Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

An Online Learning Approach to Dynamic Pricing and Capacity Sizing in Service Systems

Xinyun Chen, Yunan Liu, Guiyu Hong

informs.
https://pubsonline.informs.org/journal/opre

OPERATIONS RESEARCH
Articles in Advance, pp. 1–21
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Methods

An Online Learning Approach to Dynamic Pricing and Capacity Sizing in Service Systems

Xinyun Chen,[a] Yunan Liu,[b,*] Guiyu Hong[a]

[a] The Chinese University of Hong Kong, Shenzhen, Guangdong 518172, China; [b] Department of Industrial and Systems Engineering, North Carolina State University, Raleigh, North Carolina 27695-7906

*Corresponding author

Contact: chenxinyun@cuhk.edu.cn, https://orcid.org/0000-0003-1727-0923 (XC); yliu48@ncsu.edu, https://orcid.org/0000-0001-9961-2610 (YL); guiyuhong@link.cuhk.edu.cn, https://orcid.org/0009-0006-9616-318X (GH)

Received: September 6, 2020
Revised: January 29, 2022; September 6, 2022; April 9, 2023
Accepted: April 29, 2023
Published Online in Articles in Advance: June 12, 2023

Area of Review: Stochastic Modeling

https://doi.org/10.1287/opre.2020.612

Copyright: © 2023 INFORMS

Abstract. We study a dynamic pricing and capacity sizing problem in a $GI/GI/1$ queue, in which the service provider's objective is to obtain the optimal service fee p and service capacity μ so as to maximize the cumulative expected profit (the service revenue minus the staffing cost and delay penalty). Because of the complex nature of the queueing dynamics, such a problem has no analytic solution so that previous research often resorts to heavy-traffic analysis in which both the arrival and service rates are sent to infinity. In this work, we propose an online learning framework designed for solving this problem that does not require the system's scale to increase. Our framework is dubbed gradient-based online learning in queue (GOLiQ). GOLiQ organizes the time horizon into successive operational cycles and prescribes an efficient procedure to obtain improved pricing and staffing policies in each cycle using data collected in previous cycles. Data here include the number of customer arrivals, waiting times, and the server's busy times. The ingenuity of this approach lies in its online nature, which allows the service provider to do better by interacting with the environment. Effectiveness of GOLiQ is substantiated by (i) theoretical results, including the algorithm convergence and regret analysis (with a logarithmic regret bound), and (ii) engineering confirmation via simulation experiments of a variety of representative $GI/GI/1$ queues.

Funding: The first author acknowledges support [Grants NSFC72171205, NSFC11901493, and RCYX20210609103124047].
Supplemental Material: The e-companion is available at https://doi.org/10.1287/opre.2020.612.

Keywords: online learning in queues • service systems • capacity planning • staffing • pricing in service systems

1. Introduction

1.1. Problem Statement and Methodology

We study a service queueing model in which the service provider manages congestion and revenue by dynamically adjusting the price and service capacity. Specifically, we consider a $GI/GI/1$ queue in which the demand for service is $\lambda(p)$ per unit of time when each customer is charged by a service fee p; the cost for providing service capacity μ is $c(\mu)$, and a holding cost h_0 incurs per job per unit of time. By choosing the appropriate service fee p and capacity μ, the service provider aims to maximize the net profit, which is the service fee minus the staffing cost and penalty of congestion, that is,

$$\max_{\mu,p} \ \mathcal{P}(\mu,p) \equiv p\lambda(p) - c(\mu) - h_0\mathbb{E}[Q_\infty(\mu,p)], \qquad (1)$$

where $Q_\infty(\mu,p)$ is the steady-state queue length under service rate μ and price p.

Problems in this framework have a long history; see, for example, Kumar and Randhawa (2010), Lee and Ward (2014, 2019), Maglaras and Zeevi (2003), Nair et al. (2016), Kim and Randhawa (2018), and the references therein. Because of the complex nature of the queueing dynamics, exact analysis is challenging and often unavailable (computation of the optimal dynamic pricing and staffing rules is not straightforward even for the Markovian $M/M/1$ queue; Ata and Shneorson 2006). Therefore, researchers resort to heavy-traffic analysis to approximately obtain performance evaluation and optimization results. Commonly adopted heavy-traffic regimes require sending the arrival rate and service capacity (service rate or number of servers) to ∞. Although heavy-traffic analysis provides satisfactory results for large-scale queueing systems, approximation formulas based on heavy-traffic limits often become inaccurate as the system scale decreases.

In this paper, we propose an online learning framework designed for solving Problem (1). According to our online learning algorithm, the $GI/GI/1$ queue is operated in successive cycles, in which, in each cycle, the service provider's decisions on the service fee p and service capacity μ, deemed the best by far, are obtained using the system's data collected in previous operational cycles. Data hereby include (i) the number of customers who join for service, (ii) customer waiting times, and (iii) the server's busy time, all of which are easy to collect. Newly generated data, which represent the response from the (random and complex) environment to the present operational decisions, are used to obtain improved pricing and staffing policies in the next cycle. In this way, the service provider can dynamically interact with the environment so that the operational decisions can evolve and eventually approach the optimal solution.

At the beginning of each cycle k, the service provider's decisions (p_k, μ_k) are computed and enforced throughout the cycle. At the heart of our procedure for computing (p_k, μ_k) is to obtain a sufficiently accurate estimator H_{k-1} for the gradient of the objective function of (1), using past experience. Specifically, our online algorithm updates (p_k, μ_k) according to

$$(\mu_k, p_k) \longleftarrow (\mu_{k-1}, p_{k-1}) + \eta_{k-1} H_{k-1},$$

where η_k is the updating step size for cycle k. We call this algorithm gradient-based online learning in queue (GOLiQ).

Besides showing that, under our online learning scheme, the decisions in cycle k, (μ_k, p_k) converge to the optimal solutions (μ^*, p^*) as k increases, we quantify the effectiveness of GOLiQ by computing the regret—the cumulative loss of profit because of the suboptimality of (μ_k, p_k), namely, the maximum profit under the (unknown) optimal strategy minus the expected profit earned under the online algorithm over time. When GOLiQ's hyperparameters are chosen optimally, we show that our regret bound is logarithmic so that the service provider with any initial pricing and staffing policy (μ_0, p_0) quickly learns the optimal solutions without losing much profit in the learning process.

1.2. Advantages, Challenges, and Contributions

In what follows, we first discuss the general advantages of the online learning approach by contrasting with heavy-traffic methods; we next explain the key challenges we face in the development of online learning algorithms for queueing systems.

1.2.1. Online Learning vs. Heavy-Traffic Method. First, heavy-traffic solutions are derived from approximating models that arise as the system scale approaches infinity, so the fidelity of the solutions is sensitive to the system scale. Unlike heavy-traffic methods, online learning approaches do not require any asymptotic scaling, so they can treat service systems at any scale (small or large). Second, heavy-traffic approaches usually require the knowledge of certain distributional information a priori (e.g., moments and distribution functions of service times), which serve as critical input parameters for the heavy-traffic models. On the other hand, online learning methods require information of this kind to a lesser extent. Although certain distribution information can help fine-tune parameters of online algorithms, it is less crucial to algorithm design and implementation. So, in this sense, the dependence on the distributional information is weaker than that of heavy-traffic analysis. Finally, online learning is advantageous when the underlining problem focuses on performance optimization in the long run. Heavy-traffic analysis gives approximate solutions that are static, and in a longer time frame, the performance discrepancy (relative to the true optimal reward) should grow linearly as time increases. But online learning is a dynamic evolution, and its data-driven nature enables it to constantly produce improved solutions that eventually reach optimality. In addition, heavy-traffic solutions require the establishment of heavy-traffic limit theorems and careful analysis of the dynamics of the limit processes (e.g., fluid and diffusion). Both steps can be quite sophisticated in general. See Remark 11 and Section EC.1 for more detailed discussions; also see Section 6.3 for numerical evidence.

1.2.2. Challenges of Online Learning in Queueing Systems. Online learning in queues is by no means an easy extension of online learning in other domains; its theoretical development has to account for the unique features in queueing systems. A crucial step is to develop effective ways to control the nonstationary error that arises at the beginning of every cycle because of the policy update. Toward this, we develop a new regret analysis framework for the transient queueing performance that not only helps establish desired regret bounds for the specific online $GI/GI/1$ algorithm, but may also be used to develop online learning methods for other queueing models (see Section 4). Another challenge we have to address here is to devise a convenient gradient estimator for the online learning algorithm (essentially, an estimator for the gradient of $\mathbb{E}[Q_\infty(\mu, p)]$). The estimator should have a negligible bias to warrant a quick convergence of the algorithm, and at the same time, its computation (using previous data) should be sufficiently straightforward to ensure the ease of implementation (the detailed gradient estimator of GOLiQ for the $GI/GI/1$ system is given in Section 5).

1.2.3. Main Contributions. We summarize our contributions.

- To the best of our knowledge, the present work is the first to develop an online learning framework for joint pricing and staffing in a queueing system with logarithmic regret bound in the total number of customers served (Theorem 3). Because of the complex nature of queueing systems, previous research often resorts to asymptotic heavy-traffic analysis to approximately solve for desired operational decisions. The ingenuity of our online learning method lies in the ability to obtain the optimal solutions without needing the system scale (e.g., arrival and service rates) to grow large. The other appeal of our method is its robustness, especially in its weaker dependence on the distributions of service and arrival times.

- A critical step in the regret analysis is the treatment of the transient system dynamics because, when improved operational decisions are obtained and implemented at the beginning of a new period, the queueing performance shifts away from previously established steady-state level. Toward this, we develop a new way to treat and bound the transient queueing performance in the regret analysis of our online learning algorithm (Theorem 1). Bounding the transient error also guarantees convergence of the stochastic gradient descent (SGD) iteration (Theorem 2). Compared with previous literature (e.g., the regret bound is $O(T^{2/3})$ in Huh et al. (2009)), our analysis of the regret resulting from nonstationarity gives a much tighter logarithmic bound. In addition, the regret analysis in the present paper may be extended to other queueing systems that share similar properties to $GI/GI/1$.
- Supplementing the theoretical results of our regret bound, we evaluate the practical effectiveness of our method by conducting comprehensive numerical experiments. Our simulations draw the following two main conclusions. First, our method is robust in several dimensions: (i) GOLiQ exhibits convincing performance for $GI/GI/1$ queues having representative arrival and service distributions; (ii) GOLiQ remains effective even when certain theoretical assumptions are relaxed. Furthermore, in order to clearly highlight the advantages of our online learning approach relative to the previous results of heavy-traffic limits, we provide a careful performance comparison of these two methods. We show that GOLiQ is more effective in any one of the following three cases: the system scale is not too large, staffing cost is high, or service times are more variable.

1.3. Organization of the Paper

In Section 2, we review the related literature. In Section 3, we introduce the model assumptions and provide an outline of our online learning algorithm. In Section 4, we conduct the regret analysis for GOLiQ by separately treating the regret of nonstationarity, the part of regret arising from the transient system dynamics, and the regret of suboptimality, the part originating from the errors because of suboptimal pricing and staffing decisions. In Section 5, we give the detailed description of GOLiQ and establish a logarithmic regret bound by appropriately selecting our algorithm parameters. In Section 6, we conduct numerical experiments to confirm the effectiveness and robustness of GOLiQ. We conclude in Section 7. In the online e-companion, we give all technical proofs and provide additional numerical examples.

2. Related Literature

The present paper is related to the following three streams of literature.

2.1. Pricing and Capacity Sizing in Queues

There is a rich literature on pricing and capacity sizing in service systems under different settings. Maglaras and Zeevi (2003) study a pricing and capacity sizing problem in a processor sharing queue motivated by internet applications; Kumar and Randhawa (2010) consider a single-server system with nonlinear delay costs; Nair et al. (2016) study $M/M/1$ and $M/M/k$ systems with network effect among customers; Kim and Randhawa (2018) consider a dynamic pricing problem in a single-server system. The specific Problem (1) we consider here is most closely related to Lee and Ward (2014), that is, joint pricing and capacity sizing for the $GI/GI/1$ queue. Later, the authors extend their results to the $GI/GI/1+G$ model with customer abandonment in Lee and Ward (2019). As there is usually no closed-form solution for the optimal strategy or equilibrium, asymptotic analysis is adopted under large-market assumptions. In detail, their analysis is rooted in a deterministic static planning problem that requires both the service capacity and the demand rate to scale to infinity. Most of the papers conclude that the heavy-traffic regime is economically optimal. (There are some exceptions in which the heavy-traffic regime is not optimal; for example, Kumar and Randhawa (2010) show that an agent is forced to decrease its utilization if the delay cost is concave.) Our algorithm is motivated by the pricing and capacity sizing problem for service systems; however, as explained previously, our methodology is very different from the asymptotic analysis used in these papers.

2.2. Reinforcement Learning (RL) for Queueing Systems

Our paper is also related to a small but growing literature on RL for queueing systems. Dai and Gluzman (2021) study an actor–critic algorithm for queueing networks. Liu et al. (2019) and Shah et al. (2020) develop RL techniques to treat the unboundedness of the state space of queueing systems. Jia et al. (2021) study a price-based revenue management problem in an $M/M/c$ queue with a discrete price space; their methodology draws from the multiarmed bandit framework (with each price treated as an "arm"). Krishnasamy et al. (2021) develop bandit methods for scheduling problem in a multiserver queue with unknown service rates. Our work draws distinction from the aforementioned literature in two dimensions. To the best of our knowledge, we are the first to develop an online learning method for joint pricing and capacity sizing in queue. In addition, our method applies to settings of continuous decision variables. Compared with the more general RL literature, our algorithm design and regret analysis take advantage of the specific queueing system structure so as to establish tight regret bounds and more accurate control of the convergence rate. In some sense, the algorithm developed in the present paper may be viewed as a version of the policy gradient method, a special class of RL methods (Sutton and Barto 2018); see Remark 2 for detailed discussions.

