level parameter* $s \in (0, \infty)$ that effectively determines the variance of the present AWGN. Our construction is such because we care about the scenario in which a practitioner has an *estimate noise level* $s_0 \in (0, \infty)$ of some *underlying hidden, unknown and true noise level* $s_t \in (0, \infty)$, and such that $s_0 \approx s_t$ and $s_0 \neq s_t$.

2.1 Result 1: Characterization of the minimizers under L2-loss

We assume that a practitioner is training M using a projected Stochastic Gradient Descent (SGD) algorithm on L_2 -loss *in silico*. This means specifically that the practitioner is iterating

$$w^{\{k+1\}} = p_H(w^{\{k\}} - \varepsilon_k \nabla_w (y^{\{k\}} - M(x^{\{k\}}, w^{\{k\}}, \mathbf{N}^{\{k\}}))^2) \quad (1)$$

in silico. Here, the $w^{\{0\}}$ denote initial weights, $\mathbf{N}^{\{k\}} \sim \mathcal{N}_s$ noise realizations, $(x^{\{k\}}, y^{\{k\}}) \sim \mu$ data samples, and $\varepsilon_k > 0$ step sizes.

For $s \in (0, \infty)$, let $\mathcal{N}_s$ refer specifically to the noise distribution implied by [20, (15)] in [20, Section 2.1], and μ a data distribution on $\mathbb{R}^{d_0} \times \mathbb{R}^{d_L}$. Furthermore, let

$$\mathcal{J}_s(w) := \int \int (y - M(x, w, \mathbf{N}))^2 d\mathcal{N}_s(\mathbf{N}) d\mu(x, y) \quad (2)$$

be the *objective function at level s*. The fact that Equation (2) really is the objective function is implied by our first result: the exact description of the limiting behavior of Equation (1) in Propositions 1 and 2 of [20, Section 2.1]. Informally stated, for one for example, Proposition 2 [20] implies:

RESULT 1 (INFORMAL). For any $s \in (0, \infty)$, under typical assumptions, the limit of Equation (1) converges to a point in the set $\{w : \nabla_w \mathcal{J}_s = 0\}$.

The proofs of Propositions 1 and 2 [20] are modifications of proofs in [34], and ultimately rely on the so-called Ordinary Differential Equation (ODE) method [21]. The proof parts that deal with the AWGN are new.

Now that we have identified in [20, (15)] what the objective function is, and what the limit points of Equation (1) are, we can tackle the problem of model misspecification.

2.2 Result 2: Minimizers from a misspecified model can be improved

Assume now that $s_0 \neq s_t$, i.e., that there is model misspecification. Our second result, Theorem 1, gives conditions under which there is in fact scope for improvement. This is made explicit by the strict inequality in Equation (5). Its proof is given in [20, Section 3.3.1]:

THEOREM 1. Let $s_0, s_t \in (0, \infty)$. Assume that

$$\begin{aligned} w_0 \in \{w : \nabla_w \mathcal{J}_{s_0}(w) = 0\} \neq \emptyset \quad \text{and} \\ w_t \in \{w : \nabla_w \mathcal{J}_{s_t}(w) = 0\} \neq \emptyset. \end{aligned} \quad (3)$$

The following then holds: if for all $\zeta \in (s_0 \wedge s_t, s_0 \vee s_t)$

$$0 < |s_t - s_0| < \frac{\left\| \frac{\partial}{\partial s} \nabla_w \mathcal{J}_{s_0}(w_0) \right\|_2}{\frac{1}{2} \left| \left(\frac{\partial^2}{\partial s^2} \nabla_w \mathcal{J}_\zeta(w_0) \right)^T \left(\frac{\partial}{\partial s} \nabla_w \mathcal{J}_{s_0}(w_0) \right) \right|}, \quad (4)$$

then $w_0 \neq w_t$. Moreover,

$$\mathcal{J}_{s_t}(w_t) < \mathcal{J}_{s_t}(w_0). \quad (5)$$

2.3 Result 3: GIFT improves minimizers from a misspecified model

Our third result is Algorithm 1, nicknamed GIFT. Importantly, GIFT is designed to be implementable *in situ*. Specifically, it takes *in silico* estimates of opportune directions,

$$\begin{aligned} D_{W^{(\ell)}}^{[0]}(K_1, K_2) &= \frac{1}{K_1 K_2} \sum_{k=1}^{K_1} \sum_{m_k=1}^{K_2} R^{(\ell), \{k\}} (A^{(i-1), \{k\}})^T \\ &\times \left(\sum_{\alpha \in \mathcal{S}} (s_0^{-2} (N^{\alpha, \{m_k\}})^T N^{\alpha, \{m_k\}} - d_{f(\alpha)}) \right) \end{aligned} \quad (6)$$

and

$$\begin{aligned} D_{b^{(\ell)}}^{[0]}(K_1, K_2) &= \frac{1}{K_1 K_2} \sum_{k=1}^{K_1} \sum_{m_k=1}^{K_2} R^{(i), \{k\}} \\ &\times \left(\sum_{\alpha \in \mathcal{S}} (s_0^{-2} (N^{\alpha, \{m_k\}})^T N^{\alpha, \{m_k\}} - d_{f(\alpha)}) \right) \end{aligned} \quad (7)$$

(see Line 2), along which an *in situ* search is then performed (see Lines 5 to 9). The candidate point along the direction that scores best according to the *in situ* evaluation function

$$\text{Eval}_w(K_1, K_2) = \frac{1}{K_1 K_2} \sum_{k=1}^{K_1} \sum_{m_k=1}^{K_2} (y^{\{k\}} - \Phi^{\text{ONN}}(x^{\{k\}}, w))^2 \quad (8)$$

is returned by the algorithm. For GIFT, we prove the following performance guarantee in [20, Section 3.3.2]:

Algorithm 1 Gradient-Informed Fine-Tuning (GIFT)

Require: An estimate $s_0 \in (0, \infty)$ of s_t , parameters $\eta > 0$, $K_1, K_2 \in \mathbb{N}_+$, and a performance evaluation function $\text{Eval}_w(K_1, K_2)$ evaluable *in situ*

Initialization:

- 1: Calculate an approximate minimizer $w_0 \in \{w : \nabla_w \mathcal{J}_{s_0}(w) = 0\}$, e.g. via Equation (1), *in silico*
- 2: Calculate an estimate of $(\partial/\partial s) \nabla_w \mathcal{J}_s(w)$, say $D^{[0]}(K_1, K_2)$, *in silico*

Directional search:

- 3: Initialize $w^{[0]} \leftarrow w_0$, $i \leftarrow 0$
- 4: Evaluate $L^{[0]} \leftarrow \text{Eval}_{K_1, K_2}(w^{[0]})$ *in situ*
- 5: **repeat**
- 6: Calculate $w^{[i+1, \pm]} \leftarrow w^{[i]} \pm \eta D^{[0]}$
- 7: Evaluate $L^{[i+1, \pm]} \leftarrow \text{Eval}_{K_1, K_2}(w^{[i+1, \pm]})$ *in situ*
- 8: Increment $i \leftarrow i + 1$
- 9: **until** $L^{[i, +]} \vee L^{[i, -]} \geq L^{[0]}$

Finalization:

- 10: **Return**
- 11: $w_f \in \arg \min_{w \in \{w^{[c, \cdot]} | c \in [i], \cdot \in \{+, -\}\}} \{\text{Eval}_{K_1, K_2}(w)\}$

THEOREM 2. Let $s_0, s_t \in (0, \infty)$. Let the fine-tuned weights output by Algorithm 1 be

$$w_f = \text{GIFT}(s_0; \eta, K_1, K_2). \quad (9)$$

Assume that Equations (3) and (4) hold and that $w_0 \in \{w : \nabla \mathcal{J}_{s_0}(w) = 0\}$. Then, for sufficiently small η and sufficiently large K_1, K_2 ,

$$\mathcal{J}_{s_t}(w_f) < \mathcal{J}_{s_t}(w_0). \quad (10)$$

2.4 Conceptual explanation of Theorem 1

We found inspiration for Theorems 1 and 2 in [32]; a paper that establishes the existence of optimality gaps in appropriately dimensioned large many-server systems. As in [32], but for different reasons, there exists a performance gap when optimizing a misspecified ONN. Let us explain this using Figure 2 for a one-dimensional, stylized example.

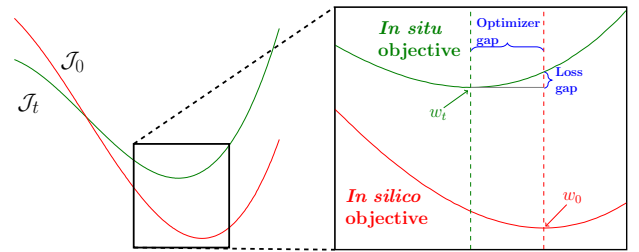


Figure 2: (Left) Schematic depictions of $\mathcal{J}_{s_0}$ and $\mathcal{J}_{s_t}$ for some $s_0 \neq s_t$. (Right) The curves around the minimizers of the *in situ* and *in silico* objectives.

In Figure 2, illustrations of $\mathcal{J}_0$ and $\mathcal{J}_t$ are shown in red and green, respectively, for some $s_0 \neq s_t$ while also $s_0 \approx s_t$. Notice that the curves are not identical because $s_0 \neq s_t$, but that they are close to each other because $s_0 \approx s_t$. Notice also that the minimum of the green curve is strictly less than its value at the minimizer of the red curve. This is the content of Theorem 1. Thus, in short, a better minimizer and better minimum can be found when there is model misspecification. Finally, observe that by moving away from the *in silico* minimizer w_0 (say experimentally towards the left and

right) and evaluating $\mathcal{J}_t$'s value by trial and error, one can indeed be led towards the actual minimizer w_t . GIFT builds on this intuition; and the fact that this idea works, is the content of Theorem 2.

Recall, however, that the stylized example in Figure 2 is one dimensional. The parameter space of a NN typically is of a much higher dimension; and the direction in which to adjust the parameters via trial-and-error is not obvious. To address this, Algorithm 1 calculates the direction of error descent as a function of the noise level s *in silico*. Algorithm 1 then explores this direction *in situ* by testing along the one-dimensional line opened by this calculated direction.

To make condition (4) in Theorem 1 more transparent, we explicitly evaluate the right-hand side of the bound for a linear network and a specific learning task in [20, Section 1.1.5]. There we find that the range specified by (4) is quite generous. In a specific scaling case, finetuning is always successful when the true system experiences less noise than assumed during training ($s_t < s_0$). Moreover, if the true noise exceeds the assumed noise in training ($s_0 < s_t$), finetuning remains guaranteed for all $s_t < 1$, even when there was no noise in training.

3. SIMULATION STUDY ON GIFT

To evaluate the effectiveness of Algorithm 1, we conducted experiments on two NN architectures: a shallower NN with layer dimensions 784–500–100–100–10 and a deeper NN with layer dimensions 784–500–250–250–100–50–10. Both architectures use hyperbolic tangent activation functions and were trained on the Modified National Institute of Standards and Technology (MNIST) classification task. The training algorithm closely follows (1), with two key deviations: the projection step is omitted and updates are performed for batches of $(x^{\{k\}}, y^{\{k\}})$ pairs rather than individual samples.

All experiments were implemented using a modified PyTorch framework and executed on systems equipped with NVIDIA Tesla V100-PCIE-16GB and NVIDIA GeForce RTX 2080 Ti GPUs.

We first apply GIFT to obtain fine-tuned weights and evaluate training performance, analyzing results for the deeper network under both noise misspecification in the AWGN model (Section 3.1) and when the AWGN assumption is violated (Section 3.2). Additional simulations assess GIFT on a shallower network [20, Fig. 4] and examine whether the fine-tuned weights w_f generalize to unseen test data [20, Fig. 6]. For each setting, we trained 10 networks per training noise level, each with different random initialization.

3.1 GIFT Performance on the Deeper NN

Figure 3 illustrates the performance improvements obtained by applying GIFT to the deeper NN.

Observe in Figure 3 that both the loss and accuracy percentage improvements increase as the magnitude of the injected noise (i.e., the variance of the AWGN used during training) increases. This positive correlation holds true irrespective of whether the subsequent true noise level is higher or lower than the noise level assumed during training. However, the trends in loss improvement and accuracy improvement exhibit distinct behaviors relative to the true noise level: the largest percentage reductions in the loss occur when the true noise level is low, whereas the greatest accuracy improvements occur when the true noise level is high.

A plausible explanation for this observed divergence arises from the fundamental differences between the accuracy and

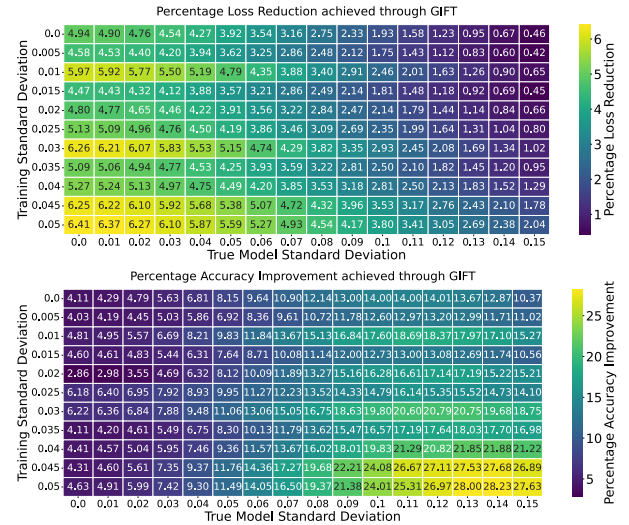


Figure 3: Percentage improvements in loss (Mean Squared Error (MSE)) and accuracy achieved with GIFT (Algorithm 1) compared to the baseline (no fine-tuning) for the deeper NN, under varying training (s_0) and true (s_t) noise standard deviations.

MSE metrics. Accuracy is threshold-based and hence, in high-noise conditions, even slight adjustments to decision boundaries can correct misclassifications, resulting in substantial relative improvements.

On the other hand, MSE measures the average squared deviation and naturally increases under higher noise due to increased random fluctuations. Consequently, even if the absolute reduction in loss remains constant across noise levels, the relative (percentage) improvement appears smaller when the baseline loss is higher.

The opposite reasoning applies for accuracy. As noise increases, baseline accuracy deteriorates, amplifying the relative importance of absolute improvements. Therefore, the largest relative accuracy improvements tend to appear in high-noise environments, even when the relative improvement in loss appears diminished.

Notably, even when $s_0 = s_t$, following the gradient direction $D^{[0]}$ in Algorithm 1 still results in substantial loss improvements. This occurs because $D^{[0]}$ reflects the gradient with respect to the full (available training) data distribution. In contrast, the original training procedure relies on iterative updates from small mini-batches; cf. the introduction to this section. GIFT applies a structured gradient-informed adjustment that follows the gradient direction as calculated for the whole data distribution.

3.2 GIFT's performance when the noise is not AWGN

To test the robustness of GIFT under different noise assumptions, we replaced AWGN with uniform noise $\mathcal{U}(-s_t, s_t)$, multiplicative Gaussian noise with standard deviation s_t , and Laplace-distributed noise with parameters 0 and s_t . Performance results for these scenarios are presented in Figure 4. Shown on the top is the loss reduction and at the bottom accuracy improvement.

Because the selection step in GIFT (line 11 in Algorithm 1) includes the initial weights w_0 as a baseline, the improvement in loss is inherently nonnegative. This guarantees that the fine-tuned model will not perform worse than the initial

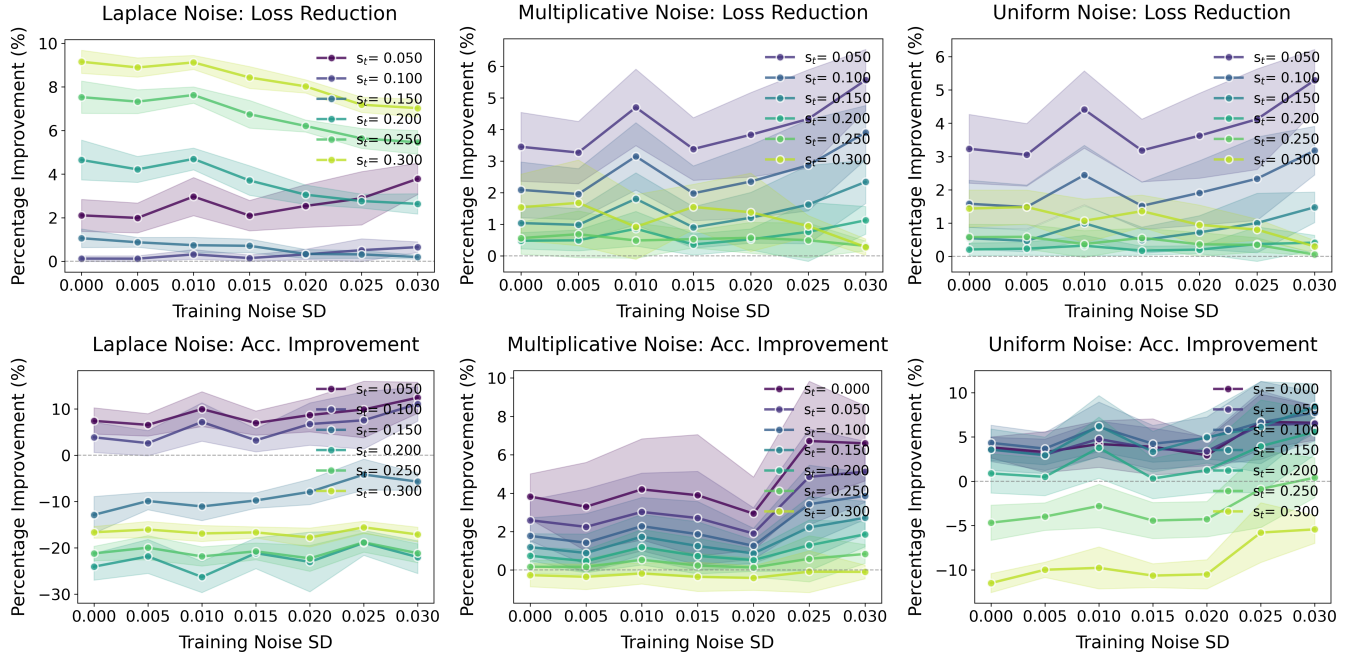


Figure 4: Percentage improvements in loss (top) and accuracy (bottom) achieved by Algorithm 1 compared to the baseline (no fine-tuning) for the deeper network under violations of the AWGN assumption. Results are shown for Laplace-distributed noise (left), multiplicative Gaussian noise (middle), and uniformly distributed noise (right). Confidence intervals represent 95%.

one, which is clearly observed in Figure 4.

Accuracy improvements (bottom row), however, require a more nuanced interpretation. In several configurations, percentage accuracy gains of approximately 10% are achievable. Yet, caution is needed, as in certain extreme cases of double misspecification (both variance and noise distribution deviating significantly from assumptions), accuracy can degrade substantially—exceeding 25% deterioration. Nonetheless, provided that the double misspecification remains moderate, GIFT consistently delivers meaningful improvements.

These results demonstrate that GIFT exhibits notable robustness even when the AWGN assumption is violated. Thus, the proposed algorithm can effectively be applied in physical ONNs systems, which typically exhibit more complex or mixed noise characteristics beyond the idealized Gaussian assumptions.

4. CONCLUSION

This work introduces GIFT (Algorithm 1) which is a lightweight and effective method for mitigating the impact of misspecification in *in silico*-trained ONNs by finetuning *in situ*. GIFT refines trained models using precomputed information that captures the interplay between noise and network structure, relying solely on on-chip inference to adjust network parameters. As a result, GIFT operates without requiring on-chip retraining, which is either infeasible in practical settings or requires complex interfaces. This makes GIFT an experimentally simple yet effective solution for adapting models to unpredictable noisy real-world conditions. Our theoretical and simulation results demonstrate its efficacy. In short, the key contributions of this paper are:

- Mathematical results establishing conditions under which GIFT improves ONNs (see Sections 2 and 3 in [20]). We provide a formal analysis of noise-aware

ONN training and theoretically prove that GIFT Algorithm 1 leads to improvements (Theorems 1 and 2).

- Empirical insights into how the magnitude of noise misspecification influences fine-tuning success (Section 3), including observed accuracy improvements of up to 28% on MNIST classification. Our results in [20] show that deeper ONNs benefit most from GIFT.

Beyond its demonstrated performance gains, GIFT also showed robustness across various noise types, including Laplace, multiplicative Gaussian, and uniform noise; despite it having been designed under the AWGN assumption. This suggests that GIFT may generalize well beyond the original training assumptions. We further hypothesize that GIFT may also compensate for mapping inaccuracies and fabrication-induced imperfections, making it a promising direction for practical ONN deployment.

Taken together, the theoretical and experimental findings presented here introduce GIFT as the first *in situ* fine-tuning approach for *in silico* trained ONNs with a proven ability to improve models trained under misspecified AWGN noise. As such, GIFT represents a valuable step toward scalable, reliable integration of ONNs in real-world applications.

Having established the foundational efficacy of GIFT, future work can explore possible extensions to other noise models. For example, one may consider more complex and realistic noise models, and device-specific distortions in ONNs. Showcasing GIFT’s usefulness in a system-level simulator or in proper hardware implementations can further underscore the effectiveness of GIFT.

Acknowledgment. This research was supported by the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement no. 945045, and by the NWO Gravitation project NETWORKS under grant no. 024.002.003.

5. REFERENCES

- [1] F. Ashtiani, A. J. Geers, and F. Aflatouni. An on-chip photonic deep neural network for image classification. *Nature*, 606(7914):501–6, 2022.
- [2] Y. Chen, M. Nazhamaiti, H. Xu, et al. All-analog photoelectronic chip for high-speed vision tasks. *Nature*, 623(7985):48–57, 2023.
- [3] W.R. Clements, P.C. Humphreys, B.J. Metcalf, et al. Optimal design for universal multiport interferometers. *Optica*, 3(12):1460–1465, 2016.
- [4] T. F. de Lima, A. N. Tait, H. Saeidi, et al. Noise analysis of photonic modulator neurons. *IEEE JSTQE*, 26(1):1–9, 2020.
- [5] R.-J. Essiambre, G. Kramer, P. J. Winzer, et al. Capacity limits of optical fiber networks. *J. Lightwave Technol.*, 28(4):662–701, 2010.
- [6] J. Feldmann, N. Youngblood, M. Karpov, et al. Parallel convolutional processing using an integrated photonic tensor core. *Nature*, 589(7840):52–58, 2021.
- [7] C. Feng, J. Gu, H. Zhu, et al. A compact butterfly-style silicon photonic-electronic neural chip for hardware-efficient deep learning. *ACS Photonics*, 9(12):3906–3916, 2022.
- [8] S.A. Fldzhyan, M.Y. Saygin, and S.P. Kulik. Optimal design of error-tolerant reprogrammable multiport interferometers. *Optics Letters*, 45(9):2632–2635, 2020.
- [9] J. Gu, Z. Zhao, C. Feng, et al. Towards area-efficient optical neural networks: an FFT-based architecture. In *Proc. 2020 ASP-DAC*, pages 476–81. IEEE, 2020.
- [10] J. Gu, Z. Zhao, C. Feng, et al. O2NN: Optical Neural Networks with Differential Detection-Enabled Optical Operands. In *Proc. 2021 DATE*, pp. 1062–67, 2021.
- [11] J. Gu, Z. Zhao, C. Feng, et al. ROQ: A Noise-Aware Quantization Scheme Towards Robust Optical Neural Networks with Low-bit Controls. In *Proc. 2020 DATE*, pages 1586–1589, 2020.
- [12] J. Gu, H. Zhu, C. Feng, et al. ADEPT: automatic differentiable DEsign of photonic tensor cores. In *Proc. 59th ACM/IEEE DAC*, pp. 937–42, 2022.
- [13] J. Hestness, S. Narang, N. Ardalani, et al. Deep learning scaling is predictable, empirically. *arXiv:1712.00409*, 2017.
- [14] J. Kaplan, S. McCandlish, T. Henighan, et al. Scaling laws for neural language models. *arXiv:2001.08361*, 2020.
- [15] M. Kirtas, A. Oikonomou, N. Passalis, et al. Quantization-aware training for low precision photonic neural networks. *Neural Networks*, 155:561–73, 2022.
- [16] M. Kirtas, N. Passalis, G. Mourgias-Alexandris, et al. Learning photonic neural network initialization for noise-aware end-to-end fiber transmission. In *Proc. 2022 EUSIPCO*, pp. 1731–35, 2022.
- [17] G. Kosmella, M. G. de Queiroz, M. Hejda, et al. Towards a noise-robust automated search of photonic circuit designs. In *IEEE Photonics Benelux Symp.*, 2024.
- [18] G. Kosmella, J. Sanders, B. Shi, et al. Higher-accuracy photonic neural networks via duplication schemes for noise reduction. In *Proc. 2023 PSC*, pages 1–4, 2023.
- [19] G. Kosmella, R. Stabile, and J. Sanders. Noise-resilient designs and analysis for optical neural networks. *Neuromorph. Comput. Eng.*, 4(4):044002, 2024.
- [20] G. Kosmella, R. Stabile, and J. Sanders. In situ fine-tuning of in silico trained optical neural networks. *arXiv:2506.22122*, 2025. Extended version of this IFIP Performance 2025 paper.
- [21] H. Kushner and G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer Science & Business Media, 2003.
- [22] X. Li, R. Mardling, and J. Armstrong. Channel capacity of im/dd optical communication systems and of aco-ofdm. In *Proc. 2007 IEEE Int. Conf. Commun.*, pp. 2128–33, 2007.
- [23] A. E.-J. Lim, J. Song, Q. Fang, et al. Review of silicon photonics foundry efforts. *IEEE JSTQE*, 20(4):405–416, 2014.
- [24] F. Marchesin, M. Hejda, T. Melendez Carmona, et al. Braided interferometer mesh for robust photonic matrix-vector multiplications with non-ideal components. *Optics Express*, 33(2):2227–2246, 2025.
- [25] P. L. McMahon. The physics of optical computing. *Nature Reviews Physics*, 5(12):717–734, 2023.
- [26] X. Meng, G. Zhang, N. Shi, et al. Compact optical convolution processing unit based on multimode interference. *Nature Communications*, 14(1):3000, 2023.
- [27] A. Mirza, F. Sunny, P. Walsh, et al. Silicon photonic microring resonators: A comprehensive design-space exploration and optimization under fabrication-process variations. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 41(10):3359–72, 2022.
- [28] G. Mourgias-Alexandris, A. Tsakyridis, N. Passalis, et al. An all-optical neuron with sigmoid activation function. *Opt. Express*, 27(7):9620–9630, 2019.
- [29] S. Ning, H. Zhu, C. Feng, et al. Photonic-electronic integrated circuits for high-performance computing and ai accelerators. *J. Lightwave Technol.*, 42(22):7834–59, 2024.
- [30] A. Oikonomou, M. Kirtas, N. Passalis, et al. A robust, quantization-aware training method for photonic neural networks. In *Eng. Appl. Neural Netw.*, pp. 427–38, 2022.
- [31] N. Passalis, G. Mourgias-Alexandris, A. Tsakyridis, et al. Training deep photonic convolutional neural networks with sinusoidal activations. *IEEE Trans. Emerg. Top. Comput. Intell.*, 5(3):384–93, 2021.
- [32] J. Sanders, S.C. Borst, A.J.E.M. Janssen, et al. Optimality gaps in asymptotic dimensioning of many-server systems. *Operations Research Letters*, 44(3):359–365, 2016.
- [33] N. Semenova, L. Larger, and D. Brunner. Understanding and mitigating noise in trained deep neural networks. *Neural Networks*, 146:151–160, 2022.
- [34] A. Senen-Cerda and J. Sanders. Almost sure convergence of dropout algorithms for neural networks. *arXiv:2002.02247*, 2023.
- [35] B. J. Shastri, A. N. Tait, T. Ferreira de Lima, et al. Photonics for artificial intelligence and neuromorphic computing. *Nature Photonics*, 15(2):102–114, 2021.
- [36] Y. Shen, N. C. Harris, S. Skirlo, et al. Deep learning with coherent nanophotonic circuits. *Nature Photonics*, 11(7):441–446, 2017.
- [37] B. Shi, N. Calabretta, and R. Stabile. Deep neural network through an inp soa-based photonic integrated

- cross-connect. *IEEE JSTQE*, 26(1):1–11, 2020.
- [38] A. Sludds, S. Bandyopadhyay, Z. Chen, et al. Delocalized photonic deep learning on the internet’s edge. *Science*, 378(6617):270–276, 2022.
 - [39] A. N. Tait, T. F. de Lima, E. Zhou, et al. Neuromorphic photonic networks using silicon photonic weight banks. *Scientific Reports*, 7(1):7430, 2017.
 - [40] A. Varri, F. Brücknerhoff-Plückelmann, J. Dijkstra, et al. Noise-resilient photonic analog neural networks. *J. Lightwave Technol.*, 42(22):7969–7976, 2024.
 - [41] Z. Wang, L. Chang, F. Wang, et al. Integrated photonic metasystem for image classifications at telecommunication wavelength. *Nature Communications*, 13(1):2131, 2022.
 - [42] L. G. Wright, T. Onodera, M. M. Stein, et al. Deep physical neural networks trained with backpropagation. *Nature*, 601(7894):549–555, 2022.
 - [43] T. Xu, W. Zhang, J. Zhang, et al. Control-free and efficient integrated photonic neural networks via hardware-aware training and pruning. *Optica*, 11(8):1039–1049, 2024.
 - [44] X. Xu, M. Tan, B. Corcoran, et al. 11 tops photonic convolutional accelerator for optical neural networks. *Nature*, 589(7840):44–51, 2021.
 - [45] Z. Zhao, J. Gu, Z. Ying, et al. Design technology for scalable and robust photonic integrated circuits: Invited paper. In *Proc. 2019 IEEE/ACM ICCAD*, pp. 1–7, 2019.
 - [46] T. Zhou, X. Lin, J. Wu, et al. Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit. *Nature Photonics*, 15(5):367–373, 2021.
 - [47] Y. Zhu, M. Liu, L. Xu, et al. Multi-wavelength parallel training and quantization-aware tuning for wdm-based optical convolutional neural networks considering wavelength-relative deviations. In *Proc. 2023 ASP-DAC*, pp. 384–89, 2023.

When Q-Learning fails: unstable behavior for infinite state spaces

U. Ayesta^{a,c}, S. Foss^d, M. Jonckheere^b and V. Puricelli^{a,b,*}

^a CNRS, IRIT, 2 rue C. Camichel, 31071 Toulouse, France.

^b CNRS, LAAS, 7 av. du colonel Roche, 31031 Toulouse, France.

^c IKERBASQUE - Basque Foundation for Science, 48011 Bilbao, Spain.

^d School of MACS, Heriot-Watt University and Maxwell Institute, EH14 4AS, Edinburgh, UK.

ABSTRACT

The Q-learning algorithm is well known for its convergence guarantees to the optimal policy in finite-state environments. In this paper, we investigate its limitations in countable infinite state spaces – a setting common in real-world problems. To this end, we introduce a simple queueing model, based on a load balancing problem, with a countably infinite state space. In this model, a dispatcher assigns incoming jobs to one of two queues by choosing between two possible actions: “red” and “green”. The “red” action leads to transient behavior, whereas the “green” action ensures stability. Our main result shows that, under certain parameter conditions, Q-learning exhibits instability and fails to converge to the optimal policy. Our findings reveal a critical gap in the theoretical understanding of model-free Reinforcement Learning methods in infinite domains. Numerical experiments illustrate that the transience also occurs with decreasing step-sizes that satisfy the usual Robbins-Monro conditions.

1. INTRODUCTION

In the realm of Reinforcement Learning (RL), simple algorithms such as Q-learning or SARSA stand out as an easily implemented and successful approaches for solving complex stochastic decision-making problems. In particular, Q-learning has garnered significant attention as an algorithm with the ability to learn optimal policies without requiring a model of the environment.

In many real-world problems, however, such as resource allocation in communication networks or control of physical systems, natural models have an infinite state space. This is the starting point of the present article, where we discuss Q-learning on discrete but infinite state spaces.

In the context Markov Decision Processes (MDPs) or model-based methods, several recent studies have explored the challenges associated with infinite state spaces. For instance, [5] addressed the convergence of the Natural Policy Gradient algorithm on some queueing MDPs in infinite state spaces, providing insights into how finite-state approaches can be extended. These generalizations typically rely on the structural properties of the state space and on the continuity

of the reward and transition functions, which allow for the extension of convergence proofs from finite to infinite environments, both for discounted and undiscounted settings.

Interestingly, our results indicate that such generalizations do not necessarily hold when dealing with model-free approaches. Model-free methods, which do not assume knowledge of the underlying MDP dynamics, face significant and different challenges from MDPs in infinite state spaces. In particular, the lack of a prior knowledge of state transitions and costs can lead to instability and to convergence issues that are not present in model-based approaches. Within the realm of model-free Reinforcement Learning, we focus on the Q-learning algorithm, which stands out as one of the most fundamental and widely studied algorithms due to its simplicity and theoretical grounding. Our goal in this paper is to formally demonstrate that, under certain parameter conditions, the Q-learning algorithms can exhibit instability in the form of *transience*, implying the inability to learn the optimal policy.

To this end, we consider a queueing system that models a load balancing problem – a widely studied scenario that is appealing due to its simplicity, despite operating in an infinite (countable) state space. In this queueing example, jobs arrive and are dispatched to one of two queues. The state of the system is represented by the number of jobs in each queue, and the goal is to balance the load between the queues to optimize performance metrics such as throughput and waiting time. The dispatcher can choose between two actions, “red” and “green”, which determine the probability of assigning jobs to the queues. The parameters are chosen so that if the “red” action were taken at every state, the number of jobs in the system would be transient, whereas with the “green” action, the number of jobs would remain stable.

The main result of this work Theorem 1 states that, under certain parameter conditions, the Q-learning algorithm exhibits transient behavior. This transience implies that the agent’s policy will not stabilize, leading to suboptimal performance. This result is significant as it highlights a fundamental limitation of Q-learning and of similar algorithms in infinite state spaces, where traditional convergence guarantees may not hold. The discrepancy between the results obtained in the MDP and RL setting highlight the need for new theoretical frameworks and algorithms specifically designed to address infinite state spaces in RL.

For a full-length version of this extended abstract see [1].

*Corresponding author. email: vpuricelli@laas.fr

2. RELATED WORK

Regarding Q-learning, as mentioned in the introduction, the first convergence results towards the optimal policy were established for step sizes satisfying the Robbins-Monro conditions and finite state MDPs in [9] and [8]. Convergence guarantees for tabular Q-learning with constant step sizes in finite state spaces were provided in [2]. Let $\hat{Q}_n$ denote the n -step estimate of Q^* , the optimal action-value function. In [2], the authors show that for any $\theta > 0$, there exists a step size $\alpha_\theta > 0$ and $K_\theta > 0$ such that for all $\alpha \leq \alpha_\theta$ and $n \geq K_\theta$, $\|\hat{Q}_n - Q^*\|_\infty \leq \theta$.

In the context of model-based approaches for infinite state spaces, convergence of the Natural Policy Gradient (NPG) algorithm is proven in [5] for queueing MDPs under the additional assumption on the initial policy being stable. In [3], again for queueing systems, the authors consider a parameterized policy class and construct a practical algorithm that iteratively improves the average reward. This algorithm is also initialized with a stable policy.

