Pricing in Network Revenue Management Systems with Reusable Resources

by

Armando Bernal

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Industrial and Operations Engineering)
at the University of Michigan
2020

Doctoral Committee:

Associate Professor Cong Shi, Chair
Professor Romesh Saigal
Assistant Professor Joline Uichanco
Professor Mark P. Van Oyen

Armando Bernal

abernal@umich.edu

ORCID iD: 0000-0002-3482-1910

To my beloved wife, sister, parents, and to those
underrepresented students from under-resourced communities.

Acknowledgements

First, I would like to express the deepest gratitude to my academic advisor Professor Cong Shi. Without him taking me in as his student, I would have definitely not finished my PhD. Not only did he go above and beyond in guiding and supporting me through my doctorate study but he also believed in my abilities even when I did not believe in them myself. He showed necessary patience for me to make progress as I was new to the whole PhD process. He pushed me to be the better version of my academic but also personal self. For that, I cannot thank him enough for his support, patience, and advice that he has provided throughout my doctorate studies. I will forever be indebted to him.

I would also like to thank my committee members Professor Romesh Saigal, Professor Mark Van Oyen, and Professor Joline Uichanco for their helpful discussions and feedback.

I would be remiss if I did not acknowledge all the wonderful colleagues, especially the 2014 cohort, I have met along the way, who helped me in various ways, without them, homework, tests, and projects would have been immensely tougher. Their impact on me academically is incalculable, but also they made the overall experience joyful and have made lifelong friends.

Last but not least, I would like to acknowledge my beautiful wife, whom I married while at the university, Mariana Caballeros, for her abounding patience and just being there throughout my academic journey. I strongly believe that without her, I might have not pursued graduate school. She pushed me and made me give it my all in my doctorate studies without saying a word. She sacrificed a lot and for that I thank her. I love you and thank you. My PhD is dedicated to you, my wonderful parents, and my little sister. I hope I have made everyone proud.

Table of Contents

ABSTRACT

This thesis focuses on the problem of pricing reusable products in the network revenue management setting. In a nutshell, dynamic pricing problem concerns pricing and selling a finite inventory of products within a given time horizon so as to maximize the total revenue. Most of the existing literature studies the setting with perishable products in which the products sold are permanently removed from inventory. In this thesis, we tackle a different and arguably more challenging problem with reusable products wherein the products are returned back to the seller upon serving a customer and can be used to serve another customer. This class of problems finds a broad range of applications including hotel management, cloud computing, workforce management, call center service, and car rental management.

In the first chapter of the thesis, we address the pricing of reusable resources with advance reservation when the demand function is known as a function of price and the demand follows a Poisson point process. We demonstrate that a simple static pricing policy is asymptotically optimal when demand and capacity are scaled without bound. The performance of the policy is measured as a ratio with respect to the policy that does not exhibit any blocking. We also show that the static policy becomes optimal at a rate close to $1/\sqrt{n}$, where n is a scaling factor. Simulation results show the asymptotic behavior but additionally, it shows that for small-scaled systems, the static pricing policy performs very well relative to the no-blocking policy.

In the second chapter of the thesis, we consider the learning variant of the same problem in which the customer's response to selling price and the demand distribution are not known a priori. Connecting this problem to multi-armed bandits (MAB), we propose a variant of the upper confidence bounds (UCB) algorithm. The setting is different from literature in that capacity constraints exist and booking profile is dynamically updated. We solve an LP in every period where the UCB estimates guides the right-hand side parameter and outputs a distribution over the finite pricing actions. We demonstrate that for some large scaling factor n, with high probability, the seller will always choose the optimum after the testing phase and will not exhibit any blocking.

In the third and final chapter of the thesis, we employ unsupervised learning methods to tackle the pricing policy from a practical point-of-view. In particular, model-free reinforcement

learning method is used to implicitly learn the transition dynamics that governs the reward process to maximize revenue. Deep neural networks are used to parametrize the action policy and value function and through a simulated environment. We show that the generated pricing policy, using purely data, achieved good performance with respect to traffic, revenue, and blocking.

List of Figures

List of Tables

CHAPTER 1

Introduction

Many problems in revenue management can be characterized as the difficulties in resource allocation under stochastic environments in which the operator seeks to allocate limited resources efficiently as possible. This problem can be seen in all aspects of life. In industries that have inventory, optimal allocation of products is needed to balance costs that can be incurred from products not selling, from holding costs to cost associated space that could have been used for other higher-generating revenue products. In whichever setting, the operator faces these challenges due to the limited quantity of resources available and the uncertainty inherent in the dynamics of the underlying system that might be complex to model. In this thesis we consider a few sources of randomness: (1) randomness in the demand process, (2) randomness in the customers' advance reservation period of products, i.e., how long in advance purchasing customers will reserve, (3) randomness in customers' length of product usage. The former owes to the various factors such as competition, product substitutability, operator market size, customers' selection process, which is unobservable. The first two chapters of this thesis develops theoretical tools to manage the aforementioned difficulties when the operator has limited resources and the resources in question are reusable.

One common tool used by businesses to allocate limited quantity of resources is via pricing. The simplest pricing mechanism is to set a price based on future expected demand made in the *present* state. A more complex tool is via dynamic pricing where prices are adjusted to reflect the state, which is continuously evolving. Dynamic pricing in the revenue management literature has seen a tremendous increase due to the massive amounts of data that companies are collecting thru various technologies that are able to capture data at a very granular level to understand their customer base and target them according to their likes. We do not develop models that take contextual features as inputs but this area of research is becoming more relevant as technologies are continuously improving to develop better predictive models. Chapter 2 of this thesis only takes sequential arrival and made purchases information into account to update the model parameters and the pricing policy.

In the traditional pricing setting, the operator has a predefined quantity of limited resources at her disposal to sell to customers and has to make pricing decisions in the face of unknown incoming demand. The goal of the operator is to employ a strategy to meet business objectives and maximize overall revenue. In much of the literature in revenue management, the prevailing assumption is that the resources are perishable, in other words, the resources are consumable such as clothing, foods, electronics. Every time a purchase of a perishable product is made, the product is consumed by the customer and results in the loss of the underlying resource.

However, in other settings, this is not the case. In these settings, the resources are "renewable", or reusable. This is the case for example, the lodging industry, car-rental industry, cloud computing resources, work staffing. In such settings, the customer makes a purchase to use the resource for a customer-specified service duration, and then the resource is returned to the operator upon completion for reuse. The uncertainties in this setting are the length of time the resource is to be utilized by each customer and the future point in time the customer will begin resource utilization. This is in contrast to the perishable setting where the resource is assumed to be lost at the point of purchase. In the reusable setting, the purchase might be made at a point in time that differs to the time at which the customer begins utilizing the resource. This introduces additional complexity to the traditional perishable model since, in addition to the risk of future demand, the random future state of the available resources must be accounted for when making pricing decisions.

1.1 Overview of Thesis

This thesis is divided into three sections with each addressing a distinct aspect of the network revenue management pricing problem with reusable resources.

In Chapter 2, we address the problem of pricing of a monopolist firm who has at its disposal the time-homogeneous demand function which only depends on price under continuous advance reservation and service time distributions with finite support. We demonstrate that a static fixed-pricing policy derived from a linear program achieves great results with respect to a policy that has infinite resources, and thus will never deny any purchasing customer service. The static pricing policy achieves optimal revenue as the demand and initial capacity get scaled without bound at a rate arbitrarily close to $O(1/\sqrt{n})$. Our computational results displays the $O(1/\sqrt{n})$ behavior. Additionally, even though nothing rigorous can be said about the behavior for finite instances of the problem, the fixed-pricing policy achieves at least 75% of the optimal revenue. The data we used is simulated with different distributions and load factors to showcase its robustness to varying uncertain environments.

Motivated by the fact that in practice the seller does not have demand information and has

to learn it over time by exploring pricing policies, in Chapter 3 we include more dynamics into the pricing problem. Fixed prices are indeed preferred by companies in practice even though the optimal pricing path might not be a fixed-price policy. But owing to the nature of how the world is evolving today with massive amounts of data in the hands of the retailers, we want to make pricing decisions as new data streams are collected over time to make improved pricing decisions that lead to better overall regret. The performance of the pricing policy is measured using regret with respect to the firm who has an unlimited quantity of resources. In the setting with no advance reservation and deterministic service time with a fixed finite set of prices to choose from, we show that after a sufficiently large n, with high probability, the firm's only regret comes initially from testing each price *and* will choose the optimal price. The computational results show this behavior and does not exhibit any blocking events.

In Chapter 4, motivated in a broad sense to relax the strong assumption of the monopolist firm and use unsupervised machine learning methods to develop good performing dynamic pricing policies without the use of a *model*. This chapter diverges from the previous theoretical nature of the previous chapters and is a simulation study to show that we can leverage a *model-free* reinforcement learning to tackle the pricing problem of reusable resources. We show using deep neural networks on conjunction with reinforcement learning algorithms, that we can develop good pricing policies that perform well with respect to a combination of relevant measures, such as revenue, traffic, and blocking. The framework is extremely modular to accommodate different performance measures by playing with the reward structure to reward (penalize) the agent in certain ways to achieve a desired goal. Similarly as before, the data was simulated and the performance was measured with respect to the fluid policy derived from a linear program and the parameters are the means. In contrast to before, the learning algorithm does not know any information about the system in advance other than the generated data. The fluid model was derived using information that the seller does not possess, or at best, have an approximation to the statistics to the environment. The simulated results show that the dynamic pricing policy from the learning algorithm did well compared to many other static policies in term of revenue, traffic, and blocking.

Finally, we conclude with chapter 5 by providing summary remarks and proposing interesting directions for future work. The more technical proofs for each chapter are provided in the associated appendices.

In the following section, we introduce ideas and references that are broadly applicable throughout this thesis.

1.2 Literature Review

We will introduce references related to the general problems of pricing and dynamic pricing variations thereof. Subsequently, we introduce references related to reusable resources in revenue management. Lastly, we will provide references related to model-free reinforcement learning in the revenue management realm. First, we will introduce literature that is relevant for analyzing the blocking probabilities in the network revenue management with reusable resources, this stream of literature is regarding loss network systems.

1.2.1 Loss Network Systems

Loss network systems without advance reservation have been extensively studied, primarily in the context of communication networks (e.g., the survey paper by Kelly (1991)). In this setting, signals/calls arrive to the communication network as a Poisson process, and the call is satisfied immediately if there are sufficient channels to connect the call, otherwise, the call is lost. Virtually, no attempt was made to consider blocking probabilities in communication networks with advance reservation. The major problems in the literature on loss network systems have been the design of heuristics for admission control (e.g., Miller (1969), Ross and Tsang (1989), Key (1990), Kelly (1991), Hunt and Laws (1997), Puhalskii and Reiman (1998), Fan-Orzechowski and Feinberg (2006)), and the development of approximations and bounds as well as sensitivity analysis of blocking probabilities with respect to input parameters and resource capacities (e.g., Erlang (1917), Sevastyanov (1957), Kaufman (1981), Burman et al. (1984), Whitt (1985), Kelly (1991), Ross and Yao (1990), Zachary (1991), Louth et al. (1994), Kumar et al. (1998), and Adelman (2006)).

From an admission control stand-point, there have been few successful endeavors to analyze loss network systems with advance reservation. Virtamo and Aalto (1991) analyzed slotted systems in which the start time is uniformly distributed over the horizon and Luss (1977) derived a model and analyzed performance metrics of certain admission policy. Our work departs from the above literature in that we don't allow service interruptions, which can be seen as a drawback, but in our intended applications, interruptions are not allowed (e.g., hospitality industry, car rental industry, etc.). Additionally, their algorithms require certain blocking probabilities which they don't determine, but do estimate via simulation, whereas we derive explicit upper bounds on the blocking probabilities and study their asymptotic behavior to deduce optimality of our ϵ-PSP policy.

Another stream of literature studies the property of the optimal admission control of loss system, including Miller (1969), Kelly (1991), Altman et al. (2001), Örmeci et al. (2001), Savin et al. (2005), Gans and Savin (2007), Papier and Thonemannm (2010), and Jain et al. (2015). However, none of them devise heuristics that are practical and provably-good. The most relevant

prior work in loss network systems with advance reservation are Levi and Radovanovic (2010) and Chen et al. (2017), where both used a simple knapsack-type linear program (LP) to devise a simple admission control policy called class selection policy for models without advance reservation Levi and Radovanovic (2010), and generalized to the advanced booking setting Chen et al. (2017).

Systems with advance reservation and deterministic sequence of arrivals have been studied extensively in the appointment scheduling literature (for an excellent survey see Gupta and Denton (2008)). The objective is mainly the minimization of costs due to idling resources and waiting times (see Kaandorp and Koole (2007), Begen and Queyranne (2011), Ge et al. (2013), Begen et al. (2012), Kong et al. (2013), Mak et al. (2014)), where resources can be an individual such as a physician and an idling physician is not generating revenue. The methods that are typically used are stochastic optimization and/or dynamic programming and usually have to make impractical assumptions on the arrival process, discreteness of the service times to allow for an intractable model, whereas our method incorporates the stochasticity of the arrival, reservation, and service time process.

Our setting relaxes many of the drawback of the previous works where we allow continuity in the sequence of arrivals instead of a deterministic sequence of arrivals, we allow continuity in advance reservation times. In practice, advance reservations are almost always discrete, i.e., in the car rental industry pick-up times are in increments of 30 minutes from open to closing times and in the hotel industry there is only one check-in time per day. Regardless, we wanted to extend Chen et al. (2017) to continuous time distributions instead of bounded support finite distributions. We wanted to see if the policy robustness still holds for varying degrees of distribution skewness and means since in practice one rarely has a hold of the true distributions and errors in the estimations at best. Continuity in the service time and advance reservation distribution adds a layer of complexity since now we have to make sure interchange of integrals holds to estimate the true arrival process and departure process but also how to bound the true blocking probability.

1.2.2 Pricing in Revenue Management

Revenue management has been a robust area of research as it has seen widespread applications including, but definitely not exhaustive, lodging and car rental industry, cloud computing, and workforce management industry. There is a significant amount of literature that provides an overview of the theory and practice of revenue management (e.g., extensive surveys by Bitran and Caldentey (2003a), Talluri and van Ryzin (2005), Özer and Phillips (2012), den Boer (2015)).

Levi and Radovanovic (2010) consider a class selection model without advance reservation, but instead the seller is exogenously given the prices each customer class pays for the resources

and so the seller's problem is to choose which classes it sells to so as to maximize the long-run average revenue. They use a knapsack-type LP which is used to derive their policy, called *class selection policy* (CSP). They show that the CSP policy is asymptotically optimal regardless if the customers either all take-up exactly one resource or differs between customer classes as long as the ratio $C/A^7 \to \infty$, where C is the capacity and A is the maximum resource required by a class.

From a pricing point-of-view, the seminal work by Gallego and van Ryzin (1994) is a pricing problem where the seller is faced with pricing a fixed inventory of a single perishable product over a finite time horizon, at which point selling stops and the leftover products are sold for a salvage value. They derive sufficient conditions for which the optimal value function satisfies, which is the Hamilton-Jacobi differential equation. They derive the optimal policy for a simple case, which turns out to be non-implementable in practice because the optimal pricing trajectory has to be dynamically updated at every point in time. But they show that a single fixed-price policy is asymptotically optimal and the performance of the fixed-price policy relative to the seller's optimal revenue decays as $1/\sqrt{n}$. Our work differs in that we are tasked with finding a pricing policy for *re-usable*, instead of perishable, resources in an uncertain environment, where the seller is additionally burdened with not knowing how long nor what time in the future customers will use the resource.

Talluri and van G. Ryzin (1998) study a network revenue management model where the seller sells multiple products produced from multiple resources and has no assumptions on the network structure and demand arrival process. The seller's aim is to maximize their expected revenue by deciding to sell or not sell a product to a customer. They propose a *bid-price control* policy derived from an LP model,. They show that as the capacity and time-horizon grow linearly, the bid-price policy is asymptotically optimal with rate $O(1/\sqrt{n})$. In their work, they study products that can be used once by a customer, where as in this thesis, we study products that are re-usable. Another difference is that in our work, we allow customers to reserve in advance, where as in Talluri and van G. Ryzin (1998) assume customers demand is satisfied immediately.

Gallego and van Ryzin (1997) study a revenue management in which the seller has fixed inventory of multiple resources to produce different products and a fixed time horizon, and the goal is to price the products to maximize expected revenue when the arrival process for each product is a Poisson process. They show that a *make-to-order* policy, which is a policy of producing products as they arrive, thus the policy name, until any of the resources required to make the products is not sufficient. The policy then prices the products according to a pricing function derived from a simple functional optimization, is asymptotically optimal as the resource capacities and arrival rates of each product are scaled. But this policy is for the finite time horizon, not infinite horizon, as in Talluri and van G. Ryzin (1998). Our work in the second chapter departs from their results

by allowing advance reservation with reusable resources. Similar to the revenue management literature mentioned above, we theoretically analyze the performance loss of our policy and prove asymptotic optimality.

Literature closely related to this thesis are Chen et al. (2017) and Lei and Jasin (2016a). Chen et al. (2017) considers a revenue management model in the admission control setting where customers arrive to the system as a Poisson process and inform the seller the service time and future time they intend to use the resource, where the service time and advance bookings are discrete and bounded, i.e. $s \in \{1,\ldots,u\}$ and $d \in \{0,\ldots,v\}$. The seller has to maximize the long-run average revenue when there are M classes, where each class pays a certain amount and is given to the seller ahead of time. They proposed the ϵ-class selection policy (ϵ-CSP), in which they showed asymptotic optimality in applying this policy.

Lei and Jasin (2016a) also considers a revenue management model from a pricing control standpoint. Their model assumes that customers arrive randomly over finite time according to a specified non-stationary rate, and can reserve *t nonrandom* time units in advance for a fixed service time known *a priori*. The seller's objective is to set the price dynamically to maximize its expected total revenues over the finite selling horizon, T. They propose two policies. First, they propose a static policy called *deterministic price control-Batch* (DPC-Batch) which they prove has an average regret in the order of $O\left(n^{-2/3}\sqrt{log(n)}\right)$, where n is the service time, or the number of periods the resource will be in use, and is the same for all customers, J_A^D is the optimal revenue from the deterministic optimization problem, and $\mathbb{E}[R^\pi]$ is the revenue received from applying DPC-Batch. The second policy updates the price dynamically based on past prices and demand observations and has a similar average regret order.

Our setting throughout the thesis, except chapter 3, departs from analyzing one product, or resource, because we analyze a network revenue management system where multiple reusable resources makes multiple products. This model contains the special case where only products are being sold, in other words, the resource is the product. The blocking probability analysis is complex for systems with multiple products since multiple products can use the same resource and so the blocking probabilities are not independent. Chapter 2 breaks down the blocking probabilities per resource and chapter 4 doesn't explicitly handle blocking probabilities, but learns how to price the products so that a good pricing policy generates close to optimal revenue. Pricing via model-free learning affords us the ability to tweak the reward terms so the agent can control how much blocking he/she tolerates so that revenue and traffic increases.

1.2.3 Reinforcement Learning

To the best of the author's knowledge, there exists no literature in the network revenue management with reusable resources using machine learning, in particular, reinforcement learning. There is limited literature in the area of revenue management using reinforcement learning to find an optimal pricing policy in the perishable case. Gosavi and Bandla (2002) used reinforcement learning to develop a strategy for seating allocation and overbooking in order to maximize the average revenue gained by an airline. In particular, Raju et al. (2006) used a reinforcement learning (Q-learning) algorithm to price products dynamically with customer segmentation.

Model-free reinforcement learning has been successfully applied in robotics Peters and Schaal (2006), machine scheduling Ye et al. (2018), playing Atari games Mnih et al. (2013), cybernetics, psychology, and computer science disciplines Sutton and Barto (1998). There has been a surge of interest in model-free reinforcement learning after it was successfully applied to learn to play many old Atari video games Mnih et al. (2013), using one generic structure with deep neural networks and Q-learning. Model-based reinforcement learning solves for the optimal policy using past experience. An advantage in using reinforcement learning is that it can adapt to a changing environment through experience.

In chapter 3, we add to the growing applications of model-free reinforcement learning. Chapter 3 proposes using model-free reinforcement learning to find a good dynamic pricing policy using available resources left as state information, without having to define the transition probabilities that govern this stochastic process. We will be optimizing over the policy space directly instead of optimizing over the action-value space as it has been shown in practice to have better convergence properties at the expense of taking longer to train. We used DDPG instead of any other policy gradient algorithms because they are stochastic, in other words, the output is a stochastic policy. A stochastic policy does not make sense in practice as it will produce different prices for the same state since we are sampling a distribution.

CHAPTER 2

Network Revenue Management with Reusable Resources and Advance Reservations

2.1 Introduction

In this chapter, we consider revenue management problems with reusable resources that make reusable products and the ability for customers to reserve the resources in advance to be used for a fixed period of time, both *unknown* to the seller. Revenue management has received considerable attention in wide-ranging domains such as hospitality industry, car rental industry, hospital room management, workforce management, cloud computing, etc. Many of the aforementioned applications have many commonalities we consider from a practical point-of-view. One is the starting finite capacity that the seller has at any point time, for example, the finite computing capabilities that Google has to offer to clients, finite number of channels in a communications network, the finite number of rooms that a hotel manages, and the number of cars that a car renting company manages. Second, it's the re-usability of these resources, in other words, the resources are "consumed" temporarily for a random amount time at some random future date, but then becomes available to be used again for future requests. Third, the resources can be booked in advance for later use, e.g. hotel rooms can be booked months in advanced. Lastly, the seller has to decide the prices of these resources before customer arrivals.

The above characteristics are seller-related. In these settings, the resources serve multiple customer classes, each class having their unique characteristics. One of those characteristics is the *valuation*, reservation price, that a customer has *a priori*, not known to the seller, of the resource. The customer resource valuation is the fair-value price the customer thinks the resource is worth; it is different for each customer, and therefore, random. Other characteristic are the arrival process of customers and the service time and advance reservation requirements, which are also random. The last characteristic is that customers might leave the system if the seller-imposed price is not in par with the customers' resource valuation, i.e. the resource is overpriced that some customers won't

bother making a reservation. The seller's aim is to devise a pricing control policy to maximize his/her long-run average revenue. The seller has minimal information to accomplish this goal, i.e. complete distribution information of the advance reservation and service times and correlation between them and the reservation price distribution. Even if we did have complete information, in which case a dynamic programming model can be used to solve the problem, the *curse of dimensionality* would make it computationally intractable to solve because of the exponential blow-up of the state-space. But as we show in this chapter, in the fluid regime, much of the randomness becomes negligible, that is to say that the problem becomes deterministic, and a simple pricing policy derived from a convex optimization problem becomes nearly optimal as the capacity and arrival rates grow proportionally.

2.1.1 Main Contributions

We propose a static pricing policy called ϵ-perturbation pricing selection policy (ϵ-PSP), wherein the fluid regime, the ϵ-PSP policy defined later in the optimization problem, is nearly optimal. This policy has a very simple pricing structure that charges a single price for each resource over the infinite horizon, making the *implementability* feasible and easy, i.e. no dynamic updating, or extra computation is required after solving a single optimization problem nor is any additional distributional information or correlation information required to devise the policy, thus, the seller avoids the risk of miscalculating customer information.

Our main contributions about the performance of the ϵ-PSP are the following:

1. The heuristic applies a single price for each resource over the infinite horizon, where the price is chosen according to an optimization problem that constrains the seller to price in such a way that leaves a buffer that depends on ϵ, for the purpose of hedging against uncertainty. The policy is asymptotically optimal with rate arbitrarily close to $1/\sqrt{n}$.

2. The performance loss of the ϵ-PSP relative to the seller's optimal revenue is upper bounded by a function that does not require the seller to have perfect information of the distributions and correlations of the advance reservation and service times but does depend on the initial capacity, the price at which no customers arrive, and the optimal objective value of an convex optimization problem to be defined later.

3. Our numerical experiments show that the performance and *robustness* of ϵ-PSP. The policy performs nearly optimal when the capacity and arrival rates are sufficiently large but also performs relatively well even when the parameters are of modest size, both in the cases when the advance reservation and service times are dependent and independent.

We analyze the performance of the ϵ-PSP policy by first showing that the policy induces a well-studied stochastic process, the *loss network system*, namely an $M/G/C/C$ loss system with advance reservation. Loss network system are concerned with the setting in which customers are served when there is sufficient inventory in stock to fulfill the demand, otherwise, the customer leaves the system entirely if he/she finds the system at capacity. To get a hold on the steady-state blocking probabilities of the $M/G/C/C$ loss system, we analyze an infinite capacity system, or $M/G/\infty$ system, that upper bounds the steady-state blocking probability of the original system. Since we consider infinite support distributions, i.e. the advance reservation and service times, we have to upper bound the steady-state blocking probability of an infinite capacity system after a certain point in time, say t^*, in which we can then focus on the steady-state blocking probability on the *finite* interval $[0, t^*]$. There has been little work in attempting to understand the steady-state blocking probability in loss networks systems with advance reservation. As seen from the literature review below, Coffman-Jr et al. (1999), Lu and Radovanovic (2007), and Chen et al. (2017) are recent works who have characterized the steady-state blocking probabilities of special cases. The assumptions in our model that are pertinent for the analysis on the steady-state blocking probability are fairly general: a time-homogeneous arrival process, a general continuous service time density function with finite mean, and general continuous reservation density function with finite mean. The assumptions that are pertinent to devising a pricing policy is the concavity of the revenue rate function from the optimization problem. The main results (formally stated in Theorem 2.2.2 and Theorem 2.2.3) stem from the analysis of the steady-state blocking probabilities. One of the major reasons why it is difficult to analyze in the advance reservation setting, is the fact that an arriving customer effectively observes a non-homogeneous Poisson process induced by the *pre-arrivals*, i.e. the customers who have already reserved resources made prior to the customer arrival. We have contributed to the analysis of the virtual blocking probability in the case when one has *continuous* densities on unbounded support in $M/G/\infty$ systems with advance reservation. Our approach is similar in nature to Chen et al. (2017), but departs from their work by not only extending the results from discrete to the continuous setting, but we also extend from bounded to unbounded support on the service and reservation density functions. One of the drawbacks of our model is that we only consider single-class instead of multiple class.

This chapter is organized as follows: The beginning of Chapter 2.2 presents the model. Section 2.2.2 analyzes the performance of the ϵ-PSP policy. Section 2.2.1 analyzes the general network model's performance with multiple resources and multiple products. Section 2.2.3 presents empirical results of the ϵ-PSP policy.

2.2 Problem Formulations

We consider a seller who provides a set of finite reusable resources $I = \{1, 2, \ldots, m\}$ used to produce a set of products $J = \{1, 2, \ldots, n\}$ to serve customers over an infinite horizon. Demand for products at time t is a multivariate, stochastic point process with Markovian intensities. At any time t, the vector of intensities $\lambda_t = (\lambda_t^1, \ldots, \lambda_t^n)$ is determined by t and the current price vector $p_t = (p_t^1, \ldots, p_t^n)$ through a demand function $\lambda_t(p_t)$. Thus, demand is a controlled Poisson process. Following Gallego and van Ryzin (1997), we assume that the demand function $\lambda_t(p_t)$ is *regular* as follows.

Assumption 1 We assume the demand function $\lambda_t(p_t)$ is known, and further, that it satisfies the following regularity conditions.

a) $\lambda_t(p_t)$ is bounded, twice differentiable, and invertible.

b) For each j, there exists a "turn-off" price $\bar{p}^j < \infty$ such that $\lambda_t^j(\bar{p}^j) = 0$.

c) $\lambda_t^j \to 0$ implies that $\lambda_t^j \cdot p_t^j(\lambda_t^j) \to 0$.

d) The revenue rate $r_t(\lambda_t) = \lambda_t^\top p_t(\lambda_t)$ is continuous, bounded, and strictly concave in λ_t, and has an interior maximizer.

e) The function $p_t(\lambda)$ is non-increasing in λ for each t.

We make additional assumptions on the service time and delay marginal distributions.