2.3. Stochastic Gradient Decent Algorithms

In general, our algorithm falls into the broad class of SGD methods. There are some early papers on SGD algorithms for steady-state performance of queues (see Fu 1990, Chong and Ramadge 1993, L'Ecuyer and Glynn 1994, L'Ecuyer et al. 1994, and the references therein). In particular, these papers establish convergence results of SGD algorithms for capacity sizing problems with a variety of gradient estimating designs. In this paper, we consider a more general setting in which the price is also optimized jointly with the service capacity. Besides, in order to establish theoretical bounds for the regret, we conduct a careful analysis on the convergence rate of the algorithm and provide an explicit guidance for the optimal choice of algorithm parameters, which is not discussed in this early literature. Our algorithm design and analysis are also related to the online learning methods in recent inventory management literature (Burnetas and Smith 2000, Huh et al. 2009, Huh and Rusmevichientong 2013, Zhang et al. 2020, Yuan et al. 2021). Among these papers, our work is perhaps most closely related to Huh et al. (2009), in which the authors develop an SGD-based learning method for an inventory model with a bounded replenishment lead time. Still, because of the unique natures of queueing models, we develop a new regret analysis framework as explain in detail in Section 1.2.3.

3. Problem Setting and Algorithm Outline

In Section 3.1, we describe the queueing model and technical assumptions. In Section 3.2, we provide a general outline of GOLiQ. Finally, in Section 3.3, we conduct preliminary analysis of the queueing performance under GOLiQ.

3.1. Model and Assumptions

We study a $GI/GI/1$ queueing system having customer arrivals according to a renewal process with generally distributed interarrival times (the first *GI*); independent and identically distributed (i.i.d.) service times following a general distribution (the second *GI*); and a single server that provides service under the first-in, first-out discipline. Each customer, upon joining the queue, is charged by the service provider a fee $p > 0$. The demand arrival rate (per time unit) depends on the service fee p and is denoted as $\lambda(p)$. To maintain a service rate μ, the service provider continuously incurs a staffing cost at a rate $c(\mu)$ per time unit.

For $\mu \in [\underline{\mu}, \overline{\mu}]$ and $p \in [\underline{p}, \overline{p}]$, the service provider's goal is to determine the optimal service fee p^* and service capacity μ^* with the objective of maximizing the steady-state expected profit (1) or, equivalently, minimizing the objective function $f(\mu, p)$ as follows:

$$\min_{(\mu,p)\in\mathcal{B}} f(\mu,p) \equiv h_0\mathbb{E}[Q_\infty(\mu,p)] + c(\mu) - p\lambda(p),$$
$$\mathcal{B} \equiv [\underline{\mu}, \overline{\mu}] \times [\underline{p}, \overline{p}]. \tag{2}$$

We impose the following assumptions on this service system throughout the paper.

Assumption 1 (Demand Rate, Staffing Cost, and Uniform Stability)

a. *The arrival rate $\lambda(p)$ is continuously differentiable and nonincreasing in p.*

b. *The staffing cost $c(\mu)$ is continuously differentiable and nondecreasing in μ.*

c. *The lower bounds $\underline{p}$ and $\underline{\mu}$ satisfy that $\lambda(\underline{p}) < \underline{\mu}$ so that the system is uniformly stable for all feasible choices of the pair (μ, p).*

Part (c) of Assumption 1 is commonly used in the literature of SGD methods for queueing models to ensure that the steady-state mean waiting time $\mathbb{E}[W_\infty(\mu,p)]$ is differentiable with respect to model parameters (see Fu 1990, Chong and Ramadge 1993, L'Ecuyer and Glynn 1994, L'Ecuyer et al. 1994; also see theorem 3.2 of Glasserman 1992). In our numerical experiments (see Online Section EC.4.1), we show that our online algorithm remains effective when this assumption is relaxed.

We do not require full knowledge of service and interarrival time distributions. But, in order to develop explicit bounds for the part of the regret resulting from the nonstationarity of the queueing processes, we require both distributions to be light-tailed. Specifically, because the actual service and interarrival times are subject to our pricing and staffing decisions, we model the interarrival and service times by two scaled random sequences $\{U_n/\lambda(p)\}$ and $\{V_n/\mu\}$, where $U_1, U_2, \ldots$ and $V_1, V_2, \ldots$ are two independent i.i.d. sequences of random variables having unit means, that is, $\mathbb{E}[U_n] = \mathbb{E}[V_n] = 1$. We make the following assumptions on U_n and V_n.

Assumption 2 (Light-Tailed Service and Interarrival Times). *There exists a sufficiently small constant $\eta > 0$ such that the moment-generating functions*

$$\mathbb{E}[\exp(\eta V_n)] < \infty \quad \textit{and} \quad \mathbb{E}[\exp(\eta U_n)] < \infty.$$

In addition, there exist constants $0 < \theta < \eta/2\overline{\mu}$, $0 < a < (\underline{\mu} - \lambda(\underline{p}))/(\underline{\mu} + \lambda(\underline{p}))$ and $\gamma > 0$ such that

$$\phi_U(-\theta) < -(1-a)\theta - \gamma \quad \text{and} \quad \phi_V(\theta) < (1+a)\theta - \gamma, \tag{3}$$

where $\phi_V(\theta) \equiv \log \mathbb{E}[\exp(\theta V_n)]$ and $\phi_U(\theta) \equiv \log \mathbb{E}[\exp(\theta U_n)]$ are the cumulant generating functions of V and U.

Note that $\phi'_U(0) = \phi'_V(0) = 1$ as $\mathbb{E}[U] = \mathbb{E}[V] = 1$. Suppose ϕ_U and ϕ_V are smooth around zero; then, we have $\phi_U(-\theta) = -\theta + o(\theta)$ and $\phi_V(\theta) = \theta + o(\theta)$ by Taylor's expansion. This implies that, for any $a > 0$, we can make θ small enough such that $\phi_U(-\theta) < -(1-a)\theta$ and $\phi_V(\theta) < (1+a)\theta$. To obtain the bound in (3), we can simply take $\gamma = \frac{1}{2}\min(-(1-a)\theta - \phi_U(-\theta), (1+a)\theta - \phi_V(\theta)) > 0$. Hence, a sufficient condition that warrants (3) is to require that ϕ_U and ϕ_V be smooth around zero, which is true for many distributions of U and V considered in common queueing models. Assumption 2 is used in our proofs to build an explicit bound for the regret of nonstationarity.

Finally, in order to warrant the convergence of our online learning algorithm, we require a convex structure for the problem in (2), which is common in the SGD literature; see Broadie et al. (2011), Kushner and Yin (2003), and the references therein.

Let $x^* \equiv (\mu^*, p^*)$ and $x \equiv (\mu, p)$. Let $\nabla f(x)$ denote the gradient of a function $f(x)$ and $\|\cdot\|$ denote the Euclidean norm.

Assumption 3 (Convexity and Smoothness). *There exist finite positive constants $K_0 \le 1$ and $K_1 > K_0$ such that, for all $x \in \mathcal{B}$,*

a. $(x - x^*)^T \nabla f(x) \ge K_0 \|x - x^*\|^2$.

b. $\|\nabla f(x)\| \le K_1 \|x - x^*\|$.

Remark 1. Our simulation experiments show that our algorithm works effectively for some representative $GI/GI/1$ queues with conditions in Assumption 3 relaxed; see Section 6 and Online Section EC.4. In addition, we later provide some sufficient conditions for Assumption 3 in the special case of $M/GI/1$ queues in Online Section EC.5.

3.2. Outline of GOLiQ

In general, an SGD algorithm for a minimization problem $\min_x f(x)$ over a compact set $\mathcal{B}$ relies on updating the decision variable via the recursion

$$x_{k+1} = \Pi_{\mathcal{B}}(x_k - \eta_k H_k), \qquad k \ge 1.$$

Here, η_k is the step size, H_k is a random estimator for $\nabla f(x_k)$, x_k is the decision variable by step k, and the projection operator $\Pi_{\mathcal{B}}$ restricts the updated decision in $\mathcal{B}$. For Problem (2), we let $x_k \equiv (\mu_k, p_k)$ represent the service capacity and price at step k. We define

$$B_k \equiv \mathbb{E}[\|\mathbb{E}[H_k - \nabla f(x_k) \,|\, \mathcal{F}_k]\|^2]^{1/2} \quad \text{and} \quad \mathcal{V}_k \equiv \mathbb{E}[\|H_k\|^2], \tag{4}$$

where $\mathcal{F}_k$ is the σ-algebra including all events in the first $k-1$ iterations. Intuitively, B_k measures the bias of the gradient estimator H_k and $\mathcal{V}_k$ measures its variability. As we see later, B_k and $\mathcal{V}_k$ play important roles in designing the algorithm and establishing desired regret bounds.

The standard SGD algorithm iterates in discrete step k. In our setting, however, the queueing system and objective function $f(\mu, p)$ are defined in continuous time (in particular, $Q_\infty(\mu, p)$ is the steady-state queue length observed in continuous time). To facilitate the regret analysis, we first transform the objective function into an expression of customer waiting times that are observed in discrete time. By Little's law, we can rewrite the objective function $f(\mu, p)$ as, for all $(\mu, p) \in \mathcal{B}$,

$$f(\mu, p) = h_0 \lambda(p) \left(\mathbb{E}[W_\infty(\mu, p)] + \frac{1}{\mu} \right) + c(\mu) - p\lambda(p), \tag{5}$$

where $W_\infty(\mu, p)$ is the steady-state waiting time under (μ, p). In each cycle k, our algorithm adopts the average of D_k observed customer waiting times to estimate $\mathbb{E}[W_\infty(\mu, p)]$, where D_k denotes the number of customers that enter service in cycle k (we refer to D_k as the cycle length or sample size of cycle k). But any finite D_k introduces a bias to our gradient estimate H_k. To mitigate the bias resulting from the transient performance of the queueing process, we let the cycle length D_k be increasing in k (in this way the transient bias vanishes eventually). We give the outline of the algorithm as follows.

3.2.1. Outline of GOLiQ.

0. Input: $\{D_k\}$ and $\{\eta_k\}$ for $k = 1, 2, .., L$, initial policy $x_1 = (\mu_1, p_1)$. For $k = 1, 2, \ldots, L$,

1. In the k^{th} cycle, operate the $GI/GI/1$ queue under policy $x_k = (\mu_k, p_k)$ until D_k customers enter service.

2. Collect and use the data (e.g., customer delays) to build an estimator H_k for $\nabla f(\mu_k, p_k)$.

3. Update $x_{k+1} = \Pi_{\mathcal{B}}(x_k - \eta_k H_k)$.

Remark 2 (Exploration vs. Exploitation). The online nature of this algorithm makes it possible to obtain improved decisions by learning from past experience, which is in the spirit of the essential ideas of reinforcement learning in which an agent (hereby the service provider) aims to trade off between exploration (step 1) and exploitation (steps 2 and 3). Effectiveness of the algorithms lies in properly choosing the algorithm parameters and devising an efficient gradient estimator H_k. For example, if D_k is too small, we are unable to generate sufficient data (we do not have much to exploit in order for devising a better policy); if D_k is too large, we incur a higher profit loss because of suboptimality of the policy in use (we do not explore enough for seeking potentially better policies). In particular, GOLiQ may be viewed as a special case of the policy gradient (PG) algorithm (the general idea of PG is to estimate the policy parameters using the gradient of the value function learned via continuous interaction with the system; see, for example, Sutton and Barto (2018)). To put this into perspective, the policy in the present paper is specified by a pair of parameters (μ, p), and in each iteration, we update the policy parameters using an estimated policy gradient H_k learned from data of the queueing model. In the subsequent sections, we give detailed regret analysis that can be used to establish optimal algorithm parameters (Section 4) and develop an efficient gradient estimator (Section 5).

3.3. System Dynamics Under GOLiQ

We explain explicitly the dynamic of the queueing system under GOLiQ with the system starting empty. We first define notations for relevant performance functions. For $k \geq 1$, let T_k be the length of cycle k in the units of time, and let D_k be the total number of customers who enter service in cycle k. For $n = 1, 2, \ldots, D_k$, let W_n^k be the waiting time of the n^{th} customer that enters service in cycle k. We define $W_0^k \equiv W_{D_{k-1}}^{k-1}$. We use the two i.i.d. random sequences V_n^k and U_n^k to construct the service and interarrival times in cycle k, $n = 1, 2, \ldots, D_k$. In particular, V_n^k corresponds to the service time of customer $n-1$, and U_n^k corresponds to the interarrival time between customers $n-1$ and n in cycle k. Let $\lambda_k \equiv \lambda(p_k)$. Finally, we use Q_k to denote the number of existing customers (those who arrive in previous cycles) at the beginning of cycle in k with $Q_1 = 0$. We have $Q_k \geq 1$ for $k \geq 2$ as we explain soon, according to our updating procedure. The detailed dynamics of the queueing system in cycle k is summarized as follows:

- Updating the control policy: In cycle k, we adopt the pricing and staffing policy (p_k, μ_k). The service time of customer $n-1$ in cycle k is $S_n^k = V_n^k/\mu_k$ for $n = 1, \ldots, D_k$. Cycle k ends as soon as a total number of D_k (of which the value is to be determined later) customers have entered service. So customer D_k receives service in cycle $k+1$ (with service time S_1^{k+1}), and the queue leftover consists of at least one customer, that is, $Q_{k+1} \geq 1$ for a new cycle $k+1$, which begins under a new policy (p_{k+1}, μ_{k+1}) as follows:
  - — Service rate: The service rate is updated to μ_{k+1} immediately as the new cycle begins so that all existing customers undergo service times with rate μ_{k+1}.
  - — Service fee: The price remains p_k at the beginning of cycle $k+1$ and evolves to p_{k+1} immediately after the first new customer arrives in the new cycle; we charge this customer with p_k (because its interarrival time is modulated by p_k) and all subsequent customers in cycle $k+1$ with p_{k+1}.
- Leftovers from previous cycles: For $k \geq 2$, at the beginning of cycle k, there are $Q_k - 1$ customers waiting in queue indexed by n from 1 to $Q_k - 1$. The customer who just enters service is indexed by zero. We update the price from p_{k-1} to p_k right after the first new customer (indexed by Q_k) arrives in a new cycle. As a consequence, the prices charged to customers $1, 2, \ldots, Q_k$ are not yet updated to p_k. Denote by p_n^k and $\lambda_n^k \equiv \lambda(p_n^k)$ as the price and arrival rate for customer n in cycle k, respectively, for $1 \leq n \leq Q_k$. The corresponding interarrival time is $\tau_n^k = U_n^k/\lambda_n^k$. In case $Q_{k-1} > D_{k-1}$, some queueing leftovers are customers from earlier cycles. So, here, $p_n^k \in \{p_1, p_2, \ldots, p_{k-1}\}$. In addition, in case $Q_k > D_k$, part of Q_k continues to remain in cycle $k+1$ and we have, for example, $p_1^{k+1} = p_{D_k+1}^k$.
- New arrivals: We denote interarrival times for new customers in cycle k by $\tau_n^k = U_n^k/\lambda_k$ for $n = Q_k + 1, \ldots, D_k$ if $D_k \geq Q_k + 1$. (As soon becomes clear, the case $D_k \leq Q_k$ is a rare event with a negligible probability under appropriate algorithm settings; see Remark 3.)
- Customer delay: Customers' waiting times in cycle k are characterized by the recursions

$$W_n^k = \begin{cases} \left(W_{n-1}^k + \dfrac{V_n^k}{\mu_k} - \dfrac{U_n^k}{\lambda_n^k} \right)^+ \\ \qquad \text{for } 1 \leq n \leq Q_k \wedge D_k; \\ \left(W_{n-1}^k + \dfrac{V_n^k}{\mu_k} - \dfrac{U_n^k}{\lambda_k} \right)^+ \\ \qquad \text{for } (Q_k + 1) \wedge (D_k + 1) \leq n \leq D_k. \end{cases} , W_0^k = W_{D_{k-1}}^{k-1}, \tag{6}$$

where $x^+ \equiv \max\{x, 0\}$.