In [7], the authors propose a model-free algorithm for discounted reward MDP with an infinite state space, under the generative framework (where one can sample transitions from any pair of state/action). The proposed algorithm operates in a forward-backward manner, where at every step the action is determined based on a simulation of the future events. They show that the process obtained by the application of this algorithm is stable.

3. THE MODEL: STATE DEPENDENT LOAD BALANCING

In this section, we present a class of two-dimensional problems arising in a queueing system, for which we will establish instability of Q-learning in Theorem 1. We have not identified a nontrivial one-dimensional example that exhibits transience phenomena. Consider jobs arriving to a dispatcher with rate λ . After a job arrives the dispatcher sends the job to one of two queues. The dispatcher can choose between actions r (“red”) and g (“green”), where action a represents selecting p_a for the probability of assigning a job to Queue 1. Meanwhile jobs leave the system in the following way: when both queues are busy a job in Queue i leaves the system with rate μ_i for $i = 1, 2$. When one (no matter which one) queue is empty a job leaves the busy queue at rate $\tilde{\mu}$.

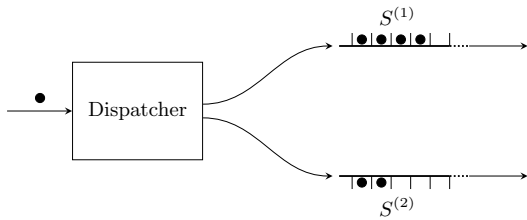


Figure 1: Load balancing

It is known from theory that given a continuous-time Markov chain on a discrete state space, one can construct a corresponding discrete-time Markov chain by observing the system at the random times when state transitions occurs. This yields what is known as the embedded Markov chain.

In this construction, the transition probabilities of the embedded chain capture the probability of jumping from one state to another, conditioned on the fact that a transition occurs. Following this approach, we consider a discrete-time MDP $(\mathcal{S}, \mathcal{A}, p)$ where: the state space is $\mathcal{S} = \mathbb{Z}_+^2$ (2-dimensional integer-valued vectors with non-negative coordinates) so each state indicates the lengths of both queues (how many jobs are in each queue), action space $\mathcal{A} = \{r, g\}$ and standard results yield that the transition probabilities $p(s' | s, a)$ defined for different states $s = (s^{(1)}, s^{(2)})$ are as follows:

1. **For states with $s^{(1)} > 0, s^{(2)} > 0$:**

$$\begin{aligned} p(s + (1, 0) | s, a) &= \frac{\lambda p_a}{\lambda + \mu_1 + \mu_2} \\ p(s + (0, 1) | s, a) &= \frac{\lambda(1 - p_a)}{\lambda + \mu_1 + \mu_2} \\ p(s + (-1, 0) | s, a) &= \frac{\mu_1}{\lambda + \mu_1 + \mu_2} \\ p(s + (0, -1) | s, a) &= \frac{\mu_2}{\lambda + \mu_1 + \mu_2} \end{aligned} \quad (1)$$

2. **For boundary states with $s^{(2)} = 0, s^{(1)} > 0$:**

$$\begin{aligned} p(s + (1, 0) | s, a) &= \frac{\lambda p_a}{\lambda + \tilde{\mu}} \\ p(s + (0, 1) | s, a) &= \frac{\lambda(1 - p_a)}{\lambda + \tilde{\mu}} \\ p(s + (-1, 0) | s, a) &= \frac{\tilde{\mu}}{\lambda + \tilde{\mu}} \end{aligned} \quad (2)$$

3. **For boundary states with $s^{(1)} = 0, s^{(2)} > 0$:**

$$\begin{aligned} p(s + (1, 0) | s, a) &= \frac{\lambda p_a}{\lambda + \tilde{\mu}} \\ p(s + (0, 1) | s, a) &= \frac{\lambda(1 - p_a)}{\lambda + \tilde{\mu}} \\ p(s + (0, -1) | s, a) &= \frac{\tilde{\mu}}{\lambda + \tilde{\mu}} \end{aligned} \quad (3)$$

4. **For the origin state $s = (0, 0)$:**

$$\begin{aligned} p((1, 0) | (0, 0), a) &= p_a \\ p((0, 1) | (0, 0), a) &= 1 - p_a \end{aligned} \quad (4)$$

3.1 The drift conditions

Our goal is to model an agent that: has no prior knowledge of the transition dynamics $p(s' | s, a)$ and that at each time step $n \geq 0$ it has to select some action A_n at state S_n , leading to a next state $S_{n+1} \sim p(\cdot | S_n, A_n)$.

We then analyze whether the stochastic process $(S_n)_{n \geq 0}$ on $\mathcal{S}$ can be unstable in the following sense:

DEFINITION 1. *We say that $(S_n)_{n \geq 0}$ is transient or unstable if for any bounded set $U \subset \mathcal{S}$, there is $s_0 \notin U$ such that*

$$\mathbb{P}(\inf\{n \geq 1 : S_n \in U\} = +\infty \mid S_0 = s_0) > 0. \quad (5)$$

We say that $(S_n)_{n \geq 0}$ is positive recurrent or stable if for any bounded set $U \subset \mathcal{S}$ and any $s_0 \notin U$

$$\mathbb{E}(\inf\{n \geq 1 : S_n \in U\} \mid S_0 = s_0) < +\infty. \quad (6)$$

Note that Definition 1 coincides with the standard definition of transience and positive recurrence in the case of irreducible Markov chains on $\mathcal{S}$.

Consider an agent always applying “red” action, yielding a Markov chain $(S_n^r)_{n \geq 0}$, and another always applying “green” action, yielding another Markov chain $(S_n^g)_{n \geq 0}$. We establish parameter conditions ensuring $(S_n^r)_{n \geq 0}$ is transient (unstable) and $(S_n^g)_{n \geq 0}$ positive recurrent (stable).

Then first condition concerns the instability of action r ,

$$\begin{cases} \mu_1 < \lambda p_r \\ \mu_2 < \lambda(1 - p_r). \end{cases} \quad (7)$$

Concerning action g we assume

$$\begin{cases} \mu_1 < \lambda p_g \\ \mu_2 > \lambda(1 - p_g). \end{cases} \quad (8)$$

and also that

$$\tilde{\mu} > \lambda \left(\frac{(\mu_1 + \mu_2)p_g - \mu_1}{\mu_2 - \lambda(1 - p_g)} \right). \quad (9)$$

Note that if $\mu_1 + \mu_2 < \lambda < \tilde{\mu}$, conditions (7), (8) and (9) are satisfied for some $p_r, p_g \in (0, 1)$.

For each action its corresponding drift D_a is given by

$$\begin{aligned} D_a(s) = & (1, 0) \times p(s + (1, 0) \mid s, a) \\ & + (0, 1) \times p(s + (0, 1) \mid s, a) \\ & + (-1, 0) \times p(s + (-1, 0) \mid s, a) \\ & + (0, -1) \times p(s + (0, -1) \mid s, a). \end{aligned} \quad (10)$$

Observe that $D_a(s)$ takes 4 different values as a function of the state according to (1), (2), (3) and (4). We refer to Figure 2 for a visual representation of $D_a(s)$ for both actions. The drift in the positive orthant (states with both coordinates strictly positive) plays an important role in Theorem 1, and we reserve for it the notation D_a . From (10) and (1):

$$D_a = \left(\frac{\lambda p_a - \mu_1}{\lambda + \mu_1 + \mu_2}, \frac{\lambda(1 - p_a) - \mu_2}{\lambda + \mu_1 + \mu_2} \right). \quad (11)$$

From Theorem 3.3.1 of [4] it follows that if (7) holds the Markov chain $(S_n^r)_{n \geq 0}$ is transient, and that if (8) and (9) hold the Markov chain $(S_n^g)_{n \geq 0}$ is positive recurrent.

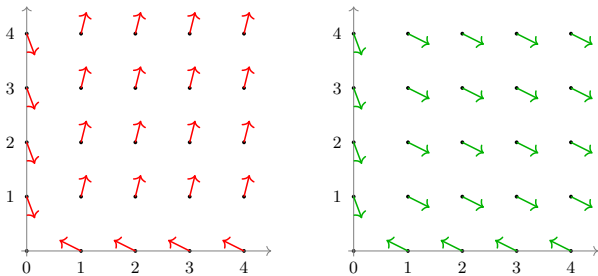


Figure 2: Drift of red (left) and green (right) action.

We still need to assume one further condition upon the parameters for our main result Theorem 1 to hold. Consider

the quantity

$$\begin{aligned} \varrho &:= \min\{D_r \cdot (1, 0), D_g \cdot (1, 0)\} \\ &= \min\left\{ \frac{\lambda p_r - \mu_1}{\lambda + \mu_1 + \mu_2}, \frac{\lambda p_g - \mu_1}{\lambda + \mu_1 + \mu_2} \right\} \\ &= \frac{\lambda p_r - \mu_1}{\lambda + \mu_1 + \mu_2}, \end{aligned} \quad (12)$$

where the last equality follows from (7) and (8). Then we will assume that the vector $(\varrho/2)D_r + (1 - \varrho/2)D_g$ has all its coordinates positive (see Figure 3). Precisely in terms of the parameters, this is equivalent to the following condition:

$$\mu_2 < \lambda(1 - p_g) + \frac{1}{2} \frac{(\lambda p_r - \mu_1)(p_g - p_r)}{\lambda + \mu_1 + \mu_2}. \quad (13)$$

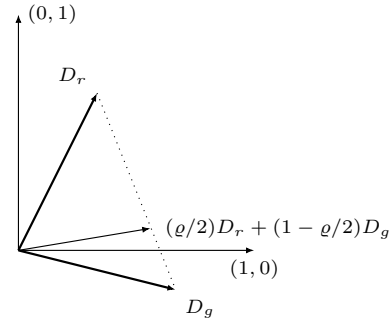


Figure 3: The configuration of the drifts derived from (7), (8) and (13).

3.2 The cost function

We end the description of our model by considering a specific cost function $c : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ given by:

$$c(s, a, s') = \left(s'^{(1)} - s^{(1)} \right) + \left(s'^{(2)} - s^{(2)} \right). \quad (14)$$

We refer this cost as being “local”, in the sense that it only depends on the transition, and not on the current state. This assumption is required in order to formally establish the main result, see Section 5 for more details. Numerical experiments reported in Section 6 illustrate that the instability of Q-learning remains when the cost function is defined as the sum of the coordinates of the current state.

In infinite-horizon cost problems the goal for an agent is to minimize for some fixed $\gamma \in (0, 1]$ the following objective:

$$J_\gamma := \limsup_n \mathbb{E} \left(\sum_{i=1}^n \gamma^i c(S_i, A_i, S_{i+1}) \right), \quad (15)$$

where J_γ is called the total discounted cost for $\gamma < 1$, while $J := J_1$ is the total cost.

We see that for the cost function defined by (14) in addition of penalizing the growth of the queues, having finite total cost prevents transience as defined by Definition 1.

Since the agent that always applies green action defines a positive recurrent Markov chain, it has finite total cost. Moreover we have that

$$J_\gamma^g := \mathbb{E} \left(\sum_{i=1}^{+\infty} \gamma^i c(S_i^g, g, S_{i+1}^g) \right) = O(1) \text{ as } \gamma \rightarrow 1. \quad (16)$$

for a complete proof of (16) we refer to [1].

4. Q-LEARNING

As underlined in the introduction, the Q-learning algorithm is a popular option to solve total or discounted cost problems. The algorithm is based on sequentially approaching the optimal state-value function $Q^* : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, which is known to be the unique solution of the so-called Bellman equation [6], that for $\gamma < 1$ takes the form:

$$Q^*(s, a) = \min_{a' \in \mathcal{A}} \left\{ \sum_{s' \in \mathcal{S}} p(s'|s, a) [c(s, a, s') + \gamma Q^*(s', a')] \right\}.$$

The knowledge of Q^* allows to obtain an optimal policy π^* (i.e. such that $J_{\gamma}^{\pi^*} \leq J_{\gamma}^{\pi}$ for all π) by considering $\mathcal{A}^*(s) = \arg \min_{a \in \mathcal{A}} Q^*(s, a)$ and setting

$$\pi^*(a | s) = \begin{cases} \frac{1}{|\mathcal{A}^*(s)|} & \text{if } a \in \mathcal{A}^*(s) \\ 0 & \text{if not.} \end{cases}$$

DEFINITION 2. Fix $\varepsilon \in [0, 1]$ and $\alpha > 0$ the exploration parameter and step-size respectively. Consider $s_0 \in \mathcal{S}$ and $\hat{Q}_0 : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$. The asynchronous Q-learning with constant step size α is the sequence $(S_n, A_n, \hat{Q}_n)_{n \geq 0}$ where $S_0 = s_0$ and for each $n \geq 0$:

$$1. \pi(a | \hat{Q}_n(S_n, \cdot)) = \begin{cases} \frac{1-\varepsilon}{|\hat{\mathcal{A}}_n|} + \varepsilon/2 & \text{if } a \in \hat{\mathcal{A}}_n \\ \varepsilon/2 & \text{if not,} \end{cases}$$

where $\hat{\mathcal{A}}_n := \arg \min_{a' \in \mathcal{A}} \hat{Q}_n(S_n, a')$,

$$2. A_n \sim \pi(\cdot | \hat{Q}_n(S_n, \cdot)),$$

$$3. S_{n+1} \sim p(\cdot | S_n, A_n),$$

and lastly

$$\begin{aligned} \hat{Q}_{n+1}(s, a) &= \hat{Q}_n(s, a) + \mathbf{1}_{\{S_n=s, A_n=a\}} \times \\ &\times \alpha [c(S_n, A_n, S_{n+1}) + \gamma \min_{a' \in \mathcal{A}} \hat{Q}_n(S_{n+1}, a') - \hat{Q}_n(S_n, A_n)]. \end{aligned}$$

This can be called $(1-\varepsilon)$ -greedy asynchronous Q-learning with constant step-size α . We recall that in [2] (see Theorem 3.4), the authors derive an upper bound for the expected error of the approximation $\mathbb{E}(\|\hat{Q} - Q^*\|_{\infty})$ in the case of finite state spaces.

5. MAIN RESULT

In this section we state our main result Theorem 1, where we show that for the model defined in Section 3, Q-learning exhibits transient behavior. This transience arises because upon the first visit to each state, the agent chooses with sufficiently high probability the “red” action and is unable to recover from its instability (see (7)), leading to a transient behavior. As a consequence, we demonstrate that Q-learning exhibits suboptimal behavior in terms of total discounted cost.

THEOREM 1. Consider $(\mathcal{S}, \mathcal{A}, p)$ as defined in Section 3. Then for all $\varepsilon \in [0, 1]$ and $\alpha > 0$ if $\hat{Q}_0 = 0$ the process $(S_n)_{n \geq 0}$ induced by the Q-learning algorithm given by Definition 2 is transient. Also

$$J_{\gamma} = \Omega\left(\frac{1}{1-\gamma}\right) \text{ as } \gamma \rightarrow 1. \quad (17)$$

We observe that this implies that for γ sufficiently close to 1, Q-learning is sub-optimal under the assumptions of Theorem 1. This sub-optimality arises because the optimal discounted cost J_{γ}^* satisfies the inequality $J_{\gamma}^* \leq J_{\gamma}^g = O(1)$ (see (16)) while from Theorem 1 we obtain $J_{\gamma} = O(1/(1-\gamma)) \rightarrow +\infty$ as $\gamma \rightarrow 1$.

A complete proof of Theorem 1 can be found in [1]. We provide here a sketch of the proof for the transience of Q-learning.

Sketch of the proof of transience. From the model described in Section 3, we define a new model enlarging the state space in such a way that we erase the effect of the boundary states. This new model is explicitly given by $(\mathcal{S}', \mathcal{A}', p')$, where $\mathcal{S}' = \mathbb{Z}^2$, $\mathcal{A}' = \{r, g\}$ and p' is given by (1) for all $s \in \mathbb{Z}^2$. Observe that in this new model each action a induces a constant drift D_a in space which by construction is given by (11). We also extend the cost function c given by (14) to all $\mathbb{Z}^2$. This allows us to consider $(S_n, A_n, \hat{Q}_n)_{n \geq 0}$ defined exactly by Definition 2 in this new model $(\mathcal{S}', \mathcal{A}', p')$. The key step of the proof is to show that there exists a uniformly positive probability on the initial condition S_0 , for $S_n > S_0$ coordinate-wise for all $n \geq 1$. By construction this property also holds for the process derived from Q-learning in the original model. To verify the transience of Q-learning (see Definition 1), we consider any bounded set U and $s_0 > s$ for all $s \in U$, and obtain

$$\begin{aligned} \mathbb{P}(\inf\{n \geq 1 : S_n \in U\} = +\infty \mid S_0 = s_0) \\ \geq \mathbb{P}(S_n > S_0 \text{ for all } n > 0 \mid S_0 = s_0) > 0, \end{aligned}$$

which follows since there is a positive probability of $S_n > S_0$ for all $n \geq 1$.

In order to derive this key step of the proof, we first show that $\delta := \lim_n \frac{1}{n} \sum_{i=0}^{n-1} \mathbf{1}_{\{A_i=r\}}$, the average number of times Q-learning applies red action, is well defined and that

$$\lim_n \frac{S_n}{n} = \delta D_r + (1-\delta) D_g \in \mathbb{R}^2 \quad \mathbb{P}\text{-a.s.} \quad (18)$$

We aim to prove that all coordinates of the limiting drift (18) are positive. This is crucial to establish that there exists a uniformly bounded positive probability, on the initial condition S_0 , for $S_n > S_0$ coordinate-wise for all $n \geq 1$. To show that all coordinates of (18) are positive, we establish the bound $\delta \geq \varrho/2$ (recall (12)) through two observations. First, ϱ is a lower bound for the average number of times the process $(S_n)_{n \geq 0}$ visits new states. Second, since $\hat{Q}_0 = 0$, at these states visited for the first time there is at least probability $1/2$ of applying red action. Finally this bound on δ implies that the limiting drift (18) lies in the line between D_r and $(\varrho/2)D_r + (1-\varrho/2)D_g$. Using (13) we deduce that both coordinates of (18) are strictly positive (see Figure 3).

We conclude giving some details on how to establish (18), the main idea is to show that $(S_n)_{n \geq 0}$ has a regenerative structure. Given the assumptions on the drifts, we focus on the process D_r and D_g have a common component towards $l = (1, 0)$, we are going to focus $(S_n \cdot l)_{n \geq 0}$, with $l = (1, 0)$, which captures the horizontal drift of the process. We then show that there exists some random time $\tau_1 \geq 0$ such that $S_{\tau_1} \cdot l > S_t \cdot l$, for all $t < \tau_1$ and $S_{\tau_1+t} \cdot l \geq S_{\tau_1} \cdot l$, for all $t \geq 0$. Due to the locality nature of the cost function (see (14)) and the way in which the Q-learning algorithm operates (see Definition 2), it follows that τ_1 is a renewal epoch for $(S_n)_{n \geq 0}$. This enables us to find a next random time $\tau_2 > \tau_1$ such that $S_{\tau_2} \cdot l > S_t \cdot l$, for all $t < \tau_2$ and

$S_{\tau_2+t} \cdot l \geq S_{\tau_2} \cdot l$, for all $t \geq 0$. Iterating this process we conclude that $(S_n)_{n \geq 0}$ has a renewal structure, with renewal epochs τ_n , $n = 0, 1, \dots$. This renewal structure implies in turn the existence of the limit and its characterization in (18).

6. NUMERICS

In this section we perform numerical experiments to illustrate the behavior of Q-learning in various settings combining constant and non-constant step size and local and non-local cost function. As non-constant step size we consider $\alpha_n = 1/(n+1)$, which satisfies the classical Robbins-Monro conditions. As a non-local cost function we consider the total number of jobs, namely $c(s, a, s') = s^{(1)} + s^{(2)}$, which is a standard objective considered in the literature. In particular, we consider the following four settings: (i) (setting of Theorem 1) constant step-size and local cost function as given by (14), (ii) constant step-size and non-local cost function, (iii) non-constant step-size and local cost function, and (iv) non-constant step-size and non-local cost function.

Throughout the experiments, we fix the following choice of parameters for the load balancing model: $\lambda = 1.5$, $\mu_1 = 0.1$, $\mu_2 = 0.35$, $p_r = 0.45$, $p_g = 0.8$ and $\tilde{\mu} = 0.01 + \lambda((\mu_1 + \mu_2)p_g)/(\mu_2 - \lambda(1 - p_g))$. We consider the discounting factor $\gamma = 0.1$, exploration parameter $\varepsilon = 0.1$, step-size $\alpha = 0.2$, initial condition $S_0 = (0, 0)$ and initialization $\hat{Q}_0(s, a) = 0$.

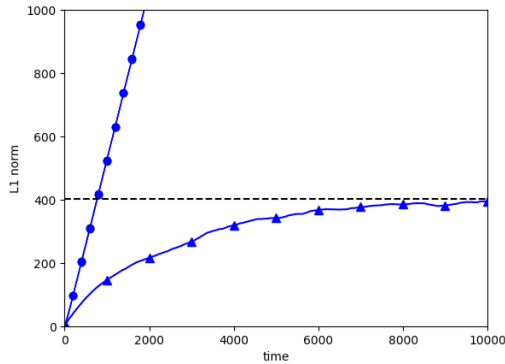


Figure 4: Evolution over time of the L1-norm of $(S_n)_{n \geq 0}$ derived from Q-learning in various settings (circles) and of $(S_n^g)_{n \geq 0}$ (triangles). The dashed line indicates the value J^g .

In Figure 4 we simulate $N = 1000$ trajectories up to time $n = 10000$ and we plot over time the average L_1 norm $\|s\|_1 = s^{(1)} + s^{(2)}$ as a function of time for $(S_n^g)_{n \geq 0}$ and $(S_n)_{n \geq 0}$ derived from Q-learning for settings (i)-(iv) with initial condition $S_0 = (0, 0)$. We observe that the qualitative behavior of Q-learning is similar across all settings, suggesting that Q-learning remains transient even when using decreasing step-sizes and non-local costs.

In Figure 5, under the setting of Theorem 1, we plot the discounted cost functions J_γ and J_γ^g as a function of γ . For this we simulate $N = 100$ values of J_γ , calculated for a $n = 10000$ path-length, and plot an average for each γ . We recall that from Equation (15), we know that as $\gamma \rightarrow 1$, $J_\gamma^g \rightarrow J^g = \lim_n \mathbb{E} \left(S_n^{(1)} + S_n^{(2)} \right)$, a value we represent with a big circle in Figure 5. As predicted by Theorem 1, we observe that J_γ increases with γ and has an asymptote at

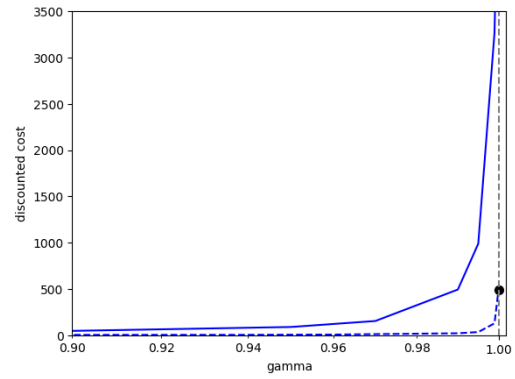


Figure 5: The discounted cost J_γ for the Q-learning algorithm (solid) and J_γ^g (dashed) as a function of γ .

$\gamma = 1$, whereas J_γ^g stays finite.

Acknowledgment

The authors would like to thank Sean Meyn for valuable discussions. Research partially supported by the French “Agence Nationale de la Recherche (ANR)” through the project ANR-22-CE25-0013-02 (ANR EPLER).

References

- [1] U. Ayesta, S. Foss, M. Jonckheere, and V. Puricelli. *On the instability of local learning algorithms: Q-learning can fail in infinite state spaces*. 2025. URL: <https://arxiv.org/abs/2510.25572>.
- [2] C. Beck and R. Srikant. “Error bounds for constant step-size Q-learning”. In: *Systems & Control Letters* 61.12 (2012), pp. 1203–1208.
- [3] J. G. Dai and M. Gluzman. “Queueing Network Controls via Deep Reinforcement Learning”. In: *Stochastic Systems* 12.1 (2021), pp. 30–67.
- [4] G. Fayolle, V. A. Malyshev, and M. V. Menshikov. *Topics in the constructive theory of countable Markov chains*. Cambridge: Univ. Press, 1995.
- [5] I. Grosz, S. T. Maguluri, and R. Srikant. “Convergence of Natural Policy Gradient for a family of infinite-state queueing MDPs”. In: *Queueing Systems* 109.22 (2025).
- [6] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. English. Wiley Ser. Probab. Math. Stat. New York, NY: John Wiley & Sons, Inc., 1994. ISBN: 0-471-61977-9.
- [7] D. Shah, Q. Xie, and Z. Xu. “Stable Reinforcement Learning with Unbounded State Space”. In: *Learning for Dynamics and Control (L4DC) 2020*. 2020.
- [8] J. N. Tsitsiklis. “Asynchronous stochastic approximation and Q-learning”. In: *Machine Learning* 16.3 (1994), pp. 185–202. ISSN: 0885-6125.
- [9] C. J. C. H. Watkins and P. Dayan. “Q-learning”. In: *Machine Learning* 8.3-4 (1992), pp. 279–292. ISSN: 0885-6125.

Alert or Noise? Reducing False Positives with Active Behavioral Analysis for Cloud Security

Dikshant
7-Eleven
Bengaluru, India
27dikshant@gmail.com

Geetika Verma
RMIT University
Melbourne, Australia
geetika.verma@rmit.edu.au

ABSTRACT

Rule-based cloud security posture management (CSPM) solutions are known to produce a lot of false positives based on the limited contextual understanding and dependence on static heuristics testing. This paper introduces a validation-driven methodology that integrates active behavioral testing in cloud security posture management solution(s) to evaluate the exploitability of policy violations in real time. The proposed system employs lightweight and automated probes, built from open-source tools, validation scripts, and penetration testing test cases, to simulate adversarial attacks on misconfigured or vulnerable cloud assets without any impact to the cloud services or environment. For instance, cloud services may be flagged as publicly exposed and vulnerable despite being protected by access control layers, or secure policies, resulting in non-actionable alerts that consume analysts' time during manual validation. Through controlled experimentation in a reproducible AWS setup, we evaluated the reduction in false positive rates across various misconfiguration and vulnerable alerts. Our findings indicate an average reduction of 93% in false positives. Furthermore, the framework demonstrates low latency performance. These results demonstrate a scalable method to improve detection accuracy and analyse productivity in large cloud environments. While our evaluation focuses on AWS, the architecture is modular and extensible to multi-cloud setups.

Keywords

Cloud Security Posture Management, False Positive Reduction, Behavioral Validation, Active Security Testing, Exploitability Assessment, Alert Triage

1. INTRODUCTION

Cloud computing revolutionized application deployment to ensure infrastructure becomes scalable, agile, and provisioned quickly. But with this agility comes greater complexity and risk, especially in securing dynamic configurations in distributed environments. Typical problems like misconfigurations, excessive permissions, and public endpoints continue and are often raised as security warnings by Cloud Security Posture Management (CSPM) solutions and Cloud-

Native Application Protection Platforms (CNAPPs).

While worthwhile, these tools often rely on static, rule-based detection methods with no contextual awareness of the specific environment or business impact [1]. As a result, they generate large amounts of false positives. For instance, critical severity risk alerts like publicly exposed storage buckets or unused access keys may pose minimal risk when compensating controls are implemented. This deluge worsens analyst fatigue, decreases productivity, and inhibits response to valid threats in a timely manner [2, 4].

At scale, triaging that level of noise is costly. Current research indicates that false positives waste analyst time and disrupt remediation efforts [1], diluting the effectiveness of threat detection, particularly when critical alerts are lost among noise.

To address this challenge, we introduce a behavioral validation mechanism that supplements static rule-based detection with lightweight, runtime exploitability analysis. By leveraging open source tool exploitation and automated validation methods, our mechanism emulates true world analysis, e.g., examining suspected resources, ascertaining exposure, and executing simple validation commands. This strategy enables prioritization based on actual observed risk instead of theoretical vulnerabilities.

This approach was evaluated by controlled experiments run in Amazon Web Services (AWS), simulating real-world misconfigurations. The results show that behavioral validation significantly reduces false positives, simplifies the alert triage process, and enables analysts to prioritize alerts with real security effects without sacrificing detection accuracy, improving cloud security controls. Although our implementation and evaluation are conducted within AWS, the underlying validation logic and modular architecture are cloud-agnostic by design. With appropriate provider-specific probes, the framework can be extended to Azure, GCP, or hybrid environments without altering the core methodology.

2. RELATED WORK

Cloud Security Posture Management (CSPM) and Cloud-Native Application Protection Platforms (CNAPP) are the building blocks of modern cloud security frameworks. However, a persistent problem identified by the corporate world and by researchers too is the overwhelming number of false positives with rule-based detection controls, which overwhelm the security team and slow down timely actions [4].

Most rule-based scanners perform only static compliance checks or signature matches without accounting for context-

tual security posture. For instance, a reported configuration might already be addressed by upstream access control or firewall rules unknown to the scanner [9]. Efforts to improve this involve augmenting rule engines with contextual enrichment or alert prioritization schemes [1, 2, 3]. However, these approaches remain passive as they do not actively validate exploitability.

Machine learning (ML) algorithms have also been investigated to minimize false positives in cloud alerting. Though promising, they typically demand enormous amounts of labeled data, are not transparent, and pose threats of false negatives, especially in mission-critical environments [6, 7, 15].

Active validation i.e., checking whether a problem faced can indeed be attacked, has been utilized in intrusion detection systems and explored in emerging cloud security validation approaches, with discussions on contextual verification also noted in CSPM literature [17]. The majority of production tools, even open-source such as Prowler and commercial CSPMs, remain dependent on static rules based compliance frameworks (e.g., CIS, NIST), and little runtime probing.

Our solution addresses this important shortfall by providing an in-line, real-time behavioral validation layer that complements rule-based alerts without modifying the underlying scanning engine. In contrast to ML-based or enrichment-only solutions, our solution actively verifies whether reported misconfigurations (e.g., open S3 buckets, exposed credentials, insecure ports) are actually reachable or exploitable. This reduces false positives without sacrificing the explainability and regulatory compliance of rule-based CSPM.

To the best of our knowledge, while prior work has applied anomaly-detection approaches (e.g., Extended Isolation Forest) to reduce alert volumes in SOCs [16], there is no existing work that has demonstrated real-world behavioral validation on top of production-level CSPM pipelines in real live clouds with negligible overhead. We test our system on AWS, with real misconfiguration scenarios, and measure its impact on false positive reduction.

Table 1 provides a comparative summary of the proposed behavioral validation method with earlier cloud security alert reduction methods. Rule-based CSPM solutions traditionally carry out static analysis and are plagued by high false positive rates, with recent research suggesting estimated false alarm rates higher than 40% in enterprise environments. Context aware enrichment steps typically a part of CNAPPs strive to enhance prioritization [2] but do not directly validate exploitability, and their effectiveness remains largely untested in peer reviewed research. Machine learning-based methods have experienced partial success in mitigating alert noise; however, they need enormous labeled datasets and tend to be black boxes, leading to explainability and reproducibility issues. Unlike those alternatives, the proposed behavioral validation framework achieves a 93% reduction in false positives using lightweight, transparent, and reproducible runtime probes, verified through controlled experimentation on AWS.

3. RESEARCH QUESTIONS AND METHODOLOGY

3.1 Research Questions

Table 1: Comparison of Behavioral Validation with Existing Alert Reduction Techniques

Feature / Capability	Rule-based CSPM	Contextual Enrichment	ML-based Filtering	Proposed Behavioral Validation
Detects Misconfigurations	Yes	Yes	Sometimes	Yes
Tests Exploitability	No	No	No	Yes
Real-time Validation	No	Limited	No	Yes
Requires Large Dataset	No	Maybe	Yes	No
Explainability / Transparency	High	Medium	Low	High
Multi-cloud Support	Tool-dependent	Yes	Depends	AWS-focused (Extensible)
Open-source Tool Integration	Yes	Partial	No	Yes
Reduction of False Positives	Limited (40%)	Moderate (Undocumented)	Good (Documented in ML studies)	Very High (93%, this paper)

This study examines how behavior validation can be added into traditional rule-based CSPM systems in order to minimize false positives. It also explores which lightweight behavior analysis techniques can effectively separate valid violations from harmless, non-exploitable misconfigurations. Finally, it considers how adding this layer affects both detection accuracy and how quickly alerts can be triaged.

3.2 Methodology

To address the limitations of traditional rule based Cloud Security Posture Management (CSPM) scanners, we introduce a validation first solution that augments static rule-based detection with dynamic behavioral verification. In this work, behavioral validation refers to the execution of targeted, automated probes that actively test whether a reported cloud misconfiguration or exposure can be exploited in practice, classifying each case as true positive, false positive, or inconclusive. This verification layer, built using a combination of open-source and proprietary tools, cross-verifies alerting with lightweight, context-aware probes that mimic adversary behavior in a secure runtime environment. While traditional CSPM scanners have wide visibility, pre-configured rule employment leads to high false positive rates because of a lack of contextual knowledge [9]. Probes returning timeouts, inconsistent results, or indeterminate permissions are marked as inconclusive and excluded from precision/recall calculations.

The mapping in Table 2 is applied consistently across all experiments to classify each finding and compute reported metrics. TP stands for true positive and FP stands for false positive.

Table 2: Alert Types, Probes, and Classification Criteria

Alert Type	Probe Action(s)	TP / FP Criteria
Public S3 Bucket	HEAD/GET (auth & unauth)	TP if unauth GET succeeds; FP if all blocked
EC2 Open Port	nmmap scan + service check	TP if service responds; FP if closed/filtered
IAM Key Exposure	get-access-key-last-used	TP if key active & used; FP if inactive/denied
Secret Leak	Trufflehog scan	TP if secret valid; FP if revoked/invalid

Figure 1 illustrates a typical CSPM architecture that produces large amounts of potentially irrelevant alerts through static analysis.

Our proposed solution is focused on lightweight, real-time verification of cloud alerts based on simulation of attack behavior [17]. This enhances detection and allows triage mechanisms to determine if an alert is truly exploitable.

3.2.1 Alert Categorization and Enrichment

The platform consolidates alerts from CSPM tools and maps them to a provider-agnostic unified schema. The alerts are enriched with contextual metadata (e.g., timestamps, account IDs, resource types) and grouped into threat categories: public resource exposure (e.g., EC2 with permissive security groups), open storage (e.g., S3 buckets), and credential leakage (e.g., keys in code repositories). This schema-

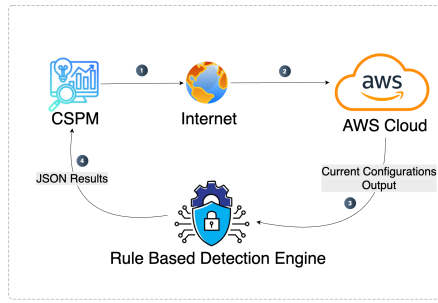


Figure 1: Rule-based CSPM architecture static detection-dependent, resulting in too many false positives. Common in open-source and commercial products.

based grouping allows for efficient verification of high-impact risks.

3.2.2 Behavioral Validation Framework

Behavioral validation actively tests for exploitability with lightweight probes simulating real-world attack behavior, in contrast to static policy validation. These probes verify whether misconfigurations are actually exploitable or mitigated by controls such as IAM policies or firewall rules.