Assumption 2

a) The marginal distributions $F_S(s)$ and $F_L(d)$ are differentiable.

b) There exists a point $u^* \in [0, \infty)$ for which $f_L(d)$ decays monotonically for $d \geq u^*$.

c) There exists a linear function $s^*(d) = c + ad$ where $\forall d \in [0, \infty)$ implies $f_{S|L=d}(s)$ decays monotonically for $s \geq s^*(d)$.

d) $M = \max_{d \in L} f_L(d) < \infty$ and $N = \max_{(s,d) \in S \times D} f_{S|L=d}(s) < \infty$.

Assumptions 1(a)-1(d) are well-known assumptions in the literature. The last assumption, though a theoretical drawback, in practice it is not an issue since if prices are high, less customers purchase and vice versa. In other words, a bigger arrival rate implies a lower price by invertibility assumption 2.2(a).

Assumption 2 is to ensure the interchange of differentiation and integration in the case the support of the distributions are unbounded. The case when both are finite, assumptions 2(b) and

2(c) are automatically satisfied. The first assumption is not restrictive since many distributions satisfy it. For example, all distributions that belong in the exponential family, beta prime distribution, F-distribution, and many more. In words, the third assumption states that the service time chosen by a customer who will delay service by d time units will likely choose a service time that is at *most* linear in the delay. In practice we won't expect that the longer the delay the bigger the service time. For example, a hotel manager wouldn't likely expect that a customer who reserves a room far out into the future would also choose a proportional service time. We would most likely have the service time not growing, i.e. concentrated, for any delay. The bivariate normal distribution with positive correlation coefficient fits this assumption, where we can take the linear function

$$s^*(d) = \mathbb{E}(S|D = d).$$

which is a linear function of d. A class of bivariate lognormal densities and bivariate F-distributions can be shown to satisfy assumption 2(c) and 2(d). The *pre-arrival* process will be described next.

We shall view the vector of intensities $\lambda_t = (\lambda_t^1, \ldots, \lambda_t^n)$ as the firm's decision variables. In this case, one can imagine the firm setting the output intensities λ and the market determining the prices p_t based on these output intensities. As price-sensitive customers *call* to the system, the seller offers a price p_t determined by the intensity λ_t. Depending on the product price, the arriving customer either requests the product for future use or if price is high enough, the customer leaves the system entirely. If the customer requests the product upon arrival, then seller reserves the product's resources for a fixed amount of service time at a future time, both of which are random quantities given by the customer from the outset. When a scheduled customer arrives in the future to *use* the product, the seller collects p_t, which is the price imposed at time t when the customer called for reservation and not the time at which the customer begins to use the product. Simultaneously, the resource(s) used to make the product will be occupied for a certain amount of time at which no one else can use the same resource(s). When the customer finishes with the product, the resource(s) will be available to be used again for future requests. We assume that the customer arrives on time and does not alter its request at any point between the time the customer's call to the system and its scheduled service

The n final products are made up by m types of reusable resources. We use $c = (c^1, \ldots, c^m) \geq \mathbf{e}$ to denote the capacity vector where c_i is the capacity of reusable resource $i \in I$. Let $A = [a_{ij}]$ represent the *bill-of-materials* matrix, where $a_{ij} \in \{0, 1\}$ represents whether resource $i \in I$ is required to make product $j \in P$. Also, let a_i be the i^{th} row of A. We assume A is binary-valued and has no zero columns; that is, each product uses at least one of the m resource types. Each arriving customer requests a service of product $j \in I$ in advance. The time between her request and the start of her service is called *lag time*, drawn from a continuous distribution L_j, and the time between the start

and the end of her requested service is called *service time*, drawn from a continuous distribution S_j. Note that L_j and S_j are *a priori* random to the system and only become realized at the moment when the request is made. We allow for an arbitrary correlation between L_j and S_j. If all the required resources of product j are available at the time of request, her reservation is successfully made at the current price; otherwise, she will leave the system. The resources of product j are released upon completion of her service and can be used to serve other customers (i.e., resources are reusable). The firm's objective is to maximize the long-run average expected total revenues by setting prices dynamically.

Let Π denote the set of all non-anticipating and state-dependent controls and let p_t^π denote the price to be applied at time t under control $\pi \in \Pi$. The optimal stochastic control formulation of our dynamic pricing problem is given by

$$\begin{aligned} J^* = \max_{\pi\in\Pi} \quad & \liminf_{T\to\infty} \frac{1}{T}\mathbb{E}^\pi\left[\int_0^T p_t^\top dN_s\right] \\ \text{s.t.} \quad & A(N(t)-D(t)) \le c, \quad \forall t \ge 0 \text{ a.s.} \\ & p_t \in P(t). \end{aligned} \tag{2.1}$$

where $N(t)$ is the arrival point process when applying the control π, and $D(t)$ is the departure process when applying the control π. The seller's objective is to find the policy $\pi \in \Pi$ which maximizes (2.1). Note that we maximize the *limit infimum* since the *limit* of the expected long-run average revenue might not exist, but the limit infimum always exists. Denote the optimal value of (2.1) as $\mathcal{R}(OPT)$. Following similar arguments as in Levi and Shi (2015), Lu and Radovanovic (2007), and Sevastyanov (1957), one can show that the induced Markov process has a unique stationary distribution which is ergodic. Invoking Little's Law and the existence of a long-run stationary distribution, whatever the state-dependent policy $\pi \in \Pi$ the seller uses, the arrival rate averages out to a single number for each product. Therefore, the *fluid* approximation/model is

$$\begin{aligned} J^D = \max \quad & \lambda^\top p(\lambda) \\ \text{s.t.} \quad & A\lambda \le c, \\ & \lambda \in \Lambda. \end{aligned} \tag{2.2}$$

The constraint $A\lambda \le c$ was attained from Little's Law. Specifically, since

$$A(L(t)-D(t)) \le c, \quad \forall t \ge 0 \text{ a.s.} \qquad \Rightarrow \qquad t^{-1}\int_0^t A(L(t)-D(t)) \le c \quad \forall t \ge 0.$$

Observe that the capacity vector c has different meanings in (2.1) and (2.2). c in (2.1) signifies that the number of each resources in use at time t is not greater than c for all times t. c in (2.2)

is interpreted on a per time basis, i.e. each resource's *rate* of usage is less than c per unit time. Additionally, note that (2.2) is a convex optimization problem since we are maximizing a concave function over a polytope.

Now we will show that a static pricing policy determined by (2.2) and inverting the function $\lambda(p)$ to find the prices that will be offered, is asymptotically optimal with rate arbitrarily close to $O(1/\sqrt{n})$.

2.2.1 Multi-Product Multiple-Resources with Advance Reservation

Before we delve into the analysis, we will define the necessary quantities that will be used throughout. If y is a vector in $\mathbb{R}^m$ and $W \subset I$, then y_W is the vector of length $|W|$ whose components are y_i for $i \in W$, and $|W|$ is the cardinality of the set W. If λ_j is the arrival rates for each product $j \in J$, then the arrival rate, α_i, for ingredient $i \in I$ is

$$\alpha_i = \sum_{j \in S_i} \lambda_j,$$

where $S_i := \{j \in J | a_{ij} > 0\}$, i.e., S_i are the products which use ingredient $i \in I$. Additionally, define M_j to be the ingredients that make-up product j.

We will solve the following perturbed version of (2.2)

$$\begin{aligned} J_\epsilon^D = \quad & \max \quad \lambda^\top p(\lambda) \\ & \text{s.t.} \quad A\lambda \leq (1-\epsilon)c, \\ & \qquad \lambda \in \Lambda. \end{aligned} \tag{2.3}$$

Let λ_ϵ^* be the optimal value of (2.3). Having defined the perturbed optimization problem, we have the following lemma whose proof is provided in the appendix:

Lemma 2.2.1. *Let λ_ϵ^* be the optimal solution to (2.3). Then the optimal objective value of (2.3) is at least $(1-\epsilon)$ times the optimal expected revenue, i.e.,*

$$J_\epsilon^D \geq (1-\epsilon)J^*.$$

The solution from solving the LP gives rise to the ϵ-*PSP* policy:

1) For each arriving customer, accept product request if there is sufficient unreserved resources available to make the product over the requested service interval. Otherwise customer is blocked.

2) Collect $p(\lambda_j^*)$ for product j purchases.

Assume now that a random customer in the *capacitated* system arrives to the system requesting product $j \in P$, then the probability of being blocked when the requested interval is $[t+d, t+d+s]$, denoted by $\mathbb{P}(B_j^{(d,s)})$, is upper bounded by

$$\begin{aligned}\mathbb{P}(B_j^{(d,s)}) &= \sum_{\pi \in \Pi(M_j)} \mathbb{P}(B_j^{(d,s)}|\pi)\mathbb{P}(\pi) \\ &\leq \sum_{\pi \in \Pi(M_j)} \mathbb{P}(\pi) \\ &\leq N \sum_{i \in I} \mathbb{P}(\pi_i),\end{aligned} \tag{2.4}$$

where π_i = {resource i is insufficient over [t+d,t+d+s]} and $\Pi(M_j)$ is the set of all combinations of ingredients that can run out at the time of reservation. For example if product 1 is made up of ingredients $\{1,2,3\}$, then $\Pi(M_1) = \{\{1\},\{2\},\{3\},\{1,2\},\{2,3\},\{1,3\},\{1,2,3\}\}$ are the possible ways that a customer can get blocked at the time of reservation since at the time of arrival and requested interval, if some combination of resource in $\Pi(M_1)$ are exhausted, then customer is blocked. The last inequality is due to summing over all ingredients $i \in I$. Going from the first to second inequality is due to the number of different ways that resources can run out for a single product. Using the same example as before, product 1 cannot be made if, say resource 2 and 3 are exhausted, so the overall blocking probability will include $\mathbb{P}(\pi_2 \cap \pi_3)$ and other terms as well. But note that trivially,

$$\mathbb{P}(\pi_2 \cap \pi_3) \leq \mathbb{P}(\pi_1) + \mathbb{P}(\pi_2)$$

Therefore, N is a finite constant and it is the maximum number of times that any of the $\mathbb{P}(\pi_i)$'s appear in $\Pi(M_j)$ for any $j \in J$. For example, using $\Pi(M_1)$ again, resources 1,2, and 3 appear four times in $\Pi(M_1)$. We check for all $\Pi(M_j)$ for all products and take the maximum over all products. It can easily be seen that

$$N = \max_i \left\{ \sum_{i=0}^{n_i} \binom{n_i}{i} \right\}$$

where $n_i = \sum_j A_{ij}$.

Observe that for each product $j \in J$ the last inequality holds since this was for arbitrary product. Looked thru a "queueing" lense, we can look at m $M/G/\infty$ queues for each resource *separately* and analyze the conditional blocking probability for each single resource. In this counterpart system, all customers are admitted since there are an infinite capacity for each resource. It is not hard to see that if a customer gets blocked in the capacitated queue, then there exists at least one

ingredient $i \in I$ that was insufficient. Then it must be that in the $|I|$ running $M/G/\infty$ queues, the customer would have been *virtually* blocked due to at least one of the queues. We will define the concept of *virtual blocking probability* for the analysis of the conditional blocking probability in the capacitated system. Therefore, we can look at the blocking probability of a product request as the sum of blocking probabilities of single ingredient requests, where the arrival rate of ingredient $i \in I$ is

$$\alpha_i = \sum_{j \in S_i} \lambda_j^*,$$

where λ^* is the solution to (2.3).

If we implement the constant pricing policy determined from (2.3), the revenue we obtain is not fully J_ϵ^D since there is the possibility of rejecting customers because the capacity is not enough. From Little's Law, the revenue from implementing the constant rate , i.e. fixed pricing policy, is

$$\mathcal{R}(\epsilon\text{-}PSP) = \sum_{i=1}^{n} \lambda_i p_i^* \int_{(S_i, L_i)} (1 - \mathbb{P}(B_{s,d})) s_i f_{S_i, L_i}(s_i, d_i),$$

where $B_{s,d}$ is the conditional blocking probability given that the customer chose to delay his service d time units into the future and use the product for s time units. If we obtain a uniform bound on $\mathbb{P}(B_{s,d})$, say $\mathbb{P}(B_{s,d}) \leq \gamma$, then we have

$$\mathcal{R}(\epsilon\text{-}PSP) \geq (1-\gamma) \sum_{i=1}^{n} \lambda_i p_i^* \mu_{S_i}.$$

Then observe that the term $\lambda_i \mu_{S_i}$ is the optimal rate determined from (2.3), i.e. $(\lambda_\epsilon^D)_i$. Lemma 2.2.1 then implies

$$\mathcal{R}(\epsilon\text{-}PSP) \geq (1-\gamma)(1-\epsilon) J^*.$$

Therefore, it is sufficient to find a uniform upper bound γ that vanishes to zero as $\epsilon \to 0$ to show that the ϵ-PSP policy is asymptotically optimal.

Theorem 2.2.2. *Consider a revenue management model (2.3) with $I = \{1, \ldots, m\}$ reusable resources and $J = \{1, \ldots, n\}$ products with advance reservation. The expected long-run average revenue loss of the ϵ-PSP relative to the optimal expected long-run average revenue has the following finite lower bound*

$$\frac{\mathcal{R}(\epsilon\text{-}PSP)}{J^*} \geq \left(1 - \sum_{r=1}^{m} \left(\frac{t_r^* - 1}{\alpha_r} - \sum_{i=0}^{t_r^*} \left(\frac{e^{\delta_{i,r}}}{(1+\delta_{i,r})^{(1+\delta_{i,r})}} \right)^{\alpha_r} - \frac{2c_r}{e^{2\alpha_r}} - \frac{e^{-\nu c_r}}{\sqrt{2\pi c_r}} l \right) \right) (1-\epsilon),$$

where $l = \sum_{j=0}^{\infty} (e\nu)^j$ *for some arbitrary* $\nu < \frac{1}{e}$, $\tilde{\alpha_r} = \min\{(1-\epsilon)c_r, a_r^\top \lambda^*\}$, $\delta_{i,r} = \left(\frac{\epsilon}{1-\epsilon} - \frac{\log(1+\tilde{\alpha_i})}{\tilde{\alpha_i} \log \theta_{i,r}^{-1}} \right)^+$, *and* $t_r^* \in \mathbb{N}$ *and* $\theta_{i,r} \in \mathbb{R}_{>0}$*'s are finite well-defined constants.*

Consider a sequence of problems indexed by $k \in \mathbb{N}$, defined by an initial capacity vector $c_k = kc$, $\lambda_k = k\lambda$, and $\epsilon_k = \frac{\epsilon}{\sqrt{n^{1-\beta}}}$ for $\beta \in (0,1)$. This generates a sequence of problems with proportionately larger sales volumes and initial stocks. It is not hard to see that if λ^* is the optimal value of (2.3) with initial stock c, then $\lambda_k^* = k\lambda^*$ for the k^{th} problems with inital stock kc. Denote the *true* optimal objective value of the scaled problem as J_n^*.

Theorem 2.2.3. *Consider a revenue management model (2.3) with* $I = \{1, \ldots, m\}$ *reusable resources and* $J = \{1, \ldots, n\}$ *products with advance reservation. For a sequence of problems where the* n^{th} *problem has parameters* $\lambda_j^{(n)} = n\lambda_j$ *for each product* $j \in J$, $c^{(n)} = nc$, *and* $\epsilon^{(n)} = \frac{\epsilon}{\sqrt{n^{1-\beta}}}$, *with* $\beta \in (0,1)$, $\nu < \frac{1}{e}$, *we have*

$$\frac{\mathcal{R}(\epsilon^{(n)}\text{-}PSP)}{J_n^*} \geq 1 - \frac{\epsilon m N}{\sqrt{n^{1-\beta}}} + o\left(\frac{1}{\sqrt{n^{1-\beta}}}\right).$$

2.2.2 Analysis of Blocking Probabilities

From the previous section's discussion, we will look at the conditional blocking probability of a capacitated system of a *single* customer class with a single product who arrives to the system as a Poisson process with rate λ, with joint distribution $F_{S,L}(s,d)$, marginal delay distribution $F_L(d)$ with marginal service time mean $\mu_S = \mathbb{E}[S]$, and marginal service time distribution $F_S(s)$ with marginal delay mean $\mu_L = \mathbb{E}[L]$. Additionally, assume that the initial capacity of the single resource is c and the traffic intensity

$$\rho = \min\{(1-\epsilon)c, \lambda\mu_S\}. \tag{2.5}$$

The steady-state blocking probability in the case when there is no advance reservation has a closed form (e.g. Levi and Radovanovic (2010)). This is due to the fact that the stochastic process in question, which can be equivalently described as a loss network queue, can be constructed from an infinite server queue system with a truncated state-space that includes only those states in which no more then C customers are in service Kelly (1991). The equilibrium distribution for the $M/G/\infty$ queue satisfies the detailed balance equations and the equilibrium distribution for the truncated stochastic process also satisfies the detailed balance equations. In the case when customers can reserve in advance, the number of customers who *arrive* by time t is no longer a Poisson process, but a non-homogeneous Poisson process (NHPP) Chen et al. (2017). "Arrive" is italicized because

the time the customer called to request a reservation is not the same time at which the customers begins to use the resource.

To analyze the blocking probability for the capacitated system, we consider the uncapacitated system, or the infinite server queue $M/G/\infty$, where all customers get to use a resource upon arrival. Note that the blocking probability in the uncapacitated system is at least as big as the blocking probability in the capacitated system. Indeed, suppose that a customer who arrives in steady-state at time t reserves d time units in advance for use of s time units, in other words, customer is requesting service in the time interval $[t+d, t+d+s]$, is blocked in the capacitated system, i.e. there are c resources that will be in use/reserved. Since *this* customer "sees" the capacity full, *this* customer in the $M/G/\infty$ would also "see" the system with *at least c* resources in use/reserved. Since the set of accepted customers in the $M/G/\infty$ system is a superset of the accepted customers in the capacitated system, for any sample path $\omega \in \Omega$, *this* customer would have been *virtually blocked*. Let B be the event $B = \{\max$ reserved capacity over $[t+d, t+d+s]$ in the capacitated system $\geq c\}$. Now, we define the *virtual blocking probability*, $\mathbb{P}(B_v)$, as

$$\mathbb{P}(B_v) = \mathbb{P}\big(\max \text{ reserved capacity over } [t+d, t+d+s] \text{ in the } M/G/\infty \text{ system} \geq c\big).$$

From the above, we have

$$\mathbb{P}(B_v) \geq \mathbb{P}(B).$$

Therefore, we will analyze upper bounds on the *virtual blocking probability*, $\mathbb{P}(B_v)$, to upper bound the true blocking probabilities. To that end, we need knowledge of the booking profile, i.e. the number of customers who are in service and the pre-arrivals who would be coming in later, at any point in time. We start by understanding the pre-arrival and departure process.

Upon a customer arrival to the system at some time t, all the starting service times of the customers who had arrived prior to time t are already known. We call these starting service times pre-arrivals. We define this process to analyze the virtual blocking probabilities. We derive the rate of the pre-arrival process, denoted by $Y(t)$, starting from *steady-state* by differentiating the mean $\mathbb{E}(Y(t))$ with respect to time t.

$$\begin{aligned}
\Lambda_Y(t) &= \frac{d}{dt}\left(\rho \int_{u=0}^{\infty} \mathbb{P}(u \leq L \leq u+t, \quad S < \infty)\right) \\
&= \frac{d}{dt}\left(\rho \int_{u=0}^{\infty} F_L(u+t) - F_L(u)\right) \\
&= \rho \bar{F}_L(t).
\end{aligned}$$

$\Lambda_Y(t)$ signifies the mean number of reserved customers t time units after a random customer arrival. Similar to the discrete case in Chen et al. (2017), the pre-arrival rate is $\rho\bar{F}_L(d)$, where ρ is the traffic intensity. The interchange above is allowed due to the following lemma

Lemma 2.2.4. $$\frac{d}{dt}\left(\int_{u=0}^{\infty} F_L(u+t) - F_L(u)\right) = \int_{u=0}^{\infty} \frac{d}{dt}\left(F_L(u+t) - F_L(u)\right) = \int_{u=0}^{\infty} f_L(u+t).$$

Proof. By Billingsley (1995), we only need to find an integrable function $g(u)$ such that $f_L(u+t) \leq g(u)$ for all $t, u \in \mathbb{R}_+$. Let $M = \max_{x\in[0,\infty)} f_L(x)$. Using assumption 3(i), the function

$$g(u) = \begin{cases} M & \text{for } u < u^* \\ f_L(d) & o/w, \end{cases}$$

is obviously integrable and upper bounds $f_L(u+t)$ for all $u, t \in [0,\infty]$ since for $u \geq u^*$ and $t \geq 0$ by assumption implies $f_L(u+t) \leq f_L(u)$. □

If we are dealing with continuous bounded support random variables, then the assumption is not required as we can bound the distribution functions by some large enough constant over the bounded support, which is obviously integrable.

We assume the support of the service time is $\mathbb{R}_+$. The departure process, which we denote $Z(t)$, is a NHPP with rate $\Lambda_Z(t) = \rho(1 - \mathbb{P}(S \leq t, L \leq t - S))$. Indeed, let us denote time 0 as the time the system reaches steady-state. Then the # of customers who will finish service in the time interval [0, 0+t] is a Poisson RV with mean

$$\begin{aligned}\mathbb{E}(Z(t)) = \rho \int_{u=0}^{\infty} \mathbb{P}(L \leq u, \quad u - L \leq S \leq u + t - L) \\ + \rho \int_{u=0}^{\infty} \mathbb{P}(u \leq L \leq u + t, \quad S \leq u + t - L).\end{aligned}$$

Taking the derivative of $\mathbb{E}(Z(t))$ and assuming that the interchanging of differentiation and integration holds, we get that the departure rate $\Lambda_Z(t)$ is

$$
\begin{aligned}
\Lambda_Z(t) &= \frac{d}{dt}\left(\underbrace{\rho \int_{u=0}^{\infty} \mathbb{P}(L \le u, \quad u-L \le S \le u+t-L)}_{A} + \underbrace{\rho \int_{u=0}^{\infty} \mathbb{P}(u \le L \le u+t, \quad S \le u+t-L)}_{B}\right) \\
&= \rho \frac{d}{dt}\left(\int_{u=0}^{\infty}\int_{d=0}^{u} \mathbb{P}(u-d \le S \le u+t-d | L=d) f_L(d) + \int_{u=0}^{\infty}\int_{d=u}^{u+t} \mathbb{P}(S \le u+t-d | L=d) f_L(d)\right) \\
&= \rho \frac{d}{dt}\left(\int_{u=0}^{\infty}\int_{d=0}^{u} \left(F_{S|L=d}(u-d+t) - F_{S|L=d}(u-d)\right) f_L(d) + \int_{u=0}^{\infty}\int_{d=u}^{u+t} F_{S|L=d}(u+t-d) f_L(d)\right) \\
&= \rho\left(\int_{u=0}^{\infty}\int_{d=0}^{u} \frac{d}{dt}(F_{S|L=d}(u-d+t) - F_{S|L=d}(u-d)) f_L(d) + \int_{u=0}^{\infty}\int_{d=u}^{u+t} \frac{d}{dt} F_{S|L=d}(u+t-d) f_L(d)\right) \\
&= \rho\left(\int_{u=0}^{\infty}\int_{d=0}^{u} f_{S|L=d}(u+t-d) f_L(d) + \int_{u=0}^{\infty}\int_{d=u}^{u+t} f_{S|L=d}(u+t-d) f_L(d)\right) \\
&= \rho\left(\int_{u=0}^{\infty}\int_{d=0}^{u} f_{S,L}(u+t-d,d) + \int_{u=0}^{\infty}\int_{d=u}^{u+t} f_{S,L}(u+t-d,d)\right) \\
&= \rho\left(\int_{x=0}^{\infty}\int_{y=0}^{\infty} f_{S,L}(x+t,y) + \int_{x=-t}^{0}\int_{y=-x}^{\infty} f_{S,L}(x+t,y)\right) \\
&= \rho\left(\mathbb{P}(S \ge t) + \mathbb{P}(S \le t) - \mathbb{P}(S \le t, L \le t-S)\right) \\
&= \rho(1 - \mathbb{P}(S \le t, L \le t-S)) \longrightarrow 0 \text{ as } t \to \infty,
\end{aligned}
$$

where A are the customers who call and arrive, i.e. request resource and start service, before time 0 but leave in the interval [0,t], and B are the customers who call before time 0 but start and end service in the time interval [0,t]. The third-to-last equality is by change of variables and Leibniz integral rule. The change of differentiation and integration can be justified by using dominated convergence theorem as proved in the next lemma:

Lemma 2.2.5.

$$\frac{d}{dt}\left(\int_{u=0}^{\infty}\int_{d=0}^{u}\Big(F_{S|L=d}(u-d+t)-F_{S|L=d}(u-d)\Big)f_L(d)+\int_{u=0}^{\infty}\int_{d=u}^{u+t}F_{S|L=d}(u+t-d)f_L(d)\right)$$
$$=\int_{u=0}^{\infty}\int_{d=0}^{u}\frac{d}{dt}(F_{S|L=d}(u-d+t)-F_{S|L=d}(u-d))f_L(d)+\int_{u=0}^{\infty}\int_{d=u}^{u+t}\frac{d}{dt}F_{S|L=d}(u+t-d)f_L(d).$$

Proof. We will prove the interchange for the first term as the second term is similar. Let $M = \sup_{s,d} f_{S|L=d}(s)$ which is finite by assumption 3(iii). Note that the first term can be written as

$$\frac{d}{dt}\int_{u=0}^{\infty}\int_{d=0}^{\infty}\Big(F_{S|L=d}(u-d+t)-F_{S|L=d}(u-d)\Big)f_L(d)\mathbf{1}_{d\in[0,u]}.$$

Similar to Lemma 2.2.4, we have to find an integrable function $g(u,d)$ which upper bounds the derivative of the integrand for all $u,d,t\in\mathbb{R}_+$. By the aforementioned assumption, there exists a function $s^*(d)=c+ad$ that satisfies assumption 3(ii). Without loss of generality, we can assume that $a\geq 1$ and $c<0$. One can easily show the following:

$$\begin{aligned}\frac{d}{dt}&(F_{S|L=d}(u-d+t)-F_{S|L=d}(u-d))f_L(d)\mathbf{1}_{d\in[0,u]}\\ &=f_{S|L=d}(u-d+t)f_L(d)\mathbf{1}_{d\in[0,u]}\\ &\leq f_L(d)\mathbf{1}_{d\in[0,u]}*\begin{cases}M & \text{for } u-d\leq s^*(d)\\ f_{S|L=d}(u-d) & \text{for } u-d>s^*(d)\end{cases}=g(u,d).\end{aligned}$$

Then $g(u,d)$ is measurable since it is the product of two measurable functions and

$$\begin{aligned}\int_{d=0}^{\infty}\int_{u=0}^{\infty}g(u,d)&=\int_{d=0}^{\infty}\int_{u=d}^{d+c+ad}Mf_L(d)+\int_{d=0}^{\infty}\int_{u=d+c+ad}^{\infty}f_L(d)f_{S|L=d}(u-d)\\ &=M\int_{d=0}^{\infty}(c+ad)f_L(d)+\int_{d=0}^{\infty}f_L(d)\bar{F}_{S|L=d}(c+ad)\\ &\leq M(c+a\mathbb{E}(L))+\int_{d=0}^{\infty}f_L(d)<\infty,\end{aligned}$$

i.e. $g(u,d)$ is integrable. By Billingsley (1995), the interchange holds. □

Note that $\Lambda_Z(t)$ approaches zero as $t \to \infty$. This supports intuition since at any random point in time t, the expected number of customers who are in service at time t is finite and the expected number of customers whom already reserved service after time t is finite. Indeed, observe that

$$1 - \mathbb{P}(S \leq t, L \leq t - S) \leq \bar{F}_L(\frac{t}{2}) + \bar{F}_S(\frac{t}{2}),$$

since

$$\{S \leq \frac{t}{2}, L \leq \frac{t}{2}\} \subset \{S \leq t, L \leq t - S\},$$

which implies

$$\{S \leq t, L \leq t - S\}^C \subset \{S \leq \frac{t}{2}, L \leq \frac{t}{2}\}^C \subset \{S \geq \frac{t}{2}\} \cup \{L \geq \frac{t}{2}\}.$$

Therefore, $\int_{t=0}^{\infty} (1 - \mathbb{P}(S \leq t, L \leq t - S))$ converges. This implies that $\Lambda_Z(t) \to 0$ as $t \to \infty$.