- Server's busy time: The age of the server's busy time observed by customer n upon arrival, which is the length of time the server has been busy since the last idleness, is given by the recursions

$$X_n^k = \begin{cases} \left(X_{n-1}^k + \dfrac{U_n^k}{\lambda_n^k} \right) \mathbf{1}_{\{W_n^k > 0\}} \\ \qquad \text{for } 1 \leq n \leq Q_k \wedge D_k; \\ \left(X_{n-1}^k + \dfrac{U_n^k}{\lambda_k} \right) \mathbf{1}_{\{W_n^k > 0\}} \\ \qquad \text{for } (Q_k + 1) \wedge (D_k + 1) \leq n \leq D_k. \end{cases} , \quad X_0^k = X_{D_{k-1}}^{k-1}, \tag{7}$$

where the indicator $\mathbf{1}_A$ is one if A occurs and is zero otherwise.

We provide explanations for (6) and (7). First, Recursion (6) simply follows from Lindley's equation. Next, Recursion (7) follows from the fact that, for customer n, if the queue is empty upon its arrival, the observed busy time is simply zero by definition; otherwise, the server must have been busy since the arrival of the previous customer, and therefore, the observed busy time by customer n should extend that of customer $n-1$ by an additional interarrival time. As we see later, both the delay and busy time observed by customers are important ingredients (i.e., data) for building the gradient estimator of the online learning algorithm.

Remark 3 (Clearance of the Leftover Q_k). As explained, Q_k is random and unbounded, whereas in our algorithm design, the cycle length D_k is deterministic. So it is indeed possible the remaining queue content may not be all cleared in cycle k (i.e., $D_k < Q_k$). We see later in the regret analysis that our choice of D_k leads to a small probability of uncleared leftovers, and thus, the impact of the rare event $\{D_k < Q_k\}$ is negligible.

In Figure 1, we further illustrate how the service price and service rate are updated by showing the ordering of all relative events as a new cycle begins. We emphasize that (i) the service rate μ_{k-1} is updated to μ_k immediately when a new cycle k begins, which is triggered as soon as the last one of D_{k-1} customers enters service, and (ii) the service price p_{k-1} is updated to p_k only after the first external arrival occurs in the new cycle k (we honor our previous prices for all customers who arrive in the previous cycle).

We end this section by providing a uniform boundedness result for all relevant queueing functions. This result is used in the next sections to establish desired regret bounds. The proof follows from a stochastic ordering approach and is given in Online Section EC.1.1.

Lemma 1 (Uniform Boundedness of Relevant Queueing Functions). *Under Assumptions 1 and 2, there exists a finite positive constant $M > 0$ such that, for any sequences $(\mu_k, p_k) \in \mathcal{B}$ and $D_k \geq 1$, we have, for all $k \geq 1$, $1 \leq n \leq D_k$ and $1 \leq m \leq 4$, and $\eta > 0$ as defined in Assumption 2,*

$$\mathbb{E}[(W_n^k)^m],\quad \mathbb{E}[(X_n^k)^m],\quad \mathbb{E}[(Q_k)^m],\quad \mathbb{E}[\exp(\eta W_n^k)] \quad \text{and} \quad \mathbb{E}[\exp(\eta Q_k)]$$

are all bounded by M.

4. Regret Analysis

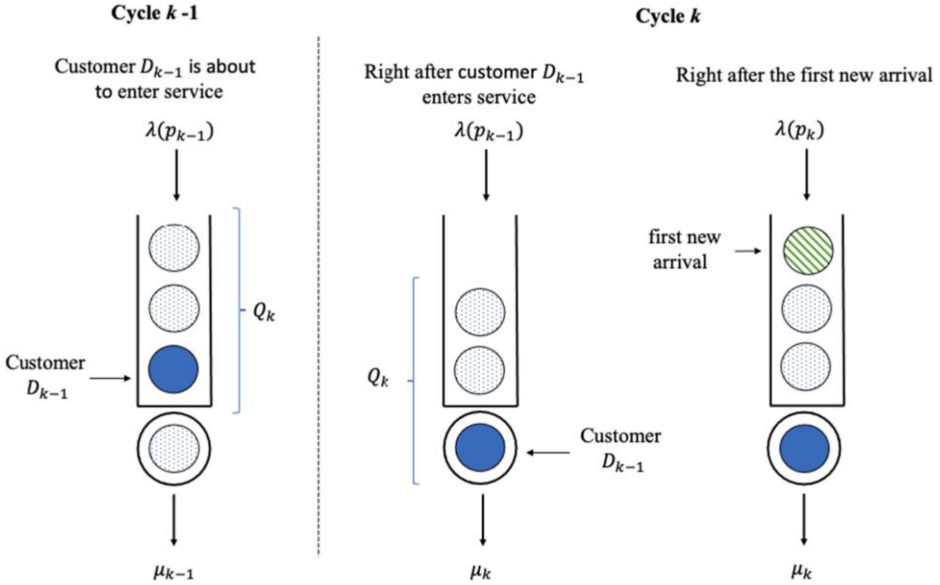
The online learning approach described in Section 3.2 is a data-driven method, and it should continue to generate improved solutions that eventually converge to the true optimal solution as the server's experience accumulates (by serving more and more customers). The performance of GOLiQ is measured by the so-called regret, which can be interpreted as the cost to pay, over the time or the number of samples, for the algorithm to learn the optimal policy. In this section, we give a formal definition of the regret and conduct the regret analysis for our online learning algorithm.

The expected net cost of the queueing system incurred in cycle k is

$$\rho_k = \mathbb{E}\left[\sum_{n=1}^{Q_k \wedge D_k} (h_0(W_n^k + S_n^k) - p_n^k) + \sum_{n=Q_k+1}^{D_k} (h_0(W_n^k + S_n^k) - p_k) + c(\mu_k)T_k\right], \tag{8}$$

where the summation $\sum_{n=Q_k+1}^{D_k} \cdot$ is zero in case $D_k < Q_k + 1$. The total regret accumulated in the first L

Figure 1. (Color online) On the Timing of the Update of p_k and μ_k Under GOLiQ



cycles is

$$R(L) \equiv \sum_{k=1}^{L} R_k, \quad \text{where} \quad R_k \equiv \rho_k - f(\mu^*, p^*)\mathbb{E}[T_k] \tag{9}$$

is regret in cycle k (the expected system cost in cycle k minus the optimal cost).

Remark 4. Following Huh et al. (2009) and Jia et al. (2021), our regret defined in (9) is computed by accumulating the difference between the steady-state maximum profit under (μ^*, p^*) and the expected profit earned under GOLiQ. However, one may find such a definition to be somewhat too demanding; it appears to be more reasonable if we were to benchmark with the nonstationary dynamics under (μ^*, p^*) rather than the steady-state performance. Nevertheless, our numerical studies confirm that the nuance of the two aforementioned regret definitions is negligible. See Online Section EC.4.5.

Separation of regret: To treat the total regret defined in (9), we separate it into two parts: regret of nonstationarity, which quantifies the error resulting from the system's transient performance, and regret of suboptimality, which accounts for the suboptimality error resulting from the present policy. In detail, we write

$$R_k = \underbrace{(\rho_k - \mathbb{E}[f(\mu_k, p_k)T_k])}_{\equiv R_{1,k}} + \underbrace{\mathbb{E}[T_k(f(\mu_k, p_k) - f(\mu^*, p^*))]}_{\equiv R_{2,k}}, \tag{10}$$

so that

$$R(L) = \sum_{k=1}^{L} R_{1,k} + \sum_{k=1}^{L} R_{2,k} \equiv R_1(L) + R_2(L). \tag{11}$$

Intuitively, $R_{1,k}$ measures the performance error resulting from transient queueing dynamics (regret of nonstationarity), whereas $R_{2,k}$ accounts for the suboptimality error of control parameters (μ_k, p_k) (regret of suboptimality).

In what follows, we analyze the two terms $R_1(L)$ and $R_2(L)$ separately. To treat $R_1(L)$, we develop in Section 4.1 a new framework to analyze the transient queueing behavior using the coupling technique (Theorem 1). The development of the theoretical bound for $R_2(L)$ is given in Section 4.2 (Theorem 2). Results in these sections provide convenient conditions that facilitate the convergence and regret bound analysis of our GOLiQ algorithm for GI/GI/1 queues (which is given in Section 5). The road map of the theoretical analysis is depicted in Figure 2.

4.1. Regret of Nonstationarity

In this part, we analyze the transient queueing dynamics, based on which we develop a theoretical upper bound for $R_1(L)$. As we see later in Section 5, this analysis is also essential to bounding the bias B_k and variance $\mathcal{V}_k$ of the gradient estimators for GOLiQ.

A crude $O(L)$ bound: Roughly speaking, because the parameters μ, p and functions $\lambda(\cdot)$, $c(\cdot)$ are all bounded, the regret $R_1(L)$ is in the same order as the transient bias of the waiting time process, that is,

$$R_1(L) \approx \sum_{k=1}^{L} O\left(\sum_{n=1}^{D_k} (\mathbb{E}[W_n(\mu_k, p_k)] - \mathbb{E}[W_\infty(\mu_k, p_k)])\right).$$

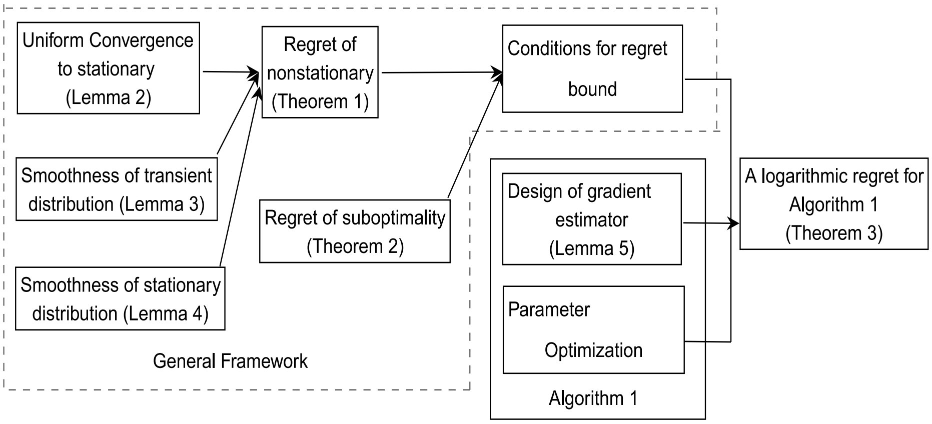
Here, we use $W_\infty(\mu, p)$ to denote the steady-state waiting time of the $GI/GI/1$ queue with parameter $(\mu, p) \in \mathcal{B}$. Under the uniform stability condition (Assumption 1), it is not difficult to show that there exist positive constants $\gamma > 0$ and $K > 0$, independent of k and (μ_k, p_k) such that

$$|\mathbb{E}[W_n^k] - \mathbb{E}[W_\infty(\mu_k, p_k)]| \le e^{-\gamma n} K.$$

Then, as a direct consequence, we have

$$\sum_{n=1}^{D_k} (\mathbb{E}[W_n(\mu_k, p_k)] - \mathbb{E}[W_\infty(\mu_k, p_k)]) \le \frac{K}{1 - e^{-\gamma}} \quad \Rightarrow \quad R_1(L) = O(L).$$

Figure 2. Road Map of Regret Analysis and Algorithm Design


Uniform Convergence to stationary (Lemma 2)
Regret of nonstationary (Theorem 1)
Conditions for regret bound
Smoothness of transient distribution (Lemma 3)
Regret of suboptimality (Theorem 2)
Smoothness of stationary distribution (Lemma 4)
General Framework
Design of gradient estimator (Lemma 5)
Parameter Optimization
Algorithm 1
A logarithmic regret for Algorithm 1 (Theorem 3)

An analogue of this $O(L)$ bound is given by Huh et al. (2009, lemma 11) in an inventory model.

An improved $o(L)$ bound: In the rest of this section, we conduct a more delicate analysis on the transient performance of the queueing system, and our analysis renders a (tighter) sublinear bound $R_1(L) = o(L)$ (of which the exact order depends on the concrete algorithm as we see later).

Theorem 1 (Regret of Nonstationarity). *Suppose that Assumptions 1 and 2 hold. In addition, assume that the following conditions are satisfied for some constant $K_2 > 0$ and $0 < \alpha \le 1$:*

a. $\lceil 6\log(k)/\min(\gamma,\eta)\rceil \le D_k \le K_2 k^{2-\alpha}$.
b. $\mathbb{E}[\|x_k - x_{k+1}\|^2] \le K_2 k^{-2\alpha}$.