Figure 2 illustrates how our architecture layers validation logic on top of current rule-based detection.

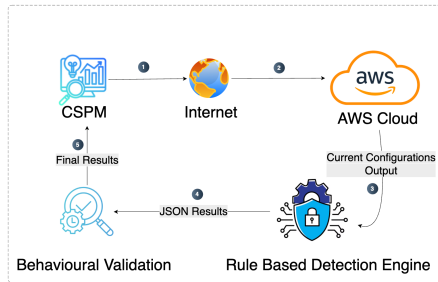


Figure 2: Improved CSPM with behavioral validation. Alerts are validated in real-time prior to escalation.

Each validation question maps to some type of misconfiguration. For example: - **S3 Exposure:** Anonymous **HEAD** and **GET** requests are issued, denial of access renders the alert unexploitable. - **EC2 Exposure:** Port scans (e.g., 22, 3389) with **nmap** confirm public access. - **Credential Leaks:** AWS CLI verifies key use and performs scoped **sts:AssumeRole** calls, disabled or least-privileged keys are reported as false positives.

All the probes are run in isolated, transient environments in order to minimize any possible impact on running systems. Each test stores metadata that contains timestamps, command output, and result status (exploitable, non-exploitable, or inconclusive).

Tools such as Nmap, AWS CLI, and TruffleHog are used within test environments to execute lightweight simulations. IAM-related notifications trigger privilege simulation probes, and compromised credentials are verified through sandboxed authentication attempts.

3.2.3 Effective, Real-Time Implementation

Each probe is time bounded and stateless. Validation occurs in isolated machines or transient virtual machines, isolated from CSPM scanning. This independence prevents failure or delay in a single probe from impeding others and enables probes to scale in parallel without stalling CSPM scans. With a 5-second per-probe limit, the architecture maintained near real-time analysis without bottlenecks at the evaluated scale. However, larger-scale deployments may require containerized/serverless probe orchestration.

Current implementations use traditional computing instances, but upcoming releases will take advantage of serverless platforms and containerized probes to enable distributed scalability. This aids performance flexibility in steady-state and dynamic alerting patterns.

Figure 3 illustrates the end-to-end validation flow, where CSPM results are enriched with behavioral metadata through custom scripts and tools, then parsed into structured outputs for accurate classification.

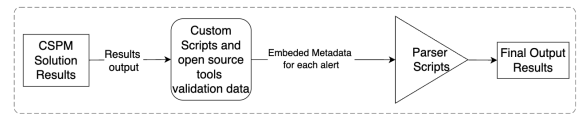


Figure 3: Validation flow: CSPM results are enriched with behavioral metadata and parsed into final output.

3.2.4 Security Controls and Constraints

To reduce risks incurred due to active probing, the system enforces various protections:

- **Scoped IAM Roles:** All probes run under least-privilege policies, tightly scoped to avoid administrative access or resource modification.

- **Request Throttling:** Rate limits ensure API usage remains within safe bounds, preventing potential slowdowns or rate-limit alarms.

- **Tagging and Transparency:** Probe requests include tags that allow monitoring systems to distinguish validation traffic from real threats. These mechanisms enable safe and responsible probe usage. In our assessment, these controls prevented privilege escalation attempts, avoided AWS rate-limit alarms, and ensured probe traffic remained distinguishable from real threats. Enhancements in the future include anomalies in probe behavior detection and tighter integration with cloud native security services.

3.2.5 Integration and Output

Validated alerts are appended with a transparent exploitability status and accompanying metadata. Outputs preserve the original CSPM data and include actionable validation evidence. It can be integrated with SIEMs and incident response tools through webhooks and APIs so that SOC teams can prioritize incidents based on confirmed exploitability [7].

3.3 Environment Configuration

To validate our behavioral validation framework, we constructed a testbed within a standalone AWS account. The configuration involved fundamental AWS services that are typically utilized in CSPM. Configuration decisions were in-

formed by the CIS AWS Foundations Benchmark v1.4 and AWS security best practices to optimize test cases against broad threat models. We established fifty S3 buckets, forty EC2 instances, and thirty IAM users with common misconfigurations in two AWS regions. Notifications were triggered using the Prowler implementation, AWS Config rule, and custom Python-based scanners that simulate CSPM logic. The entire validation process was automated using Python scripts, and the code is available at: <https://github.com/27dikshant/Noisecut>.

3.3.1 Simulated Scenarios

To stress test the validation framework, we wrote diverse misconfigurations corresponding to true and false positives. We drew examples from real-world incident reports and CSPM data. Among the tests were public S3 bucket policies, some readable, others set up to look open but locked down by IAM or bucket conditions. Another test launched EC2 instances with public IP addresses, but locked down via Security Groups or NACLs, to test alerts against shallow exposure indicators.

Credential leak scenarios involved deliberately revealing IAM access keys. Few keys were enabled, while others were disabled or had already expired, all for the sake of simulating false positives. Secrets and high-entropy tokens were cached to simulate accidental data leakage, including decoy artifacts to test noise robustness.

We also simulated access keys found in public GitHub repositories and derived variations with different statuses (active, rotated, disabled). Each scenario was designed to mimic common deterministic misconfigurations documented in benchmarks and incidents, not artificial test cases tailored to our system. These configurations tested the framework to identify edge cases where static CSPM tools misclassify alerts. To enhance credibility, these scenarios were derived from recurring misconfigurations documented in the CIS AWS Foundations Benchmark and cloud incident reports, ensuring they mirrored real-world conditions.

3.3.2 Tools and Automation

Baseline alerts were gathered from Prowler scans against the CIS AWS Benchmark. Probes were executed as stateless scripts using Python, AWS CLI and command-line utilities like `curl` and `nmap`. Visibility of S3 buckets, for example, was probed with unauthenticated `aws s3api get-object-acl` and `curl`; EC2 visibility was probed with `nmap -Pn` port 22 and port 3389 scans. Access key probes invoked `iam get-access-key-last-used` and tried scoped `sts:AssumeRole` actions. We did use custom developed scripts for parsing the results of CSPM solutions (including false positives) with the parser scripts as the part of behavioral validation as discussed in Section 3.2.3.

3.4 Evaluation Standards

We assessed behavioral validation using standard security analytics metrics. The main metrics were **False Positive Rate (FPR)** the proportion of benign alarms incorrectly labelled as threats and **True Positive Rate (TPR)** the proportion of true threats correctly labelled.

To provide context for results, we also provide **precision**, **recall**, and the **F1-score** [13]. Precision estimates the consistency of signaled alarms; recall estimates coverage of actual misconfigurations; and F1 is a balance of both.

Labels used for these metrics were derived from automated probe outcomes and cross-checked via brief manual verification by the experiment team.

Additionally, we also approximated Analyst Efficiency manual triage time reduction per alert, using time savings seen in simulated workflows on 100 randomly selected alerts. Those alerts that were marked by validation as not exploitable were assumed to take very little analyst time, approximating real-world time savings. Together, these metrics provide an overall evaluation of technical accuracy and operational relevance, measuring not just the detection quality, but the real-world utility in cloud alert fatigue reduction.

4. RESULTS AND DISCUSSION

4.1 False Positive Reduction

Before presenting quantitative results, we establish our baseline choice. We contrast behavioral validation with a popular rule-based CSPM scanner, Prowler, the de facto industry tool for cloud misconfiguration scanning. AI-powered triage or graph-based prioritization [12] is marketed by commercial CNAPPs, but these are closed-source tools with non-reproducible approaches and no public data sets to compare on a level playing field.

We know of no other existing open-source or academic platform that provides post-scan behavioral validation of CSPM alerts [2]. Thus, our system is a first of its kind solution, measured against the baseline standard of rule-only scanning. This gives us a real-world, conservative benchmark for measuring the marginal value of behavioral probes in minimizing false positives [11].

The addition of an active behavioral validation layer atop standard Cloud Security Posture Management (CSPM) alerts greatly reduced false positives for a wide range of test cases. For example, publicly exposed S3 buckets that were first identified as publicly exposed by rule-based scanning were behaviorally validated, and this revealed that approximately 93% of them were actually secured by object permissions, access controls, or bucket policy. This provided significant reduction in false alarms.

Similarly, EC2 instances with public IP addresses but shielded by firewalls or restricted security group access controls witnessed a 92% decrease in erroneous network exposure alerting. Furthermore, access keys that were identified as publicly exposed but were subsequently discovered to be disabled experienced a 94% decrease in false positives involving credentials.

Table 3: Behavioral Validation Impact on False Positives

Misconfigurations	Total Alerts	FP (Before)	FP (After)	Reduction (%)
Public S3	1000	800	50	93.72%
Public EC2	300	280	20	92.88%
Exposed Access Keys	2000	1700	100	94.12%
Exposed Credentials	200	160	10	93.75%
Total	3500	2940	180	Avg. 93.88%

4.2 Detection Accuracy and Trade-offs

Quantitatively, we ascertained that the true positive rate (TPR) was well above 0.91 across all test categories. Behavioral validation did not hinder any of the discovered alerts that were indeed exploitable during the time of annotation.

Baseline CSPM alert sets comparison, with and without the integrated behavioral validation, showed no loss of measurable recall. This confirms the assertion that the framework does not sacrifice detection accuracy while reducing the incidence of false positives [5].

Table 4: Validation Performance Metrics

Metric	Value
True Positives (TP)	360
False Positives (FP)	180
True Negatives (TN)	2960
Precision	0.667
Recall (TPR)	0.911
False Positive Rate (FPR)	0.058
F1-score	0.769

We computed standard binary classification metrics from the labels derived from the evaluation process[7]. Table 4 presents these metrics for each alert class. The behavioral validation system had a recall of 0.911, which indicates a high detection rate for real threats, with a precision of 0.667. The false positive rate decreased to 5.8%, which indicates a notable reduction in noise. The F1-score of 0.769 indicates a good balance between completeness and correctness [14].

4.3 Overview of the contribution

To quantify analyst productivity savings, we modeled representative triage workloads for each alert type with baseline rule-based CSPM output and our system. In the baseline, triage was done manually by examining alerts through the AWS Console, examining bucket policies, IAM roles, access key metadata, and firewalls processes typically taking 2–5 minutes per alert depending on complexity. Alerts identified by our behavioral validation system, however, already had exploitability ratings and logs and less than 30 seconds on average [9, 10].

The time savings were measured across 100 sample alerts with a script based stopwatch method with multiple measurements to control for variability. The results showed an observed **average triage time reduction of 86.4%**, with certain high-volume kinds of alerts (e.g., public EC2 instances or inactivated access keys) decreasing greater than 90%.

5. ANALYSIS AND LIMITATIONS

Our findings indicate that lightweight active behavioral validation meaningfully augments rule-based CSPM tools. Validating whether misconfiguration alerts are exploitable, the system significantly reduces false positives and enables security teams to focus on actual threats. The framework’s modular architecture makes it straightforward to integrate into current security infrastructures without significant architectural adjustments.

However, some limitations must be considered. Behavioral validation is most effective for deterministic forms of misconfigurations such as publicly accessible resources, overly broad IAM policies, and breached credentials. These are best resolved using active probing and regular verification practices. On the other hand, the model is less effective for identifying intricate or indirect attacks such as time based privilege escalation, insider threats, or lateral movement that depend on dynamic runtime conditions not dis-

cernible through cloud APIs or config files. Additionally, behavioral probes cannot determine exploitability when compensatory controls are applied outside of their scope, e.g., attack surface reduction products or multi-layered monitoring systems. Encrypted communications channels and obfuscation of services can also hinder the probes’ ability to identify exposure correctly [6].

Another real-world limitation is scope: the existing implementation has been validated mainly in AWS environments. Adding support for Azure, GCP, and hybrid clouds will involve probe modifications to cloud-specific APIs, resource models, and IAM infrastructures.

Finally, as cloud attackers evolve, the validation logic itself may be targeted or evaded. Keeping the integrity and security of the validation pipeline would demand constant tuning, adaptive techniques, and incorporation with anomaly detection or AI-driven techniques [8].

6. CONCLUSION AND FUTURE DIRECTIONS

The paper proposes a new active behavioral validation layer for minimizing false positives of cloud security alerts of rule-based CSPM solutions. Lightweight goal-based probes are employed for validating actual resource exposure and credential status, and the solution significantly enhances alert accuracy as well as analyst productivity without the degradation of detection accuracy with cloud native or open source solutions.

Experimental measurement in an AWS controlled environment showed substantial reductions in false positives across typical cloud misconfiguration. The results offer the promise of behaviorally validating enhanced operational efficacy at the expense of reduced alert fatigue in large-scale cloud security operations.

In line with our prior work [18], we are also exploring behavioral query driven mechanisms to further reduce false positives. This prior work is currently under peer-review, but is cited here as ongoing complementary research.

Future work will include extending behavioral validation support to Azure and GCP. This will involve incorporating cloud-specific probe implementations while preserving the modular design principles demonstrated in the AWS evaluation. Adaptive, AI-driven validation methods may be used to further improve alert prioritization and response automation. The proposed framework is a stepping stone to more contextual, dynamic cloud security posture management and reflects the need for ongoing collaboration between practitioners and academic researchers towards solving new cloud security challenges.

APPENDIX

A. REPRODUCIBILITY DETAILS

A.1 Cloud Setup

- **Platform:** AWS Free Tier
- **Regions Used:** us-east-1, us-west-2
- **Resources Created:**

- 50 S3 buckets with varied ACLs and bucket policies, files level permissions
- 40 EC2 instances (with/without public IPs and firewall rules)
- 30 IAM users (with active, rotated, and disabled access keys)
- 20 fake secrets pushed to public GitHub repositories

A.2 Alert Simulation

- **Tools Used:** AWS Config Rules, Custom Python-based CSPM Scanner, Prowler
- **Baseline Rules Simulated:**
 - s3-bucket-public-read-prohibited
 - ec2-instance-no-public-ip
 - iam-access-key-access-check
 - Custom high-entropy credential scan

A.3 Behavioral Validation Scripts

- **Public S3 Check:**
`curl -I https://<bucket>.s3.amazonaws.com/`
- **EC2 Exposure Scan:**
`nmap -Pn -p 22,3389 <public-ip>`
- **Test Credential Use:**
`curl -H "Authorization: Basic $(echo -n 'user:pass' | base64)" https://api.example.com`
- **Check Access Key Status:**
`aws iam get-access-key-last-used`
- **Secrets Discovery + Validation:**
`trufflehog git https://github.com/yourorg/test-repo.git --json > secrets.json`
Parse discovered secrets and attempt validation using:
`aws sts get-caller-identity --access-key <key> --secret-key <secret>`
- **Run Full Validation Script:**
`python3 validate_prowler.py prowler_findings.json output.json --aws-region us-east-1 --threads 10`
Replace `prowler_findings.json` with the path to the Prowler output file, `output.json` with the desired output file, and adjust the AWS region or thread count as needed.
- **EC2 Individual Port Scan:**
`python3 port_scanner.py <public-ip>`
Scans TCP ports 22 and 3389 on the specified host to detect possible SSH or RDP exposure. Reports open ports or confirms absence of these services.
- **S3 Bucket Exposure Check:**
`python3 check_s3_exposure.py <bucket_name> <region>`
Checks if the specified S3 bucket is publicly accessible by sending an HTTPS GET request to the bucket URL. Reports if the bucket is public, private (forbidden), or nonexistent.

- **AWS Credentials Validation:**

`python3 validate_aws_credentials.py`

Prompts interactively for AWS Access Key ID, Secret Access Key, and AWS region. Validates the credentials by attempting to list S3 buckets using the AWS SDK. Reports if credentials are valid or invalid.

A.4 Evaluation Process

- Parse JSON outputs from AWS CLI and scanners
- Cross-reference with alert metadata
- Generate false positive metrics (before vs. after validation)

A.5 Code Repository

The complete dataset and scripts are available at:
<https://github.com/27dikshant/Noisecut>

B. REFERENCES

- [1] Y. Liu, X. Shu, Y. Sun, J. Jang, and P. Mittal, "A context-aware clustering approach for assisting operators in classifying security alerts," *IEEE Transactions on Software Engineering*, vol. 51, no. 1, pp. 153–171, Jan. 2025.
- [2] F. Jalalvand, M. B. Chhetri, S. Nepal, and C. Paris, "Alert prioritisation in security operations centres: A systematic survey on criteria and methods," *ACM Computing Surveys*, vol. 57, no. 2, art. no. 42, pp. 1–36, Nov. 2024.
- [3] J. Ghadermazi, A. Shah, and S. Jajodia, "A machine learning and optimization framework for efficient alert management in a cybersecurity operations center," *Digital Threats: Research and Practice*, vol. 5, no. 2, pp. 1–23, 2024.
- [4] S. Tariq, M. B. Chhetri, C. Paris, and S. Nepal, "Alert fatigue in security operations centres: Research challenges and opportunities," *ACM Computing Surveys*, vol. 57, no. 9, Article 224, 2025.
- [5] X. Wang, X. Yang, X. Liang, X. Zhang, W. Zhang, and X. Gong, "Combating alert fatigue with AlertPro: Context-aware alert prioritization using reinforcement learning for multi-step attack detection," *Computers & Security*, vol. 137, art. no. 103583, Feb. 2024.
- [6] F. Voutsas, J. Violos, and A. Leivadreas, "Mitigating alert fatigue in cloud monitoring systems: A machine learning perspective," *Computer Networks*, vol. 250, art. no. 110543, 2024.
- [7] Q. Tang *et al.*, "DeepARR: Alert risk rating based on deep learning," in *Proc. Int. Symp. Parallel Distrib. Process. Appl.*, IEEE, 2024, pp. 2097–2104.
- [8] M. B. Chhetri, S. Tariq, R. Singh, F. Jalalvand, C. Paris, and S. Nepal, "Towards human-AI teaming to mitigate alert fatigue in security operations centres," *ACM Trans. Internet Technol.*, vol. 24, no. 1, pp. 1–22, 2024.
- [9] B. A. Alahmadi, L. Axon, and I. Martinovic, "99% False Positives: A Qualitative Study of SOC Analysts' Perspectives on Security Alarms," in *Proc. of the 31st USENIX Security Symposium (USENIX Security '22)*, pp. 2783–2800, 2022.

- [10] R. Rani, G. Epiphaniou, and C. Maple, "Reinforcement learning-based alert prioritisation in security operation centre: A framework for enhancing cybersecurity in the digital economy," in *IET Conference Proceedings*, vol. 2023, pp. 151-157, 2023.
- [11] M. Landauer, F. Skopik, M. Wurzenberger, and A. Rauber, "Dealing with security alert flooding: Using machine learning for domain-independent alert aggregation," *ACM Trans. Priv. Secur.*, vol. 25, no. 18, pp. 1–36, 2022.
- [12] A. Malach, P. N. Wudali, S. Momiyama, J. Furukawa, T. Araki, Y. Elovici, and A. Shabtai, "CyberShapley: Explanation, prioritization, and triage of cybersecurity alerts using informative graph representation," *Computers & Security*, vol. 150, Article 104270, 2025.
- [13] A. B. Nassif, M. A. Talib, Q. Nasir, H. Albadani, and F. M. Dakalbab, "Machine learning for cloud security: A systematic review," *IEEE Access*, vol. 9, pp. 20717–20735, 2021.
- [14] M. E. Aminanto, T. Ban, R. Isawa, T. Takahashi, and D. Inoue, "Threat Alert Prioritization Using Isolation Forest and Stacked Auto Encoder with Day-Forward-Chaining Analysis," *IEEE Access*, vol. 8, pp. 217977–217986, 2020.
- [15] A. M. Abdallah, A. S. R. O. A. Alkaabi, G. B. N. D. Alameri, S. H. Rafique, N. S. Musa and T. Murugan, "Cloud Network Anomaly Detection Using Machine and Deep Learning Techniques—Recent Research Advancements," *IEEE Access*, vol. 12, pp. 56749–56773, 2024.
- [16] T. Ahmed, A. Shah, M. Kolla and R. Yellashiri, "Reduction of Alert Fatigue using Extended Isolation Forest," *2021 International Conference on Forensics, Analytics, Big Data, Security (FABS)*, Bengaluru, India, pp. 1-5, 2021.
- [17] K. A. Torkura, A. Acharya, and C. Okoli, "CloudStrike: Chaos engineering for security and resiliency in cloud infrastructure," *IEEE Access*, vol. 8, pp. 123044–123060, 2020.
- [18] Dikshant and G. Verma, "Precision over Noise: Tailoring S3 Public Access Detection to Reduce False Positives in Cloud Security Platforms," *arXiv preprint arXiv:2508.14402*, 2025.

Convergence of off-policy TD(0) with linear function approximation for reversible Markov chains

Maik Overmars, Jasper Goseling, Richard J. Boucherie
Stochastic Operations Research, University of Twente

m.g.overmars@utwente.nl, j.goseling@utwente.nl, r.j.boucherie@utwente.nl

ABSTRACT

We study the convergence of off-policy TD(0) with linear function approximation when used to approximate the expected discounted reward in a Markov chain. It is well known that the combination of off-policy learning and function approximation can lead to divergence of the algorithm. Existing results for this setting modify the algorithm, for instance by reweighing the updates using importance sampling. This establishes convergence at the expense of additional complexity. In contrast, our approach is to analyse the standard algorithm, but to restrict our attention to the class of reversible Markov chains. We demonstrate convergence under this mild reversibility condition on the structure of the chain, which in many applications can be assumed using domain knowledge. In particular, we establish a convergence guarantee under an upper bound on the discount factor in terms of the difference between the on-policy and off-policy process. This improves upon known results in the literature that state that convergence holds for a sufficiently small discount factor by establishing an explicit bound. Convergence is with probability one and achieves projected Bellman error equal to zero. To obtain these results, we adapt the stochastic approximation framework that was used by Tsitsiklis and Van Roy [1997] for the on-policy case, to the off-policy case. We illustrate our results using different types of reversible Markov chains, such as one-dimensional random walks and random walks on a weighted graph.

1. INTRODUCTION

Temporal-difference learning (TD learning) [Sutton, 1988] is a reinforcement learning algorithm that aims to predict the expected reward of a dynamical system. The convergence of TD learning has been well studied, for instance by considering different ways to approximate the value function that represents the predicted reward, and by using either on-policy or off-policy sampling of states. We know that in the tabular case, on-policy TD learning converges to the optimal prediction [Sutton, 1988, Dayan and Sejnowski, 1994, Jaakkola et al., 1994]. Under linear value function approximation, on-policy TD learning algorithm still converges. The point of convergence is no longer optimal, but does have a projected Bellman error (PBE) of zero [Tsitsik-

lis and Van Roy, 1997, Dayan, 1992, Tsitsiklis, 1994, Tadić, 2001]. However, off-policy learning with function approximation may diverge, as shown through examples by Baird [1995] and Tsitsiklis [1994].

All of the above work considers the convergence of TD learning for a general input, i.e., for any possible Markov chain. In this work, by restricting our attention to the class of reversible Markov chains, we are able to prove convergence of the off-policy TD(0) algorithm under linear function approximation. The class of reversible Markov chains covers many application areas, such as queueing theory [Allen, 2014], statistical mechanics [Davidson, 2013], and Bayesian statistics [Bolstad and Curran, 2016]. We show that off-policy TD(0) with linear function approximation converges whenever the on-policy and off-policy chain are reversible and have the same structure, under a condition on the discount factor. In particular, the difference between the on-policy and off-policy chain imposes an upper bound on the discount factor. Finally, we show in Remark 1 that our results also apply to a more general setting, where the reversibility restriction is not necessary.

The stability of TD(0) learning is commonly analysed by considering the update rule in general form

$$w_{t+1} = w_t + \alpha_t(A(X_t)w_t + b(X_t)), \quad (1)$$

where X_t is a Markov chain describing the sampling process, w_t is the parameter vector that is learned, α_t is the step size, and $A(X_t)$ and $b(X_t)$ contain the details of the update rule. Also, define matrix A as $A(X_t)$ for X_t in steady-state. The core requirement for convergence is that matrix A is negative definite [Tsitsiklis and Van Roy, 1997, Dayan, 1992, Sutton et al., 2016, Sutton and Barto, 2018]. Matrix A is negative definite if and only if $x^T A x < 0$ for all $x \neq 0$. Tsitsiklis and Van Roy [1997] showed that this matrix is always negative definite for on-policy TD learning with linear function approximation. We generalise their result by showing that for reversible Markov chains, matrix A is negative definite for off-policy TD learning with linear function approximation. We show that $A = \Phi^T(\gamma\hat{Q}P - \hat{Q})\Phi$, where the elements of this matrix are the linear features Φ , discount factor γ , off-policy distribution $\hat{Q}$, and the transition probabilities P corresponding to the original chain. If the on-policy and off-policy processes are very different, then the difference between $\hat{Q}P$ and $\hat{Q}$ can be large, and A may not be negative definite. Decreasing the discount factor γ can mitigate this effect, since A always becomes negative definite when γ approaches zero. In this paper we address the open question as to how small the discount factor should be to achieve

convergence.

Our main contribution is that we establish convergence with probability one of off-policy TD(0) with linear function approximation for the class of reversible Markov chains. Our analysis builds on the work by Tsitsiklis and Van Roy [1997], and extends the convergence of on-policy learning to the off-policy case. We model the Markov chain of the off-policy process as a perturbation of the original chain, where this perturbation describes the difference between $\hat{Q}P$ and $\hat{Q}$. We derive an exact upper bound on the discount factor γ in terms of this perturbation, and we show that the algorithm converges when this upper bound is satisfied. Larger perturbations imply that the discount factor should be chosen small, while a large discount factor limits the size of the perturbation. The point of convergence is the fixed point of the projected Bellman operator, and this point has a projected Bellman error (PBE) of zero. Finally, we show how to apply these results to common classes of reversible Markov chains, such as one-dimensional random walks, and random walks on a weighted graph.

In the literature other approaches have been introduced to deal with the divergence of off-policy TD learning. Some convergence results are available for the off-policy linear function approximation case, but these rely on modifying the algorithm. Following his example of divergence, Baird [1995] suggested residual algorithms, but these are slow to converge and require double samples. Similarly, least-squares temporal-difference (LSTD) as proposed by Nedić and Bertsekas [2003] is stable but has quadratic per-step complexity in the number of parameters. A similar approach is used by gradient TD algorithms, like gradient temporal-difference (GTD) methods by Sutton et al. [2008] and linear TD with gradient correction (TDC) by Sutton et al. [2009], which perform true gradient descent in the Bellman error. A generalization of these algorithms to non-linear function approximation was presented by Maei et al. [2009]. These algorithms converge and do not encounter the double sample issue, but they require a second set of parameters with a second step size, and thus are more complex to implement. Importance sampling techniques [Sutton et al., 2016, Precup et al., 2001] have also been used to stabilise the algorithm by reweighing the updates, but these methods can have high variance depending on the setting. On the other hand, standard TD learning works well in many settings, has low complexity, and is easy to implement. Based on the current work, if reversibility can be assumed, the standard algorithm is guaranteed to converge. Also, our models and analysis opens the way to obtain bounds on the convergence rate similarly to Srikant and Ying [2019].

In Section 2 we specify the Markov chain model that is studied and the TD learning algorithm including linear function approximation and off-policy learning. Section 3 contains the main convergence result of this paper. In Section 4 we apply the result to two types of reversible Markov chains: one-dimensional random walks and random walks on a weighted graph. Finally, Section 5 contains the conclusion, discussion and outlook for future work.

2. MODEL FORMULATION

In this section we specify the Markov chain model that we study as well as off-policy TD learning with linear function approximation. Moreover, we connect this to a stochastic approximation result and define the matrix A , which we will

use in our analysis to establish convergence conditions. Regarding notation: Unless otherwise specified, all vectors are column vectors, and we will interchangeably use subscripts and parentheses to denote elements of vectors. Also, $\|v\|$ denotes the Euclidean norm of vector v and $\|A\|$ denotes the corresponding induced norm for matrix A .

We consider a *reversible*, irreducible and aperiodic Markov chain $\{S_t \in \mathcal{S} \mid t = 0, 1, \dots\}$ with finite state space $\mathcal{S} = \{1, \dots, n\}$ [Aldous and Fill, 2002]. The transition probability matrix is denoted by $P \in \mathbb{R}^{n \times n}$, where entry p_{ij} gives the probability of a transition from state i to state j . Let $q \in \mathbb{R}^n$ be the row vector that denotes the unique steady-state probability distribution satisfying $q = qP$. Let $Q = \text{diag}(q)$, the diagonal matrix with the elements of q on the diagonal. Markov chain S_t is reversible if and only if $q_i p_{ij} = q_j p_{ji}$ for any pair of states $i, j \in \mathcal{S}$. This implies that $p_{ij} > 0 \iff p_{ji} > 0$, such that we can define the *transition ratio*

$$\rho_{ij} := \frac{p_{ij}}{p_{ji}}. \quad (2)$$

for any $i, j \in \mathcal{S}$. As a result, $\rho_{ji} = \rho_{ij}^{-1}$ exists and we can write

$$q_j = q_i \rho_{ij}. \quad (3)$$

We denote by $r \in \mathbb{R}^n$ the reward vector over the states, where $r(i)$ gives the reward for state i . Our interest is in the expected sum of discounted rewards, defined through the value function $V : \mathcal{S} \rightarrow \mathbb{R}$, with $V(i)$ the value when starting in state i , i.e.,

$$V(i) := \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t r(S_t) \mid S_0 = i \right], \quad (4)$$

where $\gamma \in (0, 1)$ is the discount factor. We aim to approximate V using linear function approximation. In particular, let

$$V_w(i) := \sum_{k=1}^K w(k) \phi_k(i), \quad (5)$$

where $\phi_k \in \mathbb{R}^n$ for $k = 1, \dots, K$ are the feature vectors, $w \in \mathbb{R}^K$ are the weights, and $i \in \mathcal{S}$. We define $\Phi \in \mathbb{R}^{n \times K}$ as the matrix where column k corresponds to vector ϕ_k , such that we can write

$$V_w = \Phi w. \quad (6)$$

We assume that feature matrix Φ has full column rank, i.e., the basis vectors ϕ_k , $k = 1, \dots, K$, are linearly independent. We will use the notation $\phi(s) = (\phi_1(s), \phi_2(s), \dots, \phi_K(s))$, $s \in \mathcal{S}$, i.e., $\phi(s)$ is the transpose of a row in Φ .

Next, we introduce the off-policy TD(0) algorithm. This algorithm iteratively updates the weights in w with the aim of approximating V by V_w . The update rule is

$$w_{t+1} = w_t + \alpha_t d_t \phi(\hat{S}_t), \quad (7)$$

where $\alpha_t > 0$ is the step size, d_t is the temporal difference which will be specified below, and $\hat{S}_t \in \mathcal{S}$ is a state that is sampled at time t . The off-policy property means that the states $\hat{S}_t$ are sampled from a different Markov chain than $\{S_t \in \mathcal{S} \mid t = 0, 1, \dots\}$, the Markov chain that the algorithm aims to estimate V for. In particular, we have reversible, irreducible and aperiodic Markov chain $\{\hat{S}_t \in \mathcal{S} \mid t = 0, 1, \dots\}$ with transition probability matrix $\hat{P}$, steady-state probability distribution $\hat{q}$ and $\hat{Q} = \text{diag}(\hat{q})$. We assume that

$$\hat{p}_{ij} > 0 \iff p_{ij} > 0, \quad (8)$$

such that both Markov chains have the same structure. We will refer to $\{S_t \in \mathcal{S} \mid t = 0, 1, \dots\}$ and $\{\hat{S}_t \in \mathcal{S} \mid t = 0, 1, \dots\}$ as the original and the perturbed Markov chain, respectively. The temporal difference d_t is based on a one-step transition in the original Markov chain when starting from state $\hat{S}_t$, i.e. let S'_t be defined as

$$\mathbb{P}(S'_t = j \mid \hat{S}_t = i) = p_{ij}. \quad (9)$$

The temporal difference is then defined as

$$d_t := r(\hat{S}_t) + \gamma V_{w_t}(S'_t) - V_{w_t}(\hat{S}_t), \quad (10)$$

$$= r(\hat{S}_t) + \gamma \phi^T(S'_t)w_t - \phi^T(\hat{S}_t)w_t. \quad (11)$$

We assume that the step sizes α_t are positive, nonincreasing, and satisfy the Robbins-Monro conditions [Robbins and Monro, 1951]

$$\sum_{t=0}^{\infty} \alpha_t = \infty, \quad (12)$$

$$\sum_{t=0}^{\infty} \alpha_t^2 < \infty. \quad (13)$$

By rewriting (7) as

$$w_{t+1} = w_t + \alpha_t \left(A(\hat{S}_t, S'_t)w_t + b(\hat{S}_t, S'_t) \right), \quad (14)$$

with

$$A(\hat{S}_t, S'_t) := \phi(\hat{S}_t) \left(\gamma \phi^T(S'_t) - \phi^T(\hat{S}_t) \right), \quad (15)$$

and

$$b(\hat{S}_t, S'_t) := \phi(\hat{S}_t)r(\hat{S}_t), \quad (16)$$

we see that it is an instance of a stochastic approximation algorithm that is well studied. In particular, we have the following convergence result.

Theorem 1 (Theorem 17 from Benveniste et al. [2012]). *Consider an iterative algorithm of the form*

$$w_{t+1} = w_t + \alpha_t (A(X_t)w_t + b(X_t)), \quad (17)$$

where

1. the step size sequence α_t is positive, nonincreasing, and satisfies $\sum_{t=0}^{\infty} \alpha_t = \infty$ and $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$;
2. X_t is a Markov process with unique steady-state distribution, and X denotes the random variable distributed according to this steady-state distribution. Furthermore, there exists a mapping $h : \mathcal{S} \rightarrow \mathbb{R}$ satisfying the remaining conditions;
3. $A(\cdot)$ and $b(\cdot)$ are matrix and vector valued functions, for which $A = \mathbb{E}[A(X)]$ and $b = \mathbb{E}[b(X)]$ are well defined and finite;
4. matrix A is negative definite;
5. there exists constants M and g such that for all x

$$\sum_{t=0}^{\infty} \|\mathbb{E}[A(X_t) \mid X_0 = x] - A\| \leq M(1 + h^g(x)), \quad (18)$$

and

$$\sum_{t=0}^{\infty} \|\mathbb{E}[b(X_t) \mid X_0 = x] - b\| \leq M(1 + h^g(x)); \quad (19)$$

6. for any $g > 1$, there exists a constant μ_g such that for all X, t

$$\mathbb{E}[h^g(X_t) \mid X_0 = x] \leq \mu_g(1 + h^g(x)). \quad (20)$$

Then, w_t converges to w^* , where w^* is the unique vector that satisfies $Aw^* + b = 0$.

Our statement of the above theorem is in fact, a special case of the more general Theorem 17 from Benveniste et al. [2012]. The above form appears in Tsitsiklis and Van Roy [1997, Theorem 2]. Most of our analysis will deal with verifying the conditions of this theorem for our setting in which $X_t = (\hat{S}_t, S'_t)$. To set the stage for this, we define matrix A and vector b as

$$A := \Phi^T(\gamma \hat{Q}P - \hat{Q})\Phi, \quad \text{and} \quad b := \Phi^T \hat{Q}r. \quad (21)$$

Crucial parts of our analysis will be to show that $A = \mathbb{E}[A(\hat{S}, S')]$ and $b = \mathbb{E}[b(\hat{S}, S')]$, and to derive conditions under which A is negative definite, henceforth abbreviated as n.d.