The following lemma shows that the rate at which customers depart is at least as great as the number pre-arrivals, i.e. $\Lambda_Z(t) \geq \Lambda_Y(t)$. This is used to prove that the blocking probabilities vanish as the arrival rate and capacity grow without bound.

Lemma 2.2.6. *The rate function of the departure process is at least as big as the rate function of the pre-arrival process, i.e.* $\Lambda_Z(t) \geq \Lambda_Y(t)$ *for all* $t \geq 0$.

Demanding that the service distribution be at least greater than some value $\alpha > 0$, then the only thing that changes for the rate of the new departure process, $Z_{new}(t)$, is a shifted $P_Z(t)$ by α, i.e.

$$P_{Z_{new}}(t) = \begin{cases} 1, & 0 \leq t \leq \alpha \\ P_{Z_{old}}(t - \alpha), & t > \alpha, \end{cases}$$

where $P_{Z_{old}}(t) = 1 - \mathbb{P}(S \leq t, L \leq t - S)$. The next lemma demonstrates the above property.

Lemma 2.2.7. *Assume the support of the customers' service time is* $[1, \infty)$. *Then the departure rate is*

$$\Lambda_Z(t) = \begin{cases} \rho, & \text{for } 0 \leq t \leq \alpha \\ \rho(1 - \mathbb{P}(S \leq t - \alpha, L \leq t - \alpha - S)), & o.w., \end{cases}$$

i.e., a shifted version of the case when the service time support is the non-negative real line.

Observe that for any $\alpha > 0$, Lemma 2.2.6 combined with Lemma 2.2.7 imply that $\Lambda_Z(t) > \Lambda_Y(t)$. Therefore, WLOG, we can let $\alpha = 1$. Note that in Chen et al. (2017), for the finite support

and discrete case, the departure rate on the interval [d,d+1] is

$$\begin{aligned}\Lambda_Z(d) &= \sum_{s=1}^{v} \rho \mathbb{P}(S = s)\left(1 - \sum_{i=0}^{d-s} \mathbb{P}(L = i | S = s)\right) \\ &= \rho\left(1 - \sum_{s=1}^{v}\sum_{i=0}^{d-s} \mathbb{P}(L = i, S = s)\right) \\ &= \rho\left(1 - \sum_{i=0}^{d-1}\sum_{s=1}^{d-i} \mathbb{P}(L = i, S = s)\right) \\ &= \rho\left(1 - \mathbb{P}(S \le d, L \le d - S)\right).\end{aligned}$$

The departure process has a rate function of similar form to the continuous case of Lemma 2.2.7, similarly, the pre-arrival process they derived has the same form as $\Lambda_Y(t)$ derived above.

Continuing on our quest to determine asymptotic optimality of ϵ-PSP, we need to compute the virtual blocking probability for the $M/G/\infty$ queue, or equivalently, the conditional blocking probability,

$$\begin{aligned}\mathbb{P}_s^d(B_v) &= \mathbb{P}(B | S = s, L = d) \\ &= \mathbb{P}(\max_{t\in[d,d+s]} N(t) \ge c),\end{aligned}$$

where B =$\{$Customer is Blocked$\}$ and $N(t)$ is the steady-state number of customers in the system at time t. But the event

$$\{\max_{t\in[d,d+s]} N(t) \ge c\}) \subset \{\max_{t\in I} N(t) \ge c\}),$$

where I is any interval containing $[d, d+s]$. Therefore, for any interval I such that $[d, d+s] \subset I$,

$$\begin{aligned}\mathbb{P}(\max_{t\in[d,d+s]} N(t) \ge c) &\le \mathbb{P}(\max_{t\in I} N(t) \ge c) \\ &= \mathbb{P}\left(\max_{i=1,\dots,n} \{\max_{t\in I_i} N(t) \ge c\}\right) \\ &\le \sum_{i=1}^{n} \mathbb{P}(\max_{t\in I_i} N(t) \ge c),\end{aligned}$$

where I_i are intervals such that $\bigcup_{i=1}^{n} I_i = I$. The above implies we can focus on the blocking probabilities on disjoint intervals instead of one interval.

Lemma 2.2.8. *Let c be the system capacity, λ and μ be the arrival rate and mean service time of the customer class, respectively, p^* the optimal value of (2.3), and $\nu \in (0,1)$. Then, there exist a*

sufficiently large, but finite, t^* such that the conditional blocking probability is

$$\mathbb{P}_s^d(B) \leq \sum_{i=0}^{t^*-1} \mathbb{P}(\{X_i + \max_{t\in[0,1]}\{Y_i'(t) + Z_i'(1) - Z_i'(t)\}\} \geq c) + \sum_{i=c}^{\infty} \frac{e^{-\nu c}(\nu c)^i}{i!}, \tag{2.6}$$

where the $Z_i'(t)$ and $Y_i'(t)$ are the NHPP that represent the number of customers who depart and arrive, respectively, in the interval $[i, i+t]$, $t \in [0,1]$, and X_i' be the number of customers who have started service before time i and depart after time $(i+1)$. The rate of $Z_i'(t)$ and $Y_i'(t)$ for $i \in \{0,\ldots,t^*-1\}$ are $\lambda_{Z_i'} = \Lambda_Z(i+t)$ and $\lambda_{Y_i'} = \Lambda_Y(i+t)$, respectively.

The importance of Lemma 2.2.8 is in the proof. It is in the proof that one can show that the t^* from Lemma 2.2.8 can be chosen to be constant when the capacity and arrival rate increase proportionally. The next lemmas are concerned with upper bounding the terms involved in Lemma 2.2.8.

Lemma 2.2.9. *Let $X_i, Y_i'(t)$, and $Z_i'(t)$ for all $i \in \{0,\ldots,t^*-1\}$ be defined as in Lemma 2.2.8, c the capacity, and λ the arrival rate of the class. Then,*

$$\sum_{i=1}^{t^*-1} \mathbb{P}(\{X_i + \max_{t\in[0,1]}\{Y_i'(t) + Z_i'(1) - Z_i'(t)\}\} \geq c) \leq \frac{t^*-1}{\rho} + \sum_{i=1}^{t^*-1} \left(\frac{e^{\delta_i}}{(1+\delta_i)^{(1+\delta_i)}}\right)^{\rho},$$

where $\delta_i = \left(\frac{\epsilon}{1-\epsilon} - \frac{\log(1+\rho)}{\rho \log \theta_i^{-1}}\right)$ and $\theta_i = \frac{\Lambda_Y(i)}{\Lambda_Z(i)} \in (0,1)$ for all i.

The above lemma makes use of various results from, Chen et al. (2017) but we will prove it for completeness. The next lemma is based from Chen et al. (2017)

Lemma 2.2.10. *Let t^*, X_i, $\bar{Y}_i$, and $\bar{Z}_i(1) - \bar{Z}(t)$ be as in the proof of Lemma 2.2.9 for $i \in \{1,\ldots,t^*-1\}$. Also, let $\rho = \min\{(1-\epsilon)c, \sum_{i=1}^{n} \lambda\mu\}$ and $\delta_i = \left(\frac{\epsilon}{1-\epsilon} - \frac{\log(1+\rho)}{\rho \log \theta_i^{-1}}\right)$. Then,*

$$\mathbb{P}(X_i + \max_{t\in[0,1]}\{\bar{Y}_i(t) + \bar{Z}_i(1) - \bar{Z}_i(t)\} \geq c) \leq \frac{1}{\rho} + \left(\frac{e^{\delta_i}}{(1+\delta_i)^{(1+\delta_i)}}\right)^{\rho}.$$

Lemma 2.2.11. *Consider a sequence of problems where the n^{th} problem has parameters $\lambda = n\lambda$, $c^{(n)} = nc$. Choose $\nu < \min\left\{\frac{1}{c}, \frac{1}{e}\right\}$. Then*

$$\sum_{i=c^{(n)}}^{\infty} \frac{e^{-\nu c^{(n)}}(\nu c^{(n)})^i}{i!} \to 0 \text{ as } n \to \infty,$$

and the rate of convergence is $o(e^{-n})$.

Lemma 2.2.12. *Consider a sequence of problems where the* n^{th} *problem has parameters* $\lambda^{(n)} = n\lambda$, $c^{(n)} = nc$ *and* $\epsilon^{(n)} = \frac{\epsilon}{\sqrt{n^{1-\beta}}}$, *with* $\beta \in (0,1)$. *Let* t^*, X_i, Y_i', *and* $Z_i'(1) - Z_i'(t)$ *be as in Lemma 2.2.9 for* $i \in \{1,\dots,t^*-1\}$. *Also, let* $\rho^{(n)} = \min\{(1-\epsilon^{(n)})c^{(n)}, \lambda^{(n)}\mu_S\}$ *and* $\delta_i^{(n)} = \left(\frac{\epsilon^{(n)}}{1-\epsilon^{(n)}} - \frac{\log(1+\rho^{(n)})}{\rho^{(n)}\log\theta_i^{-1}}\right)$. *Then*

$$\sum_{i=1}^{t^*-1} \mathbb{P}(\{X_i + \max_{t\in[0,1]}\{Y_i'(t) + Z_i'(1) - Z_i'(t)\}\} \geq c^{(n)}) \leq \frac{t^*-1}{n\rho} + o\left(\frac{1}{n}\right).$$

We have upper bounded all RHS terms of (2.6) by expressions that converge to zero as $n \to \infty$ except for the term $\mathbb{P}(X_0 + Z_0'(1) + \max_{t\in[0,1]}\{Y_0'(t) - Z_0'(t)\} \geq c^{(n)})$. The reason why the same approach cannot be used to prove that the aforementioned term converges to zero as compared to the other t^*-1 terms is that we upper bounded the other similar t^*-1 terms with probabilitities that contained homogeneous Poisson process whose rates were determined by the values that $\Lambda_Y(i)$ and $\Lambda_Z(i)$ took, where we knew that $\Lambda_Z(i) > \Lambda_Y(i)$ for all $i \in \{1,\dots,t^*-1\}$. If we used the same approach for $i = 0$, the rates of the homogeneous Poisson processes will be equal since $\Lambda_Z(0) = \Lambda_Y(0)$. Because of this, we cannot apply the results of Chen et al. (2017). Regardless, we have the following:

Lemma 2.2.13. *Consider a sequence of problems where the* n^{th} *problem has parameters* $\lambda^{(n)} = n\lambda$, $c^{(n)} = nc$ *and* $\epsilon^{(n)} = \frac{\epsilon}{\sqrt{n^{1-\beta}}}$, *with* $\beta \in (0,1)$. *Then*

$$\mathbb{P}(X_0 + Z_0'(1) + \max_{t\in[0,1]}\{Y_0'(t) - Z_0'(t)\} \geq c^{(n)}) \leq \frac{2c^{(n)}}{e^{2\rho^{(n)}}} + \left(\frac{e^{\delta^{(n)}}}{(1+\delta^{(n)})^{(1+\delta^{(n)})}}\right)^{\rho^{(n)}}. \tag{2.7}$$

Therefore,

$$\mathbb{P}(X_0 + Z_0'(1) + \max_{t\in[0,1]}\{Y_0'(t) - Z_0'(t)\} \geq c^{(n)}) \to 0 \text{ as } n \to \infty. \tag{2.8}$$

Lemmas 2.2.8-2.2.13 imply that for any customer who requests to use the resource $L = d$ time units in the future for $S = s$ time units, the probability of being blocked becomes highly unlikely as $n \to \infty$:

Proof. Theorem 2.2.2.
The expected long-run average revenue for the ϵ-*PSP* policy is

$$\sum_{j=1}^{n} \lambda_j p_j(\lambda) \int_{(S_j, D_j)} (1 - \mathbb{P}(B_{s,d}^j)) s_j f_{S_j, D_j}(s_j, d_j),$$

where $\mathbb{P}(B^j_{s,d})$ is the blocking probability of a customer who submits a request to use product $j \in J$ from $L = d$ time units from today for $S = s$ time units. From 2.4, we can upper bound each $\mathbb{P}(B^j_{s,d})$ by the blocking probabilities of each resource. From constraints in 2.3, each resource will have an arrival rate ρ_r, capacity c_r, joint service and delay time distributions $F^r_{S,L}(s,d)$, along with their marginal distributions $F^r_S(s), F^r_L(d)$. Now, we can apply Lemmas 2.2.8-2.2.12 to bound the blocking probability of each resource and getting a uniform bound, say γ, on the blocking probability of each product. Therefore,

$$\begin{aligned}
\mathcal{R}(\epsilon\text{-}PSP) &= \sum_{i=1}^{n} \lambda_i p_i(\lambda) \int_{(S_i,D_i)} (1-\mathbb{P}(B_{s,d}))s_i f_{S_i,D_i}(s_i,d_i) \\
&\geq \sum_{i=1}^{n} \lambda_i p_i(\lambda) \int_{(S_i,D_i)} (1-\gamma)s_i f_{S_i,D_i}(s_i,d_i) \\
&= \sum_{i=1}^{n} \lambda_i p_i(\lambda)(1-\gamma) \int_{(S_i,D_i)} s_i f_{S_i,D_i}(s_i,d_i) \\
&= \sum_{i=1}^{n} \lambda_i p_i(\lambda)(1-\gamma) \int_{S_i} s_i f_{S_i}(s_i) \\
&= (1-\gamma) \sum_{i=1}^{n} \lambda_i p_i(\lambda)\mu_{s_i} \\
&\geq (1-\gamma)(1-\epsilon) J^* \\
&= \left(1 - \sum_{r=1}^{m} \left(\frac{t^*_r - 1}{\rho_r} - \sum_{i=0}^{t^*_r-1} \left(\frac{e^{\delta_{i,r}}}{(1+\delta_{i,r})^{(1+\delta_{i,r})}} \right)^{\rho_r} - \frac{2c_r}{e^{2\rho_r}} - \frac{e^{-\nu c_r}}{\sqrt{2\pi c_r}} l \right) \right) (1-\epsilon) J^*.
\end{aligned}$$

The conclusion follows. □

Proof. Theorem 2.2.3:
By Theorem 2.2.2, we have that

$$\begin{aligned}
\frac{\mathcal{R}(\epsilon^{(n)}\text{-}PSP)}{J^*_n} &\geq \left(1 - \sum_{r=1}^{m} \left(\frac{t^*_r - 1}{\rho^{(n)}_r} - \sum_{i=0}^{t^*_r-1} \left(\frac{e^{\delta^{(n)}_{i,r}}}{(1+\delta^{(n)}_{i,r})^{(1+\delta^{(n)}_{i,r})}} \right)^{\rho^{(n)}_r} - \frac{2c^{(n)}_r}{e^{2\rho^{(n)}_r}} - \frac{e^{-\nu c^{(n)}_r}}{\sqrt{2\pi c^{(n)}_r}} l \right) \right) \left(1-\epsilon^{(n)}\right) \\
&= \left(1 - \sum_{r=1}^{m} \left(\frac{t^*_r - 1}{\rho^{(n)}_r} - \sum_{i=0}^{t^*_r-1} \left(\frac{e^{\delta^{(n)}_{i,r}}}{(1+\delta^{(n)}_{i,r})^{(1+\delta^{(n)}_{i,r})}} \right)^{\rho^{(n)}_r} - \frac{2nc_r}{e^{2\rho^{(n)}_r}} - \frac{e^{-\nu n c_r}}{\sqrt{2\pi n c_r}} l \right) \right) \left(1-\epsilon^{(n)}\right) \\
&\geq \left(1 - \sum_{r=1}^{m} \left(\frac{t^*_r - 1}{n\rho_r} - \sum_{i=0}^{t^*_r-1} \left(\frac{e^{\delta^{(n)}_{i,r}}}{(1+\delta^{(n)}_{i,r})^{(1+\delta^{(n)}_{i,r})}} \right)^{n\rho_r} - \frac{2nc_r}{e^{2n\rho_r}} - \frac{e^{-\nu n c_r}}{\sqrt{2\pi n c_r}} l \right) \right) \left(1-\epsilon^{(n)}\right).
\end{aligned}$$

Equality comes from direct substitution and the last expression comes from $\rho_r^{(n)} \geq n\rho_r$ for any resource. We also have that $\frac{2nc_r}{e^{2n\rho_r}} - \frac{e^{-\nu nc_r}}{\sqrt{2\pi nc_r}} \in o\left(\frac{1}{n}\right)$ for any resource and from Chen et al. (2017) we have that $\left(\frac{e^{\delta_{i,r}^{(n)}}}{(1+\delta_{i,r}^{(n)})^{(1+\delta_{i,r}^{(n)})}}\right)^{n\rho_r} \in o\left(\frac{1}{n}\right)$. Therefore,

$$\begin{aligned}
\frac{\mathcal{R}(\epsilon^{(n)}\text{-}PSP)}{\mathcal{R}^{(n)}(OPT)} &\geq \left(1 - \sum_{r=1}^{m} \frac{t_r^* - 1}{\rho_r^{(n)}} + o\left(\frac{1}{n}\right)\right)\left(1 - \epsilon^{(n)}\right) \\
&= \left(1 - \sum_{r=1}^{m} \frac{t_r^* - 1}{\rho_r^{(n)}} + o\left(\frac{1}{n}\right)\right)\left(1 - \frac{\epsilon}{\sqrt{n^{1-\beta}}}\right) \\
&= 1 - \frac{\epsilon}{\sqrt{n^{1-\beta}}} + o\left(\frac{1}{\sqrt{n^{1-\beta}}}\right).
\end{aligned}$$

The equality follows from the property that for $\beta \in (0,1)$

$$\lim_{n\to\infty} \frac{1/n}{1/\sqrt{n^{1-\beta}}}.$$

□

The analysis follows very close that of Chen et al. (2017) but we had to bound the probability of being blocked on an unbounded interval to then focus on bounding the blocking probability on a finite interval. The analysis implies that results from Chen et al. (2017) extend for continuous and unbounded supported service time and advance reservation distributions.

2.2.3 Numerical Experiments

We will test the performance of the ϵ-PSP policy in the special case when we have multiple products and resources and each customer for a particular product $j \in J$ has a reservation distribution, $\bar{F}_j(p)$ that is known to the seller. The price reservation distributions and arrival rates of each product, capacities of each resource, will not change throughout the different experiments, other than being scaled. The optimization problem we are solving is

$$\begin{aligned}
\text{maximize} \quad & \sum_{j=1}^{n} \lambda_j p_j \bar{F}_j(p_j) \mu_{s_j} \\
\text{subject to} \quad & \sum_{j=1}^{n} a_{ij} \lambda_j \bar{F}_j(p_j) \mu_{s_j} \leq (1-\epsilon) c_j \\
& p_j \in [p_0^{(j)}, p_\infty^{(j)}].
\end{aligned} \tag{2.9}$$

If we let $q_j = \bar{F}_j(p)$, the reformulation is

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n} \lambda_j q_j \bar{F}_j^{-1}(q_j) \mu_{s_j} \\ \text{subject to} \quad & \sum_{j=1}^{n} a_{ij} \lambda_j q_j \mu_{s_j} \leq (1-\epsilon) c_j \\ & q_j \in [0,1]. \end{aligned} \tag{2.10}$$

To put it in terms of Problem 2.3, define $\beta_i = \lambda \mu_{S_i} q_i$. Then,

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n} \beta_j g_j(\beta) \\ \text{subject to} \quad & A\beta \leq (1-\epsilon) c \\ & \beta_j \in [0, \lambda_j \mu_{S_j}], \end{aligned} \tag{2.11}$$

where $g_i(\beta_i) = \bar{F}_i^{-1}(\frac{\beta_i}{\lambda_i \mu_{S_i}})$.

Our experiments will vary the service and delay times distributions to test the robustness of the policy. Therefore, consider 3 products and 5 resources. The reservation price distribution follows a truncated *Gumbel* distribution, $\bar{F}(p;\mu,\nu)$, over the price range [0,10] with the following parameters for each product $j \in \{1,2,3\}$:

1) Product 1: $\mu = 1$ and $\nu = 2$ with arrival rate $\lambda_1 = 1$

2) Product 2: $\mu = 3$ and $\nu = 4$ with arrival rate $\lambda_2 = 2$

3) Product 3: $\mu = 2$ and $\nu = 2$ with arrival rate $\lambda_3 = 1.5$

with capacity vector $c = (5,4,3,6,7)$. The *bill-of-materials* matrix is

$$A = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}.$$

The distributions for each of the 4 scenarios are:

- U: uniform random variable

	Scenario 1	Scenario 2
Product 1	L$\sim Gamma(df=2)$ S|L=d$\sim Exp((1+d)^{-1})$	L$\sim Gamma(df=2)$ S|L=d$\sim Exp((1+d)^{-1})$
Product 2	$log\mathcal{N}(\mathbf{0},\begin{pmatrix}1 & 0.8\\ 0.8 & 1\end{pmatrix})$	$log\mathcal{N}(\mathbf{0},\begin{pmatrix}1 & -0.8\\ -0.8 & 1\end{pmatrix})$
Product 3	L$\sim \mathcal{X}_2(df=4)$ S$\sim U$[1,10]	L$\sim \mathcal{X}_2(df=4)$ S$\sim U$[1,10]

Table 2.1: Distributions for scenario 1 and 2.

	Scenario 3	Scenario 4
Product 1	L$\sim TrExp(10)$ S|L=d$\sim TrExp(10, 1.7(1+d)^{-1})$	L$\sim U$[0,15] S$\sim U$[1,3]
Product 2	L$\sim U$[1,11] S|L=d$\sim TrStd\mathcal{N}$	L$\sim U$[0,20] S$\sim U$[1,5]
Product 3	L$\sim U$[1,15] S$\sim U$[1,10]	L$\sim U$[0,6] S$\sim U$[1,10]

Table 2.2: Distributions for scenario 3 and 4.

- $\mathcal{X}_2(df=2)$: chi-square random variable with 2 degrees of freedom
- $Gamma(df=2)$: gamma random variable with 2 degrees of freedom
- $TrExp(a,b)$: truncated exponential random variable with range [1,a] and scale=b, i.e. the coefficient of x in the exponent
- $Exp(a)$: exponential random variable with rate equal to a
- $Tr\mathcal{N}(\mu,\Sigma)$: normal random variable with mean μ and covariance Σ

If the conditional distribution of the service time does not depend on the delay, then they are independent. In all cases, the service time was additionally truncated so that the minimal service time is 1.

All scenarios, regardless of the dependence, or lack thereof, of the service and delay, the performance of the ϵ-PSP policy gets better. The difference between the first and second scenario is the negative service and delay correlation of product 2. Positive correlation in log-normal distributions induces conditional means that grow faster than the linear behavior of the conditional mean of a bivariate normal distribution. A negative correlation has the opposite effect as the conditional mean decays rapidly. We wanted to test if the policy performed better in the negative correlation

case relative to the positive correlated case. The top two graphs in Figure 2.1 shows that this is indeed the case as it performs better for each n. When $n = 50$, the performance of the policy was at most 6% away from optimal whereas it was at most 3.5% away from optimality. Scenarios 3 and 4 tests the cases that would most likely be encountered in practice, i.e. bounded support distributions.

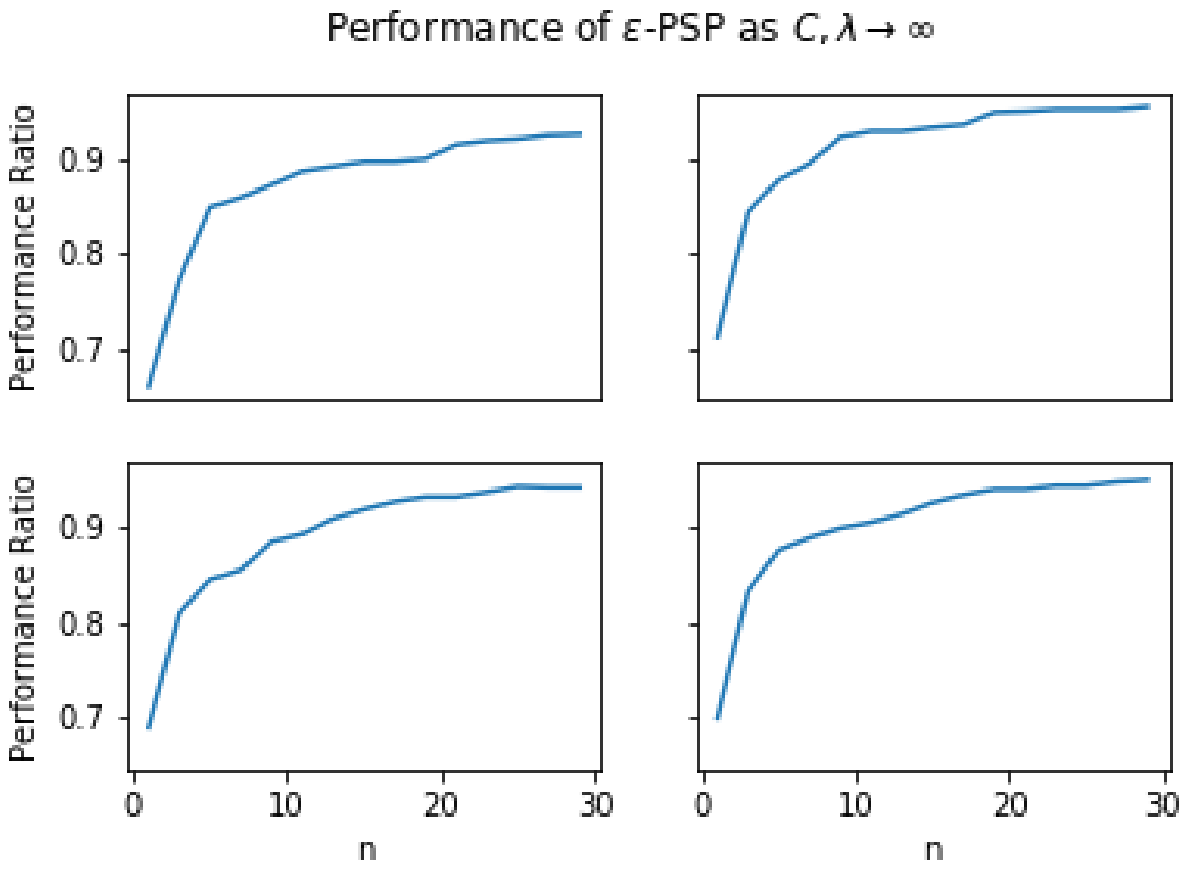

Figure 2.1: Performance ratio of the ϵ-$PS\,P$ policy under different distributional settings.

Additionally, we tested the performance of the policy by varying the load factor, defined by

$$LF = \lambda \frac{\mathbb{E}[S]}{C}.$$

The base model uses the same parameters defined above and uses the distributions described in scenario 3 described above. Table 2.3 and Table 2.4 shows when the overall total mean demand is increased and when the service time mean is increased, respectively. Each data point averaged over 20 simulations and the simulations were run long enough that the standard deviation is insignificant.