Here, the constants η and γ are defined in Assumption 2. Then, there exists a positive constant $K > 0$ such that

$$R_{1,k} \le K \cdot k^{-\alpha}\log(k), \quad k \ge 2 \quad \textit{and}$$

$$R_1(L) \le K \sum_{k=1}^{L} k^{-\alpha}\log(k), \quad L \ge 2. \tag{12}$$

Remark 5. As becomes clear later in Section 5, we obtain a bound $R_1(L) = O(\log(L)^2)$ for Algorithm 1 by validating condition (b) in Theorem 1 with $\alpha = 1$, which is much tighter than the crude $O(L)$ bound. This $O(\log(L)^2)$ bound for $R_1(L)$ is critical to achieving an overall logarithmic regret bound in the total number of served customers. An explicit expression of constant K is given in (EC.3).

4.1.1. Road Map of the Proof of Theorem 1.

Our point of departure in proving Theorem 1 is to decompose $R_{1,k}$ into three terms. We split each cycle into a warm-up period consisting the first $\tilde{d}_k = \lceil 5\log(k)/\min(\gamma,\eta)\rceil < D_k$ customers and the near-stationary period consisting of all remaining customers, in which $\gamma, \eta > 0$ are as defined in Assumption 2. The three parts are transient error in the near-stationary period (I_1), transient error in the warm-up period (I_2), and the remaining error (I_3). The detailed separation is given as

$$\begin{aligned}
R_{1,k} &= \rho_k - \mathbb{E}[f(\mu_k,p_k)T_k] \\
&= \mathbb{E}\Bigg[\sum_{n=1}^{Q_k\wedge D_k}(h_0(W_n^k + S_n^k) - p_n^k) \\
&\quad + \sum_{n=Q_k+1}^{D_k}(h_0(W_n^k + S_n^k) - p_k) + c(\mu_k)T_k - f(\mu_k,p_k)T_k\Bigg] \\
&= \underbrace{h_0\,\mathbb{E}\left[\sum_{n=\tilde{d}_k+1}^{D_k}(W_n^k - w(\mu_k,p_k))\right]}_{\equiv I_1} + \underbrace{h_0\,\mathbb{E}\left[\sum_{n=1}^{\tilde{d}_k}(W_n^k - w(\mu_k,p_k))\right]}_{\equiv I_2} \\
&\quad \underbrace{+\,\mathbb{E}\left[(D_k - \lambda_k T_k)\left(h_0 w(\mu_k,p_k) + \frac{h_0}{\mu_k} - p_k\right)\right] + \mathbb{E}\left[\sum_{n=1}^{Q_k\wedge D_k}(p_k - p_n^k)\right]}_{\equiv I_3}.
\end{aligned}$$

The term $w(\mu,p) \equiv \mathbb{E}[W_\infty(\mu,p)]$ is a function in (μ,p) and equals to the steady-state expected waiting time under parameter $(\mu,p) \in \mathcal{B}$. To prove $R_{1,k} = O(k^{-\alpha}\log(k))$, it suffices to show that $I_i = O(k^{-\alpha}\log(k))$ for $i = 1, 2, 3$. We explain the main ideas of our treatment to I_1, I_2, and I_3:

- I_1: We first show that, after serving $d_k \equiv \lceil 4\log(k)/\min(\gamma,\eta)\rceil < \tilde{d}_k$ customers, with a sufficiently high probability, all Q_k existing customers have left the system and $\{W_n^k : n = d_k, \dots, D_k\}$ follows the dynamic of a GI/GI/1 queue with arrival rate λ_k and service rate μ_k. Then, we show that W_n^k, for $n \ge d_k$, converges exponentially fast to the steady state (Lemma 2). Hence, W_n^k is close to $W_\infty(\mu_k, p_k)$ for $n \ge \tilde{d}_k$, warranting a small transient error I_1.
- I_2: Note that the $\tilde{d}_k$ customers in the warm-up period include those leftovers from previous periods, and their arrival rates λ_n^k are different from λ_k. To control the impact of such difference between λ_n^k and λ_k, we first establish almost sure Lipschitz continuity of waiting times (for queues having customer-heterogeneous arrival rates) with respect to the arrival rate sequence and the initial state (Lemma 3). As a consequence, we can prove that $|\mathbb{E}[W_n^k - w(\mu_{k-1},p_{k-1})]| = O(k^{-\alpha})$, taking advantage of the fact that the initial state $W_0^k = W_{D_{k-1}}^{k-1}$ is close to the steady state $W_\infty(\mu_{k-1},p_{k-1})$. Then, we show that the steady-state distribution is smooth in the parameter (μ,p) (Lemma 4), that is, $\mathbb{E}[|w(\mu_{k-1},p_{k-1}) - w(\mu_k, p_k)|] = O(\mathbb{E}|\mu_k - \mu_{k-1}| + \mathbb{E}|p_k - p_{k-1}|) = O(k^{-\alpha})$, which completes the analysis for I_2.
- I_3: The term I_3 is under control because $W_{D_k}^k$ is close to the steady-state (Lemma 2) and Q_k is uniformly bounded (Lemma 1).

Also see Figure 3 for a graphic illustration.

Following the road map, we next give detailed analysis for $I_i, i = 1,2,3$ by establishing three lemmas (Lemmas 2–4). We believe that these results are not only essential to the transient analysis in the present paper, but may also be of independent interest for theoretic studies of other queueing models.

4.1.1.1. Bounding I_1.

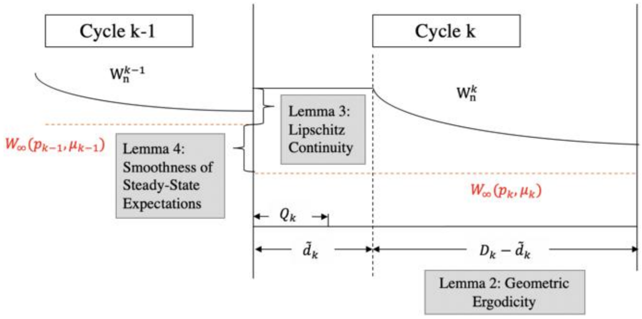
We first establish the rate at which waiting times converge to their steady state distributions. For two given sequences V_n and U_n, we say two $GI/GI/1$ queues with the same parameter $(\mu,p) \in \mathcal{B}$ are synchronously coupled if their waiting times W_n^1 and W_n^2 satisfy

$$W_n^i = \left(W_{n-1}^i + \frac{V_n}{\mu} - \frac{U_n}{\lambda(p)}\right)^+, \text{ for } i = 1,2, \text{ and } n \ge 1,$$

that is, the two systems share the same sequences of service and interarrival times (Blanchet and Chen 2015). The proof of Lemma 2 is given in Online Section EC.1.

Lemma 2 (Exponential Loss of Memory of Initial State). *Suppose two $GI/GI/1$ queues with parameter $(\mu,p) \in \mathcal{B}$ are*

Figure 3. (Color online) Road Map of the Analysis of the Regret of Nonstationarity


Cycle k-1
Cycle k
W_n^{k-1}
W_n^k
$W_\infty(p_{k-1}, \mu_{k-1})$
Lemma 4: Smoothness of Steady-State Expectations
Lemma 3: Lipschitz Continuity
$W_\infty(p_k, \mu_k)$
Q_k
$\tilde{d}_k$
$D_k - \tilde{d}_k$
Lemma 2: Geometric Ergodicity

synchronously coupled with initial waiting times W_0^1 and W_0^2, respectively. Then, for the two positive constants γ and θ defined in Assumption 2 and any $m \geq 1$, we have, conditional on W_0^1 and W_0^2,

$$\begin{aligned}\mathbb{E}[|W_n^1 - W_n^2|^m | W_0^1, W_0^2] \\ \leq e^{-\gamma n}(2 + e^{\mu\theta W_0^1} + e^{\mu\theta W_0^2})|W_0^1 - W_0^2|^m.\end{aligned}$$

In order to bound I_1, at the beginning of each cycle k, given (μ_k, p_k), we couple W_0^k with $\overline{W}_0^k$ that is independently drawn from the steady-state waiting time distribution $W_\infty(\mu_k, p_k)$. The sequence $\overline{W}_n^k$ is defined as

$$\overline{W}_n^k = \left(\overline{W}_{n-1}^k + \frac{V_n^k}{\mu_k} - \frac{U_n^k}{\lambda_k}\right)^+, \text{ for all } 1 \leq n \leq D_k.$$

Then, by definition, conditional on (μ_k, p_k), $\mathbb{E}[\overline{W}_n^k] = w(\mu_k, p_k)$ for all $1 \leq n \leq D_k$, and therefore,

$$|\mathbb{E}[W_n^k - w(\mu_k, p_k)]| \leq \mathbb{E}[|W_n^k - \overline{W}_n^k|].$$

As we show in the proof of Corollary 1, $\{W_n^k : n = d_k + 1, \ldots, D_k\}$ is coupled with $\overline{W}_n^k$ except on a set of negligible set with $d_k \equiv \lceil 4\log(k)/\min(\gamma, \eta)\rceil < \tilde{d}_k$. As a result, we can use Lemma 2 to construct a bound on $\mathbb{E}[|W_n^k - \overline{W}_n^k|]$ for $n = \tilde{d}_k + 1, \ldots, D_k$.

Corollary 1. *Under the conditions of Theorem 1, there exists a constant $A \geq 1$ independent of k and (μ_k, p_k) such that, for all $k \geq 1$ and $n \geq d_k \equiv \lceil 4\log(k+1)/\min(\gamma, \eta)\rceil$,*

$$\mathbb{E}[|W_n^k - \overline{W}_n^k|] \leq e^{-\gamma(n-d_k)}A + 2Mk^{-2}. \tag{13}$$

As a direct consequence, we have $I_1 = O(k^{-\alpha})$.

4.1.1.2. Bounding I_2.

We first show that the waiting times W_n of a queueing model having customer-heterogeneous arrival rates are Lipschitz continuous with respect to the rates (μ_n, λ_n) and the initial state almost surely.

Lemma 3 (Lipschitz Continuity). *Consider two waiting time sequences W_n and $\tilde{W}_n$ for $n \geq 1$ with initial values W_0 and $\tilde{W}_0$, respectively. Let (μ_n, λ_n) and $(\tilde{\mu}_n, \tilde{\lambda}_n) \in \mathcal{B}$ be the corresponding sequences of service and arrival rates, respectively, that is,*

$$W_n = \left(W_{n-1} + \frac{V_n}{\mu_n} - \frac{U_n}{\lambda_n}\right)^+ \quad \textit{and}$$
$$\tilde{W}_n = \left(\tilde{W}_{n-1} + \frac{V_n}{\tilde{\mu}_n} - \frac{U_n}{\tilde{\lambda}_n}\right)^+, \quad \textit{for} \;\; n \geq 1.$$

Suppose there exist two constants $c_\mu, c_\lambda > 0$ such that

$$|\mu_n - \tilde{\mu}_n| \leq c_\mu \quad \textit{and} \quad |\lambda_n - \tilde{\lambda}_n| \leq c_\lambda, \quad \textit{for all} \;\; n \geq 1.$$

Then, we have, for all $n \geq 1$,

$$\begin{aligned}|W_n - \tilde{W}_n| \leq |W_0 - \tilde{W}_0| + \left(\frac{c_\mu}{\underline{\mu}} + \frac{c_\lambda}{\underline{\lambda}}\right)\max(X_n, \tilde{X}_n) \\ + \frac{c_\mu}{\underline{\mu}}\max(W_n, \tilde{W}_n),\end{aligned}$$

where X_n and $\tilde{X}_n$ are the corresponding observed busy periods. In particular, X_n and $\tilde{X}_n$ satisfy Recursion (7) defined in Section 3.3 with any given initial values of $X_0 \geq 0$ and $\tilde{X}_0 \geq 0$.

As discussed, controlling I_2 also involves bounding the difference between the mean steady-state waiting times in two consecutive cycles. Hence, we next establish a uniform high-order smoothness result for the steady-state waiting times with respect to the model parameter (μ, p).

Lemma 4 (Smoothness in μ and p). *Suppose $(\mu_i, p_i) \in \mathcal{B}$ for i = 1, 2. Let $W_\infty(\mu_i, p_i)$ be the steady-state waiting time of the GI/GI/1 queue under parameter (μ_i, p_i), respectively. Then, the steady-state waiting times $(W_\infty(\mu_1, p_1), W_\infty(\mu_2, p_2))$ can be coupled such that there exists a constant $B > 0$ independent of (μ_i, p_i) satisfying that, for all $1 \leq m \leq 4$,*

$$\begin{aligned}\mathbb{E}[|W_\infty(\mu_1, p_1) - W_\infty(\mu_2, p_2)|^m] \\ \leq B(|\mu_1 - \mu_2|^m + |p_1 - p_2|^m),\end{aligned}$$

where a closed-form expression of constant B is given in (EC.2).

We adopt a "coupling from the past" (CFTP) approach in the proof of Lemma 4 (see Online Section

EC.1). Roughly speaking, CFTP is a synchronous coupling starting from infinite past. In the proof of Lemma 4, we explicitly explain how to construct the CFTP.

Now, we are ready to analyze I_2. Essentially, we compare $\mathbb{E}[W_n^k]$ in the warm-up period with $w(\mu_{k-1}, p_{k-1}) = \mathbb{E}[W_\infty(\mu_{k-1}, p_{k-1})]$. For each cycle k, recall that we have already coupled W_n^{k-1} with a stationary sequence $\overline{W}_n^{k-1}$ in cycle $k-1$, we then extend the sequence $\overline{W}_n^{k-1}$ to cycle k in the sense that

$$\overline{W}_{D_{k-1}+n}^{k-1} = \left(\overline{W}_{D_{k-1}+n-1}^{k-1} + \frac{V_n^k}{\mu_{k-1}} - \frac{U_n^k}{\lambda_{k-1}}\right)^+, \quad \text{for } n = 1, 2, \ldots, D_k.$$

Then, conditional on (μ_{k-1}, p_{k-1}), $\mathbb{E}[\overline{W}_{D_{k-1}+n}^{k-1}] = w(\mu_{k-1}, p_{k-1})$. So we have

$$\begin{aligned}|\mathbb{E}[W_n^k - w(\mu_k, p_k)]| &\le |\mathbb{E}[W_n^k - w(\mu_{k-1}, p_{k-1})]| \\ &\quad + \mathbb{E}[|w(\mu_{k-1}, p_{k-1}) - w(\mu_k, p_k)|] \\ &\le \mathbb{E}[|W_n^k - \overline{W}_{D_{k-1}+n}^{k-1}|] \\ &\quad + \mathbb{E}[|w(\mu_{k-1}, p_{k-1}) - w(\mu_k, p_k)|].\end{aligned}$$

Bounding the first term by Lemma 3 and the second term by Lemma 4 yields the following bound on I_2.