In addition to convergence guarantees, we study the point of convergence w^* . In particular, we will show that w^* minimises the projected Bellman error (PBE) [Sutton and Barto, 2018], which we define next. In order to do so, consider the inner product $\langle x, y \rangle_{\hat{Q}} := x^T \hat{Q}y$, and the associated weighted Euclidean norm $\|\cdot\|_{\hat{Q}} := \sqrt{\langle \cdot, \cdot \rangle_{\hat{Q}}}$ on the space $\{\Phi w \mid w \in \mathbb{R}^K\}$ of value function approximations given linear features Φ . Now,

$$\text{PBE}(w) := \|\Pi_{\hat{Q}}\mathcal{B}(\Phi w) - \Phi w\|_{\hat{Q}}, \quad (22)$$

where $\mathcal{B}$ is the Bellman operator that satisfies $V = \mathcal{B}V$, and $\Pi_{\hat{Q}}$ is the projection operator from $\mathbb{R}^n$ onto $\{\Phi w \mid w \in \mathbb{R}^K\}$ with respect to the norm $\|\cdot\|_{\hat{Q}}$.

3. MAIN RESULTS

In this section we state the main result of this paper, that is the convergence of temporal-difference learning applied on reversible Markov chains with linear function approximation and off-policy learning.

Theorem 2. *Let $c \geq 1$ satisfy*

$$\frac{1}{c}\rho_{ij} \leq \hat{\rho}_{ij} \leq c\rho_{ij}, \quad (23)$$

for all $i, j \in \mathcal{S}$ for which $p_{ij} > 0$, and let

$$\gamma < \frac{2}{c+1}. \quad (24)$$

Then w_t converges to w^* with probability 1, where w^* is the unique vector that satisfies $\text{PBE}(w^*) = 0$.

Remark 1. Careful analysis of the proof of Theorem 2 shows that we do not need reversibility of the original Markov chain. In fact, the original chain only needs to satisfy $p_{ij} > 0$ iff $p_{ji} > 0$ for any states i and j , such that ρ_{ij} exists. Thus, for any transition in the original Markov chain the reverse transition probability should also be positive.

The factor c determines the size of the perturbation, in other words how different the off-policy distribution is from the on-policy distribution. The proof of Theorem 2 and the proofs for the other results provided in this paper are given in Overmars et al. [2025]. We do, however, present the main

technical lemma that is used in the proof here. In particular, we give a sufficient condition for A as defined in (21) to be n.d.

Lemma 1. *If*

$$\gamma < \frac{2}{1 + \frac{1}{\hat{q}_i} \sum_{j=1}^n \hat{q}_j p_{ji}} \quad (25)$$

for all $i \in \mathcal{S}$, then matrix A is n.d.

The proof for this result is based on establishing diagonal dominance, see for instance Horn and Johnson [1985, Corollary 7.2.3]. Note, that this is a sufficient but not necessary condition for A to be n.d., as a matrix being n.d. does not imply that it is also diagonally dominant. To illustrate (25) consider the on-policy case where $\hat{Q} = Q$. There we have $\hat{q}_i = \sum_{j=1}^n \hat{q}_j p_{ji}$, and we can rewrite (25) as

$$\gamma < \frac{2}{1 + \frac{1}{\hat{q}_i} \sum_{j=1}^n \hat{q}_j p_{ji}} = \frac{2}{1 + \frac{1}{\hat{q}_i} \hat{q}_i} = 1, \quad (26)$$

which is always satisfied as $\gamma \in (0, 1)$. Hence, we establish the result of Tsitsiklis and Van Roy [1997] result that on-policy TD(0) learning with linear function approximation converges as a special case of Theorem 2.

Though it is not stated explicitly, Lemma 1 and its proof do not require reversibility of either the original or the perturbed Markov chain. However, Lemma 1 seems to be most useful in the case of reversible Markov chains, where limited information about the original and the perturbed chain may provide convergence guarantees. One instance of this is Theorem 2, where we only require a bound the perturbation of the transition ratios. In Section 4 we give another application of Lemma 1 that demonstrates that if more information about the process is available, tighter bounds on γ can be obtained.

4. APPLICATIONS

In this section, we first show how to apply Theorem 2 to the class of random walks on a weighted graph in the case when a bound on the perturbations of the weights is available. Next, we apply Theorem 2 to the class of one-dimensional random walks. Furthermore, using Lemma 1 we show how to obtain better bounds if more information about the process is known.

4.1 Random walk on a weighted graph

Consider a random walk on a weighted graph [Aldous and Fill, 2002]. We define the undirected, complete graph $G = (V, E)$ with vertices V and edges E . Edge $\{i, j\} \in E$ for $i, j \in V$ has weight $u_{ij} > 0$. Since edges are undirected we have $u_{ij} = u_{ji}$. We then consider the Markov chain with transition probabilities defined as

$$p_{ij} = \frac{u_{ij}}{\sum_{k=1}^n u_{ik}}. \quad (27)$$

The equilibrium distribution for this process equals

$$q_i = \frac{\sum_{k=1}^n u_{ik}}{\sum_{\ell=1}^n \sum_{k=1}^n u_{\ell k}}. \quad (28)$$

It can be easily shown that this process is reversible. Furthermore, every reversible Markov chain can be rewritten as a random walk on a weighted graph [Aldous and Fill, 2002]. This is done by setting the weights as $u_{ij} = q_i p_{ij}$, so

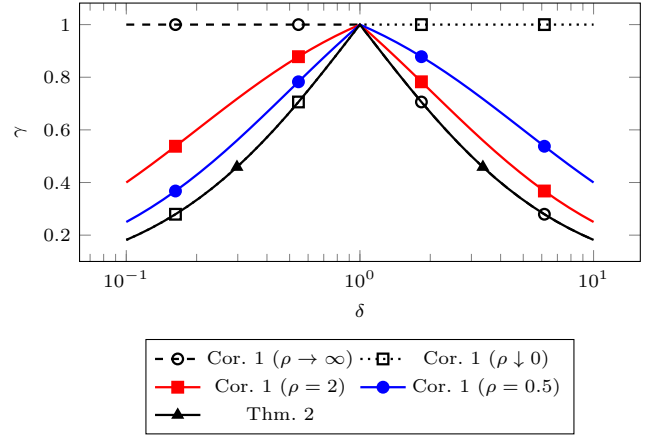


Figure 1: Upper bounds on γ given by Theorem 2 and Corollary 1 for the simple random walk.

that detailed balance immediately holds using $u_{ij} = u_{ji}$. In this way, we can apply our result to any reversible Markov chain by bounding the weights in a similar manner as shown below. However, this requires knowledge of stationary distributions q and $\hat{q}$ in order to define the weights. As a result, it is easier for a general Markov chain to apply Theorem 2 by bounding the ratios of transition probabilities directly.

Now consider a perturbed process defined on the same graph G with weights $\hat{u}_{ij} > 0$. Moreover, suppose that the difference in weights between the original and the perturbed process can be bounded as

$$\frac{1}{\delta} \leq \frac{\hat{u}_{ij}}{u_{ij}} \leq \delta, \quad (29)$$

for all edges $\{i, j\} \in E$, for some $\delta \geq 1$. Using $\hat{u}_{ij} = \hat{u}_{ji}$ and $u_{ij} = u_{ji}$ we can write

$$\frac{\hat{p}_{ij}}{\hat{p}_{ji}} = \frac{\sum_{k=1}^n \hat{u}_{jk}}{\sum_{k=1}^n \hat{u}_{ik}} \quad (30)$$

$$\leq \frac{\delta \sum_{k=1}^n u_{jk}}{\delta^{-1} \sum_{k=1}^n u_{ik}} \quad (31)$$

$$= \delta^2 \frac{p_{ij}}{p_{ji}}. \quad (32)$$

A similar argument can be made for the other direction, so that we have

$$\frac{1}{\delta^2} \leq \frac{\hat{p}_{ij}}{p_{ij}} \leq \delta^2. \quad (33)$$

Hence, we can apply Theorem 2 with $c = \delta^2$.

4.2 One-dimensional random walk

Consider the case that the original and the perturbed chain are one-dimensional random walks on n states, i.e., $p_{ij} > 0$ and $\hat{p}_{ij} > 0$ if and only if i) $|i - j| = 1$, or ii) $i = j = 1$, or iii) $i = j = n$. For these random walks it can be easily verified that detailed balance holds. As a result, they are reversible and the stationary distributions can be shown to equal

$$q_i = q_1 \prod_{j=1}^{i-1} \rho_{j,j+1} \quad (34)$$

for $i = 2, \dots, n$, where q_1 follows from normalization and where a similar expression holds for $\hat{q}_i$. Note, that the transition ratios capture the drift of the random walk to the left or right side of the state space. A transition ratio $\rho_{i,i+1} > 1$ corresponds to a drift to the right side, while a transition ratio $\rho_{i,i+1} < 1$ corresponds to a drift to the left side.

Let

$$c_{ij} = \begin{cases} \frac{\hat{\rho}_{ij}}{\rho_{ij}} & \text{if } p_{ij}, \hat{p}_{ij} > 0, \\ 1 & \text{otherwise,} \end{cases} \quad (35)$$

for $i, j \in \{1, \dots, n\}$ and

$$c = \max_{i,j \in \{1, \dots, n\}} \left\{ \max \left\{ c_{ij}, \frac{1}{c_{ij}} \right\} \right\}. \quad (36)$$

The interpretation of c is that it is the maximum relative difference in drift between the original and the perturbed chain. From Theorem 2 it follows that as long as our feature matrix is full rank and $\gamma < 2/(c+1)$, TD(0) converges.

Next, we demonstrate how Lemma 1 can be used to obtain a better bound on γ if more information about the original and perturbed process are available. In particular, we restrict our attention to the simple random walk in which the transition ratios are constant and known for all states.

Corollary 1. *Let the original and the perturbed process be simple random walks on n states with $\rho_{i,i+1} = \rho$ and $\hat{\rho}_{i,i+1} = \hat{\rho}$ for $i = 1, \dots, n-1$. Let $\delta = \hat{\rho}/\rho$ and*

$$\gamma < \min \left\{ \frac{2(\rho+1)}{\rho+2+\delta\rho}, \frac{2(\rho+1)}{2\rho+1+\delta^{-1}} \right\}. \quad (37)$$

Then w_t converges to w^ with probability 1, where w^* is the unique vector that satisfies $\text{PBE}(w^*) = 0$.*

The difference with the proof of Theorem 2, in which no structural information besides reversibility is available, is that for this corollary we use the random walk structure of the process. The key idea is that Lemma 1 imposes a condition for each state and that it turns out that the conditions imposed by boundary states 1 and n are the most stringent. Moreover, these conditions can be loosened w.r.t. the bound given in Theorem 2, most importantly because $\hat{\rho}/\rho$ is now known exactly.

We have illustrated the bounds of Corollary 1 in Figure 1 for $\rho = 2$ and $\rho = 0.5$ and the limiting behavior for $\delta \rightarrow \infty$ and $\delta \downarrow 0$. For comparison, we have also depicted the bound of Theorem 2 by using $c = \delta^{-1}$ for $\delta < 1$. The figure illustrates that Lemma 1 allows for better bounds than given in Theorem 2 by using more information. If such information is available depends on the application. The value of Theorem 2 is that it only requires a bound on the difference between the original and the perturbed process.

5. CONCLUSION AND DISCUSSION

In this paper, we have shown convergence of off-policy temporal-difference learning with linear function approximation for reversible Markov chains. Unlike previous research, which modifies the temporal-difference learning algorithm to establish convergence results, we instead consider the base algorithm. Key in our analysis is the matrix $A = \Phi^T(\gamma\hat{Q}P - \hat{Q})\Phi$, which is required to be negative definite for the algorithm to converge. In our setting, we show

that this matrix is negative definite under a condition between the discount factor γ and the size of the perturbation caused by off-policy learning.

Our work may be extended to more general settings by eliminating various assumptions at the cost of complexity of the proof. Think of the assumption of a finite state space, or linear independence of the basis functions, or aperiodicity of the underlying Markov chains, as mentioned by Tsitsiklis and Van Roy [1997] for the on-policy case.

A central assumption in this paper is reversibility of the Markov chains, and extending our results beyond this assumption would improve their applicability. Reversibility is a technical assumption in our analysis, as it allows us to manipulate the sum $\sum_{j=1}^n \hat{q}_j p_{ji}$ found in Lemma 1. This sum captures the size of the off-policy behaviour in our model. In Remark 1, we have already shown that our results also hold if the original Markov chain is not reversible, but still satisfies $p_{ij} > 0 \iff p_{ji} > 0$ for any $i, j \in \mathcal{S}$, i.e., the reverse transition always exists. It is worth investigating to extend our result to the case where the perturbed chain is also not reversible, and instead satisfies the assumption that the reverse transition exists. This extension requires changing our analysis and it seems to lead to a more complex result as we can no longer use the definition of reversibility to rewrite the terms in the sum mentioned above. As a result, we would have to add correction terms to account for this, which may in turn change the bound on γ .

Another extension is to consider the rate of convergence instead of the asymptotic convergence shown in this paper. Srikant and Ying [2019] derive finite-time error bounds for a stochastic approximation scheme, which they apply to TD learning with linear function approximation in the on-policy case. Our model follows a very similar set of assumptions, so we expect that we can extend the results from Theorems 7 and 9 in Srikant and Ying [2019] to our setting. These results place bounds on the expected error between the iterates and the TD-fixed point, and the higher moments of this error, respectively. However, Srikant and Ying [2019] mostly assume a constant step size, while we assume the Robbins-Monro conditions [Robbins and Monro, 1951]. The authors do briefly cover diminishing step sizes in Theorem 12, but this result requires technical assumptions on the step size for small values of t . Furthermore, these assumptions involve terms such as the mixing time, which depends on the structure of the Markov chain. Thus, the main challenge will be to derive corresponding assumptions for our setting.

Finally, we can test the tightness of the bound on γ as given in (24). We have performed numerical experiments to find the maximum value of γ under which A is still negative definite, which is required for convergence. We did this for the simple random walk presented in Section 4.2, such that we can also use the tighter bound given in Corollary 1. These experiments showed a significant gap between the numerically required values for γ and those obtained from our error bounds. Part of this gap is likely due to the fact that Lemma 1 is a sufficient, but not necessary condition for A to be n.d. Thus, values of γ that do not satisfy (24) can still lead to convergence. Future steps would be to derive all the terms that make up the gap, and to quantify the differences between the bounds for various settings.

References

- John N. Tsitsiklis and Benjamin Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE transactions on automatic control*, 42(5):674–690, 1997.
- Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3:9–44, 1988.
- Peter Dayan and Terrence J. Sejnowski. TD (λ) converges with probability 1. *Machine Learning*, 14:295–301, 1994.
- Tommi Jaakkola, Michael I. Jordan, and Satinder P. Singh. On the convergence of stochastic iterative dynamic programming algorithms. *Neural computation*, 6(6):1185–1201, 1994.
- Peter Dayan. The convergence of TD (λ) for general λ . *Machine learning*, 8(3-4):341–362, 1992.
- John N. Tsitsiklis. Asynchronous stochastic approximation and Q-learning. *Machine learning*, 16(3):185–202, 1994.
- Vladislav Tadić. On the convergence of temporal-difference learning with linear function approximation. *Machine learning*, 42:241–267, 2001.
- Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pages 30–37. Elsevier, 1995.
- Arnold O. Allen. *Probability, statistics, and queueing theory*. Academic press, 2014.
- Norman Davidson. *Statistical mechanics*. Courier Corporation, 2013.
- William M. Bolstad and James M. Curran. *Introduction to Bayesian statistics*. John Wiley & Sons, 2016.
- Richard S. Sutton, A. Rupam Mahmood, and Martha White. An emphatic approach to the problem of off-policy temporal-difference learning. *Journal of Machine Learning Research*, 17(73):1–29, 2016.
- Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- Angelia Nedić and Dimitri P. Bertsekas. Least squares policy evaluation algorithms with linear function approximation. *Discrete Event Dynamic Systems*, 13(1):79–110, 2003.
- Richard S. Sutton, Hamid R. Maei, and Csaba Szepesvári. A convergent $o(n)$ temporal-difference algorithm for off-policy learning with linear function approximation. *Advances in neural information processing systems*, 21, 2008.
- Richard S. Sutton, Hamid R. Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 993–1000, 2009.
- Hamid R. Maei, Csaba Szepesvari, Shalabh Bhatnagar, Doina Precup, David Silver, and Richard S. Sutton. Convergent temporal-difference learning with arbitrary smooth function approximation. In *NIPS*, pages 1204–1212, 2009.
- Doina Precup, Richard S. Sutton, and Sanjoy Dasgupta. Off-policy temporal-difference learning with function approximation. In *ICML*, pages 417–424, 2001.
- Rayadurgam Srikant and Lei Ying. Finite-time error bounds for linear stochastic approximation and TD learning. In *Conference on Learning Theory*, pages 2803–2830. PMLR, 2019.
- David Aldous and James A. Fill. Reversible Markov chains and random walks on graphs, 2002. Unfinished monograph, recompiled 2014, available at <http://www.stat.berkeley.edu/~aldous/RWG/book.html>.
- Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.
- Albert Benveniste, Michel Métivier, and Pierre Priouret. *Adaptive algorithms and stochastic approximations*, volume 22. Springer Science & Business Media, 2012.
- Maik Overmars, Jasper Goseling, and Richard Boucherie. Convergence of off-policy TD(0) with linear function approximation for reversible markov chains, 2025. URL <https://arxiv.org/abs/2510.25514>.
- Roger A. Horn and Charles R. Johnson. *Matrix analysis*. Cambridge university press, 1985.

Game of arrivals to a two queue network with heterogeneous customer routes

Agniv Bandyopadhyay * Sandeep Juneja†

ABSTRACT

We consider a queuing network that opens at a specified time, where users are non-atomic and belong to different classes. Each class has its own route, and as is typical in the literature, the costs are a linear function of waiting and service completion time. We restrict ourselves to a two class, two queue network: this simplification is well motivated as the diversity in solution structure as a function of problem parameters is substantial even in this simple setting (e.g., a specific routing structure involves eight different regimes), suggesting a combinatorial blow up as the number of queues, routes and user classes increase. We identify the unique Nash equilibrium user arrival profile when the user linear cost preferences are different. This profile is a function of problem parameters including the size of each class, service rates at each queue, and user cost preferences. When user cost preferences match, under certain parametric settings, the equilibrium arrival profiles may not be unique and may lie in a convex set. We further make a surprising observation that in some parametric settings, users in one class may arrive in disjoint intervals. Further, the two classes may arrive in contiguous intervals or in overlapping intervals, and at varying rates within an interval, depending upon the problem parameters.

Keywords: Strategic arrivals, queuing network games, population games

1. INTRODUCTION

Queueing games or games where strategic users, served by a queue or a queuing network, decide on actions such as which queue to join, whether to join, when to join, what level of priority to select and so on, are well studied in the literature (see [6, 7, 9] for surveys). In this paper we focus on the queuing arrival game where users decide on when to join a queuing network. This is in contrast to much of the literature that considers arrival games to a single queue (see, e.g., [3, 5, 8, 9, 11, 12, 13]). Applications of such arrival games to multiple queues are many: users arriving at a health facility where they queue up to meet a doctor and then may queue to get tests done and/or procure medicines; in banks where

different users may need to go to different and multiple counters; in cafeteria where users queue up for different and multiple food items, etc. In consonance with much of the existing literature, we model users as non-atomic ‘infinitesimal’ fluid particles with costs that are linear in waiting time and in time to service, users are served in a first-come-first-serve manner, and service facility opens at time zero (see [10, 11, 13]). In [12], the uniqueness of the equilibrium solution was proven in a single queue setting with stochastic service rates and a large number of users. Moreover, [12] showed that the equilibrium solution with a large number of users is well approximated by the corresponding fluid system, thus lending support to the fluid analysis. This fluid setting has resulted in unique and elegant, easily calculable user equilibrium profiles for single queues as well as certain symmetric queuing networks where users have homogeneous travel routes (see [10, 11]).

To keep the discussion simple, we focus on a two-queue, two-class user setting where each class of users has a distinct route, and users in each queue are served in a first-come-first-served manner. While this set-up may be practically interesting, our main contributions are theoretical. Our key aim is to test whether the elegance and simplicity of user equilibrium behavior to a single queue extends to more general queuing networks in the presence of heterogeneous routing.

Even in the simple two queue setting, we see that, unlike for the single queue, here the solution structure and order of arrivals in equilibrium is a function of all the problem parameters, *i.e.*, linear coefficients of the cost function, the queue service rates, and the population size of each class of users. For one set of user travel routes, we observe that, depending upon problem parameters, there exist eight distinct solution structures. This suggests that as the number of queues increases, there may be a rapid explosion in the solution structures. This may make the problem of identifying and learning the correct structure computationally prohibitive. Our other broad contributions/observations are: 1) We find that similar to the single queue setting, the equilibrium profile of arriving users is unique for a wide set of parameters. However, interestingly, when user cost preferences across classes are identical up to a constant, there may be multiple equilibrium arrival profiles, all lying in a convex set that we identify. Although there are many arrival profiles in equilibrium in this case, they all have identical social cost. 2) In [11], the equilibrium profile is determined for the case where multiple classes of users with linear costs are arriving in a single queue. They find that different classes of users arrive in

*TIFR Mumbai, India, agniv.bandyopadhyay@tifr.res.in

†Ashoka University, Sonapat, India, sandeep.juneja@ashoka.edu.in

non overlapping and contiguous intervals. In our two-queue setting, we find that depending upon the problem parameters, in equilibrium, arrivals may come in non-overlapping and contiguous intervals, in overlapping intervals, or under certain parametric settings, a class of users may even arrive in disjoint intervals. Moreover, we show that whether the classes will arrive over overlapping sets or not is independent of the population sizes and is entirely decided by the queue service rates and user preferences.

Related literature: The arrival games to queues were first considered by [5]. The concert queueing game, the fluid setting, was introduced in [11]. The arrival game in a fluid network of bottleneck queues including tandem, Trellis, and general feedforward networks was considered in [10], where they characterized the equilibrium arrival profile in each of these topologies. Our work differs from the existing literature on queueing arrival games in the aspect that we consider a two queue network with heterogeneous arrival routes, linear costs, and in this setting we are able to characterize the equilibrium user arrival profiles in a closed form, and for a large class of parameters, show that these profiles are unique.

Outline: In Section 2 we provide background information on the arrival queueing game and describe the two-class, two-line, two-route networks that we consider. We emphasize on two heterogeneous routes networks 1) where the departures of the two classes are through different queues (Heterogeneous Departure System or HDS) and 2) where the arrivals enter at different queues (Heterogeneous Arrival System or HAS). We also describe the connection between non-fluid two queue networks and the fluid model towards the end of this section. Section 3 contains our main result about the existence of equilibrium arrival profile(s) for HDS and HAS, and also contains several interesting properties of the equilibrium. In particular, for HDS, we present the four different structures that the equilibrium can have depending on the parameters defining the instance and provide outlines of the arguments behind their occurrence. Similarly for HAS, we discover that the equilibrium can have eight possible structures. Furthermore, for HDS, we identify a parametric regime that has a unique equilibrium and another parametric regime that has multiple equilibria that form a convex set. We end with a brief conclusion. In this paper, we have confined our discussion only to presenting the interesting properties of the equilibrium. The details are presented in [2].

2. PRELIMINARIES

Fluid Model: We consider a fluid model having two classes of users. The size of each class $i = 1, 2$ is given by a positive quantity $\Lambda_i > 0$. In every class $i = 1, 2$ individual users are infinitesimal and the set of all users in class i is given by the points in the interval $[0, \Lambda_i]$. We define functions $F^{(i)} : \mathbb{R} \rightarrow [0, \Lambda_i]$ for $i = 1, 2$ such that $F^{(i)}(t)$ denotes the number of users of class i who arrive by time t . We call $F^{(i)}$ the arrival profile of users in class i . Therefore, each $F^{(i)}$ is non-decreasing and satisfies $F^{(i)}(-\infty) = 0$ and $F^{(i)}(+\infty) = \Lambda_i$. We consider $F^{(i)}$ that are right-continuous and doesn't contain any singularly continuous component. As a result, $F^{(1)}, F^{(2)}$ can be expressed as a sum of a non-decreasing absolutely continuous function and a non-decreasing discrete function (by [1, Proposition 4.5.1]). We call the pair $\mathbf{F} =$

$\{F^{(1)}, F^{(2)}\}$ as the joint arrival profile of the two classes.

In this paper, we consider a network comprising of two queues in tandem, both starting service at time $t = 0$. Users can choose to arrive before both queues start at time zero. Let μ_1 and μ_2 , respectively, denote the fixed deterministic service rates in the two queues after they start service. We consider instances where two classes of users get served by the network such that: 1) The two classes get served via distinct routes, 2) Every queue serves atleast one of the classes. There can be two instances which satisfy these two criteria: 1) Classes 1 and 2 both arrive at queue 1 but depart, respectively, from queues 1 and 2, and 2) Classes 1 and 2, respectively, arrive at queues 1 and 2, but depart from queue 2. We call the first instance the Heterogeneous Departure System or HDS (illustrated in Figure 2), and the second instance as the Heterogeneous Arrival System or HAS (illustrated in Figure 4).

Waiting and Departure Times: To specify the waiting and departure times in a system, first consider a single queue setting where $A(t)$ denotes the total mass of users of all classes that have arrived at the queue by time t . Let μ denote the service rate. We use the notations $W(t)$ and $\tau(t) = W(t) + t$, respectively, to denote the waiting and departure times of a user arriving at time t . Both $W(\cdot)$ and $\tau(\cdot)$ are well-defined as functions of the arrival profile $A(\cdot)$ (see [4, Theorem 6.5]) and in particular, $W(t) = A(t)/\mu - \max\{t, 0\} + I(t)$, where $I(t)$ is the amount of time the queue remains empty in the interval $[0, t]$. $I(\cdot)$ is also well defined as a function of $A(\cdot)$. We refer the reader to [2, Section 2.2] for detailed discussion about the dependence of $W(\cdot), \tau(\cdot)$ and $I(\cdot)$ on $A(\cdot)$.

We introduce the following notation: **1)** $A_i(t)$ be the total mass of users of both the classes who have arrived at queue $i = 1, 2$ till time t (Note that while $F^{(j)}$ denotes arrival profile corresponding to user class j , A_i denotes overall arrival profile to queue i), **2)** $W_i(t)$ be the waiting time suffered by a user arriving at queue i at time t , and $\tau_i(t) = t + W_i(t)$ be the departure time of the same user from queue i , and **3)** $W_{\mathbf{F}}^{(j)}(t)$ and $\tau_{\mathbf{F}}^{(j)}(t)$ be the waiting and departure times from the network suffered by a class j user arriving at time t for $j \in \{1, 2\}$. Explicit dependence on $\mathbf{F}$ in this notation is useful to our analysis later.

For both the queues $i = 1, 2$ upon defining the arrival profile $A_i(\cdot)$, $W_i(\cdot)$ and $\tau_i(\cdot)$ are well-defined. Now we specify the waiting and departure times of both the queues in HDS and HAS as functions of time, under the assumption that the joint arrival profile $\mathbf{F} = \{F^{(1)}, F^{(2)}\}$ is absolutely continuous (we later argue by Lemma 2.1 that considering absolutely continuous joint arrival profiles are sufficient for identifying equilibrium behavior): **HDS:** Arrival profiles at individual queues are $A_1(t) = F^{(1)}(t) + F^{(2)}(t)$ and $A_2(t) = F^{(2)}(\tau_1^{-1}(t))$, where $\tau_1^{-1}(t) = \sup\{s \mid \tau_1(s) \leq t\}$. Both $A_1(\cdot)$ and $A_2(\cdot)$ are absolutely continuous. With this, $W_{\mathbf{F}}^{(1)}(t) = W_1(t)$, $\tau_{\mathbf{F}}^{(1)}(t) = \tau_1(t)$, $W_{\mathbf{F}}^{(2)}(t) = W_1(t) + W_2(\tau_1(t))$, and $\tau_{\mathbf{F}}^{(2)}(t) = \tau_1(t) + W_2(\tau_1(t))$. **HAS:** Arrival profile at individual queues are $A_1(t) = F^{(1)}(t)$ and $A_2(t) = F^{(1)}(\tau_1^{-1}(t)) + F^{(2)}(t)$ where $\tau_1^{-1}(t) = \sup\{s \mid \tau_1(s) \leq t\}$. Both $A_1(\cdot)$ and $A_2(\cdot)$ are absolutely continuous. With this, $W_{\mathbf{F}}^{(1)}(t) = W_1(t) + W_2(\tau_1(t))$, $\tau_{\mathbf{F}}^{(1)}(t) = \tau_1(t) + W_2(\tau_1(t))$, $W_{\mathbf{F}}^{(2)}(t) = W_2(t)$, and $\tau_{\mathbf{F}}^{(2)}(t) = \tau_2(t)$.

Solution Concept: We assume that every user in group

j ($j \in \{1, 2\}$) has a cost function linear in her waiting and departure times from the network given by: $C_{\mathbf{F}}^{(j)}(t) = \alpha_j \cdot W_{\mathbf{F}}^{(j)}(t) + \beta_j \cdot \tau_{\mathbf{F}}^{(j)}(t)$, where α_j and β_j are positive constants quantifying the cost suffered by a class j user for unit waiting time and delay in departure. Given an arrival profile $t \mapsto B(t)$ such that $B(+\infty) < \infty$, we use the notation $\mathcal{S}(B)$ to denote the smallest closed set having a B -measure equal to $B(+\infty)$, which is also called the *support* of B .

Definition 2.1 (Equilibrium Arrival Profile (EAP)). *The joint arrival profile $\mathbf{F}^* = \{F^{(1),*}, F^{(2),*}\}$ is an Equilibrium Arrival Profile of this game if for both the groups $j \in \{1, 2\}$: $t \in \mathcal{S}(F^{(j),*})$ and $\bar{t} \in \mathbb{R}$ implies $C_{\mathbf{F}^*}^{(j)}(t) \leq C_{\mathbf{F}^*}^{(j)}(\bar{t})$. In particular, the arrival profile is iso-cost along its support.*

Note that, the EAP doesn't change upon normalizing the cost function of both the classes $j = 1, 2$ by multiplying $1/(\alpha_j + \beta_j)$. For simplicity, and without loss of generality, we assume both the classes $j = 1, 2$ have their normalized cost function, which are: $C_{\mathbf{F}}^{(j)}(t) = \gamma_j W_{\mathbf{F}}^{(j)}(t) + (1 - \gamma_j) \tau_{\mathbf{F}}^{(j)}(t)$ where $\gamma_j = \frac{\alpha_j}{\alpha_j + \beta_j}$ quantifies the preference of every class j user. A value of γ_j close to 1 indicates users in group j prefer late departure compared to waiting a long time in the network and γ_j close to 0 implies the opposite. So, we use γ_j to quantify the *cost preference* of every group j user.

Remark 1. EAP captures the aggregate equilibrium behavior of the group. We can equivalently define Nash equilibrium at individual level where under it no individual has unilateral incentive to deviate. As is well known and discussed in more detail in [11], the two concepts are equivalent.

Lemma 2.1 helps us restrict our search of EAP only to arrival profiles $F^{(1)}$ and $F^{(2)}$ which are *absolutely continuous*.

Lemma 2.1. *In every EAP, $\mathbf{F} = \{F^{(1)}, F^{(2)}\}$ of the HDS and HAS, the arrival profiles $F^{(1)}$ and $F^{(2)}$ are absolutely continuous.*

Proof of the above lemma is similar to proof of statement (ii) of Lemma 1 in [11]. On assuming contradiction, if any of the arrival profiles have a jump, any user arriving in that jump will be strictly better off arriving slightly early and as a result the arrival profile cannot be an EAP.

[11] show that when two classes of users having cost preferences $\gamma^{(1)}$ and $\gamma^{(2)}$ arrive at a single queue with service rate μ , the EAP has a simple structure. The class with smaller $\gamma^{(j)}$ comes first at arrival rate $\mu \cdot \min\{\gamma^{(1)}, \gamma^{(2)}\}$ over an interval, while the next class arrives at a contiguous but non-overlapping interval, at rate $\mu \cdot \max\{\gamma^{(1)}, \gamma^{(2)}\}$. Figure 1 illustrates this EAP and the resulting queue length with the assumption $\gamma^{(1)} < \gamma^{(2)}$ and is useful to contrast with the various EAP structures that we find for HDS and HAS in Section 3.

Relation between fluid model and non-fluid queueing networks: [12] considered the non-fluid setting of an arrival game with a single class and single queue, constructing a sequence of non-fluid game instances converging to the corresponding fluid model with an appropriate scaling. [12] proved existence of a unique MNE of every instance in this sequence and further proved that the aggregate arrival profile of the MNE of these instances converge to the EAP of the fluid model. In [2] Section 2.4, we provide the details about how such a sequence of non-fluid instances can

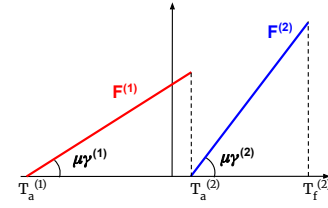


Figure 1: EAP structure when two classes of users with cost preferences $\gamma^{(1)}$ and $\gamma^{(2)}$ (assuming $\gamma^{(1)} < \gamma^{(2)}$) are arriving at a queue of capacity μ . The support boundaries are $T_f^{(2)} = \frac{\Lambda^{(1)} + \Lambda^{(2)}}{\mu}$, $T_a^{(2)} = \frac{\Lambda^{(1)}}{\mu} - \left(\frac{1}{\gamma^{(2)}} - 1\right) \frac{\Lambda^{(2)}}{\mu}$ and $T_a^{(1)} = -\left(\frac{1}{\gamma^{(1)}} - 1\right) \frac{\Lambda^{(1)}}{\mu} - \left(\frac{1}{\gamma^{(2)}} - 1\right) \frac{\Lambda^{(2)}}{\mu}$.

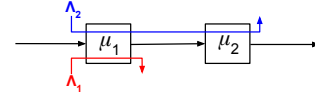


Figure 2: Heterogeneous Departure System

be constructed for HDS and HAS, so that they converge to the respective fluid models. Extending the methods of [12] to HDS and HAS is technically challenging and beyond the scope of this paper. However, we conjecture that, with appropriate scaling a similar convergence result of the MNE of the non-fluid queueing networks to the EAP of the fluid model can be proved.

3. MAIN RESULTS

Heterogeneous Departure System or HDS: We now consider the situation where the two classes arrive at the first queue and depart from different queues as illustrated in Figure 2.

Note that class 2 users, after getting served at queue 1, arrive at queue 2 at a maximum rate of μ_1 . As a result, if $\mu_1 \leq \mu_2$, queue 2 remains empty and the instance becomes equivalent to a single queue instance where the two classes get served only by a queue with service rate μ_1 . According to [11, Theorem 2], the EAP of the case $\mu_1 \leq \mu_2$ has the two classes arriving over disjoint intervals ordered in the increasing order of their preferences (for eg., the EAP will be the one in Figure 1 with $\mu = \mu_1$ when $\gamma^{(1)} \leq \gamma^{(2)}$). Therefore, we restrict our discussion only to the case $\mu_1 > \mu_2$.

Theorem 3.1 is the main result of this section.