In summary, our results demonstrate that the pricing policy performs near-optimally in the heavy traffic regime, which is consistent with our theoretical results. Moreover, the policy performs reasonably well in the light and medium traffic regimes as well, in particular, the policy performance is at least 50% optimal. Even though this was not proven in our setting, this result has been theoretically proven in various other settings such as Levi and Radovanovic (2010), Owens (2018). It is also worth noting that the performance of the policy is robust with respect to input distributions and parameters, which could be widely adopted in many practical scenarios since these parameters are rarely known exactly.

n \LF	1	4	7	10	13	16	19	22	25	28	31	34	37	40	43	46	50
3.0	82.7%	91.0%	92.9%	94.5%	94.8%	95.1%	95.7%	95.9%	96.3%	96.5%	96.6%	96.8%	96.9%	97.0%	97.0%	97.2%	97.4%
3.5	83.5%	90.8%	92.3%	93.9%	94.3%	95.1%	95.7%	95.8%	96.0%	96.2%	96.5%	96.4%	96.8%	96.9%	96.9%	97.1%	97.3%
4.0	82.6%	89.8%	92.3%	93.2%	94.0%	94.7%	95.3%	95.8%	95.8%	95.9%	96.2%	96.3%	96.3%	96.5%	96.9%	96.7%	97.0%
4.5	79.4%	88.9%	91.3%	92.9%	93.6%	94.2%	94.6%	95.1%	95.1%	95.6%	95.7%	96.2%	96.1%	96.2%	96.4%	96.7%	96.9%
5.0	80.5%	89.4%	91.3%	92.8%	93.5%	93.9%	94.8%	95.0%	95.2%	95.7%	95.6%	96.1%	96.0%	96.2%	96.4%	96.4%	96.7%

Table 2.3: Load factor when varying the *total* mean demand, λ and c.

n \LF	1	4	7	10	13	16	19	22	25	28	31	34	37	40	43	46	50
3.0	67.8%	81.3%	84.5%	86.9%	88.4%	89.2%	89.9%	90.7%	91.1%	91.5%	91.9%	92.4%	92.5%	92.9%	93.0%	93.3%	93.6%
3.5	75.8%	81.1%	83.8%	86.7%	87.5%	89.2%	89.5%	90.6%	90.8%	90.9%	91.7%	91.8%	92.3%	92.4%	92.9%	93.0%	93.2%
4.0	71.4%	78.7%	84.1%	85.3%	87.3%	87.5%	89.1%	89.4%	90.4%	90.4%	91.2%	91.3%	91.8%	91.8%	92.4%	92.3%	92.6%
4.5	64.5%	77.5%	82.1%	84.1%	85.7%	87.0%	88.1%	88.8%	89.6%	90.0%	90.3%	90.9%	91.1%	91.3%	91.5%	91.8%	92.0%
5.0	68.9%	80.3%	84.0%	83.5%	85.9%	87.2%	88.4%	89.3%	88.9%	89.5%	90.4%	90.9%	91.1%	91.1%	91.3%	91.6%	91.9%

Table 2.4: Load factor when varying the mean service time and c.

CHAPTER 3

Revenue Management with Reusable Resources using Upper Confidence Bounds (UCB)

3.1 Introduction

Much of the revenue management literature out today has analyzed the revenue management with perishable resources model, wherein the resources are bought by customers, they are consumed and cannot be resold, for example, food, electronics, etc. This area has been extensively researched and there exists a multitude of variations of the model. This chapter focuses on the non-perishable resources case, thus the word *reusable* in the title. This setting has many natural applications such as the hotel industry, where the reusable products are the hotel rooms, car rental industry, where the car is the reusable resource, cloud computing, and many more. The literature on revenue management models with reusable resources in existence are *static*, in the sense that it uses data gathered in the exploration phase to create statistical point estimate(s) and exploits this information for the rest of the time horizon. It does not use information sequentially, as data arrives. In this chapter, we will be analyzing the revenue management model with a *single* reusable resource, with finite capacity that does not change over time and post prices, from a finite set of prices, dynamically according to data that arrives in an online fashion. A possible reason for the lack of literature in this setting is the complexity of the blocking probability, i.e. the event where customers want to make a purchase but are unable to due to the seller running out goods, and thus the customer is lost, or *blocked*, and no sale occurs.

Almost all literature in revenue management in this setting assume that customers arrive according to a Poisson process or similar processes in order to get a handle on the blocking probability to prove sub-linear regret bounds. In our case we assume that a fixed, and same, number arrive in each time period. A major difference of our work is how our dynamic policy is being judged. Regret is usually measured with respect to some *clairvoyant* model which is aware of the parameters of the problem and is able to find the optimal policy that would maximize its revenue. We

will judge our policy to a fixed-price policy which not only knows the relevant parameters, but will not face any lost customers, i.e. we are comparing our resulting revenue to the revenue garnered by a *fictitious* seller with infinite capacity, which is the best one can do. In other words, the policy is not being compared to the optimal clairvoyant policy, but with the optimal clairvoyant policy with *infinite* capacity. The best clairvoyant policy *might* have blocking events associated with it wherein it might be beneficial for some customers to get blocked, but this obviously assumes that there is no cost associated with such events. That might not be the case as loss of customers can occur, customer satisfaction degrades, etc. The reason for this is to get a hold of the algorithm's convergence rate as time, capacity, and starting inventory are scaled linearly. If the clairvoyant policy contains blocking events, the analysis will be much more complex and will not be able to compare policies. In other words, we are concerned with how our performance measure scales as the scaling factor, n, gets large; this regime is known as the *fluid regime*. Another difference is the regret measure. Usually regret is the difference between absolute quantities, but our application will measure performance as the difference between terms that are quantities-per-unit-time. For example, our regret measure will be

$$Regret(T) = \mathbb{E}[\frac{Revenue^{\pi^*}}{T}] - \mathbb{E}[\frac{Revenue^{\pi}}{T}],$$

where π^* is the clairvoyant policy, π is the policy followed by our algorithm. For the n^{th} problem the regret is

$$Regret_n(T) = \mathbb{E}[\frac{Revenue_n^{\pi^*}}{Tn}] - \mathbb{E}[\frac{Revenue_n^{\pi}}{Tn}],$$

where n is the scaling of the system, the problem parameters T, capacity, and arrivals are scaled by n, and $Revenue_n^{\pi}$ is the revenue generated by applying the pricing policy π throughout the time horizon nT. As we will discuss in the coming section, the true blocking probability is intractable. Therefore, we consider the *relaxed* regret instead

$$Regret_n(T) = \mathbb{E}[\frac{\underline{Revenue}_n^{\pi^*}}{Tn}] - \mathbb{E}[\frac{Revenue_n^{\pi}}{Tn}]$$

where $\underline{Revenue}_n^{\pi^*}$ is the optimal revenue that is incurred had the seller had unlimited resources.

3.1.1 Main Results and Contribution

We summarize our high-level approach as follows.

We use *upper confidence bound* (UCB) estimates in an optimization framework to derive the randomized policy to use. The policy depends on the observations we see, i.e. purchase or no

purchase, which changes the UCB estimates. The high-level idea is that we solve a fractional 2-D Knapsack at every time period with capacity constraint that is buffered by the right order. An interpretation of this buffer is to post prices according to the constraint that we have less capacity than we actually have in stock. The order of this buffer, which is automatically supplied by the UCB estimates, turns out to surprisingly provide constant regret. As n is large enough, and computable, our policy selects the optimal price to set throughout most of the time horizon and generates the optimal revenue *rate* garnered by the clairvoyant model.

3.1.2 Organization and General Notation

This chapter is organized as follows. We formulate our problem in 2 and describe the learning algorithm. We carry out the regret analysis in 3. We show some computational performance in 5. Finally, we conclude and point out several future directions in 6.

For any $x \in \mathbb{R}$, $x^+ = \max\{0, x\}$. The indicator function $\mathbb{1}(A)$ takes value 1 if A is true and 0 otherwise. We use LHS and RHS as abbreviations for the "left-hand side" and the "right-hand side" of an equation, respectively. $[N] = \{1, \ldots, N\}$. The following notation will be useful: for real valued positive sequences a_n and b_n we write $a_n = O(b_n)$ if a_n/b_n is bounded from above for large enough values of n (i.e., $\limsup a_n/b_n < \infty$).

3.2 Literature Review

There is a plethora of literature in the perishable revenue management model and a multitude of variations arising from what information is known to the seller such as the (un)-censored demand process, demand curve structure, etc. Most, if not all, results are results in the *fluid regime*, wherein, the system parameters are scaled proportionally such that the randomness is *washed away*, roughly speaking. The seminal papers Gallego and van Ryzin (1994) and Gallego and van Ryzin (1997) derived the Hamilton-Jacobi sufficient conditions, a first-order differential equation, for the optimal value function of the model. The authors developed various pricing heuristics that approach the optimal revenue as the resource capacities and demand rate get scaled linearly, with both heuristics displaying a $O(1/\sqrt{n})$ convergence rate, where n is the scaling parameter. These papers provided the impetus for the literature that followed. The text and surveys by Talluri and van Ryzin (2005), Özer and Phillips (2012), den Boer (2015), and Bitran and Caldentey (2003b) provides excellent overviews of this area of research.

Most of the relevant literature that revolves around the revenue management models with reusable resources assumes that the arrival process is governed by a Poisson process Levi and

Radovanovic (2010), Chen et al. (2017), Bernal and Shi (2019). The assumption reflects the real-life scenarios that customers do not arrive in batches and the probability of multiple arrivals in an infinitesimal interval is infinitesimal. The assumption is made to make the model tractable to compute bounds on the blocking probabilities. The assumption we make regarding the arrival process is that customers do arrive in batches, and they will buy a resource with probability that depends on that price. Levi and Radovanovic (2010) allows for any service time distribution but no advance reservation, Chen et al. (2017) allows for any finite discrete service time and advance reservation distributions, and Bernal and Shi (2019) extends Chen et al. (2017) to allow for any continuous service time and advance reservation distributions. One of the drawbacks of our model is that we assume the service time is constant and no advance reservation is allowed. We hope that future research will extend this model to allow for these relaxations to occur. All three papers developed a static model which determined their static policy and proved asymptotic optimality of said policy in the fluid regime.

The literature that focuses on the setting where the seller uses a finite set of resources to serve customers repeatedly. Maglaras (2006) studies a setting wherein the seller is endowed with a single unit of resource that can be repeatedly used to serve multiple classes of customers, but our case serves just one class. Customers arrive according to a Poisson process and service times that are exponentially distributed. The seller's goal is to find a joint pricing and priority sequencing policy that maximizes the long-run expected profit. The author proposes a policy that is the optimal solution in the corresponding fluid model, and shows that this heuristic policy performs well. No information is used in the future to make decisions. Our work in this chapter concerns in using available streaming data to make pricing decisions from the perspective of a monopolist seller where the customers do not strategize ahead of time. Various other papers focuses on forward-looking customers where the customers can develop their own strategies to know when to buy and not buy, e.g., Chen and Shi (2016), Borgs et al. (2014).

Lei and Jasin (2016a) is one of the works that resemble the closest to our work but their model computes an LP that can have as many constraints as the length of the time horizon. They solve one LP in the beginning but don't use future information to make decisions. The buffer they computed when solving the LP is of order $O(n\log(n))$, whereas we will see later, is the same buffer order we determine in our LP. Their result is an average regret that is sublinear as n, the service time, gets large. Our result is an average regret result when the system as a whole is proportionally scaled, but we do not scale the service time.

Other work that revolves around the idea of estimation and control is Besbes and Zeevi (2011). They partition the time horizon into exploration and exploitation phases. After a set amount of exploration time, which was carefully determined and decreases as the system scales, they devise

an LP to determine the prices to set afterwards. Our work does indeed explore the prices to set and we do this in every period by solving an LP that uses UCB estimates to output a distribution over arms. This approach causes the seller to choose prices in such a way that does not create so much demand as to run out of supply but at the same time does not under price so much so as to not have low demand and little revenue. Some differences are that Besbes and Zeevi (2011) assumes a Poisson arrival process and the model is a perishable, not reusable, resource model.

3.2.1 Related Literature in Other Disciplines

The general problem of making sequential decisions with limited to no information has a vast history with the work of Robbins (1952). Lai and Robbins (1985) capture the exploration-exploitation dilemma. This model was introduced in the clinical trials statistics literature. This seminal paper proved the existence of policies that have sublinear regret. This work sparked a vast amount of research in multi-armed bandits (MAB) with different variations on the arms (e.g. continuous or discrete arms) by Auer et al. (2002a), Mandelbaum (1987), different approaches to the MAB problem (e.g. UCB or thompson sampling) by Agrawal and Goyal (2012), and the stochasticity of the reward process (e.g. stochastic or adversarial) by Auer et al. (2002b), Bubeck and Cesa-Bianchi (2012). Many more variations have spawned from the work of Lai and Robbins (1985). Our work, to the best of our knowledge, combines the UCB estimation procedure in the revenue management model with reusable resources into the optimization to help the seller in choosing the prices in such a way to maximize his/her revenue.

3.3 Problem Formulation

In this model we consider a revenue management problem where a firm sells one reusable product. Moving forward, selling is synonymous with renting. For example, DoubleTree by Hilton hotel selling a hotel room, Hertz selling a car, Amazon selling computational resources, etc. The seller has a *reusable* product to sell over a time horizon of T periods. Each period sees a *single* customer arrival in the base model, no scaling. The seller's initial inventory is x_0. Before each time period $t \in \{1, 2, \ldots, T\}$, the seller has to make a decision as to which of the K prices he/she will post to the public among the set $\{p_1, \ldots, p_K\}$, which affects the buying probability θ_k, $k = 1, \ldots, K$. We assume that the prices are normalized so that $p_k \leq 1$ for all $k \in [K]$. If a customer purchases the reusable product, then the customer utilizes the product for s time units. When the customer is done with the product, the product is released to be used again for consumption. The seller's goal is to maximize the *time-average revenue* over the time horizon. The time-average revenue will be used because of

the regret measure we will concern ourselves in this paper.

We assume that the true demand function is unknown to the seller. While the seller possesses only limited information on the demand function, s/he is able to continuously observe realized demand at all discrete time increments starting at time 0 and up until the end of the selling horizon T. We shall use π to denote a pricing policy. We impose additional assumptions to avoid trivialities and gain tractability:

Assumptions

(i) Service time is strictly greater than the initial inventory for the base case, i.e. $s > x_0$.

(ii) The optimal solution(s) have corresponding θ's strictly less than $\frac{x_0}{s}$.

The first assumption is due to the fact that blocking will never occur since in the worst and unlike case that each arriving customer decides to purchase, the seller will always satisfy demand. This case can be solved via *multi-armed bandits* (MAB) since we have K unknown mean revenues , or rewards in MAB terminology, and all the seller has to do is find the best strategy over the time horizon T. From [Auer] this has regret $O(\ln T)$. [Lai and Robbins] provided an asymptotic lower bound on the expected regret of any bandit algorithm to be $\Omega(\ln T)$. In other words, this is the best that one can do with the given information. The second assumption has to do with blocking probability as the system scales for the full information system. As will be shown, if (ii) is not satisfied, the blocking probability will be non-zero and will not dissipate as the system scales largely and this will also occur for the randomized policy. This result can be seen in the numerical illustrations of this chapter. Tractability is due to the ability to compute an upper bound on the blocking probability using our randomized policy.

Demand for products at any time $t \in [0, T]$ is given by a bernoulli process with intensity $\theta_t = \theta_t^{k_t}$, where $\theta_t^{k_t}$ is the is the buying probability given that the price was set at p_{k_t} at time t. We assume without loss of generality that $c_i \geq 0$, for all $1 \leq i \leq T$. A policy π is said to be admissible if the induced price process is non-anticipating, in other words, $\pi(t)$ is measurable with respect to the sigma algebra generated by the past decisions and arrivals, $\sigma\left(\pi(1), Bern(\theta_1^{\pi(1)}), \ldots, \pi(t-1), Bern(\theta_{t-1}^{\pi(t-1)})\right)$. The policy satisfies

$$\sum_{i=[t-s+1]^+}^{t} Bern(\theta_i^{\pi(i)}) \leq x_0, \quad \forall 1 \leq t \leq T.$$

This constraint is different than in the perishable case where instead the constraint would have been the following:

$$\sum_{i=1}^{T} Bern(\theta_i^{\pi(i)}) \leq x_0,$$

where $\pi(i)$ is the chosen arm, or price, set for the next time period. The former translates to having at most x_0 products being used at any time period. Whereas the latter means that you can only sell x_0 over the entire horizon.

When the seller uses an admissible policy π to price the single reusable resource, the performance of the policy is measured in terms of the average revenue rate

$$J^{\pi}(T) := \frac{1}{T}\mathbb{E}[\sum_{t=1}^{T} p_{\pi(t)} Bern(\theta_t^{\pi(t)})]. \tag{3.1}$$

The seller is unaware of the true buying probabilities, θ's, but the seller's goal is to optimize (3.1). Optimizing (3.1) is difficult so the seller will focus on minimizing the difference between the average revenue given that the capacity is infinite, i.e. no capacity constraint, and the average revenue with constraints. Let $J^*(T)$ be the optimal average revenue rate without capacity constraints. Then the seller's goal is to minimize regret, i.e.

$$R^{\pi}(T) := J^*(T) - J^{\pi}(T). \tag{3.2}$$

3.3.1 The Benchmark

The traditional revenue management setting which analyzes perishable resources, the *clairvoyant* model in Gallego and van Ryzin (1994) is available to use to compare any proposed policy which characterize the optimal state-dependent pricing policy using dynamic programming. Given assumption (ii), the clairvoyant policy is by assumption those instances where the optimal solution(s) from the linear program model proposed by Gallego and van Ryzin (1994) have corresponding θ's strictly less than $\frac{x_0}{s}$ and no capacity constraint.

3.4 Main Result

A relevant instance of the reusable revenue management problem occurs when the set of feasible prices is discrete and finite, say $\mathcal{D} = \{p_1, \ldots, p_K, p_\infty\}$, where $p_\infty = 1$ is the price at which there exists no demand, i.e. the buying probability is zero. We need this price so that if the system is at capacity, then the seller will want to price the resource such that no demand arises since it will not be able to meet the demand.

3.4.1 Algorithm

Let $A(t)$ be the number of arrivals at the beginning of time period t and $D(t)$ be the number of departures at the end of time period t with $D(0) = 0$. Let $T_k(t)$ be the number of times price p_k has been posted up to time t. Let $L(t)$ be the number of customers in the system at time t, then

$$L(t) = \sum_{i=[t-s+1]^+}^{t} [A(i) - D(i-1)].$$

A seller who knows the true buying probabilities for each price, θ_k, a priori can upper bound their steady-state time-average revenue by solving

$$\begin{aligned} \max \ & \sum_{k=1}^{K} \theta_k p_k q_k \\ \text{s.t.} \ & \sum_{k=1}^{K} \theta_k q_k s \leq x_0, \\ & \sum_{k=1}^{K} q_k \leq 1, \\ & q_k \geq 0, \forall k \in [K], \end{aligned} \tag{3.3}$$

where q_k is the vector of probabilities conveying the distribution over arms for all time periods. The first constraint is the stationary number of customers' average rate in the system being at most x_0. The second constraint is the summation of the distribution vector.

The intuition behind (3.3) is that the seller is trying to maximize the revenue rate with respect to the constraint that at most x_0 customers per unit time can be in the system. The distribution vector is what the seller will use to implement his/her policy, e.g. if $q_2 = 0.25$, then the seller will set the price at $p_2 = 25\%$ of the time. If the total sum of the distribution vector is less than unity, then the rest of the leftover probability mass goes towards p_∞.

The above optimization problem, (3.3), assumes the seller knows the true buying probabilities. But that is not the case and the seller will need to make decisions on which price to set as he/she observes samples of the buying probabilities. Consider the following upper confidence optimization scaled problem at time $t \in [T^{(n)}]$ (note that we got rid off the service time constant since

it does not change the optimal solution):

$$\begin{aligned} \max\ & n\sum_{k=1}^{K}\bar{\theta}_k^{(n)}(t)p_k q_k^{(n)}(t) \\ \text{s.t.}\ & \sum_{k=1}^{K}\bar{\theta}_k^{(n)}(t)q_k^{(n)}(t)s \le x_0, \\ & \sum_{k=1}^{K} q_k(t) \le 1, \\ & q_k(t) \ge 0, \forall k \in [K], \end{aligned} \tag{3.4}$$

where

$$\bar{\theta}_k(t) = \min\left\{\hat{\theta}_k(t) + \sqrt{\frac{\beta\ln(T)}{T_k(t-1)}}, 1\right\},$$

and $\hat{\theta}_k(t) = \dfrac{\sum_{i=1}^{t-1}\mathbb{1}_{A_i}(i)}{T_k(t-1)}$, where $A_i :=$ {Customer purchased on the i-th pull}. For the scaled problem, note that n customers will be arriving each period; equivalent to pulling n times each period and observing the number of customers who purchased out of the multiple pulls. Therefore, for the scaled system indexed by n:

$$\bar{\theta}_k^{(n)}(t) = \min\left\{\hat{\theta}_k^{(n)}(t) + \sqrt{\frac{\beta\ln(nT)}{nT_k(t-1)}}, 1\right\},$$

where

$$\hat{\theta}_k^{(n)}(t) = \frac{\sum_{i=1}^{t-1}\sum_{j=1}^{n}\mathbb{1}_{A_{i,j}}(i)}{nT_k(t-1)},$$

$A_{i,j} :=$ {Customer purchased on the j-th pull of the i-th period}. The difference between (3.3 and (3.4) is that the latter uses the UCB estimates of the buying probabilities, θ's.

The intuition into using UCB estimates in the constraint is due to the *buffer* that is automatically embedded in the capacity constraint. In other words, if the capacity constraint is expanded,

we get the following:

$$\sum_{k=1}^{K} \hat{\theta}_k^{(n)}(t) q_k^{(n)}(t) \leq \frac{x_0}{s} - \sum_{k=1}^{K} \sqrt{\frac{\beta \ln(nT)}{nT_k(t-1)}} q_k^{(n)}(t)$$
$$\sum_{k=1}^{K} \hat{\theta}_k^{(n)}(t) q_k^{(n)}(t) \leq \frac{x_0}{s} - \sum_{k=1}^{K} \sqrt{\frac{\beta \ln(nT)}{n}} q_k^{(n)}(t)$$
$$\sum_{k=1}^{K} \hat{\theta}_k^{(n)}(t) q_k^{(n)}(t) \leq \frac{x_0}{s} - \sum_{k=1}^{K} \sqrt{\frac{\beta \ln(nT)}{n}} q_k^{(n)}(t)$$
$$\sum_{k=1}^{K} \hat{\theta}_k^{(n)}(t) q_k^{(n)}(t) \leq \frac{x_0}{s} \left(1 - \sqrt{\frac{\beta \ln(nT)}{n}}\right).$$

This means that at each step, the algorithm is using the current empirical mean estimates of the buying probabilities to output a distribution vector such that the expected capacity rate is below the x_0/s threshold. The buffer decreases at the rate of $O\left(\sqrt{\frac{\beta \ln(n)}{n}}\right)$, which is a rate that many authors have used in the network/non-network revenue management capacity constraint for perishable resources.

Algorithm

- Pull a single arm once until all arms have been pulled and the system is empty, call this time t^*. (Note that $t^* \leq sK$ for any scaled problem, i.e. it does not change with scaled parameter n)
- For $t > t^*$, seller posts price p_k with probability $q_{t,k}^{(n)}$. Post price $p_\infty = 1$ if x_0 are in use at the end of the previous period.

3.4.2 Blocking Probability Analysis

In the base case and any scaled system, the blocking probability depends on the past $s-1$ time units, since if any customer decided to purchase the product at least s days ago from today, then that same customer will not be in the system today. Only those customers who purchased $s-1$ days ago until the day before affect the blocking probability of the purchasing customers today. In the base case, only one customer arrives each period, and n customers arrive in the scaled system. With probability which depends on the price, only a fraction actually end up purchasing. Therefore, the blocking probability is:

$$\mathbb{P}(Blocked(t)) = \mathbb{P}\left(\sum_{i=[t-s+1]^+}^{t} \sum_{k=1}^{K} Bin(n, \theta_k \mathbb{1}_{\{k\}}(i)) \geq nx_0\right). \tag{3.5}$$

Assume for now that the optimal distribution for (3.3) is q^*, i.e., at each period an arm is chosen according to q^*. Then, if q^* plays arms k, i.e. for which q^* had strictly positive entries, where the corresponding θ_k is less than or equal to $\frac{x_0}{s}$, then (3.4.2) would go to zero as $n \to \infty$:

$$\begin{aligned}
\mathbb{P}\Big(\cap_{t=1}^{nT}\{\text{No one gets blocked at time } t\}\Big) &= 1 - \mathbb{P}\Big(\cup_{t=1}^{nT}\{\text{Some one gets blocked at time } t\}\Big) \\
&\geq 1 - \sum_{t=1}^{nT} \mathbb{P}(\text{Some one gets blocked at time } t) \\
&= 1 - nT\mathbb{P}(\{\text{Some one gets blocked at time } t) \\
&= 1 - nT\mathbb{P}\left(\sum_{i=[t-s+1]^+}^{t}\sum_{k=1}^{K} Bin(n, \theta_k \mathbb{1}_{\{k\}}(i)) \geq nx_0\right) \\
&= 1 - nT\mathbb{P}\left(\sum_{j=1}^{n}\sum_{i=[t-s+1]^+}^{t} Bern_j(\theta_{k(i)}\mathbb{1}_{\{k(i)\}}(i)) \geq nx_0\right) \\
&= 1 - \\
&\quad nT\mathbb{P}\left(\sum_{j=1}^{n}\sum_{i=[t-s+1]^+}^{t} Bern_j(\theta_{k(i)}\mathbb{1}_{\{k(i)\}}(i)) - ns\theta_k \geq n(x_0 - s\theta_k)\right) \\
&= 1 - \\
&\quad nT\mathbb{P}\left(\frac{\sum_{j=1}^{n}\sum_{i=[t-s+1]^+}^{t} Bern_j(\theta_{k(i)}\mathbb{1}_{\{k(i)\}}(i))}{ns} - \theta_k \geq \frac{x_0}{s} - \theta^*\right) \\
&\geq 1 - nTe^{-2n\delta^2} \quad (\delta = \frac{x_0}{s} - \theta_k) \\
(\text{as } n \to \infty) &\to 1,
\end{aligned}$$

where the second equality is due to the fact that q^* is applied throughout the time horizon and the last inequality is due to Hoeffding's inequality. This means that as n increases, the blocking probability goes to zero exponentially fast. This is the reason why assumption (ii) was made. If "bad" arms were being selected with some non-negative probability, then the seller will definitely have customers being blocked in the long-run since if q^* is strictly positive in entry k for which

$\theta_k > \frac{x_0}{s}$, then $\forall\ t \leq nT$

$$\begin{aligned}
\mathbb{P}(\text{Some one gets blocked at time } t) &= \mathbb{P}\left(\sum_{j=1}^{n}\sum_{i=[t-s+1]^+}^{t} Bern_j(\theta_{k(i)}\mathbb{1}_{\{k(i)\}}(i)) \geq nx_0\right) \\
&\geq \mathbb{P}\left(\sum_{j=1}^{n}\sum_{i=[t-s+1]^+}^{t} Bern_j(\theta_k) \geq nx_0\right)\left(q_k^*\right)^s \\
&= \mathbb{P}\left(\frac{\sum_{i=1}^{ns} Bern_i(\theta_k)}{ns} - \theta_k \geq \frac{x_0}{s} - \theta_k\right)(q_k^*)^s \\
&\to (q_k^*)^s. \quad \text{(Due to Hoeffding's inequality)}
\end{aligned}$$

If there are customers being blocked in the long-run, the revenue will not be amenable to computation because the blocking probabilities are not tractable. Therefore, from the above analysis, the revenue the seller gets to keep by applying this policy is everything in the long-run. The issue is that this is a constant policy and the seller does not know ahead of time, which price to set since the seller is unaware of the true buying probabilities and needs to estimate these quantities by setting different prices, i.e. exploring, but at the same time the seller cannot afford to spend much time exploring and needs to exploit the current information he/she has acquired throughout the application of the policy. The result, while surprising at first, made sense intuitively because of the fact that the LP is a 2D-knapsack LP. The main result is the following:

Theorem 3.4.1. *Let* $\delta = \frac{x_0}{s} - \theta^*$, $\Delta = \min_{b\in S}\{\theta_* p_* - \theta_b p_b\}$, $\beta = \max\left\{1, \frac{\Delta^2}{2\delta^2}\right\}$, $\frac{n}{\ln(nT)} \geq \frac{4\beta}{\Delta^2}$ *and* $H^{(n)}$ *be given as (3.6). Then with probability at least*

$$1 - \frac{K-1}{(nT)^{2\beta}} - \frac{1}{(nT)^{2\beta-1}} - nTe^{-2n\delta^2},$$

we get

$$Regret_n(T) = O\left(\frac{Ks}{T}\right).$$

Theorem (3.5.1) coupled with the fact that the probability that any one gets blocked throughout the time horizon decays to zero exponentially fast implies that the regret converges to zero exponentially fast.