Corollary 2. *Under the conditions of Theorem 1, for all $k \ge 2$ and $1 \le n \le D_k$, we have*

$$\mathbb{E}[|W_n^k - w(\mu_k, p_k)|] = O(k^{-\alpha}). \tag{14}$$

As a direct consequence, $|I_2| = O(k^{-\alpha}\log(k))$.

4.1.1.3. Bounding I_3.

We complete our analysis on the regret of nonstationarity by showing that $I_3 = O(k^{-\alpha})$. The proof of Corollary 3 basically follows from Lemmas 1 and 2 with some similar argument as used in the proof of Corollary 2.

Corollary 3. *Under the conditions of Theorem 1, $|I_3| = O(k^{-\alpha})$.*

Finishing the Proof of Theorem 1. Then, Theorem 1 follows immediately from Corollaries 1–3. A complete proof of Theorem 1, including the proofs of Corollaries 1–3, is given in Online Section EC.1.5. In particular, we provide an explicit expression of the constant K in terms of the model parameters in (EC.3).

Remark 6. We advocate that Theorem 1 may apply to other queueing models (its scope is beyond the $GI/GI/1$ queue) as long as one can verify three conditions for the designated model: (i) uniform boundedness for the rate of convergence to the steady state, that is, Lemma 2; (ii) path-wise Lipschitz continuity, that is, Lemma 3; and (iii) smoothness of the stationary distributions in the control variables, that is, Lemma 4.

4.2. Regret of Suboptimality

To bound the regret of suboptimality $R_2(L)$, we need to control the rate at which x_k converges to x^*. This depends largely on the effectiveness of the estimator H_k for $\nabla f(x_k)$. In our algorithm, such effectiveness is measured by the bias B_k and variance $\mathcal{V}_k$. The following result shows that, if B_k and $\mathcal{V}_k$ can be appropriately bounded, then, x_k converges to x^* rapidly, and hence, $R_2(L)$ can be properly bounded.

Theorem 2. (Regret of Suboptimality). *Suppose Assumption 3 holds. If there exists a constant $K_3 \ge 1$ such that the following conditions hold for all k,*

a. $\left(1 + \frac{1}{k}\right)^\beta \le 1 + \frac{K_0}{2}\eta_k$.
b. $B_k \le \frac{K_0}{8}k^{-\beta}$.
c. $\eta_k \mathcal{V}_k \le K_3 k^{-\beta}$.

Here, $0 < \beta \le 1$ is a constant, and $\eta_k \to 0$ is the step size, and then, there exists a constant $C \ge 8K_3/K_0$ with an explicit expression given in (EC.5) *such that, for all $k \ge 1$,*

$$\mathbb{E}[\|x_k - x^*\|^2] \le Ck^{-\beta}, \tag{15}$$

and as a consequence,

$$R_2(L) \le CK_1 \sum_{k=1}^{L} \left(\frac{D_k}{\lambda(\overline{p})} + M\right)k^{-\beta} = O\left(\sum_{k=1}^{L} D_k k^{-\beta}\right). \tag{16}$$

Remark 7 (Selecting the "Optimal" D_k). Expression (16) indicates a trade-off in the selection of the parameter D_k. On the one hand, increasing the sample size D_k reduces the bias B_k for the gradient estimator and, hence, leads to a smaller value of $k^{-\beta}$. On the other hand, a larger D_k makes the system operate under a suboptimal decision for a longer time. To this end, one may choose an optimal order (in k) for D_k by minimizing the order of the regret as in (16).

Our proof of Theorem 2 follows an inductive approach as used in Broadie et al. (2011). Let $b_k \equiv \mathbb{E}[\|x_k - x^*\|^2]$. According to the SGD iteration $x_{k+1} = \Pi_{\mathcal{B}}(x_k - \eta_k H_k)$, we have

$$\begin{aligned}\mathbb{E}[\|x_{k+1} - x^*\|^2 | x_k] &\le \mathbb{E}[\|x_k - \eta_k H_k - x^*\|^2 | x_k] \\ &= \|x_k - x^*\|^2 - 2\eta_k \mathbb{E}[H_k | x_k](x_k - x^*) \\ &\quad + \eta_k^2 \mathbb{E}[\|H_k\|^2 | x_k].\end{aligned}$$

Then, by Assumption 3 and the definition of $B_k, \mathcal{V}_k$ by (4), we derive the following recursive inequality for b_k:

$$b_{k+1} \le (1 - K_0\eta_k + \eta_k B_k)b_k + \eta_k B_k + \eta_k^2 \mathcal{V}_k, \quad k \ge 1,$$

and we prove (15) by induction. The full proof is given in Online Section EC.1.7.

In Section 5, we apply Theorem 2 to treat our online learning algorithm (Algorithm 1) by verifying that conditions (a)–(c) are satisfied. Because, in Theorem 2, conditions (a)–(c) are stated explicitly in terms of the step size η_k, bias B_k, and variance $\mathcal{V}_k$ of the gradient estimator, these conditions may serve as useful building blocks for the design and analysis of online learning algorithms in other queueing models as well.

5. GOLiQ for the *GI*/*GI*/1 Queue

In this section, we provide a concrete GOLiQ algorithm that solves the optimal pricing and capacity sizing Problem (1) for a $GI/GI/1$ queueing system. We show that the gradient $\nabla f(\mu, p)$ can be estimated "directly" from past experience (i.e., data of delay and busy times generated under the present policy). Applying the regret analysis developed in Section 4, we provide a theoretic upper bound for the overall regret in Theorem 3.

5.1. A Gradient Estimator

Following the algorithm framework outlined in Section 3.2, we now develop a detailed gradient estimator H_k. Regarding the objective function in (5), it suffices to construct estimators for the partial derivatives

$$\frac{\partial}{\partial \mu}\mathbb{E}[W_\infty(p,\mu)] \qquad \text{and} \qquad \frac{\partial}{\partial p}\mathbb{E}[W_\infty(p,\mu)]. \tag{17}$$

Following the infinitesimal perturbation analysis (IPA) approach (see, for example, Glasserman 1992), we next show that the partial derivatives in (17) can be expressed in terms of the steady-state distributions $W_\infty(p,\mu)$ and $X_\infty(p,\mu)$ of the waiting time process W_n and observed busy period process X_n, of which the dynamics are characterized by (6) and (7).

Lemma 5. *Suppose Assumptions 1 and 2 hold. Then, for any $(\mu,p)\in\mathcal{B}$, $\mathbb{E}[W_\infty(\mu,p)]$ are differentiable in μ and p. Besides,*

$$\begin{aligned}\frac{\partial}{\partial p}f(\mu,p) &= -\lambda(p) - p\lambda'(p)\\ &\quad + h_0\lambda'(p)\left(\mathbb{E}[W_\infty(\mu,p)] + \mathbb{E}[X_\infty(\mu,p)] + \frac{1}{\mu}\right)\\ \frac{\partial}{\partial \mu}f(\mu,p) &= c'(\mu)\\ &\quad - h_0\frac{\lambda(p)}{\mu}\left(\mathbb{E}[W_\infty(\mu,p)] + \mathbb{E}[X_\infty(\mu,p)] + \frac{1}{\mu}\right).\end{aligned} \tag{18}$$

Proof of Lemma 5. To prove Equation (18), it suffices to work with the partial derivatives of the steady-state expectation $\mathbb{E}[W_\infty(\mu,p)]$. We follow the IPA analysis in Glasserman (1992) and Chen (2014).

Given (μ,p), we define $r(p) = 1/\lambda(p)$ and rewrite Recursion (6) as

$$W_n(\mu,p) = \left(W_{n-1}(\mu,p) + \frac{V_n}{\mu} - r(p)U_n\right)^+.$$

Define the derivative process $Z_n \equiv \frac{\partial}{\partial r}W_n(\mu,p)$, and then, by the chain rule, we have

$$\begin{aligned}Z_n &= \frac{\partial}{\partial r}W_n(\mu,p) = \frac{\partial}{\partial r}\left(W_{n-1}(\mu,p) + \frac{V_n}{\mu} - rU_n\right)^+\\ &= \begin{cases}\frac{\partial}{\partial r}W_{n-1} - U_n = Z_{n-1} - U_n & \text{if } W_n > 0;\\ 0 & \text{if } W_n = 0.\end{cases}\end{aligned}$$

We obtain a recursion $Z_n = (Z_{n-1} - U_n)\mathbf{1}_{\{W_n>0\}}$. Let $\tilde{Z}_n \equiv -Z_n/\lambda(p)$. Then, it is straightforward to see that $\tilde{Z}_n$ satisfies the recursion given in (7) as the observed busy period X_n, that is,

$$\tilde{Z}_n = \left(\tilde{Z}_{n-1} + \frac{U_n}{\lambda(p)}\right)\mathbf{1}(W_n > 0).$$

Under the assumption that the queueing system is stable, the limit $\tilde{Z}_\infty$ should be equal in distribution to X_∞. Therefore, we formally derive

$$\begin{aligned}\frac{\partial}{\partial r}\mathbb{E}[W_\infty(\mu,p)] &= \mathbb{E}[Z_\infty] = -\lambda(p)\mathbb{E}[\tilde{Z}_\infty]\\ &= -\lambda(p)\mathbb{E}[X_\infty(\mu,p)].\end{aligned} \tag{19}$$

These heuristics can be made rigorous by verifying exchanges of limits using the results in Glasserman (1992), and we refer the readers to Online Section EC.1.9 for detailed explanations. Using (19), we can derive the partial derivative of the steady-state waiting time with respect to price p as follows:

$$\begin{aligned}\frac{\partial}{\partial p}\mathbb{E}[W_\infty(\mu,p)] &= \frac{\partial}{\partial r}\mathbb{E}[W_\infty(\mu,p)]\frac{\partial r(p)}{\partial p}\\ &= -\lambda(p)\mathbb{E}[X_\infty(\mu,p)]\cdot -\frac{\lambda'(p)}{\lambda(p)^2}\\ &= \mathbb{E}[X_\infty(\mu,p)]\frac{\lambda'(p)}{\lambda(p)}.\end{aligned}$$

Now, we turn to $\frac{\partial}{\partial \mu}\mathbb{E}[W_\infty(\mu,p)]$. Let $\hat{Z}_n \equiv \mu W_n(\mu,p)$; it is easy to check that $\hat{Z}_n = (\hat{Z}_{n-1} + V_n - \mu U_n/\lambda(p))^+$. Then, following steps similar to those for (19), we have

$$\frac{\partial}{\partial \mu}\mathbb{E}[\hat{Z}_\infty(\mu,p)] = -\mathbb{E}[X_\infty(\mu,p)].$$

Therefore,

$$\begin{aligned}-\mathbb{E}[X_\infty(\mu,p)] &= \frac{\partial}{\partial \mu}\mathbb{E}[\hat{Z}_\infty(\mu,p)] = \frac{\partial}{\partial \mu}\mathbb{E}[\mu W_\infty(\mu,p)]\\ &= \mu\frac{\partial}{\partial \mu}\mathbb{E}[W_\infty(\mu,p)] + \mathbb{E}[W_\infty(\mu,p)],\end{aligned}$$

and hence, $\partial\mathbb{E}[W_\infty(\mu,p)]/\partial\mu = -(\mathbb{E}[X_\infty(\mu,p)] + \mathbb{E}[W_\infty(\mu,p)])/\mu$. Finally, plugging the expressions of the two partial derivatives into ∇f yields (18). □

5.2. GOLiQ: A $G/G/1$ Version

Utilizing results in Lemma 5, we are ready to design a $G/G/1$ version of the GOLiQ algorithm, in which we estimate the terms $\mathbb{E}[W_\infty(\mu,p)]$ and $\mathbb{E}[X_\infty(\mu,p)]$) in the partial derivatives (18) using the finite-sample averages of W_n^k and X_n^k observed in each cycle k. The formal description of the algorithm is given in Algorithm 1.

Algorithm 1 (GOLiQ for $GI/GI/1$ Queues)

```
Input: number of cycles $L$;
    parameters $0 < \xi < 1$, $D_k$, $\eta_k$ for $k = 1, 2, \ldots, L$;
    initial value $x_1 = (\mu_1, p_1)$.
for $n = 1, 2, \ldots, D_k$ do
    operate the system under $x_k = (\mu_k, p_k)$ until $D_k$ customers enter service;
    observe $(W_n^k, X_n^k)$ for $n = 1, 2, \ldots, D_k$;
    randomly draw $Z \in \{1, 2\}$;
    if $Z = 1$ then
        $h \leftarrow -\lambda(p_k) - p_k\lambda'(p_k) + h_0\lambda'(p_k)\Big[\frac{1}{\lceil D_k(1-\xi)\rceil}\sum_{n>\xi D_k}^{D_k}(X_n^k + W_n^k) + \frac{1}{\mu_k}\Big]$;
        $H_k \leftarrow (2h, 0)$;
    else
        $h \leftarrow c'(\mu_k) - h_0\frac{\lambda(p)}{\mu_k}\Big[\frac{1}{\lceil D_k(1-\xi)\rceil}\sum_{n>\xi D_k}^{D_k}(X_n^k + W_n^k) + \frac{1}{\mu_k}\Big]$;
        $H_k \leftarrow (0, 2h)$;
    end
    update: $x_{k+1} = \Pi_{\mathcal{B}}(x_k - \eta_k H_k)$;
end.
```

Remark 8 (On the Queueing Leftover). We elaborate more on our treatment of Q_k, the existing queue content at the beginning of cycle k. First, the content of Q_k includes customer arrivals in cycle $k-1$ and possibly even earlier cycles. Second, it is also possible to have $Q_k > D_k$. Nevertheless, these cases do not affect the implementation of Algorithm 1 (note that Algorithm 1 gives a gradient estimator using $\lceil(1-\xi)D_k\rceil$ samples without specifying any of the preceding events). Of course, the event $\{Q_k > D_k\}$ does play a role in our theoretic regret analysis, but it is a rare event with a negligible probability (in fact, we show that the probability is suppressed to $O(k^{-3})$); also see Remark 3.