Theorem 3.1. *The following statements are true about the EAP of HDS with $\mu_1 > \mu_2$: 1) If $\gamma^{(1)} \neq \gamma^{(2)}$, there is a unique EAP. Furthermore, depending on the mass of the two classes $\Lambda^{(1)}$ and $\Lambda^{(2)}$, the EAP can have four possible structures. 2) If $\gamma^{(1)} = \gamma^{(2)} = \gamma$, there is a unique EAP if $\Lambda^{(1)} < \left(\frac{\mu_1}{\mu_2} - 1\right) \Lambda^{(2)}$. Otherwise, there can be multiple EAPs and the set of all these EAPs is convex.*

Due to space constraints we hide the detailed closed forms of the EAP in the statement of Theorem 3.1. We refer the reader to [2, Theorems 3.2 and 3.3] for the closed forms of the EAPs, respectively, under the cases $\gamma^{(1)} \neq \gamma^{(2)}$ and $\gamma^{(1)} = \gamma^{(2)} = \gamma$ in Theorem 3.1. We restrict our discussion only to several interesting properties of the EAPs and provide illustrations of them whenever relevant.

Threshold behavior: In the beginning of the section, we discussed that the two classes arrive over disjoint intervals when $\mu_1 \leq \mu_2$ and $\gamma^{(1)} \neq \gamma^{(2)}$. A natural question is, whether there is a maximum value or a threshold for μ_1 , such that the two classes with unequal preferences (*i.e.*, $\gamma^{(1)} \neq \gamma^{(2)}$) arrive over disjoint intervals in EAP if and only if μ_1 is below that threshold. Lemma 3.1 provides us that threshold.

Lemma 3.1. *If $\gamma^{(1)} \neq \gamma^{(2)}$, then the two classes always arrive over contiguous intervals, *i.e.*, $\mathcal{S}(F^{(1),*})$, $\mathcal{S}(F^{(2),*})$ and $\mathcal{S}(F^{(1),*}) \cup \mathcal{S}(F^{(2),*})$ are intervals. Furthermore, $\mathcal{S}(F^{(1),*})$ and $\mathcal{S}(F^{(2),*})$ intervals overlap if and only if $\mu_1 > \mu_2 \cdot \max\left\{1, \frac{\gamma^{(2)}}{\gamma^{(1)}}\right\}$.*

[2] Lemma 3.1 and 3.2] contains the detailed version of Lemma 3.1. Here we provide a brief sketch of necessity in Lemma 3.1. Proving that the two classes arrive over contiguous intervals when $\gamma^{(1)} \neq \gamma^{(2)}$ irrespective of μ_1 and μ_2 , is done via contradiction and we skip that argument due to space constraint. If the two intervals overlap, we must have $\mu_1 > \mu_2$, otherwise, by [11, Theorem 2], the two classes arrive over disjoint intervals. Now, by Definition 2.1 of the EAP, $C_{\mathbf{F}^*}^{(1)}(\cdot)$ and $C_{\mathbf{F}^*}^{(2)}(\cdot)$ must, respectively, be constant within $\mathcal{S}(F^{(1),*})$ and $\mathcal{S}(F^{(2),*})$. Thus the derivative of the costs of the two classes in their respective arrival intervals must be zero. For the previous statement to hold, we can prove that: 1) $(C_{\mathbf{F}^*}^{(1)})'(\cdot) = 0$ implies the two classes must arrive at a joint rate of $\mu_1\gamma^{(1)}$ over $\mathcal{S}(F^{(1),*})$, and 2) $(C_{\mathbf{F}^*}^{(2)})'(\cdot) = 0$ implies class 2 users must arrive at rate $\mu_2\gamma^{(2)}$ over $\mathcal{S}(F^{(2),*})$. Therefore, if the two intervals overlap, then at the overlap, the combined rate of arrival of the two classes must be strictly larger than the rate of arrival of class 2 users, which implies, $\mu_1\gamma^{(1)} > \mu_2\gamma^{(2)}$, *i.e.*, $\mu_1 > \mu_2 \cdot \frac{\gamma^{(2)}}{\gamma^{(1)}}$. Combining this with our initial observation $\mu_1 > \mu_2$, we obtain the overall threshold $\mu_2 \cdot \max\left\{1, \frac{\gamma^{(2)}}{\gamma^{(1)}}\right\}$, such that, μ_1 must be strictly higher than the threshold for the two classes to arrive over overlapping intervals.

Ordering of classes according to preferences: By [11, Theorem 2], for single queue instances with multiple classes of users having distinct preferences (*i.e.* distinct γ 's), the EAP has all the classes arrive over disjoint intervals which are ordered according to the increasing order of their preferences. In HDS as well, we can see similar behavior in the EAP when $\gamma^{(1)} \neq \gamma^{(2)}$. We see this behavior in two separate cases: 1) $\mu_2 < \mu_1 \leq \mu_2 \cdot \max\left\{1, \frac{\gamma^{(2)}}{\gamma^{(1)}}\right\}$ and 2) $\mu_1 > \mu_2 \cdot \max\left\{1, \frac{\gamma^{(2)}}{\gamma^{(1)}}\right\}$ (recall that we are only considering HDS with $\mu_1 > \mu_2$).

In case 1, by Lemma 3.1, the two classes must arrive over disjoint intervals. Furthermore, for case 1 to happen, we must have $\gamma^{(2)} > \gamma^{(1)}$. The illustrative EAP for case 1 is in Figure 3 case a. We can observe that, class 1's arrival precedes class 2's arrival which reflects $\gamma^{(1)} < \gamma^{(2)}$. The argument behind this structure relies on proof via contradiction and [2, Lemma 3.6] provides the details. We argue that, if the two disjoint intervals are ordered otherwise, *i.e.*, class 2 users arrive before class 1 users, then: every class 2 user can strictly reduce her cost by arriving at the time when class 1 users finish arrival, which contradicts Definition 2.1 of the EAP.

In case 2, there can be two situations depending on whether $\Lambda^{(2)}/\Lambda^{(1)}$ is above or below the threshold $c(\gamma^{(1)}, \gamma^{(2)}) = \min\left\{\frac{1-\gamma^{(2)}}{1-\gamma^{(1)}}\left(\frac{\mu_1\gamma^{(1)}}{\mu_2\gamma^{(2)}} - 1\right), \frac{\mu_1}{\mu_2} - 1\right\}$ (we use c as a shorthand to denote $c(\gamma^{(1)}, \gamma^{(2)})$ for notational simplicity). If $\Lambda^{(2)} \geq c\Lambda^{(1)}$, class 1 users arrive within the support of class 2 users in the EAP (illustrative EAP is in case b of Figure 3). If $\Lambda^{(2)} < c\Lambda^{(1)}$, then arrival intervals of the two classes are ordered according to the increasing order of their preferences, even though they overlap. This can be observed in the illustrative EAPs of cases c and d in Figure 3, which, respectively, represent the situations $\gamma^{(1)} < \gamma^{(2)}$, $\Lambda^{(2)} < c\Lambda^{(1)}$ and $\gamma^{(1)} > \gamma^{(2)}$, $\Lambda^{(2)} < c\Lambda^{(1)}$. Now we provide a brief overview of the argument behind this ordering of arrivals of the two classes. In [2, Lemma 3.6], we prove that, under case 2: 1) if $\gamma^{(1)} < \gamma^{(2)}$, class 2 users finish arrival after class 1 users finish and 2) if $\gamma^{(1)} > \gamma^{(2)}$, class 2 users start arrival before class 1 users start. Thus, the ordering of one of the ends of the supports are decided by the relative ordering of the preferences. According to [2, Lemmas 3.8, 3.9], the ordering of the other end of the supports are decided according to the relative ordering between $\Lambda^{(2)}$ and $c\Lambda^{(1)}$, favoring the class having relatively larger mass. To be more specific: 1) if $\Lambda^{(2)} \geq c\Lambda^{(1)}$, the other ends of the support are ordered such that class 1's support lies inside class 2's support (case b in Figure 3), and 2) if $\Lambda^{(2)} < c\Lambda^{(1)}$, the two classes arrive in the increasing order of their preferences and their supports overlap without class 1's support being fully inside class 2's support (cases c and d in Figure 3). Therefore, as class 1's mass $\Lambda^{(1)}$ increases and exceeds $\Lambda^{(2)}/c$, the arrival of class 1 users get spilled from the left or right end of class 2's the support depending, respectively, on whether $\gamma^{(2)} > \gamma^{(1)}$ or $\gamma^{(2)} < \gamma^{(1)}$.

EAPs forming convex set under equal preferences:

When $\gamma_1 = \gamma_2 = \gamma$, we have $c(\gamma, \gamma) = \frac{\mu_1}{\mu_2} - 1$. Under the case $\Lambda^{(2)} \geq (\mu_1/\mu_2 - 1) \cdot \Lambda^{(1)}$, there is a unique EAP which can be obtained by taking $\gamma^{(1)} = \gamma^{(2)} = \gamma$ for case b in Figure 3. In the other case $\Lambda^{(2)} < (\mu_1/\mu_2 - 1) \cdot \Lambda^{(1)}$, two EAPs are the two joint arrival profiles in cases c and d of Figure 3, respectively, after taking limits $\gamma^{(1)} = \gamma, \gamma^{(2)} \rightarrow \gamma +$ and $\gamma^{(1)} \rightarrow \gamma +, \gamma^{(2)} = \gamma$. As we state in Theorem 3.1 and provide more details in [2, Theorem 3.3], there can be multiple EAPs under the case $\Lambda^{(2)} < (\mu_1/\mu_2 - 1) \cdot \Lambda^{(1)}$ and they form a convex set specified by the properties: 1) $\mathcal{S}(F^{(1)}) = [T_a, T_f]$, $\mathcal{S}(F^{(2)}) \subseteq [T_a, T_f]$, for $i = 1, 2$ $F^{(i)}(T_f) = \Lambda^{(i)}$, 2) $(F^{(1)})'(t) + (F^{(2)})'(t) = \mu_1\gamma$, and 3) $(F^{(2)})'(t) \leq \mu_2\gamma$ for every $t \in [T_a, T_f]$, where $T_a = -\left(\frac{1}{\gamma} - 1\right) \frac{\Lambda^{(1)} + \Lambda^{(2)}}{\mu_1}$ and $T_f = \frac{\Lambda^{(1)} + \Lambda^{(2)}}{\mu_1}$. This set contains both the EAPs obtained earlier by taking limits over cases c and d in Figure 3.

Heterogeneous Arrival System or HAS: We consider the instance where the two classes arrive at different queues and depart from the same queue, as illustrated in Figure 4. We restrict our discussion in this section to HAS with unequal preferences, *i.e.*, $\gamma^{(1)} \neq \gamma^{(2)}$ and provide the detailed results for equal preferences ($\gamma^{(1)} = \gamma^{(2)}$) in [2, Section 4.2]. We first describe below the main result of this section.

Theorem 3.2. *If $\gamma^{(1)} \neq \gamma^{(2)}$, HAS has a unique EAP. Furthermore, depending on the instance parameters $\Lambda^{(i)}, \mu_i, \gamma^{(i)}$ for $i = 1, 2$, the EAP can have eight different structures.*

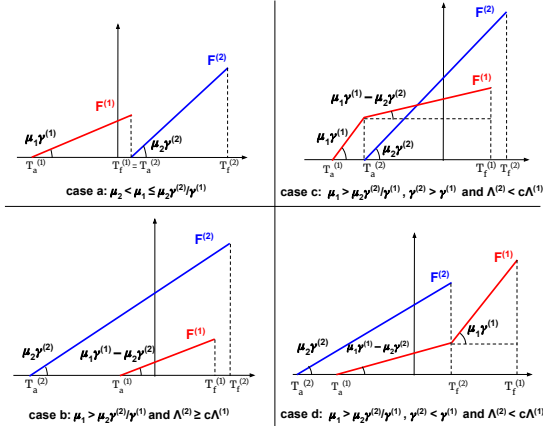


Figure 3: Illustrative EAPs of HDS when $\gamma^{(1)} \neq \gamma^{(2)}$ and $\mu_1 > \mu_2$

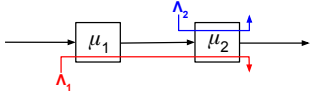


Figure 4: Heterogeneous Arrival System

For detailed closed forms of the EAP, see [2, Theorems 4.2 and 4.3]. Much of the proof arguments for HAS resemble those for HDS except for the larger no. of regimes in HAS compared to HDS. We restrict our presentation to two interesting structural properties of the EAP in HAS.

Threshold behavior: Recall that in HDS, the two classes share queue 1 in their path and Lemma 3.1 provides a threshold to μ_1 such that the two classes share queue 1 in EAP, if and only if, μ_1 is strictly above that threshold. Similarly, in HAS, the two classes share queue 2. A natural question is whether we have a similar threshold for μ_2 like the one for μ_1 in HDS. Lemma 3.2 below provides such a threshold.

Lemma 3.2. *If $\gamma^{(1)} \neq \gamma^{(2)}$, in EAP, queue 2 serves the two classes over disjoint sets of time if and only if $\mu_1 \geq \mu_2 \gamma^{(2)}$.*

Proof of Lemma 3.2 is similar to that of Lemma 3.1 and for a detailed proof see [2, Lemma 4.1]. We first show that for $C_{F^*}^{(2)}(\cdot)$ to remain constant in $\mathcal{S}(F^{(2),*})$, the combined rate of arrival of class 2 users and class 1 users coming from queue 1, must be $\mu_2 \gamma^{(2)}$. Secondly, we argue that queue 1 must remain busy, i.e., length of queue 1 must be positive when users of the two classes share queue 2. As a result, class 1 users must arrive at queue 2 from queue 1 at rate μ_1 . Since the combined arrival rate of the two classes in queue 2 must be larger than the arrival rate of the class 1 users from queue 1 to queue 2, we must have $\mu_1 < \mu_2 \gamma^{(2)}$ for the two classes to share queue 2 in equilibrium. Proving sufficiency of this condition requires proving properties related to the structure of $\mathcal{S}(F^{(1),*})$ and $\mathcal{S}(F^{(2),*})$.

Disconnected support of EAP: Under the regime $\mu_1 \geq \mu_2 \gamma^{(2)} > \mu_2 \gamma^{(1)}$ and $\Lambda^{(2)} > (1/\gamma^{(2)} - 1)\Lambda^{(1)}$ we prove that class 2 users arrive over the union of two disjoint intervals in the EAP. We provide the illustrative EAP of this regime in Figure 5 case c. This is unlike HDS, where both classes arrive over contiguous intervals in EAP. We provide a brief outline of the arguments here and the details are

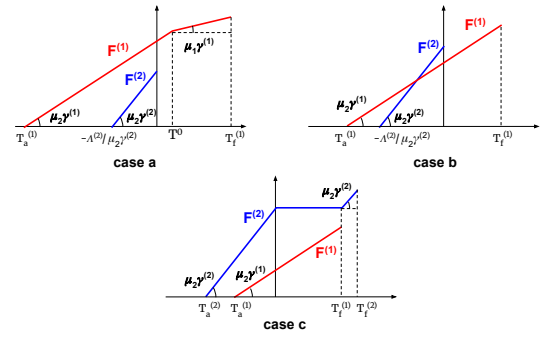


Figure 5: EAP of HAS under the regime $\mu_1 \geq \mu_2 \gamma^{(2)} > \mu_2 \gamma^{(1)}$. Cases a, b and c, respectively, represent $(\mu_2/\mu_1 - 1)\Lambda^{(1)} > \Lambda^{(2)}$, $(1/\gamma^{(2)} - 1)\Lambda^{(1)} \geq \Lambda^{(2)} \geq (\mu_2/\mu_1 - 1)\Lambda^{(1)}$, and $\Lambda^{(2)} > (1/\gamma^{(2)} - 1)\Lambda^{(1)}$.

in [2, Remark 8]. Note that the regime satisfies the condition $\mu_1 \geq \mu_2 \gamma^{(2)}$, which by Lemma 3.2 implies that the two classes cannot share queue 2 in EAP. We first argue that if $\mu_1 \geq \mu_2 \gamma^{(2)} > \mu_2 \gamma^{(1)}$, then, class 1 and 2 users must queue up, respectively, at queues 1 and 2 before time zero. As a result, from time zero onward, class 1 users arrive at queue 2 from queue 1 at rate μ_1 until queue 1 expires. We further prove that queue 1 must be empty by the time the last class 1 user arrives at the system. Hence, class 1 users arrive at queue 2 from queue 1 starting from time zero until the end of class 1's arrival interval. By Lemma 3.2, class 2 users cannot arrive at queue 2 during this period. This leaves class 2 users with two options: 1) The entire population of class 2 arrives before time zero, and 2) The class 2 population divides itself into two parts, one queueing up before time zero and the other arriving after class 1 arrival finishes. We observe this phenomenon in Figure 5 where the entire class 2 population arrives before time zero as long as $\Lambda^{(2)} \leq (1/\gamma^{(2)} - 1)\Lambda^{(1)}$. As $\Lambda^{(2)}$ exceeds $(1/\gamma^{(2)} - 1)\Lambda^{(1)}$, a fraction of the class 2 population arrives after the class 1 arrival finishes, and thus breaking class 2's support into the union of two disjoint intervals.

Conclusion: We studied a simple two-queue, two-class network with a linear cost structure, where we analyzed in detail two heterogeneous routing configurations: (1) where the user classes arrived at the same queue but departed in different queues, and (2) where the user classes arrived at different queues but departed from the same queue. We discovered non-trivial user behaviour not apparent in a single queue or in networks studied thus far in the literature. We found that although for most sets of parameters, the equilibrium profile is unique, there exist settings where the collection of equilibrium profiles is not unique but a convex set. While in a single queue, multiclass customers arrive in contiguous, nonoverlapping intervals, in our two queue setting there are regions where, in equilibrium, different class arrival times may overlap. Further, there exist regions, where a single class customer may arrive in disjoint intervals. Broadly, the paper suggests that even for more complex networks one may expect a unique equilibrium profile for most parameter settings. It also suggests that the number of different solution structures may explode with the size of the network, so learning the structure in any case may be a difficult task.

4. REFERENCES

- [1] Krishna B Athreya and Soumendra N Lahiri. *Measure theory and probability theory*. Springer, 2006.
- [2] Agniv Bandyopadhyay and Sandeep Juneja. Game of arrivals at a two queue network with heterogeneous customer routes. *arXiv preprint arXiv:2310.18149*, 2023.
- [3] Jesper Breinbjerg. Equilibrium arrival times to queues with general service times and non-linear utility functions. *European Journal of Operational Research*, 261(2):595–605, 2017.
- [4] Hong Chen and David D Yao. *Fundamentals of queueing networks: Performance, asymptotics, and optimization*, volume 4. Springer, 2001.
- [5] Amihai Glazer and Refael Hassin. $M/M/1$: On the equilibrium distribution of customer arrivals. *European Journal of Operational Research*, 13(2):146–150, 1983.
- [6] Refael Hassin. *Rational Queueing*. Chapman and Hall/CRC, 2016.
- [7] Refael Hassin and Moshe Haviv. *To queue or not to queue: Equilibrium behavior in queueing systems*, volume 59. Springer Science & Business Media, 2003.
- [8] Moshe Haviv and Binyamin Oz. A busy period approach to some queueing games. *Queueing Systems*, 97:261–277, 2021.
- [9] Moshe Haviv and Liron Ravner. A survey of queueing systems with strategic timing of arrivals. *Queueing Systems*, 99:163–198, 2021.
- [10] Harsha Honnappa and Rahul Jain. Strategic arrivals into queueing networks: The network concert queueing game. *Operations Research*, 63(1):247–259, 2015.
- [11] Rahul Jain, Sandeep Juneja, and Nahum Shimkin. The concert queueing game: to wait or to be late. *Discrete Event Dynamic Systems*, 21:103–138, 2011.
- [12] Sandeep Juneja and Nahum Shimkin. The concert queueing game: strategic arrivals with waiting and tardiness costs. *Queueing Systems*, 74:369–402, 2013.
- [13] Robin Lindsey. Existence, uniqueness, and trip cost function properties of user equilibrium in the bottleneck model with multiple user classes. *Transportation Science*, 38:293–314, 2004.

Model-Based Learning of Whittle Index

Joël Charles-Rebuffé
Univ. Grenoble Alpes, Inria,
CNRS, Grenoble INP, LIG,
38000 Grenoble, France
joel.charles-
rebuffe@univ-grenoble-
alpes.fr

Nicolas Gast
Univ. Grenoble Alpes, Inria,
CNRS, Grenoble INP, LIG,
38000 Grenoble, France
nicolas.gast@inria.fr

Bruno Gaujal
Univ. Grenoble Alpes, Inria,
CNRS, Grenoble INP, LIG,
38000 Grenoble, France
bruno.gaujal@inria.fr

Restless multi-armed bandits (**RMABs**) can model several resource allocation problems. To cite only a few applications, RMABs are used to solve optimization problems related to stochastic scheduling in a queue [1],[2], opportunistic scheduling [3],[4], machine maintenance [5], health-care [6] or recommendation systems [7].

A RMAB is composed of N independent arms, each of them being a Markov process that evolves in discrete time. At each time instant the decision maker chooses a set of arms to be activated, with the goal of maximizing rewards cumulated across all arms, while not being able to activate more than a certain number of arms at a time. Introduced by Gittins and Jones [8], the Gittins index policy and its further extension the Whittle index policy consists of associating a number – called *index* – to each state of each arm, and then to activate the arms with the highest indices in priority. Indices can be computed for all arms that satisfy a broad condition called *indexability*. If this and other index based policies are not optimal in general, they have been shown to be asymptotically optimal when the number of arms goes to infinity, under indexability and a technical condition that is often verified in practice [9],[10]. Moreover, they work extremely well in practice [11].

The computation of the indices of the states of a given arm can be done independently of the other arms and only requires to have access to the arm parameter, i.e. its transition kernel P and reward function r . There exist in the literature very efficient algorithms and implementations that compute the indices of an arm given (P, r) [12].

There are many situations, though, where the parameters (P, r) are unknown to the decision maker who can only estimate them through sampling. This is our focus: we consider a scenario where an agent interacts with an environment that can simulate an arm. The agent is faced with the learning problem of how to use these samples to estimate the indices as precisely as possible. The problem of learning Whittle indices has become popular in the recent years, and a number of algorithms have been developed, most of them relying on the popular model-free algorithm framework called **Q-learning**. One of the first papers to develop such a reinforcement learning is [13] which focuses on a particular case of monotonic bandit model. Their approach has then been extended to (unstructured) discrete state-space in [14],[15]. The convergence of a two timescales approach to

learn indices is done in [16], where the **Q-Whittle** algorithm is introduced to learn Whittle indices in the average reward setting. This algorithm has been adapted in [17] to improve convergence speed. There also exist variants of **Q-Whittle** algorithms that use function approximation, for instance linear function approximation in [18], but also neural network approximations in [19],[20],[21].

One important common denominator to all of these papers is that they are based on **Q-learning** and most of them use a two timescales approach: the learning algorithm is composed of a primary layer that learns the Q-value of a λ -penalized MDP and in parallel adapts the penalty λ at a slow rate in order to find the penalty corresponding to the indices. This approach is called model-free because the agent does not try to estimate a model of the arm but directly learns the Q-values and the Whittle indices. The advantage of such an approach is that it is quite flexible and can easily be adapted to function approximation like deep neural networks (although without theoretical guarantees in general). Yet, model-free approaches make a poor use of samples: they require very long runs of simulations in order to converge to an acceptable value. In addition, even though most **Q-learning** algorithms offer convergence guarantees under some assumptions, it is very difficult to obtain performance guarantees, i.e. sufficient conditions on the number of samples needed to get an acceptable approximation.

In this work, we propose a radically different approach and develop a new model-based approach to learn Whittle index. Our approach consists in using a simulator to sample an arm and to build estimates of the transition matrices and rewards $(\tilde{P}, \tilde{r})$. The **GKG** algorithm of [12] is the fastest known algorithm to compute Whittle indices today. Hence, it would be tempting to simply call **GKG** on $(\tilde{P}, \tilde{r})$ and return the index values of this model. Yet, we show that this *naïve* method does not work in general because the set of indexable arms is not an open-set: even if the true model (P, r) is indexable, there is no guarantee that the estimated model $(\tilde{P}, \tilde{r})$ is indexable, regardless of how close $(\tilde{P}, \tilde{r})$ is from (P, r) . This is why we introduce a generalization of **GKG**, called **Extended GKG** or **EGGK**, that takes as input any parameter value $(\tilde{P}, \tilde{r})$ and output a vector of real values **EGGK** $(\tilde{P}, \tilde{r})$. We show that if the original problem (P, r) is indexable, then **EGGK** $(\tilde{P}, \tilde{r})$ converges to the true indices **GKG** (P, r) when $(\tilde{P}, \tilde{r})$ converges to (P, r) , even when the parameters $(\tilde{P}, \tilde{r})$ are never indexable. This allows us to derive an algorithm, that we call **BLINQ** (**BLINQ Learning Is Not Q-learning**). It uses a simulator to collect samples

of (P, r) to produce $(\tilde{P}, \tilde{r})$ and sporadically calls **EGGK**. We prove that if (P, r) is indexable, then **BLINQ** converges to the true indices as the number of samples goes to infinity. We present numerical experiments that show that **BLINQ** largely outperforms **Q-learning** based approaches for any reasonable number of samples.

Summary of contributions

- Our first contribution is to show that for some indexable arms there exists non indexable arms that are arbitrarily close. As a result, we extend the **GGK** algorithm by introducing **EGGK**, which works on all arms (indexable or not) and returns values that coincide with Whittle indices when the input is indexable.
- Our second contribution is to show that when all Whittle indices are distinct for an indexable arm, then it has an indexable neighborhood.
- Our third and main contribution is the introduction and detailed analysis of **BLINQ**: a learning framework that uses **EGGK**. We provide a detailed analysis of its performance, that is, its rate of convergence as a function of the time and MDP parameters.
- As a side product, our fourth contribution is to provide a bound on the Whittle indices of an indexable arm, in terms of its parameters.

1. REFERENCES

- [1] IM Verloop. Asymptotically optimal priority policies for indexable and nonindexable restless bandits. *Annals of applied probability: an official journal of the Institute of Mathematical Statistics*, 26(4):1947–1995, 2016.
- [2] P. S. Ansell, K. D. Glazebrook, J. Niño-Mora, and M. O’Keeffe. Whittle’s index policy for a multi-class queueing system with convex holding costs. *Mathematical Methods of Operations Research*, 57(1):21–39, April 2003.
- [3] Vivek S Borkar, Gaurav S Kasbekar, Sarath Pattathil, and Priyesh Y Shetty. Opportunistic scheduling as restless bandits. *IEEE Transactions on Control of Network Systems*, 5(4):1952–1961, 2017.
- [4] Haoyue Tang, Jintao Wang, Linqi Song, and Jian Song. Minimizing age of information with power constraints: Multi-user opportunistic scheduling in multi-state time-varying channels. *IEEE Journal on Selected Areas in Communications*, 38(5):854–868, 2020.
- [5] Kevin D Glazebrook, HM Mitchell, and PS Ansell. Index policies for the maintenance of a collection of machines by a set of repairmen. *European Journal of Operational Research*, 165(1):267–284, 2005.
- [6] Arpita Biswas, Gaurav Aggarwal, Pradeep Varakantham, and Milind Tambe. Learn to intervene: An adaptive learning policy for restless bandits in application to preventive healthcare. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4039–4046. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Main Track.
- [7] Siwei Wang, Longbo Huang, and John Lui. Restless-ucb, an efficient and low-complexity algorithm for online restless bandits. *Advances in Neural Information Processing Systems*, 33:11878–11889, 2020.
- [8] John C Gittins and David M Jones. A dynamic allocation index for the discounted multiarmed bandit problem. *Biometrika*, 66(3):561–565, 1979.
- [9] Richard R. Weber and Gideon Weiss. On an index policy for restless bandits. *Journal of Applied Probability*, 27(3):637–648, 1990.
- [10] Nicolas Gast, Bruno Gaujal, and Chen Yan. Exponential asymptotic optimality of whittle index policy. *Queueing Systems*, 104(1):107–150, 2023.
- [11] José Niño-Mora. Markovian restless bandits and index policies: A review. *Mathematics*, 11(7):1639, 2023.
- [12] Nicolas Gast, Bruno Gaujal, and Kimang Khun. Testing indexability and computing whittle and gittins index in subcubic time. *Mathematical Methods of Operations Research*, 97(3):391–436, 2023.
- [13] Vivek S Borkar and Karan Chadha. A reinforcement learning algorithm for restless bandits. In *2018 Indian Control Conference (ICC)*, pages 89–94. IEEE, 2018.
- [14] Jing Fu, Yoni Nazarathy, Sarat Moka, and Peter G Taylor. Towards q-learning the whittle index for restless bandits. In *2019 Australian & New Zealand Control Conference (ANZCC)*, pages 249–254. IEEE, 2019.
- [15] Lachlan J Gibson, Peter Jacko, and Yoni Nazarathy. A novel implementation of q-learning for the whittle index. In *EAI international conference on performance evaluation methodologies and tools*, pages 154–170. Springer, 2021.
- [16] Konstantin E Avrachenkov and Vivek S Borkar. Whittle index based q-learning for restless bandits with average reward. *Automatica*, 139:110186, 2022.
- [17] Francisco Robledo, Vivek Borkar, Urtzi Ayesta, and Konstantin Avrachenkov. Qwi: Q-learning with whittle index. *ACM SIGMETRICS Performance Evaluation Review*, 49(2):47–50, 2022.
- [18] Guojun Xiong, Shufan Wang, Jian Li, and Rahul Singh. Whittle index-based q-learning for wireless edge caching with linear function approximation. *IEEE/ACM Transactions on Networking*, 2024.
- [19] Francisco Robledo, Vivek S Borkar, Urtzi Ayesta, and Konstantin Avrachenkov. Tabular and Deep Learning of Whittle Index. In *EWRL 2022 - 15th European Workshop of Reinforcement Learning*, Milan, Italy, September 2022.
- [20] Harshit Dhankar, Kshitij Mishra, and Tejas Bodas. Tabular and deep reinforcement learning for gittins index, 2024.
- [21] Yuhang Hao, Zengfu Wang, Jing Fu, and Quan Pan. A neural network-based whittle index policy for beam resource allocation in multitarget tracking. *IEEE Sensors Journal*, 2024.

On the Performance of Dynamic Matching Models with Threshold-based Policies *

Gontzal Sagastabeitia^{1,*}, Elene Anton², Josu Doncel¹

¹ *Mathematics Department, University of the Basque Country (UPV/EHU)*

² *LIUPPA, University of Pau and the Pays de l'Adour (UPPA)*

* Corresponding author: gontzal.sagastabeitia@ehu.eus

1. INTRODUCTION

We consider a dynamic matching model where a bipartite graph determines the compatibility among two set of nodes, usually referred to as “supply” and “demand” nodes. Items arrive to these nodes following a stochastic process and are matched by pairs - and leave the system - according to a matching policy. Items that are not matched queue up until they are matched later on, which entails a holding cost.

Most of the literature in dynamic matching models considers First-Come-First-Serve or Match-the-Longest [1, 5] matching policies and characterizes the steady-state distribution, which is of a product-form. Recent literature, see for instance [2–4], studies bipartite dynamic matching models where a bipartite graph describes the compatibility constraints among customers and servers. In [4] the authors consider the N -shaped bipartite matching model and show that a threshold-based policy optimizes the mean holding cost. The authors also conjecture that an analogous policy may be optimal in a more generalized setting.

In this extended abstract, we aim at analyzing the performance of threshold-based policies for the W -shaped dynamic matching model. On the one hand, we characterize the steady-state of the model for small threshold values and show that this is of a product-form. Moreover, we show that the threshold value is of fundamental importance when optimizing the weighed mean holding cost of the system. On the other hand, for general threshold values, we consider an approximation of the model and study the behavior of its weighted holding cost with respect to the threshold value.

2. MODEL DESCRIPTION

We consider a discrete-time W -shaped dynamic matching model. The model consists of two sets of nodes, $V_1 = \{v_1, v_2, v_3\}$ and $V_2 = \{v_4, v_5\}$, where items in nodes v_1 and v_3 are compatible with items in nodes v_4 and v_5 respectively, and items in node v_2 are compatible with items in any node in V_2 . Unmatched items queue up at its corresponding node.

At each time slot, two items arrive simultaneously, one to a node in V_1 and another to a node in V_2 . We denote

*This work has been partially funded by the Department of Education of the Basque Government through the Consolidated Research Group MATHMODE (IT1456-22). Elene Anton's research benefited from the support of the FMJH Program Gaspard Monge for optimization and operations research and their interactions with data science.

by p_i the probability that an item arrives to node v_i for $i = 1, \dots, 5$, where $p_1 + p_2 + p_3 = 1$ and $p_4 + p_5 = 1$. Moreover, we assume symmetric arrivals, i.e., $p_1 = p_3$ and $p_4 = p_5$. We denote by $\vec{Q} = (Q_1, \dots, Q_5)$ the queue length vector in steady-state.

We consider the following threshold-based matching policy with threshold value $T \in \mathbb{N} \cup \{\infty\}$: first match all possible items in v_1 and v_3 with those in v_4 and v_5 , respectively. Then, if the number of items in a V_2 -node exceeds the threshold value T , these items are matched to items in v_2 , up until there are $T - 1$ items in both V_2 -nodes. When $T = 1$, items in v_2 are immediately matched to any item in a V_2 -node. Hence, the system behaves as if there was no threshold.

The Markovian state descriptor of this system is given by the queue length vector at each time slot, after matching all possible items following the threshold-based policy. We denote by $t_{s_1}^{s_2}$ the 1-step transition probability from state s_1 to s_2 . Due to the symmetric arrivals and threshold-based policy, the associated Markov chain is symmetric w.r.t. queues v_1 and v_3 , and queues v_4 and v_5 . We use the latter symmetry to define a partition of the state space to apply lumpability, as defined in [6]. With some abuse of notation, we let $\vec{Q}$ denote the queue length vector in steady-state for the lumped Markov chain. Note that, $Q_1 = 0$ and $Q_2 + Q_3 = Q_4 + Q_5$, therefore $(Q_2, Q_3; Q_4)$ describes the state of the system.

We consider a linear cost function w.r.t. the queue lengths and symmetric costs for nodes v_1 and v_3 and for nodes v_4 and v_5 . We aim at analyzing the weighted mean holding cost of the form $\mathbb{E}[C(\vec{Q})] = \mathbb{E}[Q_3] + c\mathbb{E}[Q_2]$. We assume $c \in [0, 1]$; in which case, it has been conjectured in [4] that, for this matching model, threshold-based policies are optimal.

3. EXACT ANALYSIS FOR $T=1,2,3$

In this section, we assume threshold values $T = 1, 2, 3$. Let us define the path from state x to state y through a sequence of states $(x_1, \dots, x_n)$ as I_x^y , with $x_1 = x$ and $x_n = y$. We also define the set of all possible paths without cycles from state x to state y by $\mathcal{B}(x, y)$.

In the following, we provide several definitions required to characterize the product-form steady-state distribution of this model. Consider states s_0 , s_f and y as well as the path from s_0 to y , $I_{s_0}^y$.