3.5 Theoretical Analysis

Define the event $H^{(n)}$ by the following:

$$H^{(n)} = \left\{ \omega \in \Omega | \hat{\theta}_b^n p_b + \sqrt{\frac{\beta \ln(nT)}{n}} \leq \theta_* p_* \ \ \forall b \in S, \theta_* \leq \hat{\theta}_*^n(t) + \sqrt{\beta \frac{\ln(nT)}{nT_*(t)}} \ \ t^* \leq t \leq nT \right\}, \tag{3.6}$$

where S is the set of suboptimal arms and t^* is defined by the algorithm. We have the following result.

Theorem 3.5.1. *Let n be the scaling factor and define $H^{(n)}$ as above. Then if*

$$\frac{n}{\ln(nT)} \geq \frac{4\beta}{\Delta^2},$$

$H^{(n)}$ holds with probability at least

$$1 - \frac{K-1}{(nT)^{2\beta}} - \frac{1}{(nT)^{2\beta-1}}.$$

Proof.

$$\begin{aligned}
\mathbb{P}(H^{(n)}) &= 1 - \mathbb{P}(\bar{H}^{(n)}) \\
&= 1 - \mathbb{P}\left(\bigcup_{b \in S} \hat{\theta}_b^n p_b + \sqrt{\frac{\beta \ln(nT)}{n}} > \theta_* p_* \quad \bigcup \quad \bigcup_{t=t^*}^{nT} \theta_* > \hat{\theta}_*^n(t) + \sqrt{\frac{\beta \ln(nT)}{nT_*(t)}} \right) \\
&\geq 1 - \mathbb{P}\left(\bigcup_{b \in S} \left\{ \hat{\theta}_b^n p_b + \sqrt{\frac{\beta \ln(nT)}{n}} > \theta_* p_* \right\} \right) - \mathbb{P}\left(\bigcup_{t=1}^{Tn} \left\{ \theta_* > \hat{\theta}_*^n(t) + \sqrt{\frac{\beta \ln(nT)}{nT_*(t)}} \right\} \right) \\
&\geq 1 - \sum_{b \in S} \mathbb{P}\left(\hat{\theta}_b^n p_b + \sqrt{\frac{\beta \ln(nT)}{n}} > \theta_* p_* \right) - \sum_{t=1}^{Tn} \mathbb{P}\left(\theta_* > \hat{\theta}_*^n(t) + \sqrt{\frac{\beta \ln(nT)}{nT_*(t)}} \right) \\
&\geq 1 - \underbrace{\sum_{b \in S} \mathbb{P}\left(\hat{\theta}_b^n p_b + \sqrt{\frac{\beta \ln(nT)}{n}} > \theta_* p_* \right)}_{A} - \underbrace{\frac{1}{(nT)^{2\beta-1}}}_{B}.
\end{aligned}$$

Expression B follows from Hoeffding's inequality. For expression A, take a suboptimal arm $b \in S$. By assumption

$$\sqrt{\frac{\beta \ln(nT)}{n}} \leq \frac{\Delta}{2}. \tag{3.7}$$

Then

$$\begin{aligned}\mathbb{P}\left(\hat{\theta}_b^n p_b + \sqrt{\frac{\beta \ln(nT)}{n}} > \theta_* p_*\right) &= \mathbb{P}\left(\hat{\theta}_b^n p_b - \theta_b p_b > \theta_* p_* - \theta_b p_b - \sqrt{\frac{\beta \ln(nT)}{n}}\right) \\ &\leq \mathbb{P}\left(\hat{\theta}_b^n p_b - \theta_b p_b > \Delta - \sqrt{\frac{\beta \ln(nT)}{n}}\right) \\ &\leq \mathbb{P}\left(\hat{\theta}_b^n p_b - \theta_b p_b > \sqrt{\frac{\beta \ln(nT)}{n}}\right) \\ &\leq \frac{1}{(nT)^{2\beta}}.\end{aligned} \tag{3.8}$$

The last inequality follows from Hoeffding's inequality. Since $|S| \leq K-1$, we have

$$A \leq \frac{K-1}{(nT)^{2\beta}}.$$

Therefore,

$$\mathbb{P}(H^{(n)}) \geq 1 - \frac{K-1}{(nT)^{2\beta}} - \frac{1}{(nT)^{2\beta-1}}.$$

□

The above says that when n is large enough, the event $H^{(n)}$ are the instances where the UCB estimate of the optimal arm are better than the UCB estimates for the suboptimal arms throughout. This implies, via our algorithm and proved below, that the seller will choose the optimal solution throughout the time horizon. Therefore, other than the initial phase where all prices are tried once, the seller will choose the optimal solution with 100% certainty afterwards. The difference in regret will be due because of this initial testing phase. Since the blocking probability goes to zero exponentially fast, the revenue the seller would receive is essentially the same as the revenue had he/she had no capacity constraints and accepted everyone.

The optimization problem (3.4) at time t outputs a distribution over the arms $q^{(n)}(t)$, i.e. with probability $q_k^{(n)}(t)$ the seller sets the price at $\$p_k$. Each time period the seller sets price and observes the number of customers who make purchases, updates the UCB estimates $\bar{\theta}^{(n)}(t)$ and optimizes (3.4) again and repeats the same process until the end of the time horizon. Next we will show that for n large enough, with high probability the seller always chooses the optimal solution after the initial testing phase. Without loss of generality, we can assume exactly one arm is the optimal once. This is since the regret can only improve by having more optimal arms.

Theorem 3.5.2. *On $H^{(n)}$, the algorithm will always choose the optimal solution over the time horizon, except for the initial testing phase period.*

Proof. By assumption (ii), the optimal basis for (3.3) will include only the column corresponding

to the optimal solution θ_* and the slack variable. This implies that the optimal basis is of the form

$$B^* = \begin{bmatrix} \theta_* p_* & 1 \\ 1 & 0 \end{bmatrix}. \tag{3.9}$$

The inverse of B^* is

$$B^{*-1} = \begin{bmatrix} 0 & 1 \\ 1 & -\theta_* p_* \end{bmatrix}.$$

No other basis is optimal since it would either contradict the assumption or contradict non-negativity of the constraints. LP optimality is looking at the reduced costs of the non-basic variables. In this case for our maximization problem, the reduced costs take the following form:

$$\begin{aligned} \theta_b p_b - \begin{bmatrix} \theta_* p_* & 0 \end{bmatrix} B^{*-1} \begin{bmatrix} \theta_b \\ 1 \end{bmatrix} &\leq 0 \\ \theta_b p_b - \begin{bmatrix} \theta_* p_* & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & -\theta_* p_* \end{bmatrix} \begin{bmatrix} \theta^b \\ 1 \end{bmatrix} &\leq 0 \\ \theta_b p_b &\leq \theta_* p_*. \end{aligned} \tag{3.10}$$

From Theorem (3.5.1), $H^{(n)}$ occurs with high probability for a large enough n. $H^{(n)}$ implies that after large enough n,

$$\bar{\theta}_b p_b \leq \bar{\theta}_* p_*. \qquad (*)$$

Therefore, for n large enough, (*) occurs with high probability. This means that the seller implementing the algorithm, the seller will choose the optimal solution throughout the time horizon, except for the initial testing phase. The regret will be attributed almost entirely to the initial testing phase and any blocking that might occur, which happens with probability that approaches zero exponentially. □

Theorem (3.4.1) follows from the above.

Proof. Theorem (3.4.1)

Define a new event $\Phi(n) = \{H^{(n)} \ \& \ \cap_{t=1}^{nT} \{\text{No one gets blocked at time } t\}\}$. If n satisfies the inequality above, then from Theorem (3.5.1) and the blocking probability analysis done in section (3.4.2), the probability of event $\Phi(n)$ is at least

$$1 - \frac{K-1}{(nT)^{2\beta}} - \frac{1}{(nT)^{2\beta-1}} - nTe^{-2n\delta^2}.$$

Theorem (3.5.2) implies that on $\Phi(n)$, the seller will choose the optimal price $\$p^*$ after the initial testing phase. The initial testing phase is at most sK time periods so the seller will miss out on at most $n\theta_* p_* Ks$ of revenue. This implies that on $\Phi(n)$, the regret is

$$\begin{aligned} Regret_n(T) &\leq n\theta_* p_* - \frac{n\theta_* p_*(nT - Ks)}{nT} \\ &\leq \frac{Ks}{T} \\ &= O(1). \end{aligned} \tag{3.11}$$

□

If the algorithm chooses bad arms in the long-run with positive probability, then blocking will occur and will never reach the long-run average revenue rate that one can achieve without capacity constraints. In this case, we will not be able to compare the long-run average revenue rates of the seller with capacity constraints and seller without capacity constraints since the blocking probability is complex to handle. Due to assumption (ii), we can compare the long-run results since the blocking probability for both will approach zero at an exponential rate.

3.6 Experimental Results

3.6.1 Blocking Probability Scenarios

Before examining the policy from Section 3.4.1, we will examine the system state as the time horizon and initial inventory are scaled proportionally under the assumption that the optimal θ^* is strictly less than x_0/s and when it is violated. As mentioned before, if we did not include assumption (ii), then in the long-run, customers will be blocked with some non-negative probability. The figures below are two scenarios when the assumption is barely satisfied versus when it is not satisfied for different scaling factors.

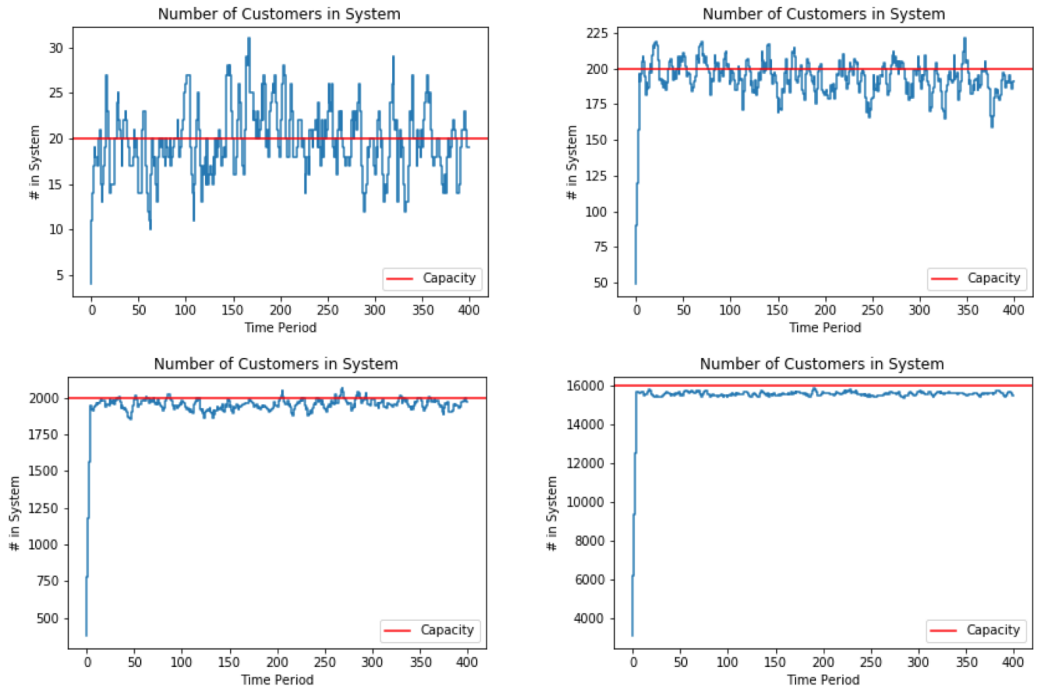

Figure 3.1: Top Left: Scaling factor n=10. Top Right: Scaling Factor n=100. Bottom Left: Scaling Factor n=1000. Bottom Right: Scaling Factor n=8000

The parameters of the artificial data are $s = 5$, $x_0 = 2$, $T = 50$, true buying probabilities $\theta = (0.6, 0.25, 0.4, 0.2, 0.39)$, and the prices for each corresponding θ, $p = (0.3, 0.7, 0.2, 0.4, 0.5)$. The optimal price corresponds to the buying probability $\theta^* = .39$, which satisfies assumption (ii). For the top left figure, the scaling factor is $n = 10$, which means that the initial inventory for the scaled problem is 20 (shown as the red line), and the same goes for the rest of the other figures. Averaging over 100 simulations, the table below demonstrates the number of periods over the 400 time periods, the number of periods where blocking would have occurred.

	Total # of Blocks
$n = 10$	154
$n = 100$	102
$n = 1000$	25
$n = 8000$	0

Table 3.1: # of blocking occurrences out of the 400 time periods.

The next set of figures is the scenario where the assumption is violated, according to Section 3.4.2, there should be some non-negative blocking probability, however small. The parameters are such the same as before but now we assume that the seller chooses $\theta^* = 0.4$, which is equal to the ratio x_0/s. The graphs show that blocking occurs even as n grows unboundedly.

	Total # of Blocks
$n = 10$	185
$n = 100$	186
$n = 1000$	206
$n = 8000$	200

Table 3.2: # of blocking occurrences out of the 400 time periods.

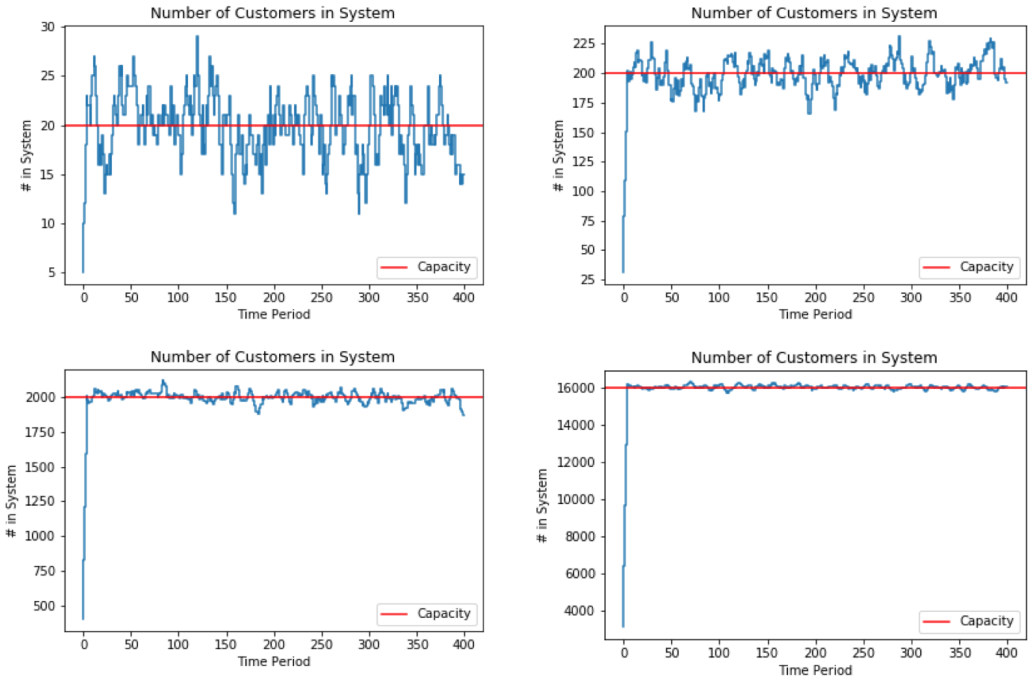

Figure 3.2: Top Left: Scaling factor n=10. Top Right: Scaling Factor n=100. Bottom Left: Scaling Factor n=1000. Bottom Right: Scaling Factor n=8000

Table 3.2 below shows the number of time periods where blocking occurred. It can be clearly seen that the number of blocking periods does not subside as the system scales compared to Table 1.

We also tested out the scenario when the fraction of time is spent on two prices, one where it chooses $\theta_1 = 0.25$ 90% of the time and $\theta_2 = 0.6$ is chosen 10% of the time. Though the time periods where blocking occurs is small, the blocking probability does not subside even as the system scales as Figure 3.2 shows. This implies that the seller will lose out on a portion of the revenue regardless of how big the system gets.

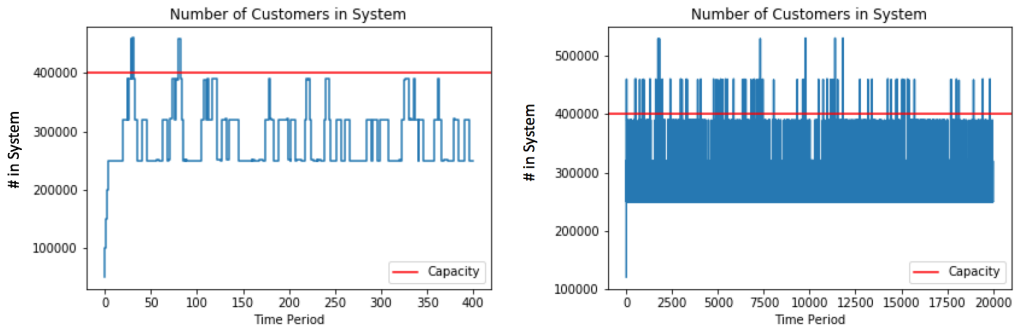

Figure 3.3: Left: Scaling factor n=200K. Right: Scaling Factor n=200K.

Figure 3.3 shows the same scaling but the right figure is run for a longer period of time to demonstrate that blocking still occurs. All scenarios have an average capacity rate that is strictly less than x_0, which is enforced via the optimization. The differences is whether blocking subsides or not, which depends on which buying probability the seller focuses on.

3.6.2 Algorithm Experiments

This section applies the algorithm on the same set of parameters as Section 3.6.1. In this case the assumptions are satisfied and the seller initially experiments with each price and and then applies prices according to the probability distribution $q^{(n)*}(t)$ provided by the output of the optimization problem (3.4), using the UCB estimates, $\bar{\theta}^{(n)}(t)$, of the buying probabilities. Figure 3.4 shows the number of customers in the system when n is small. Blocking events are evident. The seller can either decide to fulfill the demand until it runs out, and some customers will not get service. Another option is to choose the highest price when seller's capacity is less than or equal to n. In that case, all purchasing customers will get service. We chose the former. But as Figure 3.5 shows, as the system scales largely, the blocking events are rare.

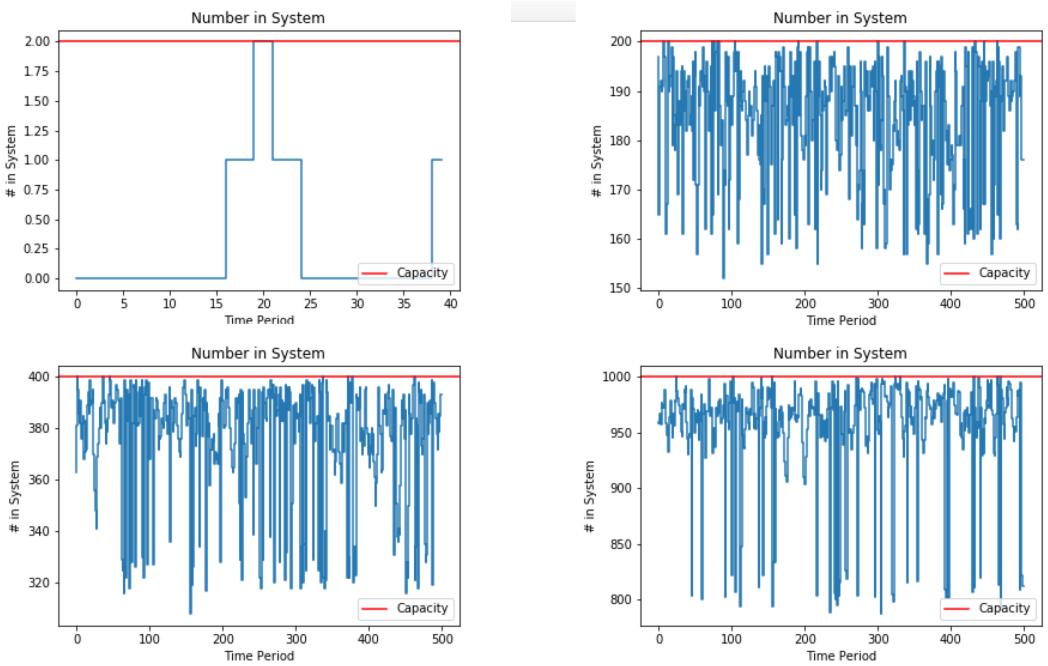

Figure 3.4: Top Left: Scaling factor n=1. Top Right: Scaling Factor n=100. Bottom Left: Scaling Factor n=200. Bottom Right: Scaling Factor n=500

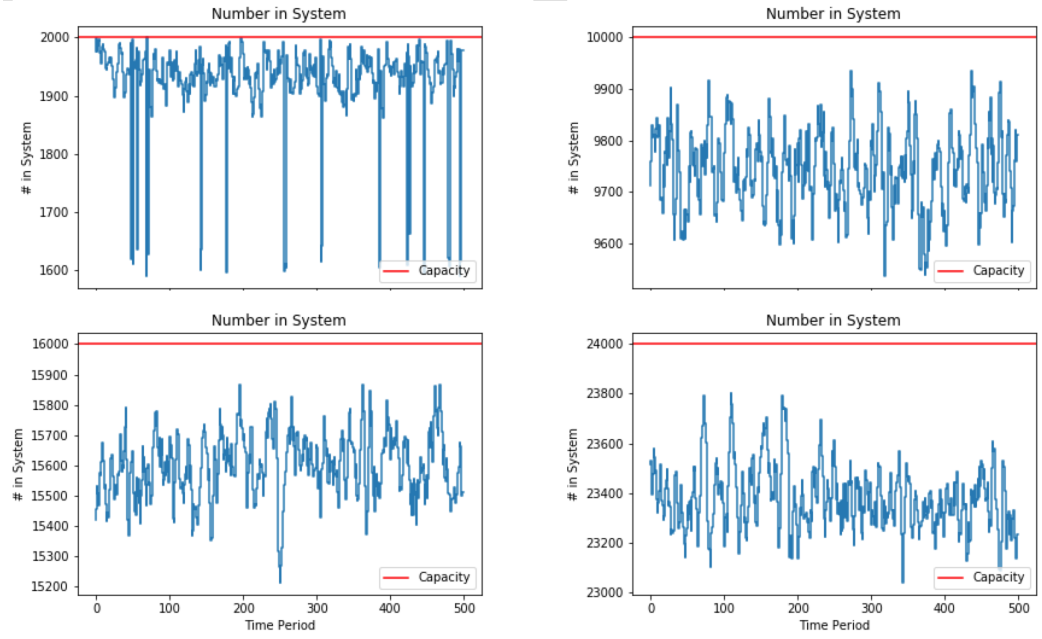

Figure 3.5: Top Left: Scaling factor n=1K. Top Right: Scaling Factor n=5K. Bottom Left: Scaling Factor n=8K. Bottom Right: Scaling Factor n=12K

The n that satisfies Theorem 3.4.1 is rather large and the simulation for just one run would have taken a really long time. From Figures 3.4 and 3.5 above and Table 3.3 below empirically

	Fraction of Time Periods
$n = 1$	4%
$n = 100$	17%
$n = 200$	15%
$n = 500$	9%
$n = 1000$	5%
$n = 5000$	0.06%
$n = 8000$	0%
$n = 12000$	0%

Table 3.3: Percentage of blocking occurrences out of the 400 time periods.

	R_π/R_*
$n = 1$	24%
$n = 100$	95.5%
$n = 200$	98.5%
$n = 500$	99.0%
$n = 1000$	99.5%
$n = 5000$	99.8%
$n = 8000$	100.0%
$n = 12000$	100.0%

Table 3.4: Ratio of seller's policy to the clairvoyant policy.

shows that the high probability event seems to happen way sooner. For $n \leq 500$, we averaged over 15 simulations and was chosen arbitrarily. But for larger n, the result is due to a single run, but blocking does not occur over the time horizon. Recall that as n increases, so does the time horizon.

The ratio of the best policy, R_*, to the seller's policy, R_π, is provided in Table 3.4. Recall that our results hold for large n given as in Theorem 3.4.1. It does not take a big system to reach a good performance.

CHAPTER 4

Reinforcement Learning in Network Revenue Management with Reusable Resources

4.1 Introduction

We consider the pricing problem that a firm that sells reusable resources faces when the objective is to maximize revenue through careful allocation of resources. Dynamic pricing is a fundamental problem faced by many firms, whether selling reusable and non-reusable resources. They have to adjust product prices accordingly to the right customer at the right time Lin (2006) based on their inventory, demand, competition, in an attempt to maximize their revenue without incurring many unhappy customers, as that is a cost in itself which is very hard to properly define and measure. The common characteristics of the industries that sell reusable resources are that the initial inventory are known and fixed at the beginning of the time horizon and no re-ordering of resources is allowed. This chapter is concerned with tackling the dynamic pricing problem using reinforcement learning to price the resources dynamically to maximize revenue over a finite time horizon in the face of uncertainties such as the demand arrival process, service time, and advance reservation times. Service time and advance reservation times are unknown to the seller until a customer makes a "purchase", at which point the customer reveals the future point in time when he/she would like to commence service and for how long the resource will be used for.

Industries that apply dynamic pricing strategies are manufactured goods Tsaia and Hung (2009), such as perishable food items and electronics. But our work focuses on the other end of the spectrum of industries that sell reusable resources. These industries includes the lodging industry, car rental, cloud computing and temporary work staffing, e.g., Levi and Radovanovic (2010), Lei and Jasin (2016b), Chen et al. (2017), Chen and Shi (2016), Bernal and Shi (2019).

The literature on network revenue management with reusable resources is dwarfed by the literature in the perishable case, with many variations of the problem. The results in the revenue management with reusable resources literature are asymptotic results, where policy performance

becomes optimal in the limit as quantities such as initial resource capacity and demand are scaled largely (see Levi and Radovanovic (2010), Lei and Jasin (2016b), Chen et al. (2017), Chen and Shi (2016), Bernal and Shi (2019)), and even for non-reusable/perishable cases (see Gallego and van Ryzin (1994, 1997), Besbes and Maglaras (2009), Besbes and Zeevi (2011)). In practice, initial capacity of reusable resources can range from a few hundreds to a few tens of thousands. From the American Hotel & Lodging Association, there are about 54,200+ hotels servicing about five million rooms. That equates to a little less than 100 rooms per hotel. For well-known hotels, with hotels around the US, an overestimate can range to a total of a few hundred thousand. It is still possible to use the pricing heuristics from the literature, but the scale at which the firm is operating might not be large enough to produce satisfactory results. Academic papers in revenue management assume a functional relationship between the price and arrival rate is known to the decision-maker Chen and Shi (2019) which makes the problem tractable and conveys great insight, but would prove ineffective in practice due to these strong assumptions. Another issue when imposing a structural form on the demand function is that it leads to model misspecification, which can provide provide suboptimal results Wang (2019).

The main objective of this chapter is to propose a model-free approach to the dynamic pricing problem, where the transition probabilities between states, in other words, demand behavior, are not specified, thus the model-free approach. The reinforcement learning control problem is a method to solve problems of optimal strategies under stochastic environments, and this can be done in a model-free way but also a model-based as well using a "model", where the "model" transition probabilities are estimated and taken to be the true model and updated. The contribution of this chapter is to propose a computational method to determine a good dynamic pricing policy to the network revenue management with reusable resources when information is incomplete and demand is stationary. In this article we use deep deterministic policy gradients (DDPG) originally proposed by de Bruin et al. (2015).

The chapter is organized as follows. Section 2 presents a literature review. Section 3 describes the model formulation and describes how reinforcement learning can solve the dynamic pricing problem. Section 4 presents numerical results of DDPG applied to the pricing problem and compare to the fluid model, which is the used model when the scale of the problem is large enough, i.e. large demand, large capacity, that the randomness of the environment does not really affect how a decision-maker should price. Essentially, the *fluid* regime, is the regime where the stochasticity is washed away and we basically have a deterministic problem.