5.2.1. Selecting the Optimal Hyperparameters. The effectiveness of Algorithm 1 largely hinges upon carefully selecting the three hyperparameters: (i) the warm-up time $\xi \in (0,1)$, (ii) the learning step size $\eta_k > 0$, and (iii) the exploration sample size $D_k > 0$. Except for ξ, which has no bearing on the theoretical order of the regret, both the other two parameters D_k and η_k play critical roles in our regret analysis. We next give the forms of the two parameters. First, The step size η_k satisfies

$$\eta_k = c_\eta/k, \qquad \text{with} \quad c_\eta \geq 2/K_0, \tag{20}$$

where K_0 is the convexity bound specified in Assumption 3. Next, the sample size D_k satisfies

$$D_k = a_D + b_D\log(k), \quad \text{with} \quad a_D \geq \frac{C_D}{\min(\gamma,\eta)\xi} \quad \text{and} \quad b_D \geq \frac{8}{\min(\gamma,\eta)\xi}, \tag{21}$$

for any warm-up parameter $\xi \in (0,1)$, where γ and η are the constants specified in Assumption 2 and the explicit formula of C_D is given in (EC.7).

The aforementioned forms of η_k and D_k are obtained from our detailed regret analysis in which we show that the structure of (20) and (21) "minimizes" the order of the overall regret (in the sense of maximizing α and β as in Theorems 1 and 2). Although the theoretical bounds of parameters a_D, b_D, and c_η are imposed to facilitate our regret analysis, our numerical experiments show that GOLiQ remains effective even when the theoretical bounds are relaxed, confirming the robustness of GOLiQ to these hyperparameters; see Online Section EC.2 for details. Next, we show that Algorithm 1 has a regret bound of $O((\log(M_L)^2)$ with $M_L \equiv \sum_{k=1}^{L} D_k$ being the cumulative number of customers served by cycle L. We do so by verifying that our choices of D_k and η_k (along with the corresponding B_k and $\mathcal{V}_k$), satisfy the conditions in Theorems 1 and 2.

Theorem 3. (Regret Bound for Algorithm 1). *Suppose Assumptions 1–3 hold, and η_k and D_k are selected according to (20) and (21).*

Then,

i. *There exists a positive constant $K_3 > 0$ such that*

$$B_k \leq \frac{K_0}{8k} \quad \textit{and} \quad \eta_k\mathcal{V}_k \leq \frac{K_3}{k}.$$

ii. *There exists a positive constant $K_2 > 0$ such that*

$$\mathbb{E}[\|x_k - x_{k+1}\|^2] \leq K_2 k^{-2}. \tag{22}$$

iii. *As a consequence of (i) and (ii), the regret for Algorithm 1*

$$R(L) \leq K_{\text{alg}}\,\log(M_L)^2 = O(\log(M_L)^2). \tag{23}$$

Remark 9 (On the Logarithmic Regret Bound (23)). We provide some additional discussions on the regret bound (23):

i. On the constant K_{alg}: The explicit expression for the constant K_{alg}, although complicated, is given by (EC.9). It involves an error bound corresponding to the transient behavior of the queueing system, the bias and variance of the gradient estimator, moment bounds on the queue length and other model parameters. One can verify that K_{alg} is decreasing in the convergence rate coefficient γ and increasing in the moment bounds of the queue length M.

ii. On the first logarithmic term: Consider an SGD algorithm in that an unbiased gradient estimator H_k with a bounded variance can be evaluated using a single data point (i.e., $B_k = 0$, $\mathcal{V}_k = O(1)$); it is proved the scaled error $k^{-1/2}(x_k - x^*)$ converges in distribution to a nonzero random variable (theorem 2.1 in chapter 10 of Kushner and Yin 2003). Hence, the convergence rate for $\|x_k - x^*\|^2$ that any SGD-based algorithm can achieve is at best $O(k^{-1})$ (yielding a cumulative regret of order $O(\log(k))$), which is exactly the rate of convergence established by our online algorithm (taking $\beta = 1$ in Theorem 3). In this sense, GOLiQ is already achieving an optimal convergence rate. We point out that, because of the nonstationary error of the queueing system, our gradient estimator is obtained using an increasing number of data points in order to guarantee a reasonably small bias.

iii. On the second logarithmic term: In order to control the regret of nonstationarity, the queueing system needs to be operated in each cycle for a duration of order $O(\log(k))$. Because the queueing performance converges to its steady state exponentially fast, this inevitably introduces an extra logarithmic term in our regret bound (which explains the "square" in $\log(M_L)^2$). The question that remains open is whether this $O(\log(M_L)^2)$ bound is optimal. We conjecture that the answer is "yes" but admit that a rigorous treatment of a lower regret bound can be quite challenging. For example, establishing a lower regret bound requires a lower bound on the convergence rate of a $GI/GI/1$ queue, which, by itself, is an open question. We leave this question to future research.

Remark 10 (Controlling the Length of Cycle k). We use D_k (the number of customers served in cycle k) instead of the clock time T_k to control and measure the regret bound. The benefit of using D_k (rather than T_k) as the cycle length is that it facilitates the technical analysis because D_k is directly related to the number of samples used to estimate our gradient estimator. In fact, using D_k instead of T_k has no bearing on the order of the regret bound. To see this, note that the arrival rate is assumed to fall into a compact set $[\lambda(\overline{p}), \lambda(\underline{p})]$. Therefore, because T_L is the total units of clock time elapsed after cycle L, we have $M_L/\lambda(\underline{p}) \le \mathbb{E}[T_L] \le M_L/\lambda(\overline{p})$ for all L.

6. Numerical Experiments

To confirm the practical effectiveness of our online learning method, we conduct numerical experiments to visualize the algorithm convergence, benchmark the outcomes with known exact optimal solutions, estimate the true regret, and compare it to the theoretical upper bounds. Our base example is an $M/M/1$ queue, having Poisson arrivals with rate $\lambda(p)$ and exponential service times with rate μ. In our optimization, we consider a commonly used logistic demand function (Besbes and Zeevi 2015)

$$\lambda(p) = n\lambda_0(p), \qquad \lambda_0(p) = \frac{\exp(a-p)}{1+\exp(a-p)}, \tag{24}$$

where n is the system scale (also referred to as the market size). We also consider the following convex cost function for the service rate:

$$c(\mu) = c_0\mu^2. \tag{25}$$

See the top left panel of Figure 4 for $\lambda(p)$ in (24). In particular, the optimal pricing and staffing problem in (1) now becomes

$$\max_{\mu,p}\left\{p\lambda(p) - c_0\mu^2 - h_0\frac{\lambda(p)/\mu}{1-\lambda(p)/\mu}\right\}. \tag{26}$$

In light of the closed-form steady-state formulas of the $M/M/1$ queue, we can obtain the exact values of the optimal solutions (μ^*, p^*) and the corresponding objective value $f(\mu^*, p^*)$, with which we are able to benchmark the solutions from our online optimization algorithm.

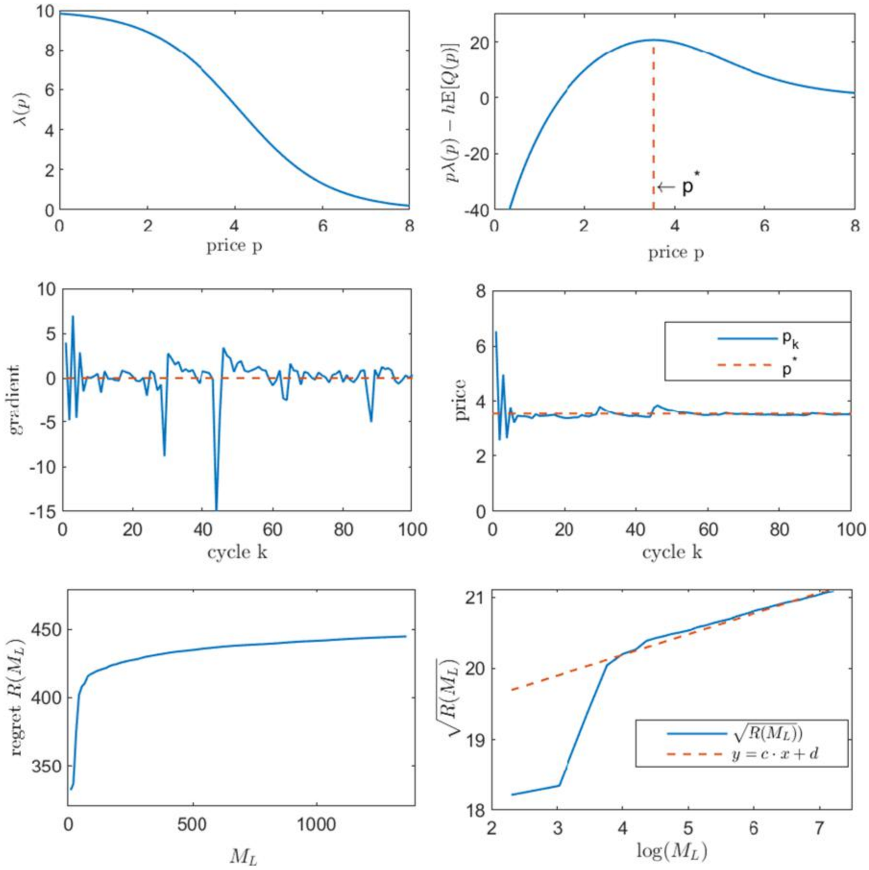
We first consider two one-dimensional online optimization problems in Section 6.1. We next treat the two-dimensional pricing and staffing problem in Section 6.2. In Section 6.3, we compare our results to previously established asymptotic heavy-traffic solutions in Lee and Ward (2014). Additional numerical experiments are provided in the e-companion: In Online Section EC.2, we investigate the robustness of GOLiQ to the hyperparameters. In Online Section EC.3, we benchmark the performance of GOLiQ to other online learning methods. Online Section EC.4 includes more experiments regarding the relaxation of uniform stability and GOLiQ's performance in queues having other interarrival and service time distributions.

6.1. One-Dimensional Online Optimizations

Algorithm 1 covers special cases in which there is only one decision variable. For example, if the service capacity μ (service fee p) is an exogenous parameter and the only decision is the service fee p (service capacity μ), then one can simply fix $Z = 1$ ($Z = 2$) throughout the learning process. The theoretical regret bound (as in Theorem 3) for these one-dimensional cases remains unchanged.

6.1.1. Online Optimal Pricing with a Fixed Service Capacity. Motivated by revenue management problems in revenue generating service systems, our first example focuses on the one-dimensional optimization of price p with service rate $\mu = \mu_0$ held fixed. In this case, we can simply omit the term $c_0\mu^2$ in (26). Fixing the

Figure 4. (Color online) Online Optimal Pricing for an $M/M/1$ Queue with Fixed Service Rate with $\mu_0 = 10, a = 4.1, p_0 = 6.5, p^* = 3.531$, $\eta_k = 1/k$, and $D_k = 10 + 10\log(k)$



Note. See (i) demand function (top left), (ii) revenue function (top right), (iii) sample path of the gradient (middle left), (iv) sample path of the price (middle right), (v) estimated regret (bottom left), (vi) square root of regret versus logarithmic of served customers with $c = 0.24$, $d = 19.04$ (bottom right).

other model parameters as $a = 4.1, n = 10, h_0 = 1$, and $\mu_0 = 10$, we first obtain the exact optimal price $p^* = 3.531$ (top right panel of Figure 4). According to Algorithm 1 and Theorem 3, we set the step size $\eta_k = 1/k$ and cycle length $D_k = 10 + 10\log(k)$. In Figure 4, we give the sample paths of the gradient H_k and price p_k as functions of the number of cycles k and the mean regret (estimated by averaging 500 independent sample paths) as a function of the cumulative number of service completions M_L. We observe that, although the objective function $f(\mu, p)$ is not convex in p, the pricing decision p_k quickly converges to the optimal value p^*, and the regret grows as a logarithmic function of M_L. In particular, a simple linear regression for the pair $(\sqrt{R(M_L)}, \log(M_L))$ (bottom right panel) verifies our regret bound given in Theorem 3.

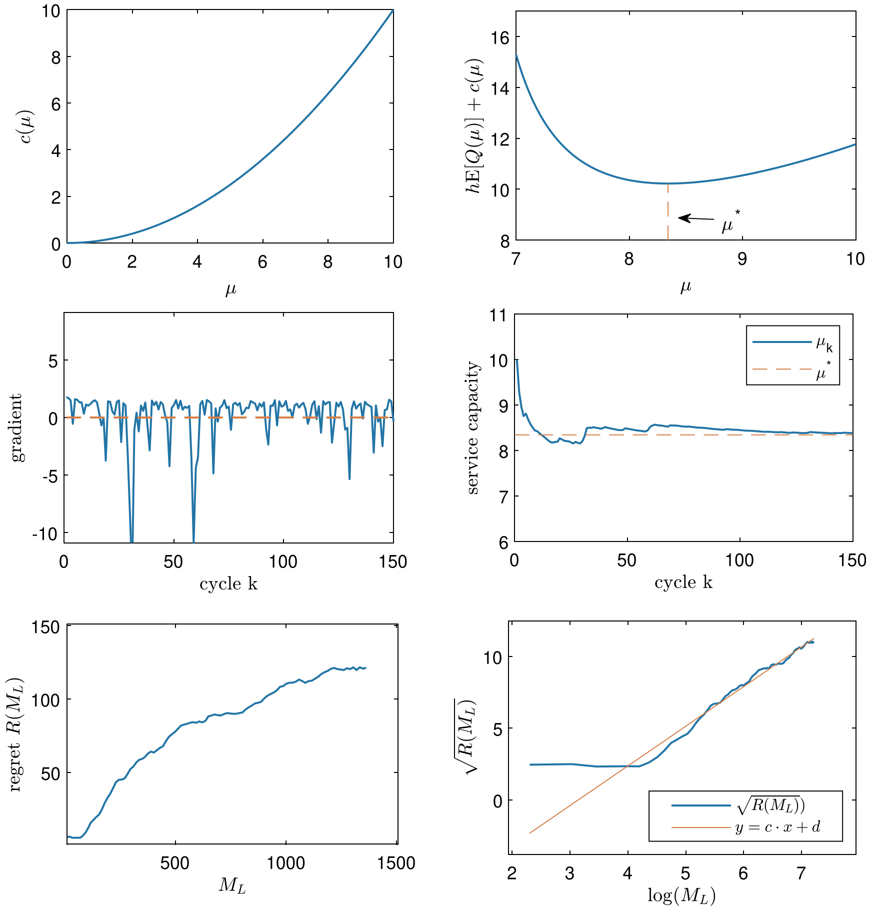
6.1.2. Online Optimal Staffing Problem with an Exogenous Arrival Rate.