We denote by $p_y^{s_f}(I_{s_0}^y)$ the probability of all the possible paths without cycles from state y to s_f , and where the path from s_0 to y is $I_{s_0}^y$. This is given by

$$p_y^{s_f}(I_{s_0}^y) = \sum_{(x_1, \dots, x_n) \in \mathcal{B}(y, s_f)} t_{x_1}^{x_2} \prod_{i=2}^{n-1} \delta\left(\frac{t_{x_i}^{x_{i+1}}}{p_{x_i}^{s_f}(I_{s_0}^{x_i})}, I_{s_0}^y\right), \quad (1)$$

where the function δ is defined as

$$\delta\left(\frac{t_{x_i+1}^{x_i}}{p_{x_i}^{s_f}(I_{s_0}^{x_i})}, I_{s_0}^y\right) = \begin{cases} \frac{t_{x_i+1}^{x_i}}{p_{x_i}^{s_f}(I_{s_0}^{x_i})}, & \text{if } I_{s_0}^y \cap I_y^{x_i} = \{y\} \\ 1, & \text{if } I_{s_0}^y \cap I_y^{x_i} \neq \{y\} \end{cases}. \quad (2)$$

Formula (1) represents the probability of all possible paths from y to s_f , omitting the portions of possible paths that intersect with $I_{s_0}^y$. I.e., when a possible path from y to s_f reaches a state in $I_{s_0}^y$, that path is not expanded further.

We denote by $p_{s_0}^{s_f}$ the probability of all the possible paths without cycles from state s_0 to s_f , such that $t_{s_0}^{s_f} > 0$ and $t_{s_f}^{s_0} > 0$. This is given by

$$p_{s_0}^{s_f} = \sum_{(x_1, \dots, x_n) \in \mathcal{B}(s_0, s_f)} t_{x_1}^{x_2} \prod_{i=2}^{n-1} \frac{t_{x_i}^{x_{i+1}}}{p_{x_i}^{s_f}(I_{s_0}^{x_i})}.$$

THEOREM 1. *Let s_0 and s_f be any two adjacent states, i.e., $t_{s_0}^{s_f} > 0$ and $t_{s_f}^{s_0} > 0$. Then, the following relation satisfies the balance equations of the system:*

$$p_{s_0}^{s_f} \pi(s_0) = p_{s_f}^{s_0} \pi(s_f).$$

Theorem 1 states that a balance property is satisfied between any two adjacent states.

From Theorem 1 we fully characterize the stationary distribution, for $T = 1, 2, 3$, and observe that it is of product-form. We also obtain explicit expressions of the weighted mean holding costs of this model. The order relation among the weighted mean holding costs depends on the holding weight c and arrival probability p_1 , as shown in Figure 1. Observe that both for small values of c and large values of p_1 , either $T = 2$ or $T = 3$ are optimal. Indeed, when p_1 is close to $\frac{1}{2}$, $T = 3$ outperforms both $T = 1, 2$.

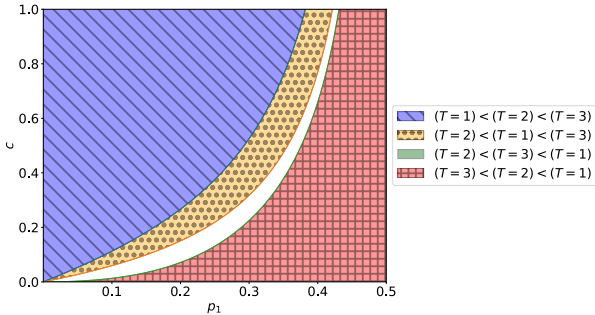


Figure 1: The order relation among the weighted mean holding costs w.r.t. the threshold value T , for $T = 1, 2, 3$.

4. APPROXIMATE ANALYSIS FOR $T > 3$

In this section, we consider a slightly different matching policy: whenever the number of waiting items in nodes v_4 and v_5 is smaller than $T - 1$, and a new item pair (v_2, v_4) or (v_2, v_5) arrives, this item pair is immediately matched and leaves the system, i.e., only (v_1, v_5) , and (v_3, v_4) pair arrivals queue up.

Under this policy, one can easily solve the balance equations of the lumped Markov chain, because there are fewer transitions among the states:

$$\pi(q_2, q_3; q_4) = \begin{cases} \left(\frac{p_2}{p_1}\right)^{q_2} \pi(0, 0; 0), & q_4, q_5 \leq T - 1 \\ \left(\frac{p_1}{p_1 + p_2}\right)^{q_4 - T + 1} \pi(0, 0; 0), & \text{otherwise} \end{cases}. \quad (3)$$

We obtain explicit expressions of the weighted mean holding cost of this modified model for any value of T . When $p_1 = p_2 = \frac{1}{3}$, the weighted mean holding cost is minimized at threshold values $T \leq 3$, for any value of $c \in [0, 1]$. In Figure 2 (left), we observe that as c increases, i.e., the weight of node v_2 increases, a smaller threshold value T is optimal. However, as T increases, node v_2 items are less likely to be matched. Hence, a smaller weight on v_2 is fairer.

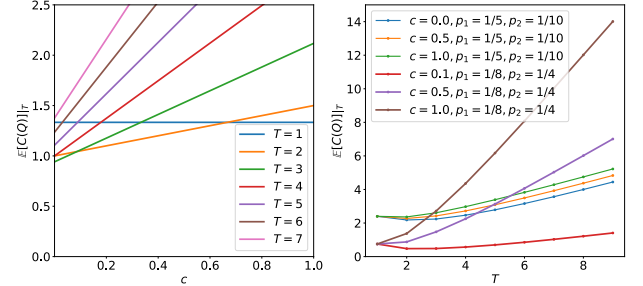


Figure 2: The weighted mean holding costs for $T = 1, \dots, 7$ with respect to c when $p_1 = \frac{1}{3}$ (left); and the weighted mean holding costs for c and p_1 , with respect to T (right).

When $p_1 \neq p_2$, the threshold value T that minimizes the weighted mean holding cost depends on both the value of c and p_1 , as shown in Figure 2 (right). Moreover, when $p_1 > p_2$ the weighted mean holding cost increases unbounded as $T \rightarrow \infty$, for any $c \in [0, 1]$. The same is true when $p_1 < p_2$ and $c > 0$. However, when $p_1 < p_2$ and $c = 0$, the weighted mean holding cost approaches 0 as $T \rightarrow \infty$.

5. FUTURE WORK

In this extended abstract, we fully characterize the stationary distribution of the W -shaped dynamic matching model for thresholds values $T = 1, 2, 3$. Generalizing these results to any threshold value remains an open problem that we aim to tackle in the near future. Moreover, we believe that the balance property that we obtain is also applicable to a broader framework, such as order independent queues.

6. REFERENCES

- [1] I. Adan, A. Bušić, J. Mairesse, and G. Weiss. Reversibility and further properties of FCFS infinite bipartite matching. *Mathematics of Operations Research*, 43(2):598–621, 2018.
- [2] I. Adan, I. Kleiner, R. Righter, and G. Weiss. FCFS parallel service systems and matching models. *Performance Evaluation*, 127:253–272, 2018.
- [3] A. Bušić, V. Gupta, and J. Mairesse. Stability of the bipartite matching model. *Advances in Applied Probability*, 45(2):351–378, 2013.
- [4] A. Cadas, J. Doncel, and A. Bušić. Analysis of an optimal policy in dynamic bipartite matching models. *Performance Evaluation*, 154:102286, 2022.
- [5] M. Jonckheere, P. Moyal, C. Ramírez, and N. Soprano-Loto. Generalized max-weight policies in stochastic matching. *Stochastic Systems*, 13(1):40–58, 2023.
- [6] J. G. Kemeny, J. L. Snell, et al. *Finite markov chains*, volume 26. van Nostrand Princeton, NJ, 1969.

Clifford Poisson Point Process for Structured Stochastic Geometry Modeling of 6G Wireless Systems

Rupei Xu
The University of Texas at
Dallas
rupeixu@gmail.com

Naofal Al-Dhahir
The University of Texas at
Dallas
aldhahir@utdallas.edu

Yuming Jiang
Norwegian University of
Science and Technology
yuming.jiang@ntnu.no

1. INTRODUCTION

Poisson point process (PPP) is fundamental in stochastic geometry for modeling random node locations in wireless networks [2, 4]. Classical PPP is inherently scalar and isotropic. Although they capture spatial randomness, other important information in 6G wireless communication, such as orientation, polarization, and geometric semantics of electromagnetic (EM) fields, is not considered. Existing extensions, such as marked PPP, Boolean models, and determinantal point processes, remain limited because they lack an internal algebraic structure capable of representing such information. To bridge the gap, this work introduces an innovative Clifford-algebraic model called the Clifford Poisson Point Process (Clifford PPP), where each point is marked with a Clifford multivector, as shown in Figure 1. In this model, in addition to location, other properties of a node, such as those related to an EM field, including its scalar, vector, bivector and higher multigrade components [1], are represented by a Clifford structure.

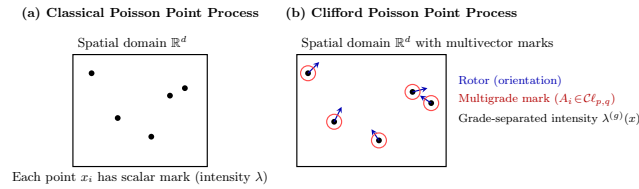


Figure 1: Comparison between (a) the traditional scalar Poisson Point Process (PPP) and (b) the proposed Clifford Poisson Point Process (Clifford PPP). Classical PPP models random spatial locations with scalar intensity λ , whereas Clifford PPP endows each point with a multivector mark $A_i \in \mathcal{C}_{p,q}$ carrying grade-wise information and rotor orientation.

2. CLIFFORD POISSON POINT PROCESS MODEL

2.1 Clifford Algebra and Multivectors

Let V be a real vector space of dimension $n = p + q$ equipped with a quadratic form $Q : V \rightarrow \mathbb{R}$ of signature (p, q) , where $Q(v) = v^2 = \langle v, v \rangle$. The Clifford algebra $\mathcal{C}_{p,q}$ is the unital associative algebra generated by V subject to

the relation $v^2 = Q(v)$, $\forall v \in V$. This implies the anticommutation relation for orthogonal basis vectors $e_i, e_j \in V$,

$$e_i e_j + e_j e_i = 2\eta_{ij},$$

where $\eta_{ij} = \text{diag}(\underbrace{+1, \dots, +1}_{p \text{ times}}, \underbrace{-1, \dots, -1}_{q \text{ times}})$, encodes the metric signature.

The reversion of a multivector R is denoted by $\tilde{R}$ and obtained by reversing the order of all basis vector products. For example, if $R = e_1 e_2 e_3$, then $\tilde{R} = e_3 e_2 e_1$. The spin group is $\text{Spin}(p, q) = \{R \in \mathcal{C}_{p,q}^+ \mid R\tilde{R} = 1\}$, representing rotations (and boosts) via $x' = Rx\tilde{R}$. In particular, $\text{Spin}(3) \simeq \text{SU}(2)$ for 3D spatial rotations. $\text{SU}(2)$ denotes the special unitary group of degree 2, $\text{SU}(2) = \{U \in \mathbb{C}^{2 \times 2} \mid U^\dagger U = I_2, \det U = 1\}$, where $U^\dagger$ denotes the Hermitian conjugate (also called the conjugate transpose) of a complex matrix U .

A general element of $\mathcal{C}_{p,q}$ is a *multivector*:

$$A = \sum_{k=0}^n \langle A \rangle_k = a_0 + a_1^i e_i + a_2^{ij} e_i e_j + \dots + a_n^{1\dots n} e_1 \dots e_n,$$

where $\langle A \rangle_k$ denotes the *grade- k* component. The full space $\mathcal{C}_{p,q}$ is 2^n -dimensional and decomposable into grades $0 \leq k \leq n$.

The *geometric product* of multivectors combines the symmetric inner product and the antisymmetric outer (wedge) product: $ab = a \cdot b + a \wedge b$. The Clifford algebra thereby encodes both algebraic and geometric relations and supports operations such as grade projection, reversion, and rotor transformations. It is particularly suited for modeling fields with direction, orientation, and phase, such as EM fields. For details, see [5].

2.2 Clifford Poisson Point Process

Let $(\mathbb{R}^d, \mathcal{B}, \mu)$ be a standard Borel measure space, and $\mathcal{C}_{p,q}$ be the Clifford algebra over $\mathbb{R}^{p+q}$ with $n = p + q$. Define the space of marked points: $\mathcal{S} := \mathbb{R}^d \times \mathcal{C}_{p,q}$.

Definition 1 (Clifford Poisson Point Process). A Clifford Poisson Point Process (Clifford PPP) Φ on $\mathcal{S}$ with intensity measure Λ is a random countable subset of $\mathcal{S}$ satisfying:

- For any disjoint measurable sets $B_1, \dots, B_k \subset \mathcal{S}$, the random variables $\Phi(B_i)$ are independent.
- For any measurable set $B \subset \mathcal{S}$, the number of points $\Phi(B)$ follows a Poisson distribution:

$$\mathbb{P}[\Phi(B) = k] = \frac{(\Lambda(B))^k}{k!} e^{-\Lambda(B)}.$$

Copyright is held by author/owner(s).

Each realization of Φ is a multiset $\{(x_i, A_i)\}_{i \in \mathbb{N}}$, where $x_i \in \mathbb{R}^d$ is the spatial location and $A_i \in \mathcal{C}_{p,q}$ is the multivector-valued mark. In the classical measure-theoretic formulation of probability established by Kolmogorov [6], any mark space M that is measurable (with a σ -algebra M) is admissible. Assigning Clifford algebra marks to spatial points is entirely legitimate in it: Clifford algebras are measurable vector spaces, so they serve as valid mark spaces for point processes. Unlike classical marked PPP [3], the mark space is non-commutative and graded, supporting modeling of polarization, angular phase, and internal EM structure.

2.3 Grade-Valued Intensity Function

To account for the algebraic decomposition of the field, we define an intensity function that encodes the distribution of multivector components per grade.

Let $\mathcal{G} = \{0, 1, \dots, n\}$ denote the set of grades in $\mathcal{C}_{p,q}$. Define the canonical grade projection $\pi_g : \mathcal{C}_{p,q} \rightarrow \mathcal{C}_{p,q}^{(g)}$ and write:

$$A = \sum_{g=0}^n \pi_g(A) = \sum_{g=0}^n A^{(g)}.$$

Definition 2 (Grade-Valued Intensity Measure). *Let $\lambda^{(g)} : \mathbb{R}^d \rightarrow \mathbb{R}_{\geq 0}$ be the grade- g intensity function defined by:*

$$\lambda^{(g)}(x) := \mathbb{E} \left[\sum_{(x_i, A_i) \in \Phi} \delta_{x_i}(x) \cdot \|A_i^{(g)}\|^2 \right].$$

Then, the total intensity is:

$$\lambda_{\text{total}}(x) := \sum_{g=0}^n \lambda^{(g)}(x).$$

This definition generalizes classical intensity to a grade-separated structure: $\lambda^{(0)}(x)$ corresponds to scalar field energy at x , $\lambda^{(1)}(x)$ corresponds to vector components (e.g., electric/magnetic), $\lambda^{(2)}(x)$ captures bivector content (e.g., angular momentum), and higher grades model curvature, torsion, or multiscale semantic structure.

These grade-wise intensities can be used to quantify signal strength, grade-coherent beamforming, and semantic compression via multigrade filtering.

2.4 Rotor-Valued Marks and Encoding

Each point $(x_i, A_i) \in \Phi$ is assumed to encode a geometric transformation via a **rotor** $R_i \in \text{Spin}(p, q) \subset \mathcal{C}_{p,q}$. A rotor is a multivector satisfying:

$$R_i \tilde{R}_i = 1, \quad R_i = \exp(-\frac{1}{2} B_i), \quad B_i \in \mathcal{C}_{p,q}^{(2)},$$

where B_i is a bivector representing a plane of rotation and $\tilde{R}_i$ is the Clifford reverse.

Let $A_i = \rho_i R_i$, where $\rho_i \in \mathbb{R}_{\geq 0}$ is a scalar amplitude. Then, R_i determines the local polarization or phase transformation applied by the RIS element at x_i . The field induced by the process becomes:

$$F(x) = \sum_{(x_i, A_i) \in \Phi} G(x - x_i) A_i,$$

where $G(x - x_i)$ is a grade-preserving Green's kernel.

The field $F(x)$ is thus a stochastic Clifford-valued random field: $F : \mathbb{R}^d \rightarrow \mathcal{C}_{p,q}$, whose multigrade statistics (mean,

covariance, entropy) depend on the rotor distribution, grade intensity, and spatial correlations.

Theorem 1 (Rotor-Grade Coupling in Clifford PPP). *Let $F(x)$ be the induced multivector field from a Clifford Poisson Point Process Φ with rotor-valued marks $A_i = \rho_i R_i$, where $\rho_i \in \mathbb{R}_+$ is a scalar weight and $R_i \in \text{Spin}(p, q)$ is a Clifford rotor. Then the expected energy in grade- g is given by:*

$$\mathbb{E} \|\pi_g F(x)\|^2 = \sum_{g'=0}^n \int_{\mathbb{R}^d} \|\pi_g [G(x - y) R^{(g')}]\|^2 \lambda^{(g')}(y) dy,$$

where $G(x - y)$ is a fixed Green's kernel, $\lambda^{(g')}(y)$ is the intensity measure of grade- g' sources, and π_g is the projection onto grade- g components.

3. DISCUSSION

The following provides a brief discussion on how Clifford PPP may be applied to help model some of the emerging 6G techniques, including reconfigurable intelligent surfaces (RIS) and six-dimensional movable antennas (6DMA).

RIS. Each RIS element acts as a rotor-valued scatterer, encoding programmable geometric phase and polarization. Clifford PPP can be used to capture grade-wise reflection, interference leakage, and alignment efficiency, increasing modeling accuracy than scalar PPP.

6DMA. In six-dimensional movable antennas, each transceiver possesses both translation and rotation. Clifford PPP can model this by rotor marks $R_i \in \text{Spin}(3)$, yielding closed-form expressions for capacity under orientation uncertainty and for grade-dependent Multiple Input Multiple Output (MIMO) rank degradation.

4. CONCLUSION

The proposed Clifford PPP model generalizes stochastic geometry to the multigrade domain, combining spatial randomness with algebraic field structure. It provides a mathematically rigorous and physically interpretable framework that opens up a new way to model and analyze 6G systems. Example applications include modeling programmable EM environments and enabling grade-aware interference analysis which are crucial for wireless systems like RIS and 6DMA. The approach may also be extended to the design and optimization of such systems.

5. REFERENCES

- [1] J. W. Arthur. *Understanding Geometric Algebra for Electromagnetic Theory*. Wiley, 2011.
- [2] Francois Baccelli and Bartłomiej Błaszczyszyn. *Stochastic geometry and wireless networks, Volume I: Theory*. Now Publishers Inc., 2009.
- [3] François Baccelli, Bartłomiej Błaszczyszyn, and Mohamed Karray. Random measures, point processes, and stochastic geometry, 2024.
- [4] Martin Haenggi. *Stochastic geometry for wireless networks*. Cambridge University Press, 2012.
- [5] David Hestenes and Garret Sobczyk. *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics*. Reidel, 1984.
- [6] Andrei N. Kolmogorov. *Grundbegriffe der Wahrscheinlichkeitsrechnung*. Julius Springer, Berlin, 1933. English translation: *Foundations of the Theory of Probability*, 2nd ed., Chelsea Publishing, 1956.

Subgradient Methods for Nonsmooth Convex Functions with Adversarial Errors*

Martijn Gösgens
CWI

research@martijngosgens.nl

Bart P.G. Van Parys
CWI

bart.van.parys@cwi.nl

ABSTRACT

We consider minimizing nonsmooth convex functions with bounded subgradients. However, instead of directly observing a subgradient at every step, the optimizer receives an *adversarially corrupted* subgradient. The adversary's power is limited to a finite corruption budget, with which the adversary can strategically time their perturbations. We show that the averaged subgradient descent method, which is optimal in the noiseless case, has worst-case performance that deteriorates *quadratically* with the corruption budget. Using performance optimization programming, (i) we construct and analyze the performance of three novel subgradient descent methods, and (ii) propose a novel lower bound on the worst-case suboptimality gap of any first-order method satisfying a mild cone condition proposed by [5]. The worst-case performance of each of our methods degrades only *linearly* with the corruption budget. Furthermore, we prove that all three proposed subgradient descent methods are *near-optimal* and suffer a worst-case suboptimality gap which asymptotically matches our lower bound. Our subgradient descent methods achieve near-optimal performance without needing momentum nor averaging. This suggests that these techniques are not necessary in this context, which is in line with recent results by [9].

1. INTRODUCTION

We consider a nonsmooth convex function $f : X \rightarrow \mathbb{R}$ with bounded subgradients. Instead of directly observing an *exact* subgradient $g_k \in \partial f(x_k)$ at every step k , we assume that the optimizer receives a *corrupted* subgradient $\tilde{g}_k = g_k + e_k$, where e_k is an adversarially chosen perturbation. We consider *subgradient decent* methods where we are given an initial iterate $x_0 \in X$ and construct iterates $x_{k+1} = x_k - \alpha_k \tilde{g}_k$ by moving in a negative subgradient direction with stepsize α_k . Classically, in an *averaging step*, a final iterate x_{N+1} is constructed as a weighted average of the iterates $x_0, \dots, x_N$. Within this class of subgradient methods parameterized by their stepsize schedule, we seek to construct a method that minimizes the worst-case suboptimality gap Δ . That is, a method which guarantees that

$$f(x_{N+1}) - \min_{x \in X} f(x) \leq \Delta$$

*Full article available at <https://arxiv.org/abs/2510.03072>

for any admissible $f, x_0, \tilde{g}_0, \dots, \tilde{g}_{N-1}$. We assume that the distance between the initial iterate x_0 and a minimizer is at most R , the subgradients of f have norm at most L and

$$\sum_{k=0}^{N-1} \|e_k\|^2 \leq \gamma^2.$$

That is, the adversary has a total *perturbation budget* γ^2 and can time their perturbations strategically. Such adversaries are of fundamental interest and have received a surge of recent attention in the optimization [1], bandit learning [6], and adversarial neural networks [8] communities.

Define $\sigma = \gamma/L$. For $\sigma \geq \sqrt{N}$, it is easy to see that no method can hope to make progress as the adversary can simply pick $e_k = -g_k$ so that $\tilde{g}_k = 0$ for all steps k . When $\sigma < \sqrt{N}$, progress does become possible and *performance estimation programming* introduced by [3, 7] can be used to characterize the worst-case suboptimality gap Δ for a given subgradient method as a convex semidefinite optimization problem. However, the search for a subgradient method with smallest suboptimality gap results in a nonconvex optimization problem [2]. We introduce a novel change of variables which however renders the search for an optimized subgradient method as a convex semidefinite optimization problem. This allows for numerical construction of optimized subgradient methods as a function of the problem parameters L, R, γ, N . Via a Lorentz cone approximation, we construct a stepsize schedule $(\alpha_k^L)_{k=0}^{N-1}$ of the form

$$\alpha_k^L = \frac{R(N-k)}{L(N+1)^{3/2}} \cdot \frac{y_k}{y_0 u_N^L(\sigma)}, \quad (1)$$

where $(u_N^L(\sigma))^2 \geq 1$ is the minimum of the resulting second-order cone optimization problem and $(y_k)_{k=0}^N$ is an auxiliary sequence satisfying the recursion $y_{k+1} = y_k + y_k^2$. We prove that the step size schedule (1) results in a suboptimality gap

$$\Delta \leq \frac{RL}{\sqrt{N+1}} u_N^L(\sigma), \quad (2)$$

We finally suggest a third sequence of step sizes that are given analytically as

$$\alpha_k' = \frac{R}{L} \frac{N-k}{(N+1)^{3/2}} \frac{u(\sigma)}{u(\sigma)^2 - (u(\sigma)^2 - 1)^{\frac{k}{N+1}}} \cdot \xi_N(\sigma), \quad (3)$$

where $u(\sigma) \geq 1$ is the solution of

$$\sigma^2 = u^2 - 1 - 2 \log u, \quad (4)$$

and

$$\xi_N(\sigma) = \sqrt{\frac{u(\sigma)^2 - 1}{\sigma^2 + \sum_{k=1}^{N+1} \frac{u(\sigma)^2 - 1}{k(u(\sigma)^2 - 1) + N + 1}}} \in \left[1, 1 + \frac{2}{N}\right],$$

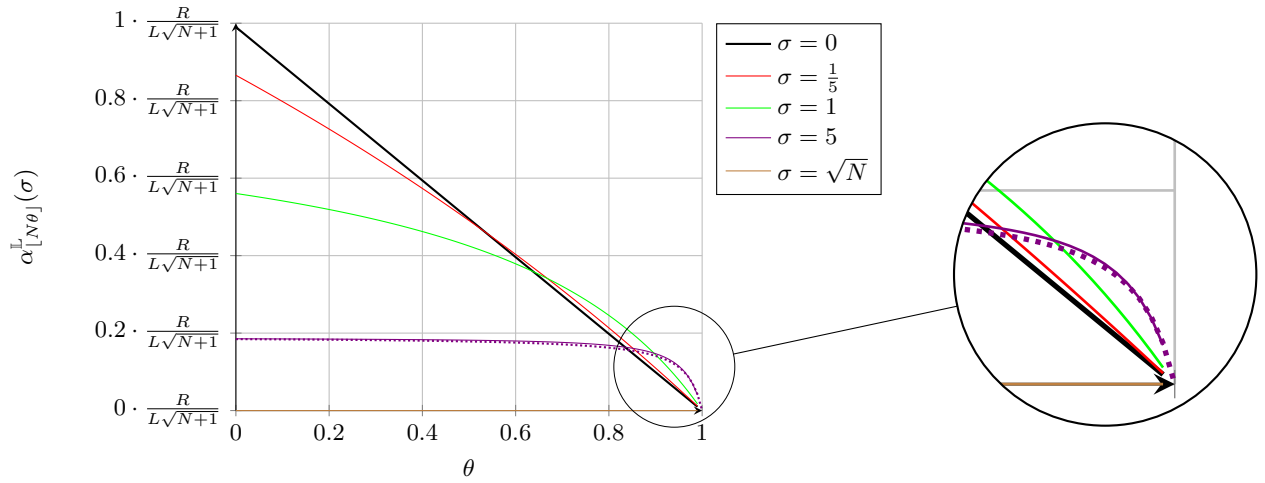


Figure 1: The conic combination $\alpha^L_{[N\theta]}(\sigma)$ (reindexed in $\theta \in [0, 1)$) for $N = 100$ and various noise levels σ . The dashed line corresponds to the step sizes α'_k for $\sigma = 5$.

so that this factor is negligible for large N . This step size schedule (3) achieves

$$\Delta \leq \frac{RL}{\sqrt{N+1}} u(\sigma).$$

The step sizes (1) and (3) are depicted in Figure 1 for various levels of gradient corruption. Unsurprisingly, increased gradient corruption begets a less aggressive overall step size schedule. However, the proposed subgradient method is observed to be much more cautious in earlier iterations than in later ones where it is in fact, more aggressive than in the absence of corruption. Notably, our proposed subgradient methods neither use iterate averaging or momentum. The same observation was recently also made in the absence of gradient corruption [9] who introduced an optimal subgradient method which our subgradient method can be interpreted to generalize.

Finally, we prove a novel lower bound on the worst-case suboptimality gap

$$\Delta \geq \frac{RL}{\sqrt{N+1}} \ell_N(\sigma).$$

Our lower bound holds for any first-order optimization method satisfying the cone conditions $x_0 - x_k \in \text{cone}(\tilde{g}_0, \dots, \tilde{g}_{k-1})$ for all steps k . This class contains any subgradient method with non-negative step sizes, the Nesterov accelerated gradient descent method and most practically relevant variable step size algorithms; see also [5]. For $\gamma = 0$ (uncorrupted), this performance bound coincides with the known universal lower bound for (uncorrupted) nonsmooth optimization [4]. We prove analytically that

$$\left(1 - \frac{5 \log(N+1)}{2N}\right) u(\sigma) \leq \ell_N(\sigma) \leq u_N^L(\sigma) \leq u(\sigma).$$

In particular, this means that the relative difference between the worst-case suboptimality gap of each of the proposed methods and the lower performance bounds vanishes to zero at rate $\mathcal{O}(\log(N)/N)$. This tells us that the proposed methods asymptotically match the proven theoretical lower bound.

2. REFERENCES

- [1] F.-C. Chang, F. Nabiei, P.-Y. Wu, A. Cioba, S. Vakili, and A. Bernacchia. Gradient descent: Robustness to adversarial corruption. In *Optimization for Machine Learning (NeurIPS 2022 Workshop)*, 2022.
- [2] S. Das Gupta, B. P. G. Van Parys, and E. K. Ryu. Branch-and-bound performance estimation programming: a unified methodology for constructing optimal optimization methods. *Mathematical Programming*, 204:567–639, 2024.
- [3] Y. Drori and M. Teboulle. Performance of first-order methods for smooth convex minimization: a novel approach. *Mathematical Programming*, 145:451–482, 2014.
- [4] Y. Drori and M. Teboulle. An optimal variant of Kelley’s cutting-plane method. *Mathematical Programming*, 160:321–351, 2016.
- [5] I. Fatkhullin, F. Hübler, and G. Lan. Can SGD handle heavy-tailed noise? *arXiv preprint arXiv:2508.04860*, 2025.
- [6] T. Lykouris, V. Mirrokni, and R. Paes Leme. Stochastic bandits robust to adversarial corruptions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 114–122, 2018.
- [7] A. B. Taylor, J. M. Hendrickx, and F. Glineur. Smooth strongly convex interpolation and exact worst-case performance of first-order methods. *Mathematical Programming*, 161:307–345, 2017.
- [8] Y. Wang, P. Mianjy, and R. Arora. Robust learning for data poisoning attacks. In *International Conference on Machine Learning*, pages 10859–10869. PMLR, 2021.
- [9] M. Zamani and F. Glineur. Exact convergence rate of the last iterate in subgradient methods. *SIAM Journal on Optimization*, 35(3):2182–2201, 2025.

Efficiency of an RL-based Algorithm for Models with an Absorbing Set of States*

Christian Carballo Lozano ^{† a,b}

Josu Doncel ^b

^a Plain Concepts, 48001 Bilbao, Spain

^b UPV/EHU, University of the Basque Country, 48940 Leioa, Spain

1. INTRODUCTION

Reinforcement Learning (RL) has emerged as a fundamental approach for solving sequential decision-making problems in environments characterized by uncertainty, complexity, and dynamic evolution. Unlike classical optimization techniques, RL does not require any knowledge of the model under investigation but instead learns optimal policies through interaction and feedback. This makes RL particularly suitable for real-world applications where modeling the entire system accurately is infeasible. Over the past decades, RL has achieved significant success in diverse fields such as robotics [1], healthcare [6], cybersecurity [3], and communication networks [2].

Our work presents an RL-based algorithm that applies to a wide family of models. Specifically, our model only assumes the existence of a set of states $\mathcal{S}^a$ which satisfies the following property: when the agent is in a state of $\mathcal{S}^a$ and takes any of the possible actions, the state that the environment returns also belongs to $\mathcal{S}^a$. This occurs, for instance, in any epidemic model without external infections.

We present an algorithm that leverages the structure of these models. Our numerical results show that our algorithm learns the optimal policy in a more efficient manner than the Q-learning algorithm.

2. MODEL DESCRIPTION

We consider a system that evolves randomly in discrete time. We denote by X_n the state of the system at time slot n and by $\mathcal{S}$ the set of states. In state $s \in \mathcal{S}$, an action $a \in A_s$ is chosen. We assume that $\mathcal{S}$ is finite and that A_s is a countable set for all $s \in \mathcal{S}$. We consider that being in state $s \in \mathcal{S}$ incurs a cost of $c(s)$. Moreover, when in state $s \in \mathcal{S}$ an action $a \in A_s$ is chosen, the state in the next time step is $s' \in \mathcal{S}$ with probability $p_{s,a}(s')$. The decision maker decides on a deterministic policy π , which determines the action to be taken in each state, i.e., $\pi : \mathcal{S} \rightarrow A_s$.

An optimal policy, which is denoted as π^* , is defined as the policy that minimizes the value function for all $s \in \mathcal{S}$. The standard way to compute π^* is by solving the Bellman equations [4]. However, in this work, we consider that we

do not have any knowledge about the transition probabilities and the costs of the model. Thus, we tackle the problem of finding π^* using an alternative approach based on RL.

The Q-learning algorithm [5] is an RL-based algorithm that provides $\hat{Q}$, which is an estimator of the action-value function. Let s_t and a_t be, respectively, the state and the action taken at time t . The Q-learning algorithm follows the next update rule:

$$\hat{Q}(s_t, a_t) \leftarrow \hat{Q}(s_t, a_t) + \eta_t \left(c(s_t) + \delta \min_a \hat{Q}(s_{t+1}, a) - \hat{Q}(s_t, a_t) \right), \quad (1)$$

where $\eta_t \in (0, 1]$ and $\delta \in [0, 1]$.

In this work, we make the following assumption.

ASSUMPTION 1. *We assume that there exists a set of absorbing states $\mathcal{S}^a$, i.e., if $s \in \mathcal{S}^a$, then for all $a \in A_s$, $\sum_{s' \in \mathcal{S}^a} p_{s,a}(s') = 1$.*

Under this assumption, it is easy to see that the Q-learning algorithm does not converge to the global optimum in the following cases: (i) when there is a single episode whose length is infinite, and (ii) when in each episode the initial state belongs to $\mathcal{S}^a$. As a consequence, we present an alternative approach in the following section.

3. OUR APPROACH

We notice that, under Assumption 1, the optimal action-value function of the states in $\mathcal{S}^a$ can be learned with any knowledge of the action-value function of the states that are not in $\mathcal{S}^a$. Therefore, we propose an RL-based algorithm that learns the optimal policy in a more efficient manner than the Q-learning algorithm. The pseudocode of our algorithm is shown in Algorithm 1. This algorithm initializes the Q-values randomly and, then, it works in two phases:

- In the first stage, we apply the update rule (1), where the starting state of each episode is a state of $\mathcal{S}^a$ drawn at random. This stage is executed for a fixed number of steps K^a . We consider that, in this stage, there are several episodes to make sure that all the states are visited a minimum number of times.
- In the second stage, we apply the update rule (1) and we consider that an episode ends when one of the following conditions is satisfied: (i) s_{t+1} , i.e., the state of the system at time $t + 1$, belongs to $\mathcal{S}^a$ or (ii) the maximum number of steps in that episode is reached. Then, a new episode starts in a randomly selected state

*This work has been partially funded by the Department of Education of the Basque Government through the Consolidated Research Group MATHMODE (IT1456-22).

[†]Corresponding author. Email: ccarballo002@ikasle.ehu.eus

Algorithm 1 Our approach

Parameters: learning rate $\eta \in (0, 1]$, $\epsilon > 0$, number of steps in the first stage K^a , number of steps of the simulation K^{sim} :
Initialize $Q(s, a)$ arbitrarily and $k = 0$
while $0 \leq k < K^a$ **do** // beginning of the first stage
 Initialize $s_0 \in \mathcal{S}^a$ randomly
 for each step t of episode **do**
 Apply Q-learning update rule (1) with ϵ -greedy policy
 $k \leftarrow k + 1$
 if episode ends **then**
 Initialize new episode
 end if
end for
end while // end of the first stage
while $k < K^{sim}$ **do** // beginning of the second stage
 Initialize $s_0 \in \mathcal{S} \setminus \mathcal{S}^a$
 for each step t of episode **do**
 Apply Q-learning update rule (1) with ϵ -greedy policy
 $k \leftarrow k + 1$
 if $s_{t+1} \in \mathcal{S}^a$ or episode ends **then**
 Initialize new episode
 end if
 end for
end while // end of the second stage

in $\mathcal{S} \setminus \mathcal{S}^a$. The second stage ends when the number of iterations equals K^{sim} .