4.2 Model Formulation

4.2.1 Markov Decision Process

In this chapter, we model the dynamic pricing problem of a reusable resource as a finite horizon, discrete state Markov decision process (MDP). We aim to use reinforcement learning to approximate the optimal pricing strategy that maximizes the revenue given a fixed time period and initial capacity.

We consider the time horizon to be one year, analogous to firms fiscal year reporting on their annual performance. Prices are allowed to change daily. They depend on the current inventory and time left until the end of the horizon. Our setup is as follows. There are m products that the firm sells and each product is made up of n resources. There is a bill-of-materials matrix A where $A_{i,j}$ is the number of resources i required to make product j. We assume that A has all integer entries. The firm additionally have an initial capacity of c_i for each resource i. We assume that the arrival, or demand, process is a Poisson process with rates $\lambda_1, \ldots, \lambda_m$. The prices for each product belong in an m-dimensional box, i.e. $p \in P = [a_1, b_1] \times \ldots \times [a_m, b_m]$, where if any of the products are priced at the upper limit, then there is no demand. The relevant elements of the MDP are:

- State space $S = \{s \in \mathbf{Z}_+^n | s = c - Ax, x \in \mathbf{Z}_+^m\}$ represents the remaining capacity at the end of the day.
- $T = \{1, 2, \ldots, 365\}$ represents the set of time steps at which pricing decision will be applied. The time horizon is 365 days, representing one year, or what is called an episode in the reinforcement learning vernacular.
- $a(s_t) \in P$ denotes the set of available actions at the beginning of time period t.
- Transition probabilities $P_t(s_{t+1}|s_t, a_t)$ which denotes the probability of going to state s_{t+1} at time period $t+1$ given that action a_t was taken in state s_t in time period t.
- Revenue function $r(s_t, a_t, s_{t+1})$.

The optimal pricing policy can be computed via the the Bellman optimality equations

$$V(s_t) = \max_{a(s_t) \in P} \mathbb{E}_{s_{t+1} \sim T(s_t, a_t)}[r(s_t, a(s_t), s_{t+1}) + V(s_{t+1})]. \tag{4.1}$$

The issue is getting a hold on the transition probabilities since many arrivals can occur between decision periods. Even if the transition probabilities were available, the state space can explode when either more resources are considered and/or initial resource capacity is large. This is where

model-free reinforcement learning comes in to bypass the need for these transition probabilities. In other words, and a popular form in the reinforcement learning literature, we want to maximize the total expected revenue collected throughout the episode, i.e., we want to find a policy π that maximizes

$$V^{\pi}(s_0) = \mathbb{E}_{\pi}[\gamma^0 r(s_0, a_0) + \gamma(s_1, a_1) + \ldots + \gamma^{365} r(s_{365}, a_{365})|s_0], \tag{4.2}$$

where γ is the discount factor, usually 0.99, s_0 is the initial number of resources for each product. The policy π is a function that maps an element of the product space $S \times T$ to an element in the action space $P.$, i.e. $\pi : S \times T \rightarrow P$.

4.2.2 Background on Reinforcement Learning

In reinforcement learning, the goal is to learn a policy to control a system with states $s \in S$ and actions $a \in P$ in an stochastic environment, so as to maximize the expected sum of returns according to the reward function $r(s, a)$. The dynamical system is defined by an initial state distribution $p(s_1)$ and transition distribution $P(s_{t+1}|s_t, a_t)$. At each time step $t \in [1, |T|]$, the agent chooses an action a_t according to its current policy $\pi(a_t|s_t)$, and observes a reward $r(s_t, a_t)$. The agent then experiences transitions to a new state sampled from the transition distribution, and we can express the resulting state visitation frequency of the policy π as $\rho^{\pi}(s_t)$. Let $R_t = \sum_{i=t}^{|T|} \gamma^{i-t} r(s_t, a_t)$. The goal is to maximize the expected sum of returns. We use a finite horizon for all of the tasks in our experiments. The expected return R_1 can be optimized using a variety of model-free and model-based algorithms. In this section, we review the model-free framework used in our work.

4.2.3 Model-Free Reinforcement Learning

When the system transition dynamics $P(s_{t+1}|s_t, a_t)$ are not known, as is often the case with physical systems such as robots, policy gradient methods Peters and Schaal (2006) and value function, or Q-function, learning with function approximation Sutton et al. (1999) are often preferred. Policy gradient methods provide a simple, direct approach to RL, which can succeed on high-dimensional problems, but potentially requires a large number of samples. Off-policy algorithms that use value or Q-function approximation can in principle achieve better data efficiency Lillicrap et al. (2016). Although, Q-learning is usually adapted in the finite state and action space, there are extensions of the Q-learning algorithm for continuous action space, our work implements actor-critic learning known DDPG. For continuous action problems, Q-learning becomes difficult, because it requires maximizing a complex, nonlinear function at each update. For this reason, continuous domains are often tackled using actor-critic methods, e.g., Lillicrap et al. (2016), Silver et al. (2014), Hafner

and Riedmiller (2011). Actor-critic learning has a major advantage over current implementations of Q-learning; the ability to respond to smoothly varying states with smoothly varying actions. Actor-critic systems can form a continuous mapping from state to action and update this policy based on the local reward signal from the critic. However, adapting such methods to continuous tasks typically requires optimizing two function approximators on different objectives as compared to one objective for Q-learning.

4.2.4 DDPG for Dynamic Pricing in Network Revenue Management with Reusable Resources

Model-free reinforcement learning has been successfully applied in robotics Peters and Schaal (2006), machine scheduling Ye et al. (2018), playing Atari games Mnih et al. (2013), cybernetics, psychology, and computer science disciplines Sutton and Barto (1998). There has been a surge of interest in model-free reinforcement learning after it was successfully applied to learn to play many old Atari video games Mnih et al. (2013), using one generic structure with deep neural networks and Q-learning. Model-based reinforcement learning solves for the optimal policy using past experience. An advantage in using reinforcement learning is that it can adapt to a changing environment through experience. Here we propose optimizing over the policy space directly instead of optimizing over the action-value space as it has been shown in practice to have better convergence properties at the expense of taking longer to train. We used DDPG instead of any other policy gradient algorithms because they are stochastic, in other words, the output is a stochastic policy. A stochastic policy does not make sense in practice as it will produce different prices for the same state since we are sampling a distribution.

During the learning process, the agent is exposed to the environment gaining experience and collecting rewards. Since we will be optimizing over the policy space, we need a performance objective with respect to the actions, which itself will be parameterized by a neural network. The performance we define is

$$J(\mu_\theta) = \mathbb{E}[R_0|\mu] = \mathbb{E}[\gamma^0 r(s_0, a_0) + \gamma(s_1, a_1) + \ldots + \gamma^{365} r(s_{365}, a_{365})|\mu]. \tag{4.3}$$

The above does not work because the target policy we are trying to learn is changing every time we update the parameters θ. The beauty of DDPG is that we can learn a good pricing policy when the agent follows a different policy, called the *behavior* policy, which it uses to navigate the environment. This is known as *off-policy* method. By updating the performance objective to average over the behavior policy, denoted as β from here on out, instead of the target policy we are

trying to learn, we get

$$J_\beta(\mu_\theta) = \mathbb{E}_{s\sim\beta}[R_0|\mu] = \mathbb{E}_{s\sim\beta}[Q^\mu(s,\mu_\theta(s))]. \tag{4.4}$$

Silver et al. (2014) proved that the gradient of (4.4) exists and approximately equals

$$\nabla_\theta J_\beta(\mu_\theta) = \mathbb{E}_{s\sim\beta}[\nabla_\theta \mu_\theta(s) \nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)}]. \tag{4.5}$$

This means we can use experience through simulation to approximate the expected policy gradient using the behavior policy. The behavior policy is where stochasticity is injected to induce action exploration in the environment while approximating our target pricing policy. Now that we have the policy gradient, the algorithm is the following, taken from Lillicrap et al. (2016):

Algorithm 4.1 DDPG Algorithm

Randomly initialize critic network $Q(s,a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ.
Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$ and $\theta^{\mu'} \leftarrow \theta^\mu$.
Initialize replay buffer R
for $episode = 1 : M$ **do**
    Initialize a random process $\mathcal{N}$ for action exploration
    Receive initial observation state s_1
    for $t = 1 : T$ **do**
        Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to current policy and exploration noise
        Execute action a_t and observe reward r_t and observe new state s_{t+1}
        Store transition (s_t, a_t, r_t, s_{t+1}) in R
        Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from R
        Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
        Update critic by minimizing the loss $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
        Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q(s_i, a_i|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}$$

        Update the target networks:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$

$$\theta^{\mu'} \leftarrow \tau\theta^\mu (1-\tau)\theta^{\mu'}$$

    end for
end for

4.3 Experimental Results

4.3.1 Data

The data structure defined in Section 4.2.1 is, to our knowledge, nonexistent. Therefore, we created a simulator that takes the following as inputs to get initialized to simulate:

- Number of resources, m, and number of products, n
- Initial resource capacity, $c \in \mathbb{R}^m$
- Max demand rate, $\lambda \in \mathbb{R}^n$
- Min and max price intervals for all products, i.e. $[p^i_{min}, p^i_{max}]$ for all products i
- Bill-of-materials integer matrix $A \in \mathbb{R}^{m \times n}$, where A_{ij} denotes the number of resources i required to make one j product.
- Service time distribution for each product
- Advance reservation distribution for each product

When the simulator is initialized, it acts as a function. It will take as input an action, in this case, feasible prices for each of the products, then it will simulate the arrivals, departures, advance reservation, service. The specific quantities for the above parameters are the following:

- m=4, n=3, with bill-of-materials matrix $A = \begin{pmatrix} 1 & 1 & 1 \\ 3 & 2 & 3 \\ 1 & 3 & 1 \\ 3 & 1 & 1 \end{pmatrix}$
- Initial resource capacity $c = (12, 36, 30, 20)$
- Minimum and maximum product prices $p_{min} = (1, 2, 1)$ and $p_{max} = (4, 7, 5)$, respectively.
- Max demand rate for each product: $\lambda^* = (4, 8, 4)$ customers per day
- Service time distribution with support $\{1, \ldots, 5\}$ with distribution $(.3, .25, .35, .05, .05)$
- Advance reservation distribution with support $\{0, 1 \ldots, 10\}$ with distribution $(.25, .20, .2, .1, .08, .06, .05, .03, .01, .01, .01)$

Note that $A\lambda^* = (16, 40, 32, 24)^T$, which is greater than the initial resource capacity. Additionally, the maximum demand rate occurs when the prices are at its lowest. The demand rate varies with pricing, implying that the Poisson process mean demand rate will vary with pricing. In particular, the demand rate for each product varies inversely proportional to the exponential function, i.e. $\lambda_i \propto k_i e^{-p_i}$. The constant parameters were set such that at the minimum price, the mean demand rate is λ_i^*, and at the maximum price, the mean demand rate is zero.

From Algorithm 4.1, both the critic network and actor network will be deep neural networks with the following parameters:

Critic Network

- 2 hidden layers with 300 hidden nodes each.
- ReLu activation function and no activation at the final layer
- Learning rate = 1E-3

Actor Network

- 2 hidden layers with 250 hidden nodes each.
- ReLu activation function
- Sigmoid final layer activation
- Learning rate = 1E-4

These parameters were arrived at using grid search on $H \times A$ where $H = \{150, 250, 300\}$ are the number of hidden nodes and $A = \{ReLu, Sigmoid\}$ are the activation functions. The learning rates used are standard learning rates for the critic and actor networks. No batch normalization was used for the actor network since the states (i.e. remaining capacity) are on the same scale. One note of importance is due to the training phase. Unlike supervised learning where training is stopped when the validation error is small and either stops decreasing further or starts to increase. In our case, DDPG has two deep neural networks, one for the actor policy and one for the action-value function, Q. The actor policy network has no stopping criteria and the critic network does minimize a loss function, but the stopping criteria we used was when the number of blocks stabilized in the validation data. We stopped as soon as number of blocks stabilized over five episodes. Figure 4.1 shows the number of blocks during the training phase. It took approximately 24 hours for training over 40 episodes. From Figure 4.2, the reward reaches a steady state early on, and had we stopped early, our validation data would have incurred blocking. But training kept on until the number of blocks stabilized over 5 episodes, leading to virtually no blocking in the validation set.

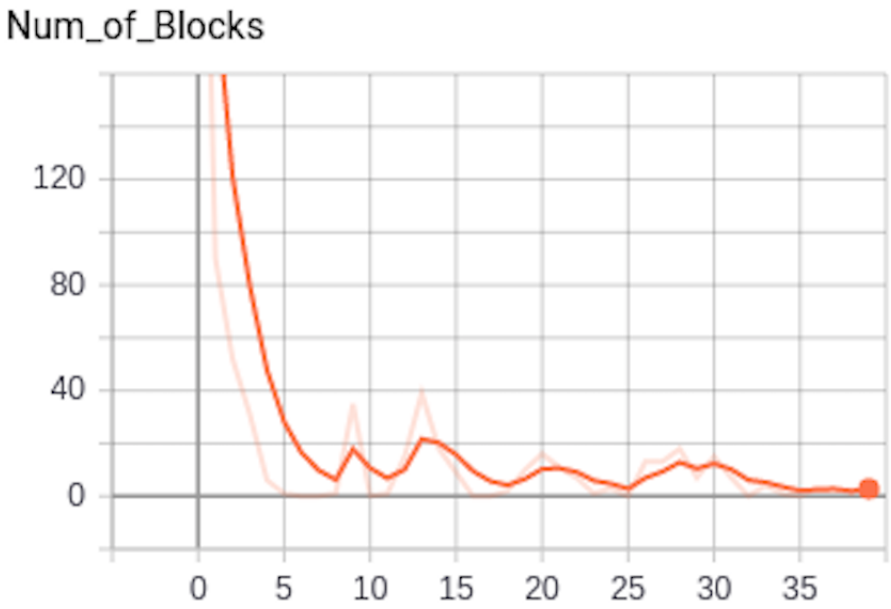

Figure 4.1: Tensorboard's record of total number of blocks in the training phase.

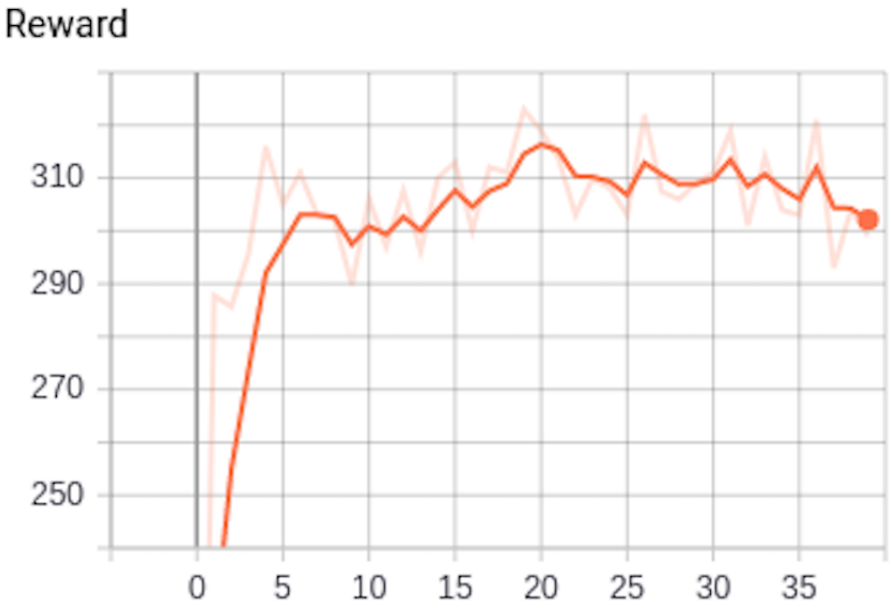

Figure 4.2: Tensorboard's record of number of total reward in the training phase.

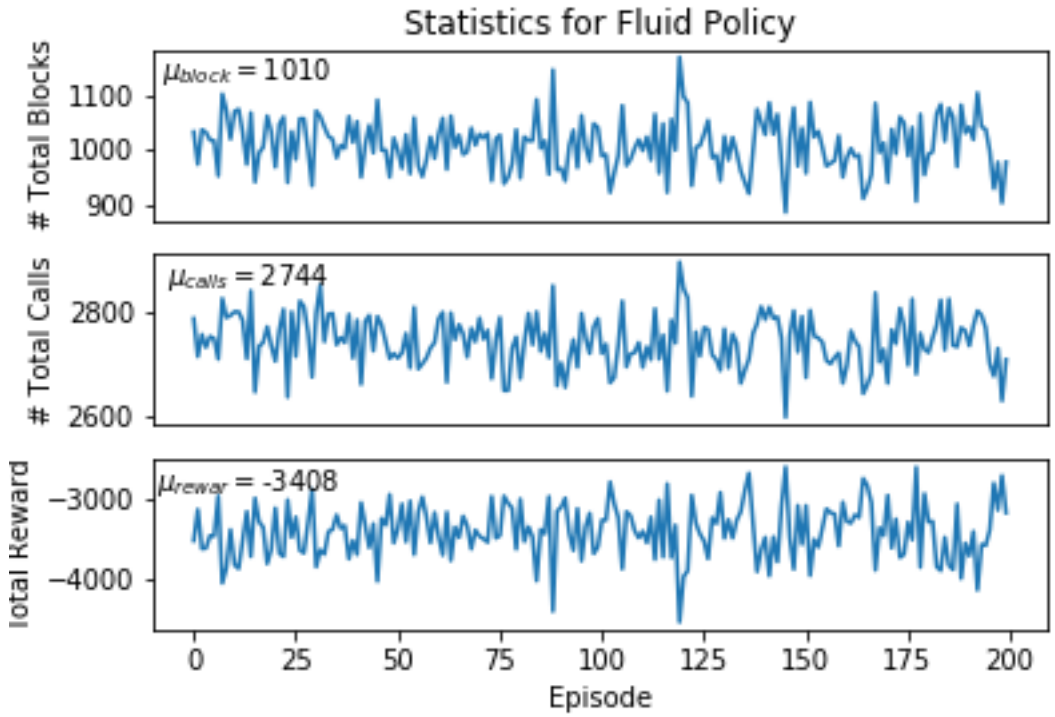

Figure 4.3: Performance of fluid pricing policy.

4.3.2 Simulation Results

We compare the results to the fluid model, where it looks to naively optimize the revenue rate subject to using less than the available initial capacity *in expectation*. Since the constraints are in expectation, we expect that there will be a large amount of blocking. Figure 4.3 confirms that indeed there will be a significant, and unacceptable, amount of blocking. When running the DDPG algorithm, we penalize the agent by $5 \max_i \{p_i\}$ where i ranges over the products. If from one state to the other, i.e., from one day to the next, profit generated from purchasing customers outweighed aggregate penalties, then the reward is 1. Otherwise, if the aggregate penalties outweighed generated profit, the reward to the agent is -1. Given this reward structure, the DDPG pricing policy is shown in Figure 4.4. On average, rounding to the nearest integer, the DDPG pricing policy rarely generated any blocking events *per* episode, i.e. annually. Analyzing the pricing policy further, it is seen that the prices are not static, i.e. fixed, but it is not erratic as well. The policy's pricing varied to only a few prices. The pricing policy seemed to be pricing in such a way as to always have sufficient resources at hand for the next period. But one can see from Figure 4.3 to 4.4, the average number of calls into the system decreased drastically, from 2744 calls to 472 calls.

Intuitively, it makes sense that if we penalize the agent less, then there would be an increase

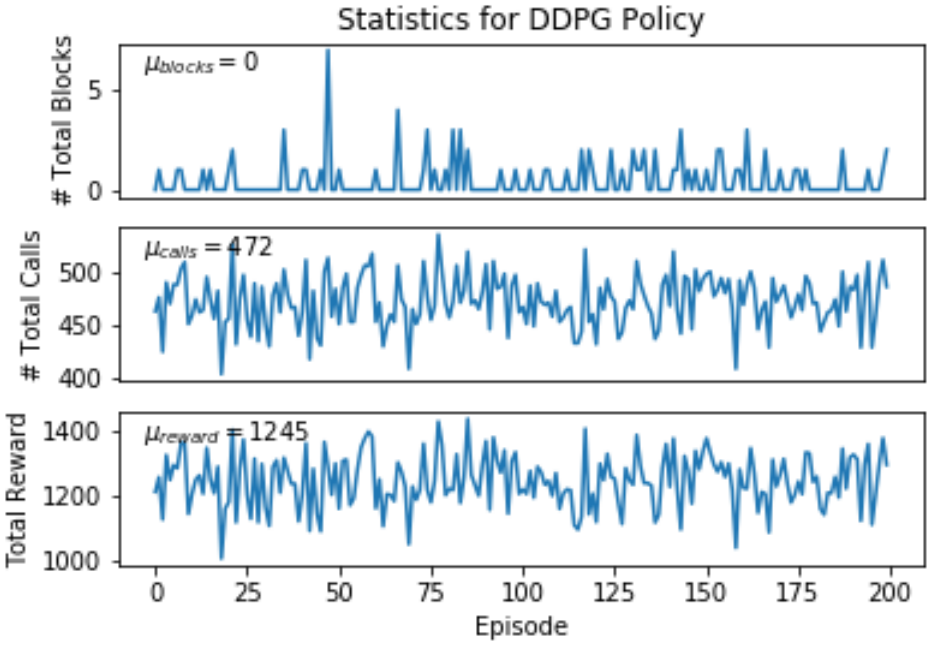

Figure 4.4: Performance of DDPG pricing policy with penalization $5 \max_i\{p_i\}$.

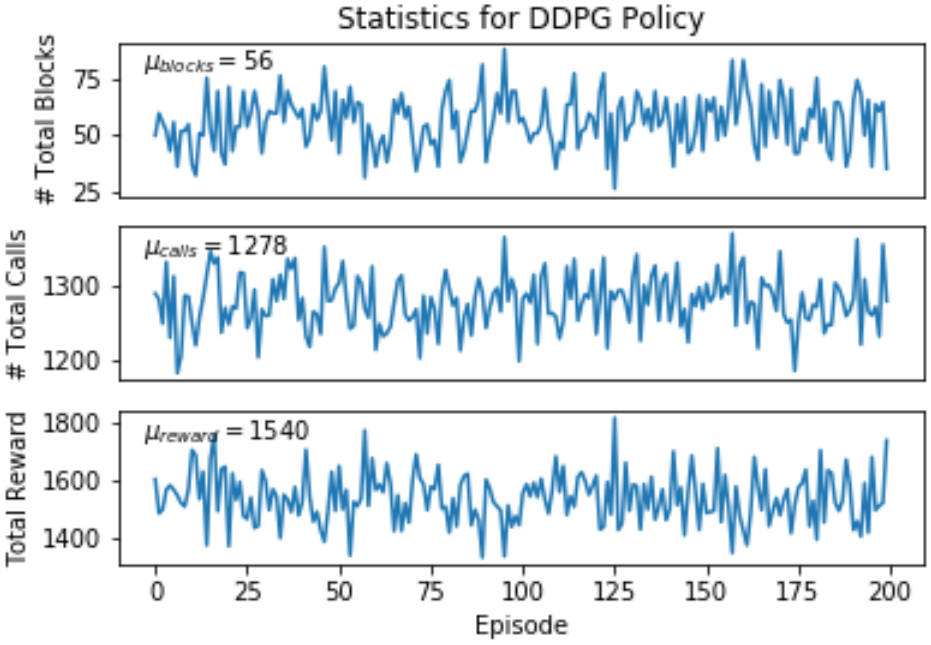

Figure 4.5: Performance of DDPG pricing policy with penalization $\max_i\{p_i\}$.

in calls, and subsequently an increase in blocks and total reward. To confirm this hypothesis, we penalized the agent by $\max_i\{p_i\}$ for every blocking occurrence. The DDPG algorithm outputted results that confirmed our intuition. Figure 4.5 confirms this. The DDPG algorithm, if configured well, finds a good but suboptimal policy. We uniformly sampled the pricing feasible set P, and took that as a fixed pricing policy to test out the statistics of these policies to compare to the DDPG policy. The results are recorded in Table 4.1. We only took those samples that generated positive rewards and for each we averaged over 100 episodes. One observation is that there are there are fixed pricing policies that perform better in terms of rewards and blocks, but the total number of generated calls is small. This would be an issue if the seller wants traffic.

Price($)	Reward	# of Blocks	# of Calls
[3.26, 2.98, 4.58]	2414	10	835
[3.74, 3.73, 2.16]	1997	0	670
[2.31, 4.38, 1.32]	2081	49	1287
[2.46, 2.75, 1.77]	1644	341	2009
[2.91, 4.24, 2.78]	1679	0	494
[2.82, 4.54, 2.49]	1458	0	456
[1.96, 4.54, 1.75]	2303	27	1108

Table 4.1: Performance of uniformly generated static policies to compare to DDPG.

The pricing strategy varied drastically in their determinism. In other words, the pricing strategy derived from the base model is almost like a fixed policy since the price does not deviate within five cents for one product, and in particular, does not deviate more than two cents for two products. This is seen in Figure 4.6. Two price trajectories, i.e. samples, are shown for the base model where the agent is penalized $5\max_i\{p_i\}$ when blocking incurs more costs than it brought in revenue. When the agent is penalized less for blocking events, the pricing policy essentially alternates between two distinct upper and lower bounds for two products and the last product is essentially at its upper limit. This is shown in Figure 4.7. One price trajectory was plotted; more trajectories would make the outputted graphs very unclear. This pricing policy generated significant revenue above the base model with more traffic and a very slight increase in blocking events.

4.4 Conclusion and Future Research

This chapter was concerned with using model-free reinforcement learning to solve the dynamic pricing policy of a firm who has a fixed finite number of *reusable* resources with which he/she uses to make products. We have shown through simulation results that DDPG converge to a suboptimal

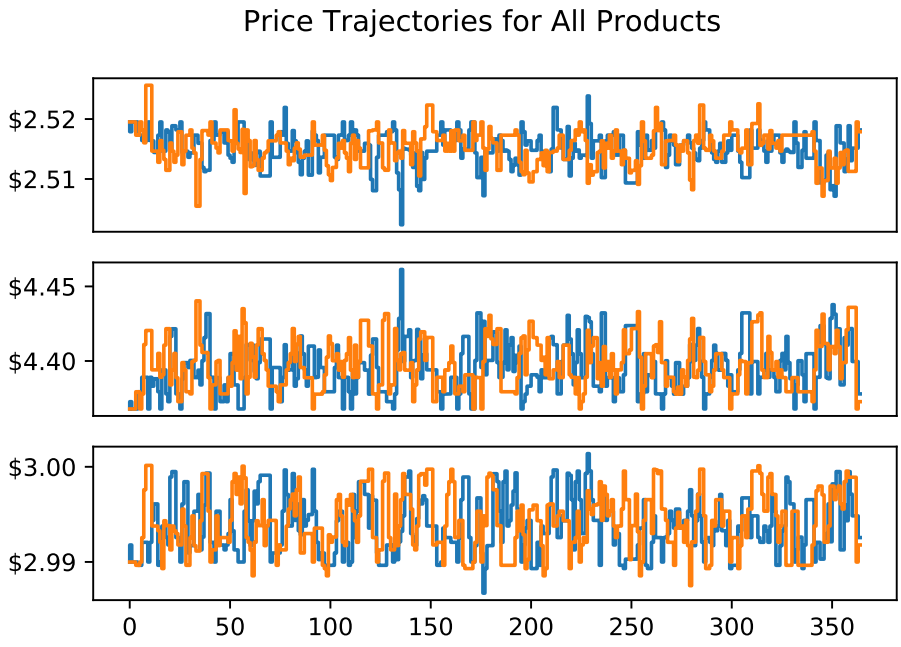

Figure 4.6: Two price trajectories for all 3 products for base model.