Motivated by conventional service systems in which customers are served based on goodwill (e.g., hospitals), we next solve an online optimal staffing problem, with the objective of minimizing the combination of the steady-state queue length (or equivalently the delay) and the staffing cost with the arrival rate (or, equivalently, the price p) held fixed. Namely, we omit the term $p\lambda(p)$ in (26). Fixing $\lambda = \lambda_0 = 6.385$, $h_0 = 1$, and $c_0 = 0.1$, we obtain the exact optimal service capacity $\mu^* = 8.342$ (top right panel of Figure 5). Also by Algorithm 1 and Theorem 3, we set the step size $\eta_k = 0.4k^{-1}$ and cycle length $D_k = 10 + 10\log(k)$ with initial service rate $\mu_0 = 10$. In Figure 5, we again give sample paths of the gradient H_k and service capacity μ_k and estimation of the regret. As the number of cycles k increases, our stage-k staffing decision μ_k quickly converges to μ^* (bottom right panel), and the regret also grows as a logarithmic function of M_L (bottom left panel).

6.2. Joint Pricing and Staffing Problem

We next consider a joint staffing and pricing problem having the objective function in (26) with the logistic demand function in (24) and parameters $a = 4.1$, $n = 10$, $h_0 = 1$, and $c_0 = 0.1$. The optimal price $p^* = 4.02$ and

Figure 5. (Color online) Online Optimal Staffing for an M/M/1 Queue with Fixed Price with $\lambda_0 = 6.385, M = 10, \eta_k = 0.4k^{-1}$, and $D_k = 10 + 10\log(k)$



Note. See (i) staffing cost (top left), (ii) cost function (top right), (iii) sample path of gradient (middle left), (iv) sample path of service capacity (middle right), (v) estimated regret (bottom left), (vi) square root of regret versus logarithmic of served customers with $c = 2.76$, $d = -8.68$ (bottom right).

service rate $\mu^* = 7.10$ are given as benchmarks (top right panel in Figure 6). In Figure 6, we show that μ_k and p_k converge quickly to their corresponding optimal target levels μ^* and p^* (although the objective $f(\mu, p)$ is not always convex when $\mu > \lambda(p)$). And similar to the one-dimensional cases, the regret grows as a logarithmic function of M_L (bottom left panel).

6.3. Comparison with Heavy-Traffic Methods

In this section, we provide numerical analysis to contrast the performance of GOLiQ to that of the heavy-traffic approach in Lee and Ward (2014). In Lee and Ward (2014), the objective is to find the optimal decisions p^* and μ^* for the $GI/GI/1$ optimization Problem (1) with a linear staffing cost $c(\mu) = c\mu$. Because this problem is not amenable to analytic treatments (because of the complex $GI/GI/1$ queueing dynamics), the authors resort to the heavy-traffic approximation by constructing a sequence of $GI/GI/1$ queues indexed by a scaling factor n, in which the n^{th} model has an arrival rate $\lambda^n(p) \equiv n\lambda_0(p)$, which grows to infinity as n increases. The authors propose an asymptotically optimal solution:

$$(\tilde{p}^{(n)}, \tilde{\mu}^{(n)}) = \left(\hat{p}^*, n\hat{\mu}^* + \sigma\sqrt{\frac{h_0 n}{2c}}\right), \tag{27}$$

where $\sigma = \sqrt{\text{Var}\,(U_i) + \text{Var}(V_i)}$, U_i and V_i are defined in Assumption 2, and $(\hat{p}^*, \hat{\mu}^*)$ solves a deterministic static planning problem:

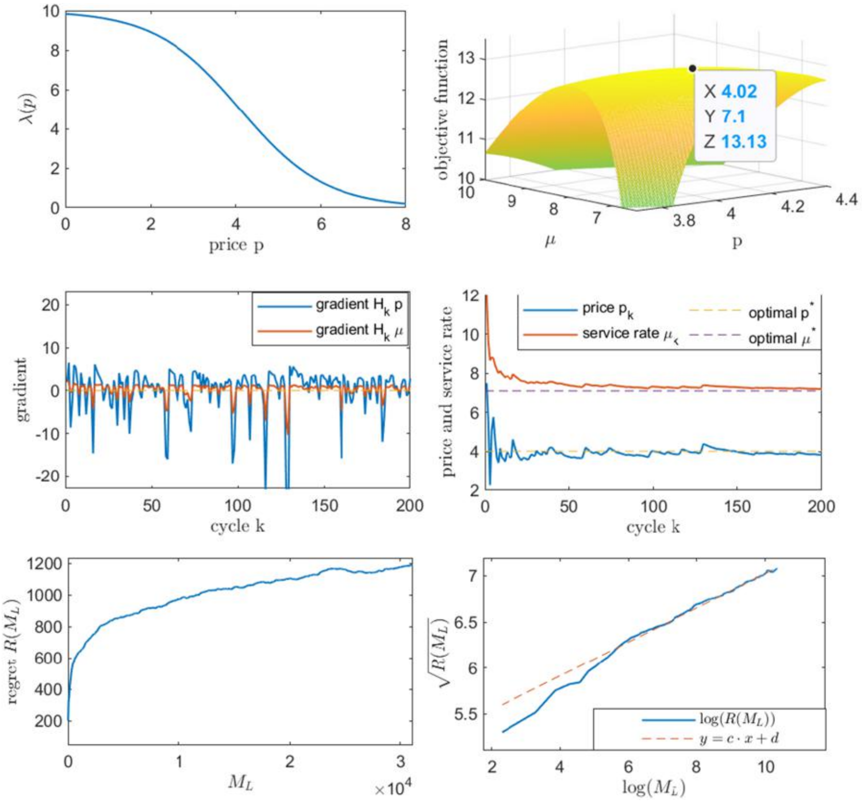
$$\min_{p,\mu} f_0(p, \mu) = -p\lambda_0(p) + c\mu. \tag{28}$$

We remark that the solution in Lee and Ward (2014) requires the precise knowledge of the second moments of service and arrival times (e.g., the term σ in (27)), but such information is not needed in GOLiQ.

6.3.1. Experimental Settings.

We consider an $M/GI/1$ model with a phase-type service-time distribution and

Figure 6. (Color online) Joint Pricing and Staffing for an $M/M/1$ Queue with $p_0 = 7.5, \mu_0 = 12, \eta_k = 1/k$, and $D_k = 10 + 10\log(k)$



Note. See (i) demand function (top left), (ii) revenue function (top right), (iii) sample path of gradient (middle left), (iv) sample path of decision parameters (middle right), (v) estimated regret (bottom left), (vi) square root of regret versus logarithmic of served customers with $c = 0.186$, $d = 5.17$ (bottom right).

a logit demand $\lambda(p) = n\lambda_0(p)$ in (24), where the base demand rate $\lambda_0(p)$ has $a = 4.1$ and the market size n plays the role of the scaling factor. We fix the delay cost $h_0 = 1$ throughout this experiment. To quantify the regret, we obtain the exact optimal policy using the Pollaczek–Khinchine formula for the queue-length function

$$\mathbb{E}[Q_\infty(p,\mu)] = \rho + \frac{\rho^2}{1-\rho}\frac{1+c_s^2}{2}, \tag{29}$$

where $c_s^2 \equiv Var(U_i)/\mathbb{E}[U_i]^2$ is the squared coefficient of variation (SCV) for the service time. We next describe the detailed settings for comparing GOLiQ to heavy-traffic solution in Lee and Ward (2014), dubbed LW. In order to benchmark the regret of our GOLiQ to that of LW, we continue to consider a dynamic environment in which the number of cycles k increases. In the k^{th} cycle,

- The LW policy remains fixed at $(\tilde{p}^{(n)}, \tilde{\mu}^{(n)})$ as in (27) (it does not evolve with k).
- Our online learning policy is dynamically updated according to GOLiQ (Algorithm 1).

Because the LW policy is an approximation, it yields a linear regret as k increases. But LW's linear regret should not be too steep when n is large enough. In contrast, although GOLiQ is guaranteed to generate a sublinear regret, it is expected to have a larger regret increment at the earlier "exploration" stage because it is learning without the supervision of the fluid or diffusion limits (as in the LW approach). Nevertheless, we expect that GOLiQ will eventually outperform the LW method (exhibiting a lower regret level) when k is sufficiently large. We next numerically study how soon GOLiQ surpasses LW and the impact of the following three parameters:

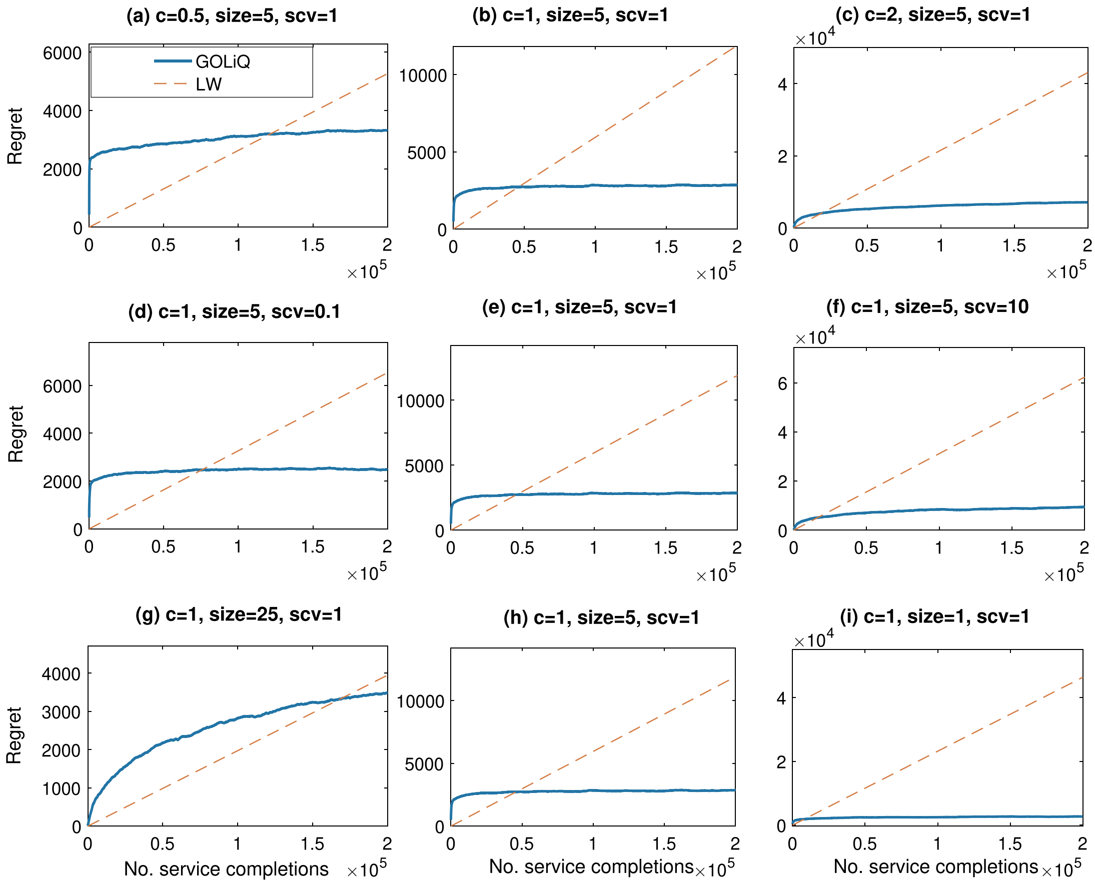
i. Staffing cost c.
ii. Service-time SCV c_s^2.
iii. Market size n (i.e., system scale).

We intentionally set the initial decision (μ_0, p_0) of GOLiQ far from the optimal solution (μ^*, p^*) in the experiment.

6.3.2. Experiment Results.

In Figure 7, we report results of regret for both GOLiQ and LW. For the three factors c, c_s^2, and n, we change one at a time (with the other two held fixed). In panels (a)–(c), we vary the staffing cost c from 0.5 to 2. In panels (d)–(f), we vary the service-time SCV c_s^2 from 0.1 to 10. Here, the cases

Figure 7. (Color online) Regret Comparison with Heavy-Traffic Approximation with Varying (i) Staffing Cost c (Panels (a)–(c)), (ii) Service Variability c_s^2 (Panels (d)–(f)), and (iii) Market Size n (Panels (g)–(i))



Notes. Hyperparameters are $\eta_k = 5k^{-1}$and $D_k = 10 + 10\log(k)$ for all instances. All regret is estimated by averaging 500 independent simulation runs.

$c_s^2 = 0.1$, 1, and 10 are achieved by considering Erlang, exponential, and hyperexponential service-time distributions. In panels (g)–(i), we vary the system scale n from 1 to 25. In all of the cases, we use hyperparameter $\eta_k = 5k^{-1}$ and $D_k = 10 + 10\log(k)$. Monte Carlo estimates of the regret curves are obtained by averaging 100 independent runs.