4. APPLICATION

We consider a variant of the SIR model with N homogeneous elements that evolve in continuous time. An element encounters another element at a rate γ . If a susceptible element encounters an infected element, then it becomes infected. An infected element gets recovered at rate ρ . A recovered individual cannot be infected until it gets susceptible again at rate β . Moreover, with a vaccination rate α , a susceptible individual becomes recovered directly without being infected. We uniformize the continuous-time Markov chain with a uniformization constant $\Omega < (N(\gamma + \rho + \beta + \alpha))^{-1}$ and obtain a discrete-time Markov chain. In this model, susceptible elements follow a confinement policy (or strategy) $\pi: \mathbb{N} \rightarrow [0, 1]$, where $\pi(t)$ is a probability that indicates the level of exposure to the epidemic of that element at time t . More precisely, when $\pi(t) = 1$, susceptible elements are completely exposed to the epidemic at time zero, whereas when $\pi(t) = 0$ they are protected from taking the infection, i.e., they are confined. We consider that there is a confinement cost that applies to each susceptible element. We assume that the confinement cost of a susceptible element at time t is $c_L - \pi(t)$, where $c_L \geq 1$ (remember that $\pi(t)$ is the strategy of susceptible elements at time t). We also consider that there is a cost associated to each infected element; more precisely, each infected element leads to a cost of $c_I > 0$ per unit of time.

We consider that the action space is $\{0, 1\}$, i.e., susceptible elements are confined or fully exposed to the epidemic.

In this model, $\mathcal{S}^a$ is the set of states where there are no infected elements.

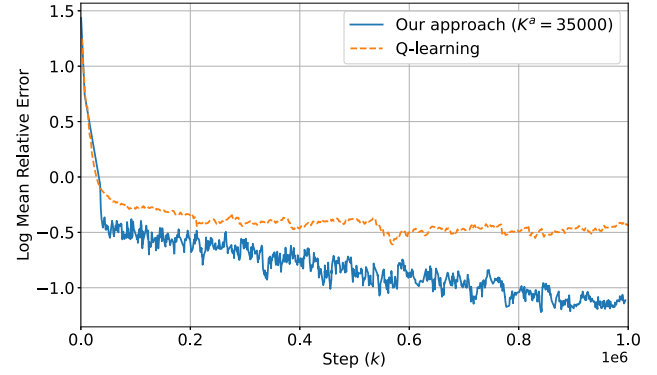


Figure 1: Logarithm of the mean relative error of Q-learning (dashed orange line) and our algorithm (blue line).

As a representative instance of the general pattern, let us consider the following configuration: $N = 5$, $\gamma = 1.1$, $\rho = 0.6$, $\alpha = 0.2$, $\beta = 0.3$, $c_I = 2$, and $c_L = 1.001$. The discount factor is set to $\delta = 0.99$ and $T = 2000$. In our experiments, we also consider $K^{sim} = 10^6$. We first study the impact of K^a in the performance of our approach, and then we will show how the presented algorithm outperforms Q-learning.

We compare the performance of our algorithm with that of Q-learning for this model. Figure 1 shows the evolution of the logarithm of the mean relative error. The dashed orange line corresponds to Q-learning: the error decreases quickly at first, but then slows down. The blue line shows our algorithm with $K^a = 35000$. This illustration shows that both methods perform similarly in the initial phase, but afterwards our approach achieves a faster and more sustained error reduction. The plotted case corresponds to the best performance observed among the tested values of K^a .

5. REFERENCES

- [1] P. Abbeel, A. Coates, M. Quigley, and A. Ng. An application of reinforcement learning to aerobatic helicopter flight. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006.
- [2] E. Ghadimi, F. Davide Calabrese, G. Peters, and P. Soldati. A reinforcement learning approach to power control and rate adaptation in cellular networks. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–7, 2017.
- [3] T. T. Nguyen and V. J. Reddi. Deep reinforcement learning for cyber security. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):3779–3795, 2023.
- [4] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. J. W. & Sons, 2014.
- [5] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018.
- [6] M. Tejedor, A. Z. Woldaregay, and F. Godtliebsen. Reinforcement learning application in diabetes blood glucose control: A systematic review. *Artificial Intelligence in Medicine*, 104:101836, 2020.

A Study on a Two-Threshold Queue Departure Process and its application to IEEE 802.15 MAC

Jie Chen
Computing Department
Guangdong University of Science and Technology

cjie3331@outlook.com

ABSTRACT

This paper provides a probabilistic analysis for the departure process of a standalone two-threshold queue and the derived calculation formula is then applied to select the optimal workload aware threshold sequence for IEEE 802.15 medium access control so the effectiveness of a dual-objective optimization scheme is guaranteed.

1. DEPARTURE PROCESS ANALYSIS

The expectation of inter-departure time is decomposed into the probabilistic weighted summation of expectations of two exclusive and complimentary events.

$$E[\Delta t_d^{k_1}] = (1 - p_0)A + p_0B \quad (1)$$

Let A represents the expectation of the inter departure time when a packet arrives at non-empty queue. Let B represents the event of packet arrival at an empty queue. $A = E[t_E]$, $B = E[t_a^{k_2} - t_d^{k_2-1} + t_E]$,

$$B|t^{k_2} - > \infty = \frac{1}{p_0\lambda} - \frac{t_E}{p_0} + t_E \quad (2)$$

$$(3)$$

Fig 1 shows both the simulation and analytical results for a two threshold queuing system and that the derived formulas for both event A and event B are consistent with simulation [1]. It evaluates the waiting time for the system with both an empty queue and non empty queue.

2. DEPARTURE PROCESS DISTRIBUTION

The author attains the departure process distribution consisting of two parts. That's

$$f(n) = f_E(n) + f_{NE}(n) \quad (4)$$

Given the condition that $\sum f_E(n) = p_0$ and $\sum f_{NE}(n) = 1 - p_0$ where $f_E(n)$ is the probability mass function when the packet arrives at an empty queue and $f_{NE}(n)$ is the PMF when the packet arrives at a non-empty queue.

$f_E(t) = Be^{-\beta t}[u(t - t_0) - u(t - t_1)] + Ce^{-\alpha t}u(t - t_1)$ with $\alpha = \lambda, \beta = \frac{2\lambda\mu}{\lambda + \mu}$ where $\mu = \frac{1}{t_E}, t_0 = E(t_{E_1})$ and $t_1 = 0.011$.

Copyright is held by author/owner(s).

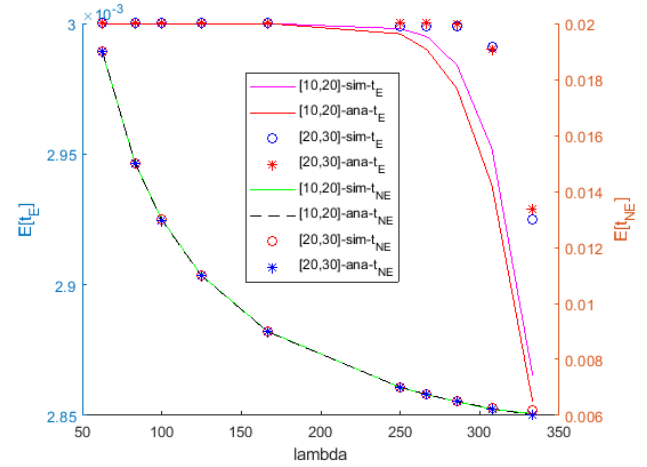


Figure 1: Model Validation - Inter-departure Time (E-NE)

It is easy to deduce that $f_{NE}(n)$ is the sum of probabilistic weights of three delta function $\delta(n - n_{t_k})$ with t_T corresponding to the three service time options. To be precise

$$f_{NE}(n) = \sum_i \sum_n p_n \delta(n - n_{t_i}) \quad (5)$$

And

$$p_n = \begin{cases} \frac{\sum_{i=1}^{L_1} p_i}{1 - p_0}, & \text{if } n = n_{t_1} \\ \frac{\sum_{i=L_1+1}^{L_2} p_i}{1 - p_0}, & \text{if } n = n_{t_2} \\ \vdots \\ \frac{\sum_{i=L_{i-1}+1}^{L_i} p_i}{1 - p_0}, & \text{if } n = n_{t_i} \\ \vdots \\ \frac{\sum_{i=L_T+1}^K p_i}{1 - p_0}, & \text{else} \end{cases} \quad (6)$$

The following fig 2 shows that the above analytical formula is consistent with simulation. The left y-axis shows the probability that the queue works in the first segmentation where queue length is from 1 to L_1 while the right y-axis shows the probability that the queue works in the second segmentation where queue length is from L_1 to L_2 .

As the inter-departure time PMF when the queue is empty can be further decomposed into weighted summation of the PMF when the consecutive two packets arrive at an empty

Asymptotically optimal reinforcement learning in Block Markov Decision Processes

Thomas van Vuren,^{*1}

Fiona Sloothaak,¹

Maarten G. Wolf,²

Jaron Sanders¹

¹Eindhoven University of Technology

²Koninklijke KPN N.V.

This extended abstract gives an overview of [6]’s results.

1. MOTIVATION

The field of machine learning, and artificial intelligence more broadly, has taken society by storm. One of its key techniques, **Reinforcement Learning (RL)**, is widely applied, e.g., in healthcare, industrial control, and robotics.

Despite its promise, **RL** still faces many practical challenges. Among these is the curse of dimensionality, which makes learning in large state spaces intractable without structure. This is problematic because the state space size is often exponential in the number of system components.

Fortunately, many environments do exhibit exploitable structure. For instance, when different states share similar transition probabilities, then we could first learn a clustering of the state space. This reduces its effective size, mitigating the curse of dimensionality’s impact.

2. STATE-SPACE REDUCTION IN BLOCK MARKOV DECISION PROCESSES

In [6], we conduct a regret analysis for an episodic learning strategy that exploits a state-space reduction technique tailored to **Block Markov Decision Processes (BMDPs)**.

BMDPs are **MDPs** in which observations are drawn from a large set of contexts. The transitions dynamics, however, depend on just a few unobserved latent states. Specifically, each context x is associated to a latent state $s = f(x)$, and the system moves to a new latent state s' according to a transition kernel $p(s' | s, a)$. The agent then observes only a context $y \in f^{-1}(s')$ drawn from an emission kernel $q(\cdot | s')$. See Figure 1 for further summary.

Recently, a clustering method developed for Block Markov Chains [5] was generalized to **BMDPs** [4]. It was then used in [4] to study the sample complexity of reward-free **RL**. A regret analysis when exploiting these techniques, however, remained open. This gap is addressed in [6].

3. SUMMARY OF RESULTS

Conducting a regret analysis for **BMDPs** poses unique challenges. Exploiting the block structure can reduce variance and tighten confidence bounds through data aggregation. However, errors in the learned latent clustering can bias estimates and complicate efficient exploration.

^{*}t.p.a.v.vuren@tue.nl

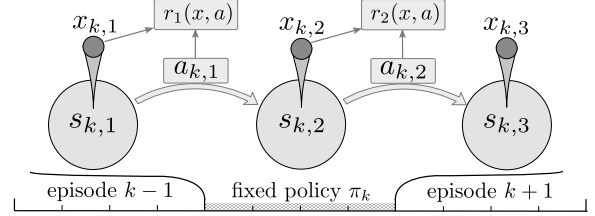


Figure 1: Shown from bottom to top are the sequences of hidden latent states, selected actions, observed contexts, and gathered rewards. Every H transitions, the environment resets to a uniformly random context. This defines an episode, and each episode the agent chooses a fixed policy.

[6] addresses these challenges with a two-phase approach building on [4]. Under [4]’s structural assumptions, clustering techniques accurately recover the latent structure based on random exploration. This avoids biases, and makes subsequent learning more efficient. Together with a regret lower bound established under these same assumptions, [6] obtains a sharp characterization of the regret in such **BMDPs**.

3.1 Structural properties

[6] analyzes the following subclasses of **BMDPs**:

Definition 3.1 (η -Reachability). *Let $\eta > 1$ be independent of n . A **BMDP** is η -reachable if for all sufficiently large n :*

$$\max_a \max_{s_1, s_2, s_3} \left(\frac{p(s_2 | s_1, a)}{p(s_3 | s_1, a)} \vee \frac{p(s_1 | s_2, a)}{p(s_1 | s_3, a)} \right) \leq \eta, \quad (1)$$

$$\max_s \max_{x, y \in f^{-1}(s)} \frac{q(x | s)}{q(y | s)} \leq \eta, \quad \text{and} \quad \max_{s_1, s_2} \frac{|f^{-1}(s_1)|}{|f^{-1}(s_2)|} \leq \eta.$$

Definition 3.2 ($\mathcal{I}$ -Identifiability). *Let $\mathcal{I} > 0$ be independent of n . An η -reachable **BMDP** is $\mathcal{I}$ -identifiable if, for all sufficiently large n , an information-theoretic measure of latent state separability is bounded from below by $\mathcal{I}$.*

Here, reachability ensures that random exploration observes every context. Identifiability ensures that the latent states are information-theoretically distinguishable.

3.2 Regret lower bound

[6, Theorem 1.3] is a regret lower bound for η -reachable and $\mathcal{I}$ -identifiable **BMDPs**. To state it, let n , S , A , K , and H , denote the number of contexts, latent states, actions, episodes, and transitions per episode, respectively, and let $T_K = KH$. Let $\text{Reg}_K(\mathcal{L}; \Phi)$ denote the expected regret after K episodes of a learning algorithm $\mathcal{L}$ on **BMDP** Φ .

Result ([6, Thm. 1.3]). Assume that the following hold:

$$S = o(n), \quad S/4 \in \mathbb{N}_{\geq 2}, \quad H \geq 2, \quad \text{and} \quad A \geq S/2. \quad (2)$$

For any learning algorithm $\mathcal{L}$, there exists an η -reachable and $\mathcal{I}_\eta$ -identifiable **BMDP** Φ , such that if $T_K = \Omega(n \vee A)$,

$$\text{Reg}_K(\mathcal{L}; \Phi) = \Omega(\sqrt{AT_K} + n). \quad (3)$$

The lower bound in (3) identifies two contributions to the regret. The $\Omega(\sqrt{AT_K})$ term captures the cost of learning the latent state transition kernel. The $\Omega(n)$ term captures the cost of learning the latent structure.

3.3 Two-phase **RL** algorithm and regret upper bound

[6, Algorithms 1 to 3] contain an explicit two-phase algorithm achieving near-optimal worst-case regret in η -regular and $\mathcal{I}$ -identifiable **BMDPs**; see [6, Theorem 1.4].

The first phase explores by taking actions uniformly at random, and then estimates the latent structure using the method of [4]. The second phase uses an optimism-based strategy to balance exploration and exploitation while refining transition kernel estimates. We adapt an algorithm designed for ordinary **MDPs** [1] by incorporating improved value estimates and smaller exploration bonuses that exploit the learned structure.

Result ([6, Thm. 1.4]). Assume that the following hold:

$$T_K = \Omega(H^4 S^3 A), \quad T_K \vee A = n^{O(1)}, \quad \text{and} \quad SH = O(n). \quad (4)$$

Assume, moreover, that Φ is η -reachable and $\mathcal{I}$ -identifiable. If [6, Alg. 1] explores for $nS^3 A \ln^2 n$ timesteps, then

$$\text{Reg}_K(\mathcal{L}; \Phi) = \tilde{O}(H\sqrt{SAT_K} + n(S^3 A + SH^3)). \quad (5)$$

Here, the notation $\tilde{O}(\cdot)$ suppresses polylogarithmic factors.

The upper bound in (5) is optimal in that it matches (3)'s dependence on n and T_K up to polylogarithmic factors. Compared to the best-possible $\Omega(\sqrt{nHAT_K})$ regret on **MDPs** without structure [3], exploiting the latent structure significantly reduces the regret for large time horizons.

The bound also improves over its closest analog, which is $\tilde{O}(DS\sqrt{AT} + n^2 SA)$ [2]. Here, D replaces H in their non-episodic setting. We specifically improve the n -dependence from $\tilde{O}(n^2)$ to $\tilde{O}(n)$, at the cost of a greater dependence on S , A , and H . This gives a better bound for large observation spaces. We achieve this by incorporating dedicated exploration and using [4]'s efficient clustering techniques.

In addition, (5) improves the leading-order regret for large T_K by $\sqrt{S}$. This is significant because [2]'s rewards depend solely on the latent state. This removes the need to learn the emission kernel, simplifying the analysis. The improvement stems from leveraging the algorithms and proof techniques in [1], rather than the earlier methods used in [2].

4. SUMMARY OF THE ANALYSIS

4.1 Proof of the regret lower bound

We prove [6, Thm. 1.3] in [6, Sec. 3] by defining a class of hard-to-learn **BMDPs**, adapted from [4], and showing that it includes instances inducing $\Omega(\sqrt{AT_K} + n)$ regret.

The $\Omega(\sqrt{AT_K})$ term is proven using techniques developed for **MDPs** without structure [3]. However, the constraints

posed by **Definitions 3.1** and **3.2** prevent direct application of these techniques. We propose a modified construction that satisfies these conditions, while still exhibiting the long-term $\Omega(\sqrt{T_K})$ growth of the regret.

The $\Omega(n)$ term follows from a new lower bound on the worst-case number of misclassifications of any latent state estimator on **BMDPs** in our class. Existing lower bounds [5, 4] apply to any given **BMDP**, but only for algorithms satisfying a consistency condition. By contrast, our class is such that any algorithm violating this condition performs poorly on at least one instance.

4.2 Proof of the regret upper bound

We prove [6, Thm. 1.4] in [6, Sec. 5] by first establishing accurate recovery of the latent structure after the first phase. We subsequently bound the regret during the second phase.

The analysis of the first phase refines [4]'s performance guarantees by deriving explicit probability bounds under weaker assumptions. These refinements are necessary for bounding the expected regret under [6, Thm. 1.3]'s assumptions, and establishing the optimality of both regret bounds.

The analysis of the second phase exploits these guarantees to broadly follow the optimism-based framework of [1]. The main new difficulty arises from the structure of the **BMDP** kernel, involving both a latent state transition kernel and emission probabilities. Their estimators are statistically dependent, and a key part of our analysis is a decomposition that disentangles their contributions to the regret.

Acknowledgements. This work is part of Valuable AI, a research collaboration between the Eindhoven University of Technology and the Koninklijke KPN N.V. Parts of this research have been funded by the Eindhoven Artificial Intelligence Systems Institute's IMPULS program.

References

- [1] M. G. Azar, I. Osband, and R. Munos. "Minimax Regret Bounds for Reinforcement Learning". *Proceedings of the 34th International Conference on Machine Learning*. 2017.
- [2] K. Azizzadenesheli, A. Lazaric, and A. Anandkumar. *Reinforcement Learning in Rich-Observation MDPs using Spectral Methods*. 2016. arXiv: [1611.03907](#).
- [3] O. D. Domingues, P. Ménard, E. Kaufmann, and M. Valko. "Episodic Reinforcement Learning in Finite MDPs: Minimax Lower Bounds Revisited". *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*. 2021.
- [4] Y. Jedra, J. Lee, A. Proutière, and S. Yun. "Nearly Optimal Latent State Decoding in Block MDPs". *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. 2023.
- [5] J. Sanders, A. Proutière, and S. Yun. "Clustering in Block Markov Chains". *The Annals of Statistics* (2020).
- [6] T. van Vuren, F. Sloothak, M. G. Wolf, and J. Sanders. *Asymptotically optimal reinforcement learning in Block Markov Decision Processes*. 2025. arXiv: [2510.13748](#).

A Variational Framework for the RLGL Algorithm

Konstantin Avrachenkov
Centre Inria d'Université Côte
d'Azur, NEO team, Sophia
Antipolis, France
konstantin.avrachenkov@
inria.fr

Lorenzo Gregoris
Department of Mathematics
and Computer Science,
Eindhoven University of
Technology
Eindhoven, The Netherlands
l.gregoris@tue.nl

Nelly Litvak
Department of Mathematics
and Computer Science,
Eindhoven University of
Technology
Eindhoven, The Netherlands
n.v.litvak@tue.nl

1. MOTIVATION AND PROBLEM FORMULATION

Computing the stationary distribution π of a large, irreducible Markov chain with transition matrix P is a fundamental task in applied mathematics and computer science [4]. Solving the eigenvalue problem $\pi = \pi P$ is typically intractable for direct solvers in large dimensions, leaving iterative methods as the only viable option [7].

The classical power method, $x_{t+1} = x_t P$, is simple but not always efficient. The recently proposed *Red Light Green Light* (RLGL) algorithm [2] is a more flexible and often faster alternative. It performs block-coordinate updates of the form

$$x_{t+1} = x_t + r_t I_{B_t}, \quad r_t = x_t(P - I),$$

where r_t is the residue vector at iteration t and I_{B_t} is a projection onto a selected subset of coordinates B_t . While heuristics for choosing B_t (e.g., selecting coordinates with the largest residue) perform remarkably well, a formal convergence analysis has remained elusive. Similar coordinate selection heuristics have been studied in the context of block coordinate descent methods [3, 5].

This work frames RLGL as a coordinate descent algorithm on a suitable energy functional, giving new insight on update schedules. Using perturbative arguments, we also extend this perspective to some irreversible chains.

2. DIRICHLET ENERGY MINIMIZATION

The primary challenge in formulating an optimization problem is that the matrix $I - P^\top$ is generally not symmetric. Standard least-squares approaches [1] that minimize $\|(I - P^\top)x^\top\|^2$ involve the matrix $(I - P)(I - P^\top)$, which can be dense and ill-conditioned.

Our contribution is to provide a theoretical foundation by framing RLGL as a block descent method on a suitable energy functional, connecting it to the literature on block coordinate descent [5, 6].

Let π be the stationary distribution of P , and $\Pi = \text{diag}(\pi)$. The time-reversal is $P^* = \Pi^{-1}P^\top\Pi$. The reversible part of P is

$$P_S = \frac{1}{2}(P + P^*).$$

The coordinate transformation $y = x\Pi^{1/2}$ symmetrizes P_S .

Copyright is held by author/owner(s).

The *symmetrized Laplacian* is

$$\mathcal{L} = I - \Pi^{1/2}P_S\Pi^{-1/2}.$$

It is symmetric, PSD, and has kernel spanned by $\sqrt{\pi} = \mathbf{1}\Pi^{1/2}$. The associated *Dirichlet energy* is

$$\mathcal{E}(y) = \frac{1}{2}y\mathcal{L}y^\top.$$

Its gradient is $\nabla\mathcal{E}(y) = y\mathcal{L} = r\Pi^{-1/2}$. By minimizing $\mathcal{E}(y)$ we can recover the stationary distribution.

3. CONVERGENCE VIA COORDINATE DESCENT

The RLGL updates in x_t correspond to coordinate descent steps on $\mathcal{E}(y)$ with $y_t = x_t\Pi^{1/2}$. For reversible chains, residue r_t and gradient $\nabla\mathcal{E}(y_t)$ are closely linked.

Since $\mathcal{E}$ satisfies the Polyak–Lojasiewicz inequality with Lipschitz constants $L_i = 1 - P_{ii}$, we obtain:

Theorem 1. *Let P be reversible. If the selection rule ensures $|\partial_{i_t}\mathcal{E}(y_t)|^2 \geq \beta\|\nabla\mathcal{E}(y_t)\|_2^2$ for some $\beta > 0$, then*

$$\mathcal{E}(y_{t+1}) \leq \left(1 - \frac{\beta\mu}{L_{\max}}\right)\mathcal{E}(y_t),$$

where $L_{\max} = \max_i L_i$ and μ is the spectral gap of P .

In particular, this proves exponential convergence for RLGL with Gauss–Southwell and random selection heuristics. This upper bound on the rate is often faster than Power Iteration (see Figure [1]).

4. RESCALED GAUSS–SOUTHWELL HEURISTIC

The steepest decrease rule is $i_t = \arg\max_i |(y_t\mathcal{L})_i|$. In x -coordinates:

$$i_t = \arg\max_i |(r_t\Pi^{-1/2})_i|, \quad r_t = x_t(I - P).$$

We call this heuristic *rescaled Gauss–Southwell* rule, since it uses rescaled residues with weights $\pi_i^{-1/2}$. In practice, π is unknown, so proxies such as x_t are used (see Figure [2]).

5. EXTENSION TO NEARLY REVERSIBLE CHAINS

Write

$$S = \frac{1}{2}\Pi^{1/2}(P + P^*)\Pi^{-1/2}, \quad A = \frac{1}{2}\Pi^{1/2}(P - P^*)\Pi^{-1/2}.$$

Then $I - S$ is symmetric PSD and A is antisymmetric.

Definition 1 (Nearly Reversible Chains). A chain with transition matrix P is η -nearly reversible if for $\eta \in [0, 1)$

$$\eta = \max_{i \in [n]} \frac{\kappa_i}{\mu}, \quad \kappa_i = \|A_{i,:}\|_2 = \frac{1}{2\sqrt{\pi_i}} \left(\sum_{j=1}^n \frac{1}{\pi_j} (\pi_j p_{ji} - \pi_i p_{ij})^2 \right)^{1/2} [1]$$

where $\mu = \gamma(P)$ is the Poincaré constant [8].

Theorem 2 (Convergence of RLGL). Let P have no self-loops and Poincaré constant μ . If P is η -nearly reversible with $\eta < \sqrt{\beta}$, and the update sequence has fraction $\rho > 0$ of good updates with parameter β , then RLGL converges exponentially with rate at least

$$-\rho \log \left(1 - \beta\mu + \frac{\kappa^2}{\mu} \right).$$

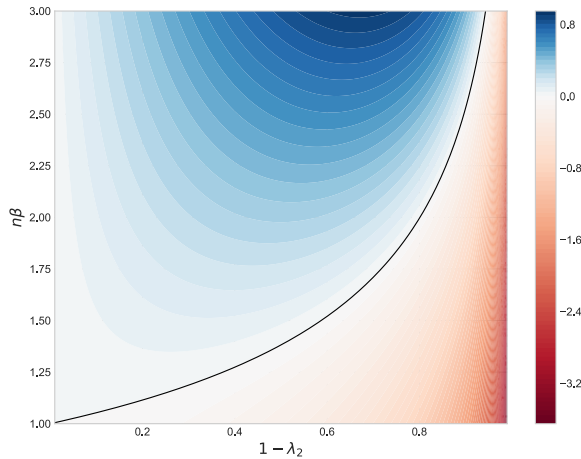


Figure 1: Difference of asymptotic convergence rate comparison of RLGL and Power Iteration on reversible chains. On the horizontal axis the spectral gap μ , on the vertical axis the lower bound on the residue of updated nodes β rescaled by the size n . In blue where RLGL is faster, in red where PI is faster. The black curve corresponds to equal rates.

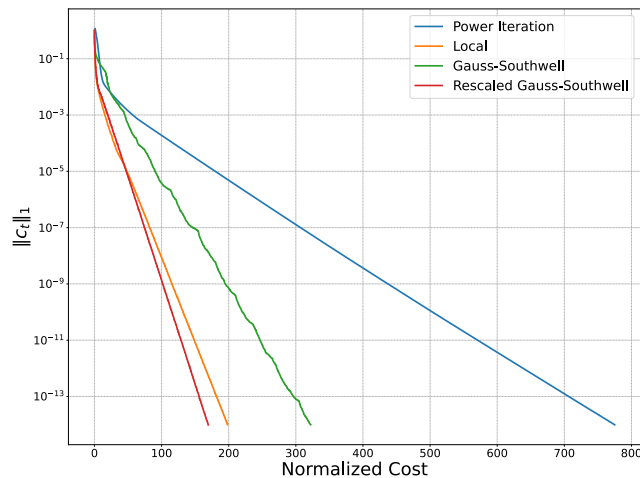


Figure 2: Empirical comparison of convergence rates for different RLGL heuristics versus Power Iteration, computed on the largest strongly connected component of the Harvard500 web graph. On the vertical axis the ℓ_1 norm of the residue, on the horizontal axis the number of normalized iterations, i.e., the fraction of activated edges.

References

- [1] Anton Anikin et al. “Efficient Numerical Methods to Solve Sparse Linear Equations with Application to PageRank”. In: *Optimization Methods and Software* 37.3 (2022-05-04), pp. 907–935. ISSN: 1055-6788, 1029-4937. DOI: [10.1080/10556788.2020.1858297](https://doi.org/10.1080/10556788.2020.1858297) arXiv: [1508.07607](https://arxiv.org/abs/1508.07607) [math]. URL: <http://arxiv.org/abs/1508.07607>.
- [2] Konstantin Avrachenkov, Patrick Brown, and Nelly Litvak. “Red Light Green Light Method for Solving Large Markov Chains”. In: *Journal of Scientific Computing* 93.1 (2022-10), p. 18. ISSN: 0885-7474, 1573-7691. DOI: [10.1007/s10915-022-01976-8](https://doi.org/10.1007/s10915-022-01976-8). URL: <https://link.springer.com/10.1007/s10915-022-01976-8>.
- [3] Amir Beck and Luba Tretuashvili. “On the Convergence of Block Coordinate Descent Type Methods”. In: *SIAM Journal on Optimization* 23.4 (2013), pp. 2037–2060. DOI: [10.1137/120887679](https://doi.org/10.1137/120887679). eprint: <https://doi.org/10.1137/120887679>. URL: <https://doi.org/10.1137/120887679>.
- [4] Sergey Brin and Lawrence Page. “The anatomy of a large-scale hypertextual web search engine”. In: *Computer networks and ISDN systems* 30.1-7 (1998), pp. 107–117.
- [5] Yu Nesterov. “Efficiency of coordinate descent methods on huge-scale optimization problems”. In: *SIAM Journal on Optimization* 22.2 (2012), pp. 341–362.
- [6] Julie Nutini et al. “Coordinate descent converges faster with the Gauss–Southwell rule than random selection”. In: *Proceedings of the 32nd International Conference on Machine Learning*. PMLR, 2015, pp. 1632–1641.
- [7] Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.
- [8] Justin Salez. “Modern aspects of Markov chains: entropy, curvature and the cutoff phenomenon”. In: *arXiv preprint arXiv:2508.21055* (2025).

Reliable estimation of small exceedance probabilities for frequency reserve dimensioning in electricity grids

D. Mastropietro¹ and A. Zocca²

¹CNRS-IRIT, Université de Toulouse INP, 31071 Toulouse, France (daniel.mastropietro@gmail.com)

²Department of Mathematics, Vrije Universiteit Amsterdam, The Netherlands (a.zocca@vu.nl)

ABSTRACT

We propose using Fleming-Viot (FV) particle systems to estimate small exceedance probabilities of the electricity grid frequency for the dynamic dimensioning of frequency containment reserves (FCR). The methodology is proposed as a simpler alternative to importance sampling for diffusion processes, which generally requires the optimization of a functional. Under equal sample sizes, the FV estimator can reliably estimate probabilities that are smaller than those estimated with vanilla Monte Carlo by a factor of 10^{-4} .

1. INTRODUCTION

The frequency of power grids has become increasingly volatile in recent years due to the growing penetration of renewable energy sources. Significant deviations of the frequency from its nominal value can compromise the stable operation of the grid. To mitigate this risk, transmission system operators (TSOs) deployed energy reserve strategies that serve as backup mechanisms. These reserves are typically allocated as a constant volume throughout the year, based on historical volatility levels. However, because frequency volatility varies over time, such static strategies are likely to be suboptimal — reserves may be oversized during low-volatility periods and insufficient during high-volatility ones.

Our work is inspired by that of Janssen et al. [3], who propose a dynamic dimensioning of energy reserves as an improvement over static allocation. Their method relies on two key components: (i) an estimate of the future frequency volatility, and (ii) an estimate of the probability that frequency deviations exceed TSO-defined thresholds, triggering reserve activation. They model frequency deviations using an Ornstein–Uhlenbeck (OU) diffusion process and capture the corrective effect of reserve activation by incorporating a regressive drift term that pulls the frequency back toward its stationary mean of 50 Hz.

As shown in [3], within this framework, the exceedance probabilities can be expressed in closed form and computed via numerical integration. Although the OU process provides a reasonable first-order approximation of frequency dynamics, it fails to reproduce the heavy-tailed distributions commonly observed in empirical data. More realistic stochastic models can account for these heavy tails, but they typically pre-

clude closed-form expressions for exceedance probabilities, necessitating estimation from simulated data.

To address this limitation, we propose a simulation-based approach that yields reliable estimates of exceedance probabilities when analytical computation is infeasible. We leverage Fleming-Viot (FV) particle systems, a multidimensional stochastic process constrained to evolve outside regions frequently visited by the underlying process. This mechanism facilitates efficient exploration of rare regions using a conceptually simpler framework than importance sampling, which, in the diffusion context, may pose practical challenges such as the minimization of a functional [6].

We adapt the approach developed in [5] —originally formulated for estimating rare-event probabilities in continuous-time Markov processes on finite state spaces (e.g., queueing networks)— to the diffusion setting. We test our proposed FV-based method on two different diffusion models for frequency evolution: the standard OU process with Brownian noise introduced by [3], used here as validation, and a diffusion OU process with Lévy noise, which better captures the heavy tails typically observed in practice [4]. In our numerical studies, we compare our FV estimation method with a traditional Monte Carlo (MC) approach and quantify the smallest probability that can be reliably estimated by each method under equal sample sizes.

2. PROPOSED METHODOLOGY

We adopt the same stochastic model for frequency evolution proposed by [3], which mimics the frequency containment reserve (FCR) strategies used by Fingrid, one of the TSOs of the Nordic grid, where different designated reserve volumes are activated at different thresholds. The resulting model is a stochastic OU process described by the SDE

$$dF_t = -\alpha(F_t)F_t dt + \sigma dW_t, \quad (1)$$

where F_t is the frequency deviation at time t from its stationary mean, W_t is a noise term with unit variance, and $\alpha(F_t)$ is a regressive drift function that captures the effects of reserves (see [3] for details). As argued earlier, TSOs need to reliably estimate the probability that the frequency F_t exceeds specific thresholds for a given reserve allocation policy $\alpha(\cdot)$. Our estimation method leverages the regenerative characterization of the stationary probability of a set $\mathcal{C}$ for an ergodic Markov process [1, Ch. 6] which, in continuous time, is given by:

$$p(\mathcal{C}) = \frac{\mathbb{E}_A \int_0^{T_A} \mathbf{1}_{\{F_t \in \mathcal{C}\}} dt}{\mathbb{E}_A T_A}, \quad (2)$$

Copyright is held by author/owner(s).

where A is an event defining a regenerative cycle and T_A is the regenerative cycle time.