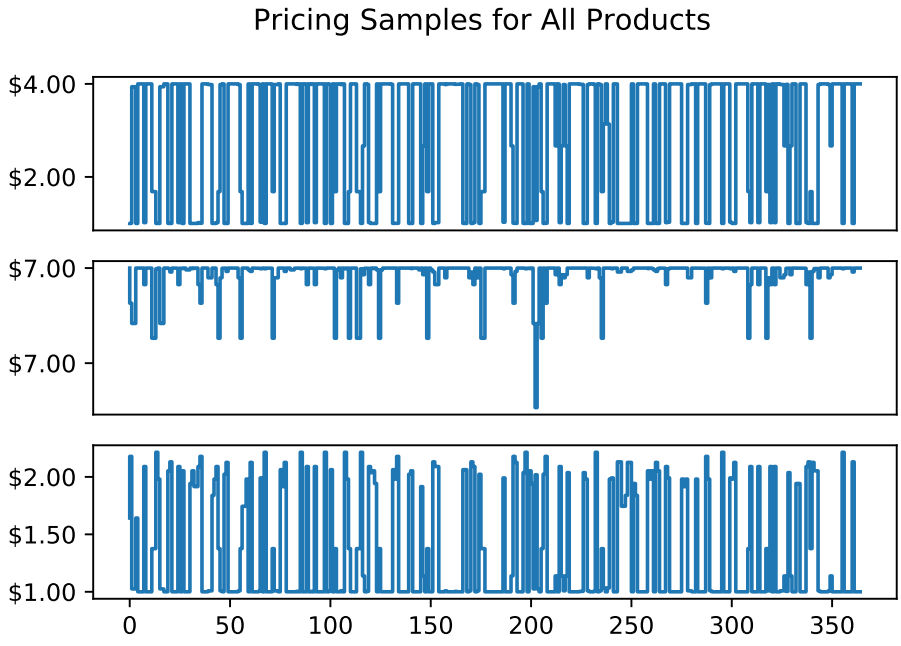

Figure 4.7: Two price trajectories for all 3 products when penalization is $\max_i\{p_i\}$.

but good pricing policy. DDPG, and model-free reinforcement learning in general, offer many advantages. One being that a model does not need to be specified; in the case of MDP, this translates to not needing the transition probabilities. The second one being that it learns the pricing policy through experience, in other words, interacting with customers, optimizing each time a transition occurs instead of waiting to the end of the episode to update parameters.

A possible future direction could look into incorporating price ladders since companies do not price products at any price that is deemed optimal. Indeed, looking at either perishable or non-perishable/reusable products, prices of big ticket items are in the form of either whole numbers, this is seen when searching for hotels or car rentals, and for other small ticket items the prices are usually in the form $xx.99. Another future direction of research is the impact of the learning agents using many engineered features as inputs instead just simply the available stock at any given point time. Features such as competitor's resource price, economic states such as interest rates, and even more granular data. This is possible since data collection technologies are becoming more sophisticated in their data gathering process that enable them to capture more pin-pointed data about consumers. Taking advantage of these technologies will enable to engineer more sophisticated features to feed into the machine learning algorithms. It would be interesting to find the impact of the pricing algorithms taking more information into account.

CHAPTER 5

Concluding Remarks

In the previous chapters we proposed a few tools for the problem of pricing reusable resources to maximize revenue in the face of uncertainty, both theoretical and practical. In this chapter, we summarize our results and propose interesting directions for future work.

5.1 Summary

In Chapter 2 we looked at the setting where the monopolist firm has at its disposal the monotone demand function as a function of price and the uncertainties of the environment are due to the unknown advance reservation and service time distributions. We develop a static pricing policy derived from a convex optimization problem. This policy was shown to generate the optimal revenue rate in the limit as the mean demand rate and initial capacity are scaled without bound. The policy was proven to converge at a rate arbitrarily close to $o(1/\sqrt{n})$. Simulation studies showed the robustness of the derived policy thru subjecting it to different correlation strength and correlation direction between the service time and advance reservation distributions and with different means. The policy was also shown to display the $1/\sqrt{n}$ convergence behavior to the optimal revenue rate. Lastly, even though theoretically we cannot state anything rigorously about the performance of our policy for fixed regimes, our simulation results seem to indicate that the policy started to perform really well really quickly, i.e. it reached performance of at least 80% from optimal revenue for small values of n.

Motivated to develop a dynamic pricing policy instead of a static policy, we incorporated UCB bounds from the MAB literature to guide the development of the pricing policy. The setting is simpler, without advance reservation and a single deterministic service time. We developed a dynamic pricing policy from a linear program that uses the updated UCB demand estimates. These results reaffirm what many previous authors have concluded, which is: In the fluid regime when the system scale is large, a simple heuristic policy becomes optimal. In this particular problem,

with high probability, the seller always selects the best price option as a seller who has no capacity constraints would price, and the regret is due only to the initial testing phase where the seller experiments with all arms. This gives rise to $O(1)$ regret with high probability for a large enough n, that we compute.

Lastly, Chapter 3 we used model-free reinforcement learning to tackle the problem of pricing of reusable resources. Model-free reinforcement learning was necessitated since the transition probabilities are difficult to get traction of and this approach afforded us the ability to bypass the need of specifying a model and learning exclusively through data, in our case, simulated data. We showed that the reinforcement learning algorithm achieved good performance compared to the fluid model pricing policy, all without specifying any model. Another convenience that the reinforcement learning approach afford us is that one can tweak the reward function, or how the agent is rewarded and/or penalized. By changing the extent of the rewards and penalties we were able to derive a better pricing policy that not only generated more revenue, but also increase traffic without significantly increasing blocking. One observation about the pricing policy was that it was not erratic and exhibited some constant behavior with price changes occurring casually. This is something preferred by companies. But it also seemed learned to anticipate how much to keep in reserve for incoming expected arrivals.

5.2 Future Research Directions

The work in this thesis adds to the small growing literature in revenue management with reusable resources. Our work still leaves open many interesting avenues for further research in revenue management systems with reusable resources. One avenue that can be explored in the future is learning while optimizing in the realm of reusable resources. There has been recent work but in the domain of perishable resources. Another possible direction of research is relaxing the assumption of the demand process stationarity. In practice, we observe seasonality and trends and these factors would definitely seem to impact how to price resources accordingly based on the companies' objectives.

Another, but different, stream of research that can be explored is the notion of fairness. There has been displeasure of customers of being charged a higher price for the same product(s). The issue at hand is that of price discrimination. It would be interesting to determine the performance of pricing policies that exhibit some pre-defined measure of fairness in relation to policies that don't exhibit fairness. Finally, on the theoretical front, it would be interesting to explore the performance of these policies when viewed through a different measure, in particular, measuring performance using competitive ratio. Competitive ratio is often used to bound the performance gap between the

proposed policy and the optimal policy under all possible demand realizations.

With the advent of new, fast, and robust artificial intelligent systems, another interesting research direction is how to incorporate contextual information, or learn the important features that affects the companies' bottom-line to make better improved pricing decisions. Our work using model-free reinforcement learning only used the remaining inventory as stateful information. It would be interesting to determine the gain in performance by taking into account much more context than what was considered in this work.

Bibliography

Abramowitz M, Stegun IA (1974) *Handbook of Mathematical Functions* (Dover Publications, Inc., New York).

Adelman D (2006) A simple algebraic approximation to the Erlang loss system. *Operations Research Letters* 36(4):484–491.

Agrawal S, Goyal N (2012) Analysis of thompson sampling for the multi-armed bandit problem. *Foundations and Trends in Machine Learning* 23:1–26.

Altman E, Jiménez T, Koole G (2001) On optimal call admission control in a resource-sharing system. *IEEE Transactions on Communications* 49(9):1659–1668.

Auer P, Cesa-Bianchi N, Fischer P (2002a) Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47(2):235–256.

Auer P, Cesa-Bianchi N, Freund Y (2002b) The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing* 32(1):48–77.

Begen MA, Levi R, Queyranne M (2012) A sampling-based approach to appointment scheduling. *Operations research* 60(3):675–681.

Begen MA, Queyranne M (2011) Appointment scheduling with discrete random durations. *Mathematics of Operations Research* 36(2):240–257.

Bernal A, Shi C (2019) Network revenue management with reusable resources and advanced reservations. *Production and Operations Management,* Working paper.

Besbes O, Maglaras C (2009) Revenue optimization for a make-to-order queue in an uncertain market environment. *Operations Research* 57(6):1438–1450.

Besbes O, Zeevi A (2011) Blind network revenue management. *Operations Research* 60(6):1537–1550.

Billingsley P (1995) *Probability and Measure* (John Wiley & Sons, Inc., New York).

Bitran G, Caldentey R (2003a) An overview of pricing models for revenue management. *M&SOM* 5(3):203–229.

Bitran G, Caldentey R (2003b) An overview of pricing models for revenue management. *Manufacturing & Service Operations Management* 5(3):203–229.

Borgs C, Candogan O, Chayes J, Lobel I, Nazerzadeh H (2014) Optimal multiperiod pricing with service guarantees. *Management Science* 60(7):1792–1811.

Bubeck S, Cesa-Bianchi N (2012) Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning* 5(1):1–122.

Burman DY, Lehoczky JP, Lim Y (1984) Insensitivity of blocking probabilities in a circuit-switching network. *J. Appl. Probab.* 21(4):850–859.

Chen Y, Levi R, Shi C (2017) Revenue management of resuable resources with advanced reservation. *Production and Operations Management* 26(5):836–859.

Chen Y, Shi C (2016) Optimal pricing policy for service systems with reusable resources and forward-looking customers, working paper, University of Michigan.

Chen Y, Shi C (2019) Network revenue management with online inverse batch gradient descent method, working paper, University of Michigan.

Coffman-Jr EG, Jelenkovic P, Poonen B (1999) Reservation probabilities. *Adv. Perf. Anal.* 2:129–158.

de Bruin T, Kober J, Tuyls K, Babuska R (2015) The importance of experience replay database composition in deep reinforcement learning. *Deep Reinforcement Learning Workshop, NIPS*.

den Boer AV (2015) Dynamic pricing and learning: Historical origins, current research, and new directions. *Surveys in operations research and management science* 20(1):1–18.

Erlang AK (1917) Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges. *Elektrotkeknikeren* 13:5–13.

Fan-Orzechowski X, Feinberg EA (2006) Optimality of randomized trunk reservation for a problem with a single constraint. *Adv. Appl. Probab.* 38(1):199–220.

Gallego G, van Ryzin G (1994) Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management Science* 40(8):999–1020.

Gallego G, van Ryzin G (1997) A multiproduct dynamic pricing problem and its applications to network yield management. *Operations Research* 45(1):24–41.

Gans N, Savin SV (2007) Pricing and capacity rationing for rentals with uncertain durations. *Management Science* 53(3):390–407.

Ge D, Wan G, Wang Z, Zhang J (2013) A note on appointment scheduling with piecewise linear cost functions. *Mathematics of Operations Research* 39(4):1244–1251.

Gosavi A, Bandla T (2002) A reinforcement learning approach to a single leg airline revenue management problem with multiple fare classes and overbooking. *IIE Transactions.* 34:729–752.

Gupta D, Denton B (2008) Appointment scheduling in health care: Challenges and opportunities. *IIE transactions* 40(9):800–819.

Hafner R, Riedmiller M (2011) Reinforcement learning in feedback control. *Machine learning* 84(1-2):137–169.

Hunt PJ, Laws CN (1997) Optimization via trunk reservation in single resource loss systems under heavy traffic. *Ann. Appl. Probab.* 7(4):1058–1079.

Jain A, Moinzadeh K, Dumrongsiri A (2015) Priority allocation in a rental model with decreasing demand. *M&SOM* 17(2):236–248.

Kaandorp GC, Koole G (2007) Optimal outpatient appointment scheduling. *Health Care Management Science* 10(3):217–229.

Kaufman JS (1981) Blocking in a shared resources environment. *IEEE Trans. Comm.* 29:1474–1481.

Kelly FP (1991) Effective bandwidths at multi-class queues. *Queueing Systems* 9(1-2):5–16.

Key P (1990) Optimal control and trunk reservation in loss networks. *Probab. Engrg. Informs. Sci.* 4:203–242.

Kong Q, Lee CY, Teo CP, Zheng Z (2013) Scheduling arrivals to a stochastic service delivery system using copositive cones. *Operations research* 61(3):711–726.

Kumar S, Srikant R, Kumar PR (1998) Bounding blocking probabilities and throughput in queueing networks with buffer capacity constraints. *Queueing Systems* 28(1-3):55–77.

Lai TL, Robbins H (1985) Asymptotically efficient adaptive allocation rules. *Bull. Amer. Math. Soc.* 6:4–22.

Lei Y, Jasin S (2016a) Real-time dynamic pricing for revenue management with reusable resources and deterministic service time requirements, working paper, University of Michigan, available at http://dx.doi.org/10.2139/ssrn.2816718.

Lei Y, Jasin S (2016b) Real-time dynamic pricing for revenue management with reusable resources and deterministic service time requirements, working paper, University of Michigan, available at https://ssrn.com/abstract=2816718.

Levi R, Radovanovic A (2010) Provably near-optimal LP-based policies for revenue management in systems with reusable resources. *Operations Research* 58(2):503–507.

Levi R, Shi C (2015) Dynamic allocation problems in loss network systems with advanced reservation, working paper, University of Michigan, available at arXiv:1505.03774.

Lillicrap P, Erez T, Tassa Y, Silver D, Wierstra D (2016) Continuous control with deep reinforcement learning. *International Conference on Learning Representations, ICLR*.

Lin K (2006) Dynamic pricing with real-time demand learning. *European Journal of Operational Research* 174:522–538.

Louth G, Mitzenmacher M, Kelly F (1994) Bounding blocking probabilities and throughput in queueing networks with buffer capacity constraints. *Theoret. Comput. Sci.* 125:45–59.

Lu Y, Radovanovic A (2007) Asymptotic blocking probabilities in loss networks with subexponential demands. *J. Appl. Probab.* 44(4):1088–1102.

Luss H (1977) A model for advanced reservations for intercity visual conferencing services. *Journal of the Operational Research Society* 28(2):275–284.

Maglaras C (2006) Revenue management for a multiclass single-server queue via a fluid model analysis. *Operations Research* 54(5):914–932.

Mak HY, Rong Y, Zhang J (2014) Appointment scheduling with limited distributional information. *Management Science* 61(2):316–334.

Mandelbaum A (1987) Continuous multi-armed bandits and multiparameter processes. *Ann. Probab.* 15(4):1527–1556.

Miller B (1969) A queueing reward system with several customer classes. *Management Science* 16(3):234–245.

Mnih X, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, Riedmiller M (2013) Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* .

Örmeci EL, Burnetas A, Wal Jvd (2001) Admission policies for a two class loss system. *Stoch. Models* 17(4):513–540.

Owens Z (2018) *Revenue Management and Learning in Systems with Reusable Resources*. Ph.D. thesis, Massachusetts Institute of Technology.

Özer Ö, Phillips R (2012) *The Oxford handbook of pricing management* (Oxford University Press).

Papier F, Thonemannm UW (2010) Capacity rationing in stochastic rental systems with advance demand information. *Operations Research* 58(2):274–288.

Peters J, Schaal S (2006) Policy gradient methods for robotics. *International Conference on Intelligent Robots and Systems, IROS*.

Puhalskii AA, Reiman MI (1998) A critically loaded multirate link with trunk reservation. *Queueing Systems* 28(1-3):157–190.

Raju C, Narahari Y, Ravikumar K (2006) Learning dynamic prices in electronic retail markets with customer segmentation. *Annals of Operations Research.* 59–75.

Robbins H (1952) Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.* 58:527–535.

Ross K, Tsang D (1989) The stochastic knapsack problem. *Management Science* 37(7):740–747.

Ross K, Yao D (1990) Monotonicity properties for the stochastic knapsack. *IEEE Trans. Inform. Theory* 36(5):1173–1179.

Ross S (2010) *Introduction to Probability Models* (Academic Press, Burlington, MA).

Savin SV, Cohen MA, Gans N, Katalan Z (2005) Capacity management in rental businesses with two customer bases. *Operations Research* 53(4):617–631.

Sevastyanov BA (1957) An ergodic theorem for markov processes and its application to telephone systems with refusals. *Theory of Probability and its Applications* 2(1):104–112.

Silver D, Lever G, Heess N, Degris T, Wierstra D, Riedmiller M (2014) Deterministic policy gradient algorithms. *International Conference on Machine Learning, ICML.*

Sutton R, Barto A (1998) *Reinforcement learning: an Introduction* (Bradford, Cambridge.).

Sutton R, McAllester D, David A, Singh S, Mansour Y (1999) Policy gradient methods for reinforcement learning with function approximation. *In Advances in Neural Information Processing Systems.* 99:1057–1063.

Talluri K, van G Ryzin (1998) An analysis of bid-price controls for network revenue management. *Management Science* 44(11):1577–1593.

Talluri KT, van Ryzin G (2005) *The theory and practice of revenue management* (Springer, New York.).

Tsaia W, Hung SJ (2009) Dynamic pricing and revenue management process in internet retailing under uncertainty: An integrated real options approach. *Omega* 37:471–481.

Virtamo JT, Aalto S (1991) Stochastic optimization of reservation systems. *European Journal of Operational Research* 51(3):327–337.

Wang LDNH (2019) Dynamic learning and pricing with model misspecification. *Management Science* 65(11):4980–5000.

Whitt W (1985) Blocking when service is required from several facilities simultaneously. *AT&T Tech. J.* 64:1807–1856.

Ye Y, Ren X, Wang J, Xu L, Guo W, Huang W, Tian W (2018) A new approach for resource scheduling with deep reinforcement learning. *Working paper, University of Electronic Science and Technology of China, Chengdu, China* URL http://arxiv.org/abs/1806.08122.

Zachary S (1991) On blocking in loss networks. *Adv. Appl. Probab.* 23(2):355–372.

Appendix

Appendix A

Technical Proofs for Chapter 2

A.1 Proof of Lemma 2.2.1

Proof. Consider the original unperturbed fluid model (2.2) and the perturbed problem (2.3).

$$\begin{aligned} J^D = \quad & \max \quad \lambda^\top p(\lambda) \\ & \text{s.t.} \quad A\lambda \le c, \\ & \qquad \lambda \in \Lambda. \end{aligned} \tag{A.1}$$

The only difference between (2.2) and (2.3) is the RHS of the constraint. Suppose the optimal solution to (2.2) is $\{\hat{\lambda}_i\}$ and consider the point $\{\tilde{\lambda}_i\} = \{(1-\epsilon)\hat{\lambda}_i\}$. Since the constraint is linear and by assumption 2.2(d) the function $r(\lambda) = \lambda p(\lambda)$ has an interior maximizer, $\{\tilde{\lambda}_i\}$ is a feasible solution to (2.3) for a small enough ϵ. Then we have

$$\lambda^{*\top} p(\lambda^*) \ge \tilde{\lambda}^\top p(\tilde{\lambda}) = (1-\epsilon)\hat{\lambda}^\top p((1-\epsilon)\hat{\lambda}). \tag{A.2}$$

Assumption 1(e) implies

$$p((1-\epsilon)\hat{\lambda}) \ge p(\hat{\lambda}).$$

Therefore, (A.2) implies

$$J^D_\epsilon = \lambda^{*\top} p(\lambda^*) \ge (1-\epsilon)\hat{\lambda}^\top p(\hat{\lambda}) \ge (1-\epsilon)J^*.$$

The last inequality holds since the optimal objective value of (A.1) provides an upper bound on the optimal expected revenue rate since the capacity constraint is enforced on expectation rather than holding on every sample path. □

A.2 Proof of Lemma 2.2.6

Proof.

$$
\begin{aligned}
\Lambda_Z(t) &= \rho(1-\mathbb{P}(S \le t, L \le t-S)) \\
&= \rho\left(1-\int_{s=0}^{t} F_{L|S=s}(t-s)f_S(s)\right) \\
&\ge \rho\left(1-\int_{s=0}^{t} F_{L|S=s}(t-s)f_S(s)-\int_{s=t}^{\infty} F_{L|S=s}(t)f_S(s)\right) \\
&\ge \rho\left(1-\int_{s=0}^{t} F_{L|S=s}(t)f_S(s)-\int_{s=t}^{\infty} F_{L|S=s}(t)f_S(s)\right) \\
&= \rho\left(1-\int_{s=0}^{\infty} F_{L|S=s}(t)f_S(s)\right) \\
&= \rho\left(\int_{s=0}^{\infty} f_S(s)-\int_{s=0}^{\infty} F_{L|S=s}(t)f_S(s)\right) \\
&= \rho\left(\int_{s=0}^{\infty} \bar{F}_{L|S=s}(t)f_S(s)\right) \\
&= \rho\int_{s=0}^{\infty} \mathbb{P}(L \ge t|S=s)f_S(s) \\
&= \rho\bar{F}_L(t) \\
&= \Lambda_Y(t).
\end{aligned}
$$

□

A.3 Proof of Lemma 2.2.7

Proof. We break up the time interval to analyze two cases.

Case $t \le \alpha$: We have to take the derivative of the mean arrival function. Since the departure process, $Z(t)$, is a NHPP, the mean at t is

$$
\begin{aligned}
\mathbb{E}(Z(t)) = \rho &\int_{u=\alpha-t}^{\infty} \mathbb{P}(u-\alpha \le L \le u+t-\alpha, \quad S \le u+t-L) \\
&+ \int_{u=\alpha}^{\infty} \mathbb{P}(L \le u-\alpha, \quad u-L \le S \le u+t-L),
\end{aligned}
$$

where the first term are the customers who start service between $[-\alpha, -(\alpha - t)]$ and end service in the interval $[0, t]$ and the second term are the customers who start service between $(-\infty, -\alpha]$ and end service in the interval $[0, t]$. Note that we cannot take into account the customers who start service between $[-(\alpha - t), 0]$ since the service time is at least α, i.e., they will not end service in $[0, t]$.

Therefore, taking the derivative of $\mathbb{E}(Z(t))$ will provide the rate function of $Z(t)$.

$$
\begin{aligned}
\Lambda_Z(t) &= \frac{d}{dt}\mathbb{E}(Z(t)) \\
&= \rho\frac{d}{dt}\left(\int_{u=\alpha}^{\infty} \mathbb{P}(L \le u-\alpha, \quad u-L \le S \le u+t-L)\right. \\
&\qquad\qquad \left. + \int_{u=\alpha-t}^{\infty} \mathbb{P}(u-\alpha \le L \le u+t-\alpha, \quad S \le u+t-L)\right) \\
&= \rho\frac{d}{dt}\left(\int_{u=\alpha}^{\infty}\int_{d=0}^{u-\alpha} \mathbb{P}(u-d \le S \le u+t-d | L=d) f_L(d)\right. \\
&\qquad\qquad \left. + \int_{u=\alpha-t}^{\infty}\int_{d=u-\alpha}^{u+t-\alpha} \mathbb{P}(S \le u+t-d | L=d) f_L(d)\right) \\
&= \rho\frac{d}{dt}\left(\int_{u=\alpha}^{\infty}\int_{d=0}^{u-\alpha} \left(F_{S|L=d}(u+t-d) - F_{S|L=d}(u-d)\right) f_L(d)\right. \\
&\qquad\qquad \left. + \int_{u=\alpha-t}^{\infty}\int_{d=u-\alpha}^{u+t-\alpha} F_{S|L=d}(u+t-d) f_L(d)\right) \\
&= \rho\left(\int_{u=\alpha}^{\infty}\int_{d=0}^{u-\alpha} \frac{d}{dt}(F_{S|L=d}(u+t-d) - F_{S|L=d}(u-d)) f_L(d)\right. \\
&\qquad\qquad \left. + \int_{u=\alpha-t}^{\infty}\int_{d=u-\alpha}^{u+t-\alpha} \frac{d}{dt} F_{S|L=d}(u+t-d) f_L(d)\right) \\
&= \rho\left(\int_{u=\alpha}^{\infty}\int_{d=0}^{u-\alpha} f_{S|L=d}(u+t-d) f_L(d) + \int_{u=\alpha-t}^{\infty}\int_{d=u-\alpha}^{u+t-\alpha} f_{S|L=d}(u+t-d) f_L(d)\right) \\
&= \rho\left(\int_{u=\alpha}^{\infty}\int_{d=0}^{u-\alpha} f_{S,L}(u+t-d, d) + \int_{u=\alpha-t}^{\infty}\int_{d=u-\alpha}^{u+t-\alpha} f_{S,L}(u+t-d, d)\right).
\end{aligned}
$$

Now, let $x = u - d$ and $y = d$. Then, the set $\alpha \le u \le \infty$, $0 \le d \le u - \alpha$ in the (u, d)-plane maps to $\alpha \le x \le \infty$, $0 \le y \le \infty$ in the (x, y)-plane. The set $\alpha - t \le u \le \infty$, $u - \alpha \le d \le u + t - \alpha$ in the (u, d)-plane

maps to $\alpha - t \leq x \leq \alpha$, $0 \leq y \leq \infty$ in the (x,y)-plane. Therefore,

$$\begin{aligned}\Lambda_Z(t) &= \rho\left(\int_{x=\alpha}^{\infty}\int_{y=0}^{\infty} f_{S,L}(x+t,y) + \int_{x=\alpha-t}^{\alpha}\int_{y=0}^{\infty} f_{S,L}(x+t,y)\right), \quad \text{(Change of variables)}\\ &= \rho\left(\mathbb{P}(S \geq \alpha+t) + \mathbb{P}(S \leq \alpha+t)\right)\\ &= \rho\end{aligned}$$

Case $t > \alpha$:

$$\begin{aligned}\Lambda_Z(t) &= \frac{d}{dt}\mathbb{E}(Z(t))\\ &= \rho\frac{d}{dt}\left(\int_{u=0}^{\infty} \mathbb{P}(L \leq u-\alpha, \quad u-L \leq S \leq u+t-L) + \int_{u=0}^{\infty} \mathbb{P}(u \leq L \leq u+t-\alpha, \quad S \leq u+t-L)\right),\end{aligned}$$

where the first term are the customers who start service in the interval $[-\infty, 0]$ and the second term are the customers who start service in the interval $[0, t-\alpha]$. Therefore, using Leibniz integral rule and change of variables as above, then

$$\Lambda_Z(t) = \rho(1 - \mathbb{P}(S \leq t, L \leq t - S)).$$

But observe that $S = S_{old} + \alpha$, S_{old} is the previous service time random variable with support $\mathbb{R}_+$. Therefore,

$$\Lambda_Z(t) = \rho(1 - \mathbb{P}(S_{old} \leq t-\alpha, L \leq t-\alpha-S_{old})).$$

In other words, we get an α-shifted version of the rate function where the support is the non-negative real line. The change in differentiation is justified by Lemma 2.2.5.