We can see from Figure 7 that, in all cases, GOLiQ eventually establishes a lower regret level than the LW policy. Varying these three factors clearly has a significant impact on how soon GOLiQ outperforms LW. Our findings are summarized:

- Staffing cost c: Figure 7 shows that GOLiQ intends to outperform LW when c is relatively large. We provide our explanations. First, a larger staffing cost c induces a smaller μ^*, which leads to a longer waiting queue. On the other hand, note that the LW solution is primarily based on solving the deterministic static problem (28), and unlike the stochastic revenue optimization Problem (1), the objective function of (28) overlooks the queue-length holding cost. This explains why GOLiQ gains its advantage over LW as c increases. See panels (a)–(c) of Figure 7.
- Service SCV c_s^2: When the service-time SCV is smaller, the LW method intends to work better because the basic idea of LW stems from solutions of a fluid model (in which the service times are assumed deterministic). On the other hand, when c_s^2 is larger, the system becomes more variable so that our learning-based algorithm begins to excel (because GOLiQ takes into account real-time information dynamically). See panels (d)–(f) of Figure 7.
- Market size n: When n is small, LW loses its advantages because it arises from the large-scale limit of the $GI/GI/1$ queue, which requires n to be sufficiently large, whereas the performance of our GOLiQ is robust to the system scale. See panels (g)–(i).
- Performance in the long run: GOLiQ is a more effective approach in the long run because the LW solution remains static, and its error grows linearly as time increases.

Remark 11 (Different Philosophies: Online Learning vs. Heavy Traffic). We emphasize that online learning and heavy-traffic analysis are two methodologies developed based on distinct philosophies. First, when the system size is large, heavy-traffic models are able to produce high-fidelity solutions, but they require more prior knowledge of the system as inputs. On the other hand, online learning requires less prior understanding of the system because the data-driven nature allows it to dynamically evolve and improve (whereas heavy-traffic solutions are static). Second, the notions of asymptotic optimality are different. As an approximate method, heavy-traffic analysis is said to be asymptotically optimal in the sense that, as the system size grows large, its solution becomes close to the true optimal solution. On the other hand, the solution of the online learning method converges to the true optimal solution as the server's experience accumulates (by serving more and more customers).

6.4. A *GI*/*GI*/1 Example

So far, our numerical experiments focus on the $M/GI/1$ examples. In this section, we test GOLiQ using a $GI/GI/1$ model. Specifically, we consider an $E_2/H_2/1$ queue with Erlang-2 (the E_2) interarrival times and hyper-exponential (the H_2) service times with $c_s^2 = 2$, for which we solve the optimal price with the service rate held fixed (as in Section 6.1.1).

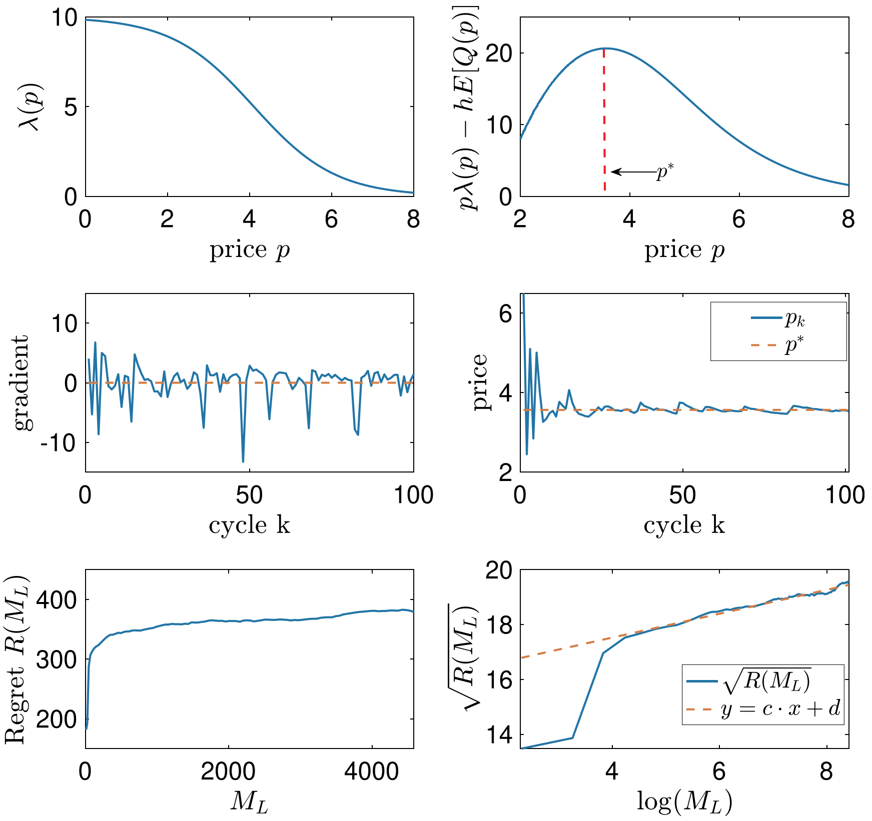
Because there is no closed-form solution for the performance function of the $E_2/H_2/1$ system, we model the queue length process as a quasi–birth-and-death process and adopt the matrix-geometric method (Latouche and Ramaswami 1999) to numerically solve the optimal solution p^*. Here, we continue to use the logit demand function (24) with $M = 10$, $a = 4.1$, $n = 1$, $h_0 = 1$, and $\mu = \mu_0 = 10$. This gives the optimal price $p^* = 3.567$ (top right panel of Figure 8). The algorithm hyperparameters are $\eta_k = 1/k, D_k = 10 + 10\log(k)$, and $p_0 = 6.5$ (which are identical to those as in Section 6.1.1). From Figure 8, we observe that, although the objective function remains a nonconvex function, GOLiQ continues to perform well with fast convergence.

An additional $LN/LN/1$ example with log-normal interarrival times and service times is given in Online Section EC.4.3.

7. Conclusion

In this paper, we develop an online learning framework designed for dynamic pricing and staffing in queueing systems. The ingenuity of this approach lies in its online

Figure 8. (Color online) Online Optimal Pricing for an $E_2/H_2/1$ Queue with Fixed Service Rate with $\mu_0 = 10, a = 4.1, p_0 = 6.5$, $p^* = 3.567$, $\eta_k = 1/k$, and $D_k = 10 + 10\log(k)$



Note. See (i) demand function (top left), (ii) revenue function (top right), (iii) sample path of the gradient (middle left), (iv) sample path of the price (middle right), (v) estimated regret (bottom left), (vi) square root of regret versus logarithm of served customers with $c = 0.43$, $d = 15.79$ (bottom right).

nature, which allows the service provider to continuously obtain improved pricing and staffing policies by interacting with the environment. The environment here is interpreted as everything beyond the service provider's knowledge, which is the composition of the random external demand process and the complex internal queueing dynamics. The proposed algorithm organizes the time horizon into successive operational cycles and prescribes an efficient way to update the service provider's policy in each cycle using data collected in previous cycles. Data include the number of customer arrivals, waiting times, and the server's busy times.

A key appeal of the online learning approach is its insensitivity to the scale of the queueing system as opposed to the heavy-traffic analysis, which requires the system to be in large scale (with the arrival and service rates both approaching infinity). Effectiveness of our online learning algorithm is substantiated by (i) theoretical results, including the algorithm convergence and regret analysis, and (ii) engineering confirmation via simulation experiments of a variety of representative $GI/GI/1$ queues. Theoretical analysis of the regret bound in the present paper may shed lights on the design of efficient online learning algorithms (e.g., bounding gradient estimation error and controlling proper learning rate) for more general queueing systems.

There are several venues for future research. One natural extension would be to develop new regret analyses that do not require the uniform stability condition. Another interesting and promising direction is to develop an online learning method without assuming the knowledge of the arrival rate function $\lambda(p)$, where the learner (hereby the service provider), during the interactions with the environment, has to resolve the tension between obtaining an accurate estimation of the demand function and optimizing returns over time. A third dimension is to extend the methodology to more general model settings (e.g., queues having customer abandonment and multiple servers), which make the framework more practical for service systems such as call centers and healthcare. In this regard, results in the present paper may serve as useful foundations; in particular, Theorems 1 and 2 help construct desired regret bounds as long as their associated conditions can be verified. Doing so usually requires two main steps in a new queueing model: (i) proving a new ergodicity (or rate of convergence to stationarity) result that can be used to bound the regret of nonstationarity and (ii) designing a new gradient estimator, which is easily computed from data (here, a good gradient estimator should have small bias and variance subject to conditions in Theorem 2).

Acknowledgments

The authors thank the editors and anonymous referees for their helpful comments.

References

Ata B, Shneorson S (2006) Dynamic control of an M/M/1 service system with adjustable arrival and service rates. *Management Sci.* 52(11):1778–1791.

Besbes O, Zeevi A (2015) On the (surprising) sufficiency of linear models for dynamic pricing with demand learning. *Management Sci.* 61(4):1211–1224.

Blanchet J, Chen X (2015) Steady-state simulation of reflected Brownian motion and related stochastic networks. *Ann. Appl. Probab.* 25(6): 3209–3250.

Broadie M, Cicek D, Zeevi A (2011) General bounds and finite-time improvement for the Kiefer-Wolfowitz stochastic approximation algorithm. *Oper. Res.* 59(5):1211–1224.

Burnetas AN, Smith CE (2000) Adaptive ordering and pricing for perishable products. *Oper. Res.* 43(3):436–443.

Chen X (2014) Exact gradient simulation for stochastic fluid networks in steady state. Tolk A, Diallo SY, Ryzhov IO, Yilmaz L, Buckley S, Miller JA, eds. *Proc. Winter Simulation Confernce 2014* (IEEE, Piscataway, NJ), 586–594.

Chong EKP, Ramadge PJ (1993) Optimization of queues using an infnitesimal perturbation analysis-based stochastic algorithm with general update times. *SIAM J. Control Optim.* 31(3):698–732.

Dai JG, Gluzman M (2021) Queueing network controls via deep reinforcement learning. *Stochastic Systems* 12(1):30–67.

Fu MC (1990) Convergence of a stochastic approximation algorithm for the GI/G/1 queue using infinitesimal perturbation analysis. *J. Optim. Theory Appl.* 65(1):149–160.

Glasserman P (1992) Stationary waiting time derivatives. *Queueing Systems* 12:369–390.

Huh WT, Rusmevichientong P (2013) Online sequential optimization with biased gradients: Theory and applications to censored demand. *INFORMS J. Comput.* 26(1):150–159.

Huh WT, Janakiraman G, Muckstadt JA, Rusmevichientong P (2009) An adaptive algorithm for finding the optimal base-stock policy in lost sales inventory system with censored demand. *Math. Oper. Res.* 34(2):397–416.

Jia H, Shi C, Shen S (2021) Online learning and pricing for service systems with reusable resources. Working paper.

Kim J, Randhawa RS (2018) The value of dynamic pricing in large queueing systems. *Oper. Res.* 66(2):409–425.

Krishnasamy S, Sen R, Johari R, Shakkottai S (2021) Learning unknown service rates in queues: A multiarmed bandit approach. *Oper. Res.* 69(1):315–330.

Kumar S, Randhawa RS (2010) Exploiting market size in service systems. *Manufacturing Service Oper. Management* 12(3):511–526.

Kushner HJ, Yin GG (2003) *Stochastic Approximation and Recursive Algorithms and Applications* (Springer, New York).

Latouche G, Ramaswami V (1999) *Introduction to Matrix Analytic Methods in Stochastic Modeling* (SIAM, Philadelphia).

L'Ecuyer P, Glynn PW (1994) Stochastic optimization by simulation: Convergence proofs for the GI/GI/1 queue in steady state. *Management Sci.* 40(11):1562–1578.

L'Ecuyer P, Giroux N, Glynn PW (1994) Stochastic optimization by simulation: Numerical experiments with the M/M/1 queue in steady-state. *Management Sci.* 40(10):1245–1261.

Lee C, Ward AR (2014) Optimal pricing and capacity sizing for the GI/GI/1 queue. *Oper. Res. Lett.* 42(8):527–531.

Lee C, Ward AR (2019) Pricing and capacity sizing of a service facility: Customer abandonment effects. *Production Oper. Management* 28(8):2031–2043.

Liu B, Xie Q, Modiano EH (2019) Reinforcement learning for optimal control of queueing systems. Liberzon D, Dominguez-Garcia A, eds. *57th Annual Allerton Conf. Communication, Control, and Computing* (IEEE, Piscataway, NJ), 663–670.

Maglaras C, Zeevi A (2003) Pricing and capacity sizing for systems with shared resources: Approximate solutions and scaling relations. *Management Sci.* 49(8):1018–1038.

Nair J, Wierman A, Zwart B (2016) Provisioning of large-scale systems: The interplay between network effects and strategic behavior in the user base. *Management Sci.* 62(6): 1830–1841.

Shah D, Xie Q, Xu Z (2020) Stable reinforcement learning with unbounded state space. Preprint, submitted June 8, https://doi.org/10.48550/arXiv.2006.04353.

Sutton RS, Barto AG (2018) *Reinforcement Learning: An Introduction*, 2nd ed. (The MIT Press, Cambridge, MA).

Yuan H, Luo Q, Shi C (2021) Marrying stochastic gradient descent with bandits: Learning algorithms for inventory systems with fixed costs. *Management Sci.* 67(10):6089–6115.

Zhang H, Chao X, Shi C (2020) Closing the gap: A learning algorithm for lost-sales inventory systems with lead times. *Management Sci.* 66(5):1962–1980.

Xinyun Chen is an assistant professor in the school of data science at the Chinese University of Hong Kong, Shenzhen. Her research interests include stochastic simulation, queueing theory, and reinforcement learning with applications in healthcare and telecommunication networks.

Yunan Liu is an associate professor in the Fitts Department of Industrial and Systems Engineering at North Carolina State University. His research interests include queueing theory, stochastic modeling, applied probability, simulations, optimal control, online learning, and their applications in call center and healthcare. His work was awarded first place in the INFORMS Junior Faculty Interest Group Paper Competition in 2016.

Guiyu Hong is a PhD student in the school of data science at the Chinese University of Hong Kong, Shenzhen. His research interests include queueing theory, online learning, stochastic modeling, and applications in telecommunication networks.