The FV estimation approach builds on the now matured use of FV particle systems to consistently estimate quasi-stationary distributions of Markov processes with absorption [2]. As shown in [5], the numerator in (2) can be written in terms of an approximation of the quasi-stationary distribution of a process that is absorbed when visiting a set $\mathcal{A}$, which in turn defines the regenerative cycle of the original Markov process $\mathcal{M}$. Thus, the event A in (2) becomes the *absorption event*, and $\mathcal{A}$ is called the *absorption set*. The FV particle system consists of N copies of $\mathcal{M}$ constrained to evolve outside $\mathcal{A}$ by an appropriate restarting mechanism of a particle touching $\mathcal{A}$. To achieve our goal of accurately estimating the probability of extreme deviations of the process from its mean, we define the absorption set $\mathcal{A}$ as a set of frequently visited states by $\mathcal{M}$, which enables the FV particle system to collect further samples from rarely visited regions. Using the convergence guarantees of the FV estimator of the quasi-stationary distribution as the number of particles $N \rightarrow +\infty$ [2], a consistent estimator of the stationary probability of a set $\mathcal{C}$, *outside* $\mathcal{A}$, can also be constructed [5]. When $\mathcal{C}$ is a low-probability set, this FV estimator is expected to provide a more accurate estimation of $p(\mathcal{C})$ than using (2) on samples from an unconstrained MC exploration by $\mathcal{M}$.

Besides the estimation of the approximated quasi-stationary distribution, the FV estimator requires the estimation of the expected reabsorption time into $\mathcal{A}$, the denominator of (2). The estimation is carried out from data collected from an unconstrained MC exploration by $\mathcal{M}$. An important requirement is that this exploration must observe exit events from $\mathcal{A}$, because the estimated exit state distribution is used for the selection of the starting positions of the N particles. Thus, as the probability of $\mathcal{A}$ increases, the exploration time T also needs to increase.

3. EXPERIMENTAL SETUP AND RESULTS

We design a set of experiments that allows us to analyze the method's performance in terms of decreasing event probabilities. We consider the OU process defined in (1) with Brownian noise, and leverage its symmetry to reflect it at zero, in order to simplify the estimation problem on events defined on the positive axis. We consider 8 nested events of interest of the form $\mathcal{C}_k := (c_k, +\infty)$, whose probabilities are linearly spaced on logarithmic scale, from 10^{-1} to 10^{-8} . Each c_k is computed using the formula for the exceedance probability $p_k := \mathbb{P}(\mathcal{C}_k)$ given in [3], adapted to the reflected OU process. For each k , the absorption set $\mathcal{A}_k$ is heuristically set as $(0, a_k)$ with $a_k = 0.80 c_k$, and the FV hyperparameter T_k , which defines the number of steps in the unconstrained MC simulation, is dimensioned using a very rough approximation of the exit event as the success event of independent Bernoulli trials with parameter equal to the probability q_k that the OU process is outside $\mathcal{A}_k$, namely as $q_k := 1 - \mathbb{P}(\mathcal{A}_k)$. The value of T_k is derived by imposing a minimum value M of the expected number of exit events during the T_k discrete steps of the OU process simulation. Hence, given M and a_k , we set $T_k := M/(1 - \mathbb{P}((0, a_k)))$, capped to a minimum of 500. To check the consistency of the estimator, we considered four different N values for each k : $N = 20, 40, 80, 160$. For each replication in each experiment, an MC estimator is also calculated using (2), on the basis of data with the same sample size as the FV estimator.

k	c_k	p_k	a_k	q_k	$T_k := 20/q_k$
1	0.071	1.09×10^{-1}	0.057	2.21×10^{-1}	90, cap. to 500
2	0.102	1.18×10^{-2}	0.082	5.83×10^{-2}	342, cap. to 500
3	0.125	1.07×10^{-3}	0.100	1.40×10^{-2}	1,423
4	0.142	1.11×10^{-4}	0.114	3.73×10^{-3}	5,366
5	0.158	1.01×10^{-5}	0.126	9.38×10^{-4}	21,324
6	0.169	1.31×10^{-6}	0.135	2.91×10^{-4}	68,628
7	0.181	1.34×10^{-7}	0.145	8.00×10^{-5}	249,744
8	0.192	1.07×10^{-8}	0.154	1.90×10^{-5}	1,034,972

Table 1: Setup of the 8 experiments run with $M = 20$ on the reflected version of the OU process in (1).

Figure 1 compares the FV and MC estimators of the exceedance probabilities defined in Table 1. We observe that MC starts to partially fail at probabilities of the order of 10^{-4} and completely fails for smaller probabilities by always giving a null estimate. In contrast, FV starts to partially fail only at probabilities of the order of 10^{-8} . When estimators do not fail, violin plots tend to get smaller as N increases while including the true probability, suggesting their consistency. Although not presented here, the results for the diffusion with Lévy noise are very similar to the Brownian case.

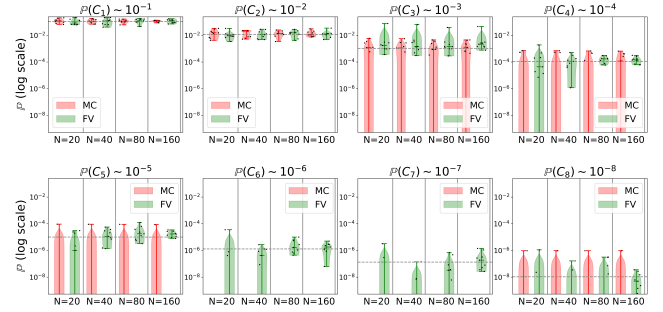


Figure 1: Violin plots on 9 replications comparing our FV estimator (green) vs. the MC estimator (red) of the eight exceedance probabilities $p := \mathbb{P}(\mathcal{C})$ defined in Table 1, marked by the horizontal gray line in each subplot.

4. REFERENCES

- [1] S. Asmussen. *Applied Probability and Queues*. Springer, 2003.
- [2] A. Asselah, P.A. Ferrari, and P. Groisman. Quasistationary distributions and Fleming-Viot processes in finite spaces. *J. Appl. Probab.*, 48(2):322–332, 2011.
- [3] J. Janssen, A. Zocca, B. Zwart, and J. Kazempour. Dynamic dimensioning of frequency containment reserves: The case of the nordic grid. *arXiv:2411.11093*, 2025.
- [4] D. Kraljic. Towards realistic statistical models of the grid frequency. *IEEE Transactions on Power Systems*, 38(1):256–266, 2022.
- [5] D. Mastropietro, U. Ayesta, M. Jonckheere, and S. Majewski. Fast-exploring reinforcement learning with applications to stochastic networks. *Queueing Systems*, 109(3), 2025.
- [6] E. Vanden-Eijnden and J. Weare. Rare event simulation of small noise diffusions. *Communications on Pure and Applied Mathematics*, 65(12):1770–1803, 2012.

Optimal Location and Allocation in Shared Mobility Services via Semi-discrete Optimal Transport and Simulated Annealing

Kun Jin*, Andre Carrel†, Simon Zhang*, Cathy Xia‡

*Department of Computer Science and Engineering

†Department of Civil, Environmental, and Geodetic Engineering

‡Department of Integrated Systems Engineering

The Ohio State University, Columbus, Ohio, USA

‡xia.52@osu.edu

ABSTRACT

Shared mobility services, including car-sharing, bike-sharing, and microtransit, are reshaping urban transportation by improving accessibility and addressing the first/last mile gap. We study the design of a hub-to-spoke service model, where users access designated locations rather than point-to-point rides, offering a more cost-effective alternative with competitive performance. The operator must decide both the placement of service hubs and the allocation of users, a problem that is NP-hard under dense demand. To address this, we approximate the dense demand as a continuous density function and study the continuous variant of the problem. We introduce a framework combining Optimal Transport theory with Simulated Annealing to solve this location-allocation problem efficiently. Using 2.2 million trips from the New York City taxicab dataset, we demonstrate that our algorithms can scale to city-level demand and provide robust solutions for shared mobility services.

1. INTRODUCTION

Shared mobility services, a key trend in the evolution of urban transportation, aim to provide flexible, efficient, and accessible alternatives to private vehicle use [1]. These services include car-sharing, bike-sharing, scooter-sharing, microtransit, peer-to-peer sharing, and other demand-responsive modes that can either operate independently or in coordination with public transit agencies and private employers. Cities across the U.S. and beyond are increasingly adopting such services to build more connected, reliable, and sustainable urban mobility networks. In practice, their design and evaluation often rely on tools such as Geographic Information System (GIS) software, simulation platforms (*e.g.*, MATSim), mobility analytics systems (*e.g.*, Remix, Populus), which may involve vast quantities of sensor data since the demand is dense.

In this work, we study the optimal design of hub-to-spoke shared mobility services within a city. The design problem involves two central challenges: identifying the optimal service locations (*e.g.*, stops, docks, or hubs) and determining how users should be allocated to these locations. The two

subproblems are often considered jointly, known as the location allocation problem. With dense discrete demand data, it is also computationally challenging to solve the problem since it is well known that the location allocation problem is NP-hard [2].

In this paper, we approximate the dense demand as a continuous density function, and formulate the problem as a two-stage location-allocation optimization program as follows. We frame the allocation problem as a semi-discrete optimal transport (OT) problem [3], which maps continuous demand distributions onto a discrete set of candidate service locations. For the location problem, we develop a Monte Carlo Markov Chain (MCMC) to model the discrete optimization problem, and solve it by exploiting the Metropolis-Hastings algorithm and simulated annealing techniques [4].

To demonstrate scalability and practicality of our method, we conduct large-scale numerical experiments, including an application to 2.2 million trips from the New York City taxicab dataset. Our results show that our algorithm can efficiently handle city-scale demand and produce robust solutions.

2. MODEL FORMULATION

Two-Stage Location-Allocation Optimization Problem Consider a mobility sharing service to be launched in a city over a compact region Y in space R^d (*e.g.*, $d = 2$ for planar models), where the demand distribution on Y is given by a continuous measure ν_Y . Given a set of potential service locations $\Omega = \{s_1, \dots, s_n\}$, and their associated weights $a(s_i)$, $i = 1, \dots, n$. Assume the facility maintenance and operation cost at location s_i is $g(s_i)$, $i = 1, \dots, n$. The transportation cost from user location $y \in Y$ to facility s_i is $c(y, s_i)$, where $c(y, s_i)$ is a function of the distance between y and s_i thus c is symmetric. For example, if $c(x, y) = \|x - y\|_2$, then it represents the Euclidean distance. Other distance metrics can be applied similarly. The urban designer needs to identify a subset $S \subset \Omega$ of facility locations to offer the service, where $|S| = m$. Given a selection $S \subset \Omega$, let $\mu_S = \sum_{s_i \in S} \lambda_i \delta_{s_i}$ be a discrete measure of S , where $\lambda_i = \frac{a(s_i)}{\sum_{s_i \in S} a(s_i)} =: \mu(s_i)$. That is, there is a mass $\mu(s_i)$ associated with location s_i . Given a section S , let $T: Y \mapsto S$ be a continuous map that allocates $y \in Y$ to stop $T(y) \in S$, and denote $T^{-1}(s_i) = \{y \in Y : T(y) = s_i\}$. The location-allocation problem can then be formulated as the following

Copyright is held by author/owner(s).

two-stage optimization programming:

$$\begin{aligned} \text{(LOC)} \quad \min_{S \subset \Omega} Z(S) &= \alpha L(S) + W(\nu_Y, \mu_S) \quad (1) \\ \text{where} \quad L(S) &= \sum_{i \in S} g(s_i) \end{aligned}$$

$$\begin{aligned} \text{(ALLOC)} \quad W(\nu_Y, \mu_S) &= \min_T \int_Y c(y, T(y)) \nu(y) dy, \\ \text{subject to} \quad \int_{T^{-1}(s_i)} \nu(y) dy &= \mu(s_i), \quad \forall s_i \in S \end{aligned}$$

In the above two-stage optimization program, the first stage focuses on the location problem (LOC) with the objective to find the subset $S \subset \Omega$ of service locations such that the total cost $Z(S)$ is minimized where $Z(S)$ consists of two parts: (1) facility and operational cost at all selected locations; and (2) transportation cost (residents inconvenience) which is a function of the distance from his/her position to a service location.

Solution Methods to (ALLOC) In this paragraph, we discuss how to solve (ALLOC) using first-order optimization methods. As established in the previous subsection, the solution to (ALLOC) can be derived by solving the dual of the semi-discrete OT problem:

$$\sup_{\phi \in R^m} F(\phi; S) = \sum_{s_i \in S} \left(\lambda_i \phi_i + \int_{\text{Lag}_S^{\phi}(i)} [c(s_i, y) - \phi_i] \nu(y) dA(y) \right) \quad (3)$$

which is a concave function and can be resolved by first-order gradient ascent method: $\nabla_{\phi_i} F(\phi; S) = \frac{\partial F}{\partial \phi_i} = \lambda_i - \int_{\text{Lag}_S^{\phi}(i)} d\nu(y) = 0$.

Location Problem as a Metropolis Chain Given n potential stop locations $\Omega = \{s_1, s_2, \dots, s_n\}$, let $\mathbf{x}(S) = (x_1, \dots, x_n)$ be a binary vector associated with a subset $S (\subset \Omega)$, where $x_i = 1$ if $s_i \in S$. Let $\mathcal{S} = \{\mathbf{x} \in \{0, 1\}^n : \sum_{i=1}^n x_i = m\}$. The location problem (LOC) then becomes the following discrete optimization problem: $\min_{\mathbf{x} \in \mathcal{S}} Z(\mathbf{x})$, where $S_{\mathbf{x}} := \{s_i \in \Omega : x_i = 1, i = 1, \dots, n\}$ denotes the subset of locations selected by $\mathbf{x}$. Construct a graph $G(\mathcal{S}, E)$ whose node set is $\mathcal{S}$, and the arcs $(\mathbf{x}, \mathbf{x}')$ denote the neighboring relationship between the nodes. We say $\mathbf{x}$ and $\mathbf{x}'$ are neighbors if by interchanging a "0" position with a "1" position of $\mathbf{x}$, we get $\mathbf{x}'$. For example, $(1, 1, 0, 0)$ and $(1, 0, 1, 0)$ are neighbors. A Markov chain $\{X_n\}$ on graph $G(\mathcal{S}, E)$ can then be constructed as follows. (1). When $X_n = \mathbf{x}$, pick one of $\mathbf{x}$'s neighbors in graph $G(\mathcal{S}, E)$ randomly, say $\mathbf{x}'$. Let $X_{n+1} = \mathbf{x}'$ with probability $\alpha(s, \mathbf{x}')$, and $X_{n+1} = \mathbf{x}$ with probability $1 - \alpha(\mathbf{x}, \mathbf{x}')$. where $\alpha(\mathbf{x}, \mathbf{x}') = \min(\exp(-\frac{Z(\mathbf{x}') - Z(\mathbf{x})}{T}), 1)$, and $T > 0$ is a fixed parameter called *temperature*.

OTSA We are now ready to present our overall algorithm which solves the two-stage location allocation optimization by iteratively solving the semi-discrete Optimal Transport problem and exploring the location graph via Simulated Annealing. The main idea is as follows: first we select an initial set of service locations S_0 , given which the allocation problem (ALLOC) is solved via optimal transport and the associated transportation cost W_0 is obtained. This gives the total cost $Z_0 = L(S_0) + W_0$. In the next iteration, we follow the Metropolis algorithm with the cooling strategy of geometric decay and select a new set of stop locations S_1 , again a new cost Z_1 would be obtained, and we record the smaller cost by comparison of Z_0 and Z_1 . Iteratively, we can

finally reach a global minimal total cost Z^* .

Initialization Strategy The initialization is to select m service locations S_0 out of n candidate sites. We propose a clustering-based initialization strategy to generate m clusters with respective m centroids. We then use a greedy strategy to match m locations with the m centroids so that the total distance between the matched points is minimized. We also explore other initialization strategies based on the submodular surrogates in second stage.

3. EXPERIMENT

In this study, we apply OTSA to a real dataset, a publicly available dataset of taxi trips in New York City (NYC), which records a large number of taxi trips (over 100M a year). We retrieve taxi records in Manhattan during the period from January 1, 2016 to June 30, 2016. From these data, we extract the pickup requests (pickup-latitude, pickup-longitude). Suppose we plan to launch a fictitious micro-transit service in Manhattan Area of NYC. To model the demand, we extract around 2.2M taxi trips in Manhattan Area of NYC between 01/01/2016 and 06/31/2016. The streets of Manhattan are extracted from the shapefile of NYC. Using the GIS information, a total of 596 candidate stop locations are selected along the streets distributed in Manhattan area of NYC. The goal is to select 100 stops out of the 596 candidate sites and allocate the demand to stops. The cost function used in the semi-discrete OT is the geodesic distance (haversine) between geographical coordinates in terms of latitude and longitude, also the haversine distance is the shortest distance over the earth's surface. We assume that the facility maintenance cost is constant for all sites. Our experiments show that our algorithms will converge and can scale to city-level demand and provide robust solutions for shared mobility services.

4. CONCLUSION

In this work, we study the city-scale service location selection and users allocation. When the demand is discrete and dense, the problem is NP-hard and computationally challenging. We propose to consider the continuous approximation of the demand density function and exploiting the optimal transport theory to solve the allocation problem. We further develop an integrated framework that combines optimal transport with simulated annealing. Our numerical experiments demonstrate that our algorithm converges to the optimal solution and can handle the urban design problem at city scale.

5. REFERENCES

- [1] Sharon Feigon, Colin Murphy, and Taylor McAdam. Private transit: Existing services and emerging directions. Technical report, 2018.
- [2] Nimrod Megiddo and Kenneth J Supowit. On the complexity of some common geometric location problems. *SIAM journal on computing*, 13(1):182–196, 1984.
- [3] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.
- [4] Yaghout Nourani and Bjarne Andresen. A comparison of simulated annealing cooling strategies. *Journal of Physics A: Mathematical and General*, 31(41):8373, 1998.

Learning payoffs while routing in skill-based queues

Sanne van Kempen,^{*1}

Jaron Sanders,¹

Fiona Sloothak,¹

Maarten G. Wolf²

¹ Eindhoven University of Technology

² Koninklijke KPN N.V.

1. INTRODUCTION

Skill-based queues are used in, for example, data centers, service systems, and healthcare applications. In such systems, customers must be handled by servers with the right skill set. An illustration is in Figure 1. Maximizing the quality of service is essential for system performance.

Canonical routing policies including First-Come-First-Served (FCFS) Assign-the-Longest-Idle-Server (ALIS) and Serve-the-Longest-Queue are typically static by design. Although these policies are easy to implement, they do not account for long-term performance tradeoffs or stability constraints. Moreover, they offer limited robustness and adaptability against changes in environment. Other policies such as the $c\mu$ rule and MaxWeight scheduling rely on system parameters which might *a priori* be unknown in real-world applications. This limits their applicability. To address these challenges, there is growing interest in data-driven control policies that can deal with unknown system parameters.

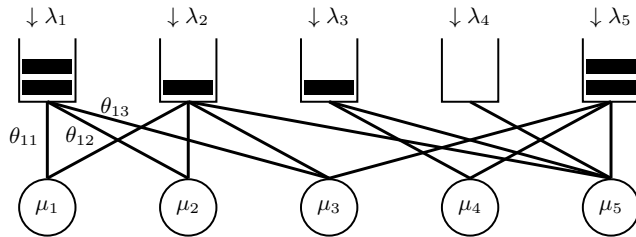


Figure 1: A skill-based queue with customer-server compatibility lines. Here, λ_i and μ_i denote the arrival and service rates and θ_{ij} are payoff parameters.

In [1], we construct a machine learning algorithm called UCB-Queue Routing (UCB-QR) that learns the payoff parameters adaptively while maximizing total payoff. We show that UCB-QR is asymptotically optimal, up to logarithmic terms, by deriving regret bounds. UCB-QR leverages the basic feasible solutions of a static linear program (LP) as the action space. We find the optimal action in a similar fashion as in Multi-Armed Bandits (MABs) [2]. To tackle the exploration-exploitation tradeoff, we use Upper Confidence Bounds (UCBs) for the payoff parameters [3]. The

regret analysis overcomes the challenging interplay between queueing and learning by analyzing the convergence of the queue length process to its stationary behavior.

In [4], we conduct a case study using a real-world data set and demonstrate the performance of UCB-QR. The experiments show that UCB-QR learns efficiently, adapts to changing environments, and outperforms static benchmark policies. This indicates its potential for live implementation.

2. THE UCB-QR ROUTING ALGORITHM

Consider a queueing system with customer types $\mathcal{I} = \{1, \dots, I\}$, servers $\mathcal{J} = \{1, \dots, J\}$, and compatibility lines $\mathcal{L} = \{(ij) : i \in \mathcal{I}, j \in \mathcal{J}\}$. Type- i customers arrive according to a Poisson process with rate $\lambda_i > 0$. Service times at server j are exponentially distributed with rate $\mu_j > 0$. Upon departure of a type- i customer at server j , a random payoff $Y_{ij} \sim \text{Ber}(\theta_{ij})$ is obtained. Critically, we assume that θ_{ij} is unknown to the operator. Denote the set of servers that are compatible with customer type i by $\mathcal{S}_i = \{j \in \mathcal{J} : (ij) \in \mathcal{L}\}$, and similarly $\mathcal{C}_j = \{i \in \mathcal{I} : (ij) \in \mathcal{L}\}$ for the reverse.

Consider the following related optimal transport LP for $\varepsilon > 0$ sufficiently small (see [1, Lemma 3]) and $\mu_j^\varepsilon := \mu_j - \varepsilon$:

$$\text{LP}(\theta, \varepsilon) : \max_{x \geq 0} \sum_{(ij) \in \mathcal{L}} \theta_{ij} x_{ij}, \quad (1a)$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{S}_i} x_{ij} = \lambda_i, \forall i, \quad \sum_{i \in \mathcal{C}_j} x_{ij} \leq \mu_j^\varepsilon, \forall j. \quad (1b)$$

In short, UCB-QR works as follows. UCB-QR uses the set of basic feasible solutions of $\text{LP}(\theta, \varepsilon)$ as action space, denoted by $\mathcal{A}$. Each $a \in \mathcal{A}$ is associated with a unique set of rates $\{x_{ij}^a\}_{(ij) \in \mathcal{L}}$ satisfying (1b). At the start of episode k , UCB-QR chooses action a with maximal (estimated) reward $\sum_{(ij) \in \mathcal{L}} x_{ij}^a U_{ij}(k)$. Here, $U_{ij}(k) = \hat{\theta}_{ij}(k) + \sqrt{\ln(k)/T_{ij}(k)}$ is the UCB estimator of payoff θ_{ij} , determined by the empirical estimator $\hat{\theta}_{ij}(k)$ and the number of collected samples $T_{ij}(k)$. During the episode, customers are routed according to the rates x_{ij}^a of the chosen action. At the end of the episode, UCB indices are updated using the collected samples.

2.1 Regret Analysis

In [1], we provide an upper bound for the regret for UCB-QR, $R^\theta(t) := \sum_{(ij) \in \mathcal{L}} \theta_{ij} (tx_{ij}^\theta - \mathbb{E}^\theta(D_{ij}(t)))$, where $D_{ij}(t)$ is the number of type- (ij) departures up to time t . The following result summarizes [1, Thm. 2]. The bound depends

^{*}s.f.m.v.kempen@tue.nl

on the minimum nonzero rate $x_{\min} := \min_{a \in \mathcal{A}} \{x_{ij}^a \neq 0\}$ and a constant $\alpha_0 > 0$ determined by system parameters.

RESULT 1. Let $\beta \in \mathbb{R}_{>1}$, $\alpha \geq \alpha_0$, and $H_0 \geq \min(4/x_{\min}, 1)$. Let the episode lengths H_k satisfy

$$H_k = \tau_k + H_0 \quad \text{with} \quad \tau_k = \alpha \ln^\beta(2|\mathcal{J}|k). \quad (2)$$

Then, the regret $R^\theta(t)$ of UCB-QR satisfies

$$\lim_{t \rightarrow \infty} \frac{R^\theta(t)}{\alpha \ln^{2\beta}(t)|\mathcal{A}|} \leq \sum_{(k\ell) \in O^c(\theta)} \phi_{k\ell}^\theta \mu_\ell^\varepsilon + \sum_{j \in \mathcal{J}} w_j^\theta \mu_j^\varepsilon. \quad (3)$$

$O^c(\theta)$ is the set of suboptimal lines with no mass in the solution of (1), $\phi_{ij}^\theta \geq 0$ is the suboptimality gap of line (ij) , and $w_j^\theta \geq 0$ is the solution of the dual program of (1).

The regret analysis relies on concentration inequalities that bound the probability that the empirical average $\hat{\theta}_{ij}(k)$ deviates far from its mean θ_{ij} . However, the analysis differs from the standard MAB techniques since the routing decisions and therefore the number of samples $T_{ij}(k)$ are subject to queueing dynamics. Because a new action is selected in every episode, the queueing dynamics may vary from one to the next. We overcome this challenge by adding a ‘warm-up’ phase τ_k to each episode (see Figure 2). τ_k increases in k (see (2)) so that the probability that the queue length process is not stationary before τ_k decreases sufficiently fast.

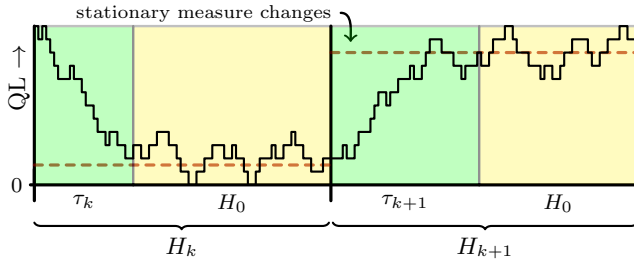


Figure 2: One possible queue length realization. Each episode has a warm-up and stationary phase.

The regret accumulation in (3) is slightly worse than the lower bound in [1, Thm. 1], implying that UCB-QR is asymptotically optimal up to logarithmic factors. This lower bound is summarized in the following result. In (4), $K(\theta, ij)$ is the minimum Kullback-Leibler divergence between θ_{ij} and the set of parameter values that make (ij) optimal.

RESULT 2. For any consistent and stabilizing policy π and for any $\theta \in \mathbb{R}_{\geq 0}^L$,

$$\lim_{t \rightarrow \infty} \frac{R_\pi^\theta(t)}{\ln(t)} \geq \sum_{(k\ell) \in O^c(\theta)} \frac{\phi_{k\ell}^\theta}{K(\theta, k\ell)} > 0. \quad (4)$$

3. CASE STUDY ON REAL-WORLD DATA

In [4], we adapt UCB-QR for implementation in practice. This includes reducing the action space and dealing with temporary system instability and time-varying arrival and service rates. Moreover, we simultaneously take into account other objectives such as queue length minimization and fairness. The simulation uses data from a U.S. call center from Technion laboratory for Service Enterprise Engineering.

We demonstrate that the modified UCB-QR routing algorithm can be implemented in a large, realistic queueing system, even when the environment is subject to change. The algorithm accumulates high payoff while learning the payoff parameters and estimating the (time-varying) arrival and service rates. We show that the algorithm outperforms the benchmark policies in terms of payoff accumulation, including a ‘greedy’ policy and a variant of the $c\mu$ rule that is modified for payoff maximization. In one scenario, our algorithm closes the optimality gap by over 70% when compared to naive random routing (see Figure 3). By also developing a new heuristic routing rule called UCB-QR-Tree (see [4, Algorithm 3]), customer waiting times are reduced, as shown in Figure 4. Lastly, we show that UCB-QR is not computationally heavy: runtime is comparable with those of other routing policies like FCFS—ALIS and random routing.

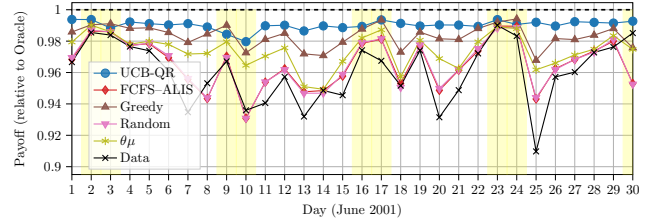


Figure 3: Average payoff relative to the Oracle policy based on June 2001. Weekends are highlighted.

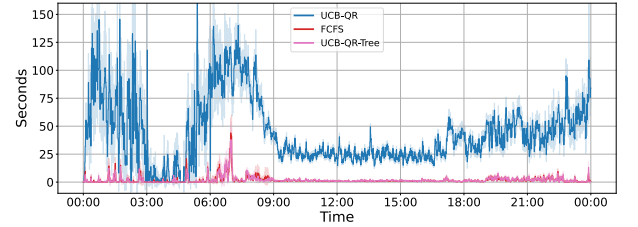


Figure 4: Customer waiting times of UCB-QR-Tree are comparable to those of FCFS.

In summary, [1] and [4] provide theoretical guarantees and show practical effectiveness of UCB-QR.

References

- [1] S. van Kempen, J. Sanders, F. Sloothaak, and M. G. Wolf, ‘Learning payoffs while routing in skill-based queues’, 2024. arXiv: [2412.10168](#).
- [2] T. L. Lai and H. Robbins, ‘Asymptotically efficient adaptive allocation rules’, *Advances in applied mathematics*, 1985.
- [3] P. Auer, N. Cesa-Bianchi, and P. Fischer, ‘Finite-time analysis of the multiarmed bandit problem’, *Machine learning*, 2002.
- [4] S. van Kempen, J. Sanders, F. Sloothaak, and M. G. Wolf, ‘Demonstration of effective UCB-based routing in skill-based queues on real-world data’, 2025. arXiv: [2506.20543](#).

This work is part of *Valuable AI*, a research collaboration between the Eindhoven University of Technology and the Koninklijke KPN. N.V.

join today!

SIGMETRICS & ACM

www.acm.org/sigmetrics

www.acm.org

The **ACM Special Interest Group on Measurement and Evaluation** (SIGMETRICS) fosters research in performance analysis techniques as well as the advanced and innovative use of known methods and tools, seeking a balance between theoretical, methodological, and practical issues. Members' interests typically include advancing the state-of-the-art, as well as applying new performance evaluation tools and techniques in practice. SIGMETRICS sponsors an annual conference. The quarterly newsletter, *Performance Evaluation Review*, includes the annual conference proceedings.

The **Association for Computing Machinery** (ACM) is an educational and scientific computing society which works to advance computing as a science and a profession. Benefits include subscriptions to *Communications of the ACM*, *MemberNet*, *TechNews* and *CareerNews*, full and unlimited access to online courses and books, discounts on conferences and the option to subscribe to the ACM Digital Library.

- ☐ SIGMETRICS (ACM Member) \$ 20
- ☐ SIGMETRICS (ACM Student Member & Non-ACM Student Member) \$ 10
- ☐ SIGMETRICS (Non-ACM Member) \$ 20
- ☐ ACM Professional Membership (\$99) & SIGMETRICS (\$20) \$ 119
- ☐ ACM Professional Membership (\$99) & SIGMETRICS (\$20) & ACM Digital Library (\$99) \$ 218
- ☐ ACM Student Membership (\$19) & SIGMETRICS (\$10) \$ 29
- ☐ *Performance Evaluation Review* only \$ 25
- ☐ Expedited Air for *Performance Evaluation Review* (outside N. America) \$ 11

payment information

Name _____
ACM Member # _____
Mailing Address _____

City/State/Province _____
ZIP/Postal Code/Country _____
Email _____
Mobile Phone _____
Fax _____

Credit Card Type: ☐ AMEX ☐ VISA ☐ MC
Credit Card # _____
Exp. Date _____
Signature _____

Make check or money order payable to ACM, Inc

ACM accepts U.S. dollars or equivalent in foreign currency. Prices include surface delivery charge. Expedited Air Service, which is a partial air freight delivery service, is available outside North America. Contact ACM for more information.

Mailing List Restriction

ACM occasionally makes its mailing list available to computer-related organizations, educational institutions and sister societies. All email addresses remain strictly confidential. Check one of the following if you wish to restrict the use of your name:

- ☐ ACM announcements only
- ☐ ACM and other sister society announcements
- ☐ ACM subscription and renewal notices only

Questions? Contact:

phone: 212-626-0500

fax: 212-944-1318

email: acmhelp@acm.org

Remit to:

ACM
General Post Office
P.O. Box 30777
New York, NY 10087-0777



Association for
Computing Machinery

www.acm.org/joinsigs

SIGAPP

Advancing Computing as a Science & Profession

TABLE OF CONTENTS (continued)

Session 4: Approximation and Numerical Methods

- 36** On Efficient Solutions of General Structured Markov Processes in Quantum Computational Environments *V. Kalantzis, M. Squillante, S. Ubaru*
- 38** Comparing approximations for the ASIP tandem queue *W. Geelen, M. Vlasiou, Y. Yeager*
- 40** On Mixed-Precision Iterative Methods and Analysis for Nearly Completely Decomposable Markov Processes *V. Kalantzis, M. Squillante, C. Wu*

Session 2.A: Queueing Systems and Scheduling Policies II (Short Papers)

- 42** Size-Aware Dispatching to Fluid Queues *R. Xie, E. Hyttiä, R. Richter*
- 49** Large-scale analysis of load-balancing policies for free-floating car-sharing models *C. Fricker, H. Mohamed, A. Rigonat*
- 57** On the geometry of the stability regions of randomly modulated queueing systems *N. Soprano-Loto, U. Ayesta, I. Verloop*
- 62** Optimal service and switching strategies for polling systems *P. Joshi, M. Boon, E. Winands*
- 67** Markov-modulated dependencies for a multiplicative Lindley-type recursion *I. Dimitriou*

Session 2.B: Learning, Game Theory, and Distributed Systems (Short Papers)

- 72** In situ fine-tuning of in silico trained Optical Neural Networks *G. Kosmella, R. Stabile, J. Sanders*
- 79** When Q-Learning fails: unstable behavior for infinite state spaces *U. Ayesta, S. Foss, M. Jonckheere, V. Puricelli*
- 84** Alert or Noise? Reducing False Positives with Active Behavioral Analysis for Cloud Security *Dikshant, G. Verma*
- 91** Convergence of off-policy TD(0) with linear function approximation for reversible Markov chains *M. Overmars, J. Goseling, R. Boucherie*
- 97** Game of arrivals at a two-queue network with heterogeneous customer routes *A. Bandyopadhyay, S. Juneja*

Session: Posters

- 103** Model-Based Learning of Whittle Index *J. Charles-Rebuffé, N. Gast, B. Gaujal*
- 105** On the Performance of Dynamic Matching Models with Threshold-based Policies *G. Sagastabeitia, E. Anton, J. Doncel*
- 107** Clifford Poisson Point Process for Structured Stochastic Geometry Modeling of 6G Wireless Systems *R. Xu, N. Al-Dhahir, Y. Jiang*
- 109** Subgradient Methods for Nonsmooth Convex Functions with Adversarial Errors *M. Gösgens, B. van Parys*
- 111** Efficiency of an RL-based Algorithm for Models with an Absorbing Set of States *C. Lozano, J. Doncel*
- 113** A Study on Two-thresholded Queue Departure Process and its application to IEEE 802.15 MAC *J. Chen*
- 115** Asymptotically optimal reinforcement learning in Block Markov Decision Processes *T. van Vuren, F. Sloothak, M. Wolf, J. Sanders*
- 117** A Variational Framework for the RLGL Algorithm *K. Avrachenkov, L. Gregoris, N. Litvak*
- 119** Reliable estimation of exceedance probabilities for frequency reserve dimensioning in the electricity grid *D. Mastropietro, A. Zocca*
- 121** Optimal Location and Allocation in Shared Mobility Services via Semi-discrete Optimal Transport and Simulated Annealing *K. Jin, A. Carrel, S. Zhang, C. Xia*
- 123** Learning payoffs while routing in skill-based queues *S. van Kempen, J. Sanders, F. Sloothak, M. Wolf*