Observe that by Lemma 2.2.7, the new $\Lambda_Z(t)$ is a shifted version of $\Lambda_{Z_{old}}(t)$, where $\Lambda_{Z_{old}}(t)$ is derived using the service distribution S_{old} with support $\mathbb{R}_+$. Then by Lemma 2.2.6, $\Lambda_Z(t) > \Lambda_Y(t)$ for all $t > 0$. Finally, observe that the event $\{S \leq t, L \leq t-S\}^C = \{L \geq t, S < \infty\} \cup \{L \leq t, S \geq t-L\}$, where A^C is the complement of the event A. The event $\{S \leq t, L \leq t-S\}^C$ can also be written as, but will not use here, $\{S \geq t, L < \infty\} \cup \{S \leq t, L \geq t-S\}$ Therefore,

$$\begin{aligned}\Lambda_Z(t) &= \rho(1 - \mathbb{P}(S \leq t, L \leq t-S))\\ &= \rho\mathbb{P}(\{S \leq t, L \leq t-S\}^C)\\ &= \rho\mathbb{P}(L \geq t, S < \infty) + \rho\mathbb{P}(L \leq t, S \geq t-L)\\ &= \rho\bar{F}_L(t) + \rho\mathbb{P}(L \leq t, S \geq t-L).\end{aligned}$$

This is yet another way of showing that the departure rate $\Lambda_Z(t)$ is at least as great as the pre-arrival rate, $\Lambda_Y(t)$. Additionally, this shows that $\Lambda_Z(t)$ will be at least as big as $\Lambda_Y(t)$ even if the the support

of the advance reservation is $[\psi, \infty]$ for any $\psi > 0$. Integrating $\Lambda_Z(t)$ over $[0, \infty)]$ provides

$$\begin{aligned}
\int_{t=0}^{\infty} \Lambda_Z(t) &= \int_{t=0}^{\infty} \rho \bar{F}_L(t) + \int_{t=0}^{\infty} \rho \mathbb{P}(L \le t, S \ge t - L) \\
&= \rho \mathbb{E}(L) + \rho \int_{t=0}^{\infty} \int_{d=0}^{t} \mathbb{P}(S \ge t - L | L = d) f_L(d) \\
&= \rho \mathbb{E}(L) + \rho \int_{d=0}^{\infty} \int_{t=d}^{\infty} \mathbb{P}(S \ge t - d | L = d) f_L(d) \\
&= \rho \mathbb{E}(L) + \rho \int_{d=0}^{\infty} f_L(d) \int_{t=d}^{\infty} \mathbb{P}(S \ge t - d | L = d) \\
&= \rho \mathbb{E}(L) + \rho \int_{d=0}^{\infty} f_L(d) \mathbb{E}(S | L = d) \\
&= \rho \mathbb{E}(L) + \rho \mathbb{E}(S).
\end{aligned}$$

In other words, if a customer was to "call", not arrive, in the steady-state and counted all customers who were already using the resource and counted all pre-arrivals, i.e. customers who have reserved their spot for future use, on average the customer would count $\rho(\mathbb{E}(L) + \mathbb{E}(S))$ on average. □

A.4 Proof of Lemma 2.2.8

Proof. Recall that $\Lambda_Z(t)$ approaches zero as $t \to \infty$ and $\int_{t=0}^{\infty} \Lambda_Z(t) = \rho(\mathbb{E}(S) + \mathbb{E}(L))$ by Lemma 2.2.7. Therefore, there exists a time t_1 really large, such that the total number of departures and pre-arrivals after t_1 is small, i.e., $\int_{t=t_1}^{\infty} \Lambda_Z(t)$ is small. Intuitively this makes sense because with respect to the time of the customer arrival, we wouldn't expect a large number of customers reserving too far into the future nor many customers prior to the customer arrival to have very large service times. Let us choose a sufficiently large time $t_1 \in \mathbb{N}_+$ such that $\int_{t=t_1}^{\infty} \Lambda_Z(t) = \rho \int_{t=t^*}^{\infty} (1 - \mathbb{P}(S \le t, L \le t - S)) \le \frac{\nu c}{2}$. Note that t_1 will remain unchanged when the capacity and arrival rates are scaled by n. This will become important later on when we consider a sequence of problems when the capacity and arrival rates are scaled. Additionally, we can also find a t_2 such that

$$\beta = \int_{t=t_2}^{\infty} \Lambda_Y(t) = \rho \int_{t=t_2}^{\infty} \bar{F}_L(t) \le \frac{\nu c}{2},$$

which again will remain unchanged when the capacity and arrival rates are scaled by n. Choose

$t^* = \max\{t_1, t_2\}$. Therefore,

$$\begin{aligned}\mathbb{P}(\text{Customer blocked in } [t^*, \infty)) &= \mathbb{P}(\max_{t\in[t^*,\infty)} \{Y(t) - Z(t)\} \geq c) \\ &\leq \mathbb{P}(Y^* + W^* \geq c) \\ &\leq \sum_{i=c}^{\infty} \frac{e^{-\nu c}(\nu c)^i}{i!},\end{aligned} \tag{A.3}$$

The processes $Y(t)$ and $Z(t)$ are dependent on $[t^*, \infty]$ as the number of customers who depart in this interval depends on how many arrived for service in this interval. Note that the maximum number of customers a person would "see" in the steady-state system cannot exceed $W^* + Y^*$, where W^* is the number of customers who reserve service before time t^* but depart after t^*, which has mean less than or equal to the mean of total departures in the interval $[t^*, \infty)$, i.e. $\mathbb{E}[W^*] = \mathbb{E}[Z_\infty]$. Y^* are the customers who pre-arrive after time t^* and will obviously depart after time t^*. Additionally, W^* and Y^* are independent. Since $\Lambda_Z(t) > \Lambda_Y(t)$, Y^* is a poisson random variable with mean less than $\nu c/2$ and and W^* is also a poisson random variable with mean less than $\nu c/2$, this implies that

$$W^* + Y^* \sim Poisson(\nu c).$$

Now, let us define $Z_s^d(t)$ and $Y_s^d(t)$ to be the number of customers who will depart in the interval $[d, d+t]$ and the number of customers who will start service in the interval $[d, d+t]$, $t \in [0, s]$, respectively. Now, X_s^d is a RV which represents the number of customers who start service before time d and depart after time $d+s$, which is a poisson RV. Then since $\mathbb{P}_s^d(B) \leq \mathbb{P}_I(B)$ for any interval I that contains $[d, d+s]$, we will choose the contiguous intervals $\{[i, i+1]\}$ for $i \in \{\lfloor d \rfloor, \ldots, t^* - 1\}$ and the interval $[t^*, \infty)$ that cover the interval $[d, d+s]$. Therefore, for $i \in \{0, \ldots, t^* - 1\}$, let us define $Z_i'(t)$ to be the number of customers who depart in the interval $[i, i+t]$, $t \in [0, 1]$, $Y_i'(t)$ be the number of customers who arrive in the interval $[i, i+t]$, $t \in [0, 1]$, and X_i' to be the number of customers who have started service before time i and depart after $(i+1)$.

$$\begin{aligned}\mathbb{P}_s^d(B) &= \mathbb{P}(X_s^d + Z_s^d(s) + \max_{t\in[0,s]} \{Y_s^d(t) - Z_s^d(t)\} \geq c) \\ &\leq \mathbb{P}\left(\max\left\{\max_{i=\lfloor d \rfloor,\ldots,t^*-1} \{X_i + Z_i'(1) + \max_{t\in[0,1]} \{Y_i'(t) - Z_i'(t)\}\}, \quad Z^* + \max_{t\in[t^*,\infty)} \{Y(t) - Z(t)\}\right\} \geq c\right) \\ &\leq \sum_{i=\lfloor d \rfloor}^{t^*-1} \mathbb{P}(\{X_i + Z_i'(1) + \max_{t\in[0,1]} \{Y_i'(t) - Z_i'(t)\}\} \geq c) + \mathbb{P}(Z^* + \max_{t\in[t^*,\infty)} \{Y(t) - Z(t)\} \geq c) \\ &\leq \sum_{i=0}^{t^*-1} \mathbb{P}(\{X_i + Z_i'(1) + \max_{t\in[0,1]} \{Y_i'(t) - Z_i'(t)\}\} \geq c) + \sum_{i=c}^{\infty} \frac{e^{-\nu c}(\nu c)^i}{i!}.\end{aligned}$$

Note that for any $i \in \{0, 1, \ldots, t^* - 1\}$, the processes $Y_i'(t)$ and $Z_i'(t)$ are independent NHPP in the interval $[i, i+1]$ with rates $\Lambda_{Y_i'}(t) = \Lambda_Y(i+t)$ and $\Lambda_{Z_i'}(t) = \Lambda_Z(i+t)$, respectively, and independent of X_i Ross (2010). Note that Lemma 2.2.6 and Lemma 2.2.7 imply $\Lambda_{Y_i'}(t) < \Lambda_{Z_i'}(t)$ for $t \in [0, 1]$ and $i \in \{1, \ldots, t^* - 1\}$ and $\Lambda_{Y_i'}(t) < \Lambda_{Z_i'}(t)$ for $t \in (0, 1]$, but $\Lambda_{Y_0'}(t) = \Lambda_{Z_0'}(t)$ at $t = 0$.

□

A.5 Proof of Lemma 2.2.9

Proof. Recall that X_i represents the number of customers who are in service by time i and are still in service by time $i+1$. Also, recall that $Z_i'(1)$ represent the number of customers who depart the system in the interval $[i, i+1]$. Then, since X_i and $Z_i'(1)$ are independent Poisson random variables Ross (2010), $X_i + Z_i'(1)$ is also a Poisson random variable that represents the number of customers who depart the system after time i. Note, that any pre-arrival in the interval $[i, i+1]$ will not leave the system in the same interval since $\alpha = 1$, i.e. customers will use the resource for at least one time unit. By Ross (2010)

$$\begin{aligned}
\mathbb{E}(X_i + Z_i'(1)) &= \rho \int_{u=0}^{\infty} \mathbb{P}(L \le i+u, S \ge i+u-L) \\
&= \rho \int_{u=0}^{\infty} \int_{d=0}^{i+u} \mathbb{P}(L \le i+u, S \ge i+u-L) \\
&= \rho \int_{u=0}^{\infty} \int_{d=0}^{i+u} \bar{F}_{S|L=d}(i+u-d) f_L(d) \\
&= \rho \int_{d=0}^{i} \int_{u=0}^{\infty} \bar{F}_{S|L=d}(i+u-d) f_L(d) + \rho \int_{d=i}^{\infty} \int_{u=d-i}^{\infty} \bar{F}_{S|L=d}(i+u-d) f_L(d) \\
&= \rho \int_{d=0}^{i} \int_{u=0}^{\infty} \bar{F}_{S|L=d}(i+u-d) f_L(d) + \rho \int_{d=i}^{\infty} f_L(d) \mathbb{E}(S|L=d).
\end{aligned}$$

Therefore, if $i = 0$, then the expected number of customers who are in service by time 0 is $\rho E(S)$. Also, one can easily see that $\mathbb{E}(X_i + Z_i'(1)) \ge \mathbb{E}(X_{i+1} + Z_{i+1}'(1))$, which implies $\mathbb{E}(X_i + Z_i'(1)) \le \rho \mathbb{E}(S)$ for all $i \in \{0, \dots, t^* - 1\}$. Therefore, let $\bar{Z}(t)$ be a homogeneous Poisson process with rate $\rho \mathbb{E}(S)$ independent of $Y_i'(t)$ for all i. For any $i \in \{1, \dots, t^* - 1\}$

$$\mathbb{P}(X_i + \max_{t \in [0,1]} \{\bar{Y}_i(t) + \bar{Z}_i(1) - \bar{Z}_i(t)\} \ge c) \ge \mathbb{P}(X_i + \max_{t \in [0,1]} \{Y_i'(t) + Z_i'(1) - Z_i'(t)\} \ge c),$$

where $\bar{Y}_i(t)$ is a homogeneous Poisson process with rate of $\Lambda_{Y_i'}(0) = \Lambda_Y(i)$ and is always at least as big as the rate of $Y_i'(t)$ over the interval $[0,1]$ and $\bar{Z}_i(1) - \bar{Z}_i(t)$, defined as in Chen et al. (2017), is a mirrored Poisson process with rate $\Lambda_{Z_i'}(0) = \Lambda_Z(i)$ which is always at least as big as the rate of $Z_i'(1) - Z_i'(t)$. Recall, that Lemma 2.2.6 and 2.2.7 imply $\Lambda_{\bar{Z}_i}(i) > \Lambda_{\bar{Y}_i}(i)$. Therefore, there exists $\theta_i \in (0,1)$ such that

$$\theta_i \Lambda_{\bar{Z}_i}(i) = \Lambda_{\bar{Y}_i}(i),$$
$$\theta_i = \frac{\Lambda_Y(i)}{\Lambda_Z(i)}.$$

As will be shown in Lemma 2.2.10, Lemmas **4** and **5**, and Proposition **2** of Chen et al. (2017)

implies the following

$$\mathbb{P}(X_i + \max_{t\in[0,1]}\{\bar{Y}_i(t) + \bar{Z}_i(1) - \bar{Z}_i(t)\} \geq c) \leq \frac{1}{\rho} + \left(\frac{e^{\delta_i}}{(1+\delta_i)^{(1+\delta_i)}}\right)^{\rho},$$

where $\delta_i = \left(\frac{\epsilon}{1-\epsilon} - \frac{\log(1+\rho)}{\rho \log \theta_i^{-1}}\right)$ and $\rho = \min\{(1-\epsilon)c, \lambda\bar{F}(p^*)\mu_s\}$. The conclusion follows.

□

A.6 Proof of Lemma 2.2.10

Proof. Let $i \in \{1, \ldots, t^* - 1\}$. By Lemma 2.2.9, there exists $\theta_i \in (0,1)$ such that

$$\theta_i \Lambda_{\bar{Z}_i}(i) = \Lambda_{\bar{Y}_i}(i).$$

Define $T = \max_{t\in[0,1]}\{\bar{Y}_i(t) + \bar{Z}_i(1) - \bar{Z}_i(t)\}$. Consider the merged Poisson process of two independent Poisson processes $\bar{Y}_i(t)$ and $\bar{Z}_i(1) - \bar{Z}_i(t)$. Let $\mathcal{N} = \bar{Y}_i(1) + \bar{Z}_i(1)$ denote the total number of occurrences (pre-arrivals and departures) over $[0,1]$ of the two independent Poisson counting processes. Additionally, let X_i be a poisson random variable independent of T. Let us associate a +1 when the jump occurs from $\bar{Y}_i$ and a -1 when a jump occurs from $\bar{Z}_i$. Conditioning on $\mathcal{N} = n$, we induce a random walk with the probability of a downward jump being strictly greater than an upward jump. In particular, each of these points has independent probability $p = \frac{\theta_i}{1+\theta_i} < \frac{1}{2}$ to be from the $\bar{Y}_i$ process and probability $1 - p_i$ from process $\bar{Z}_i$. Then (*see* Lemma **4** in Chen et al. (2017))

$$T_n = (T|\mathcal{N} = n) = G_n + M_n \textit{ almost surely},$$

where M_n denote the maximum level attained by the random walk of length n, and G_n denote the overall number of down-steps taken by the random walk. Observe that X_i is independent of T_n for all n. Then,

$$\begin{aligned}
\mathbb{P}(X_i + T_n \geq c) &= \mathbb{P}(X_i + G_n + M_n \geq c) \\
&= \mathbb{P}(X_i + G_n + M_n \geq c \cap M_n \geq -\frac{\log\rho}{\log\theta_i}) + \mathbb{P}(X_i + G_n + M_n \geq c \cap M_n < -\frac{\log\rho}{\log\theta_i}) \\
&\leq \mathbb{P}(M_n \geq -\frac{\log\rho}{\log\theta_i}) + \mathbb{P}(X_i + G_n \geq c + \frac{\log\rho}{\log\theta_i}) \\
&\leq \mathbb{P}(M_\infty \geq -\frac{\log\rho}{\log\theta_i}) + \mathbb{P}(X_i + G_n \geq c + \frac{\log\rho}{\log\theta_i}) \\
&\leq \frac{1}{\rho} + \mathbb{P}(X_i + G_n \geq c + \frac{\log\rho}{\log\theta_i}),
\end{aligned}$$

where M_∞ is the maximum number of the random walk when the random walk has taken an infinite number of steps. The last and second-to-last inequality is from Lemma **5** of Chen et al. (2017).

Therefore,

$$\begin{aligned}
\mathbb{P}(X_i + \max_{t\in[0,1]}\{\bar{Y}_i(t) + \bar{Z}_i(1) - \bar{Z}_i(t)\} \geq c) &= \mathbb{P}(X_i + T \geq c) \\
&= \sum_{n=0}^{\infty} \mathbb{P}(X_i + T_n \geq c)\mathbb{P}(\mathcal{N} = n) \\
&\leq \frac{1}{\rho} + \sum_{n=0}^{\infty} \mathbb{P}\left(X_i + G_n \geq c + \frac{\log\rho}{\log\theta_i}\right)\mathbb{P}(\mathcal{N} = n) \\
(X_i \text{ is independent of } G_n \text{ for all } n) &= \frac{1}{\rho} + \mathbb{P}\left(X_i + Poi(\Lambda_{\bar{Z}_i}) \geq c + \frac{\log\rho}{\log\theta_i}\right) \\
&\leq \frac{1}{\rho} + \mathbb{P}\left(Poi(\rho) \geq c + \frac{\log\rho}{\log\theta_i}\right) \\
&= \frac{1}{\rho} + \mathbb{P}\left(Poi(\rho) \geq c - \frac{\log\rho}{\log\theta_i^{-1}}\right) \\
&\leq \frac{1}{\rho} + \mathbb{P}\left(Poi(\rho) \geq \frac{\rho}{1-\epsilon} - \frac{\log\rho}{\log\theta_i^{-1}}\right) \\
&\leq \frac{1}{\rho} + \mathbb{P}\left(Poi(\rho) \geq \frac{\rho}{1-\epsilon} - \frac{\log(1+\rho)}{\log\theta_i^{-1}}\right).
\end{aligned}$$

The second inequality follows since G_n is the total number of down steps when we conditioned on $\mathcal{N} = n$, therefore, it is equal to the total number of down steps in $[0,1]$ when unconditioned, which is a Poisson random variable with rate $\Lambda_{\bar{Z}_i}$ and independent of X_i. From Lemma 2.2.9, $X_i + Poi(\Lambda_{\bar{Z}_i})$ is a Poisson random variable with rate $\mathbb{E}(X_i + Poi(\Lambda_{\bar{Z}_i})) \leq \lambda\mathbb{E}(S) = \rho$. The third inequality is from the capacity constraint in (2.10) that $\rho \leq (1-\epsilon)c$. Chen et al. (2017) showed that

$$\mathbb{P}\left(Poi(\rho) \geq \frac{\rho}{1-\epsilon} - \frac{\log(1+\rho)}{\log\theta_i^{-1}}\right) \leq \left(\frac{e^{\delta_i}}{(1+\delta_i)^{(1+\delta_i)}}\right)^{\rho}.$$

The conclusion follows. □

A.7 Proof of Lemma 2.2.11

Proof. Since $\nu < \min\left\{\frac{1}{c}, \frac{1}{e}\right\} < 1$, by (see 6.5.34 of Abramowitz and Stegun (1974)),

$$\lim_{c\to\infty} \sum_{j=c}^{\infty} \frac{e^{-\nu c}(\nu c)^j}{j!} \to 0. \tag{A.4}$$

This implies

$$\lim_{n\to\infty} \sum_{j=c^{(n)}}^{\infty} \frac{e^{-\nu c^{(n)}}(\nu c^{(n)})^j}{j!} \to 0. \tag{A.5}$$

Next, we characterize the convergence rate. As n is sufficiently large, we make use of Stirling's approximation for the denominator, i.e.,

$$\begin{aligned}
\sum_{j=c^{(n)}}^{\infty} \frac{e^{-\nu c^{(n)}}(\nu c^{(n)})^j}{j!} &\approx \sum_{j=nc}^{\infty} \frac{e^{-\nu nc}(\nu nc)^j}{\sqrt{2\pi j}\left(\frac{j}{e}\right)^j} \\
&= \frac{e^{-\nu nc}}{\sqrt{2\pi}} \sum_{j=nc}^{\infty} \frac{(\nu nc)^j}{\sqrt{j}\left(\frac{j}{e}\right)^j} \\
&\leq \frac{e^{-\nu nc}}{\sqrt{2\pi nc}} \sum_{j=nc}^{\infty} \frac{e^j(\nu nc)^j}{j^j} \\
&\leq \frac{e^{-\nu nc}}{\sqrt{2\pi nc}} \sum_{j=nc}^{\infty} \frac{e^j(\nu nc)^j}{(nc)^j} \\
&\leq \frac{e^{-\nu nc}}{\sqrt{2\pi nc}} \sum_{j=nc}^{\infty} (e\nu)^j \\
&= \frac{e^{-\nu nc}}{\sqrt{2\pi nc}} l,
\end{aligned}$$

where $l = \sum_{j=0}^{\infty} (e\nu)^j$ is finite since $\nu e < 1$, so the summation in the last inequality is a geometric series and does not depend on n. Therefore, convergence rate is $o(e^{-n})$ as $n \to \infty$. □

A.8 Proof of Lemma 2.2.12

Proof. By Lemma 2.2.9

$$\sum_{i=1}^{t^*-1} \mathbb{P}(\{X_i + \max_{t\in[0,1]} \{Y_i'(t) + Z_i'(1) - Z_i'(t)\}\} \geq c^{(n)}) \leq \frac{t^*-1}{\rho^{(n)}} + \sum_{i=1}^{t^*-1} \left(\frac{e^{\delta_i^{(n)}}}{(1+\delta_i^{(n)})^{(1+\delta_i^{(n)})}} \right)^{\rho^{(n)}}.$$

Following Equation 2.5, we have

$$\rho^{(n)} = \min\left\{(1-\epsilon^{(n)})c^{(n)}, \lambda^{(n)}\mu\right\} \geq \min\left\{(1-\epsilon)c^{(n)}, \lambda^{(n)}\mu\right\} \geq n\rho.$$

Now, we can apply the results of Chen et al. (2017), which showed

$$\left(\frac{e^{\delta_i^{(n)}}}{(1+\delta_i^{(n)})^{(1+\delta_i^{(n)})}}\right)^{\rho^{(n)}} = o\left(\frac{1}{n}\right).$$

Therefore,

$$\begin{aligned}\sum_{i=1}^{t^*-1}\mathbb{P}(\{X_i+\max_{t\in[0,1]}\{Y_i'(t)+Z_i'(1)-Z_i'(t)\}\geq c^{(n)}) &\leq \frac{t^*-1}{\rho^{(n)}}+\sum_{i=1}^{t^*-1}\left(\frac{e^{\delta_i^{(n)}}}{(1+\delta_i^{(n)})^{(1+\delta_i^{(n)})}}\right)^{\rho^{(n)}}\\ &\leq \frac{t^*-1}{n\rho}+o\left(\frac{1}{n}\right).\end{aligned}$$

□

A.9 Proof of Lemma 2.2.13

Proof. Conditioning on the occurrence time of the first event, W_1, i.e. an arrival or departure, for any given $n \geq 1$

$$\begin{aligned}&\mathbb{P}(X_0+Z_0'(1)+\max_{t\in[0,1]}\{Y_0'(t)-Z_0'(t)\}\geq c^{(n)})\\ &\quad=\int_{t=0}^{\infty}\mathbb{P}(X_0+Z_0'(1)+\max_{t\in[0,1]}\{Y_0'(t)-Z_0'(t)\}\geq c|W_1=t)f_{W_1}(t)\\ &\quad=\int_{t=0}^{1}\mathbb{P}(X_0+Z_0'(1)+\max_{t\in[0,1]}\{Y_0'(t)-Z_0'(t)\}\geq c^{(n)}|W_1=t)f_{W_1}(t)\\ &\qquad+\int_{t=1}^{\infty}\mathbb{P}(X_0+Z_0'(1)+\max_{t\in[0,1]}\{Y_0'(t)-Z_0'(t)\}\geq c^{(n)}|W_1=t)f_{W_1}(t)\\ &\quad\leq\int_{t=0}^{1}f_{W_1}(t)+\mathbb{P}(X_0+Z_0'(1)\geq c^{(n)}).\end{aligned}$$

To find the density of W_1, we compute the infinitesimal probability of the event {First arrival $\in [t, t+h)$}. Since the NHPP $Y(t)$ and $Z(t)$ are independent on $[0,1]$, $N(t)$ is a NHPP with rate $\Lambda(t) = \Lambda_Y(t) + \Lambda_Z(t)$. Note that $\Lambda(0) = 2\rho^{(n)}$. The next lemma will show that the density of W_1 is

$$f_{W_1}(t) = e^{-\int_{x=0}^{t}\Lambda(x)}\Lambda(t).$$

Therefore,

$$\begin{aligned}
\mathbb{P}(X_0 + Z_0'(1) + \max_{t\in[0,1]}\{Y_0'(t) - Z_0'(t)\} \geq c^{(n)}) &\leq \int_{t=0}^{1} f_{W_1}(t) + \mathbb{P}(X_0 + Z_0'(1) \geq c^{(n)}) \\
&= \int_{t=0}^{1} \Lambda(t) e^{-\int_{x=0}^{t}\Lambda(x)} + \mathbb{P}(X_0 + Z_0'(1) \geq c^{(n)}) \\
&\leq \Lambda(0) \int_{t=0}^{1} e^{-\int_{x=0}^{t}\Lambda(x)} + \mathbb{P}(X_0 + Z_0'(1) \geq c^{(n)}) \\
&\leq \Lambda(0)e^{-\Lambda(0)} + \mathbb{P}(X_0 + Z_0'(1) \geq c^{(n)}) \\
&= \frac{2\rho^{(n)}}{e^{2\rho^{(n)}}} + \mathbb{P}(X_0 + Z_0'(1) \geq c^{(n)}) \\
&\leq \frac{2(1-\epsilon^{(n)})c^{(n)}}{e^{2n\rho}} + \mathbb{P}(Poi(\rho^{(n)}) \geq c^{(n)}) \\
&\leq \frac{2nc}{e^{2n\rho}} + \mathbb{P}(Poi(\rho^{(n)}) \geq c^{(n)}) \\
&= \mathbb{P}(Poi(\rho^{(n)}) \geq c^{(n)}) + \frac{2nc}{e^{2n\rho}} \\
&\leq \mathbb{P}\left(Poi(\rho^{(n)}) \geq \frac{\rho^{(n)}}{1-\epsilon^{(n)}}\right) + \frac{2nc}{e^{2n\rho}} \\
&\leq \mathbb{P}\left(Poi(\rho^{(n)}) \geq \frac{\rho^{(n)}}{1-\epsilon^{(n)}} - \frac{\log(1+\rho^{(n)})}{\log\theta^{-1}}\right) + \frac{2nc}{e^{2n\rho}}.
\end{aligned}$$

The second and third inequality is since $\Lambda(t)$ is a decreasing function with $\Lambda(0)$ being its maximum value. The fourth inequality is from (2.2.9) since $\mathbb{E}(Poi(\rho^{(n)})) \geq \mathbb{E}(X_0 + Z_0'(1))$, $\rho^{(n)} \leq (1-\epsilon^{(n)})c^{(n)}$ due to the optimization constraint, and $\rho^{(n)} \geq n\rho$. The last inequality is because $\dfrac{\log(1+\rho^{(n)})}{\log\theta^{-1}} \geq 0$, for any $\theta \in (0,1)$, and therefore,

$$\left\{Poi(\rho^{(n)}) \geq \frac{\rho^{(n)}}{1-\epsilon^{(n)}}\right\} \subset \left\{Poi(\rho^{(n)}) \geq \frac{\rho^{(n)}}{1-\epsilon^{(n)}} - \frac{\log(1+\rho^{(n)})}{\log\theta^{-1}}\right\}.$$

We have from Chen et al. (2017) that

$$\mathbb{P}\left(Poi(\rho^{(n)}) \geq \frac{\rho^{(n)}}{1-\epsilon^{(n)}} - \frac{\log(1+\rho^{(n)})}{\log\theta^{-1}}\right) \leq \left(\frac{e^{\delta^{(n)}}}{(1+\delta^{(n)})^{(1+\delta^{(n)})}}\right)^{\rho^{(n)}},$$

and $\left(\frac{e^{\delta^{(n)}}}{(1+\delta^{(n)})^{(1+\delta^{(n)})}}\right)^{\rho^{(n)}} \in o\left(\frac{1}{n}\right)$. Therefore, we have

$$\begin{aligned}\mathbb{P}(X_0 + Z_0'(1) + \max_{t\in[0,1]}\{Y_0'(t) - Z_0'(t)\} \geq c^{(n)}) &\leq \left(\frac{e^{\delta^{(n)}}}{(1+\delta^{(n)})^{(1+\delta^{(n)})}}\right)^{\rho^{(n)}} + \frac{2nc}{e^{2n\rho}} \\ &= o\left(\frac{1}{n}\right).\end{aligned}$$

The last equality is since the second term also belongs to $o\left(\frac{1}{n}\right)$. □

Lemma A.9.1. *Let $N(t)$ be a NHPP with intensity $\Lambda(t)$ and $W_1(t)$ be the time of the first arrival. Then*

$$f_{W_1}(t) = e^{-\int_{x=0}^{t}\Lambda(x)}\Lambda(t).$$

Proof.

$$\begin{aligned}\mathbb{P}(\text{First arrival } \in [t,t+h)) &= \mathbb{P}(N(t)=0, N(t+h)-N(t)=1) \\ &= \mathbb{P}(N(t)=0)\mathbb{P}(N(t+h)-N(t)=1) \quad \text{(by independent increments)} \\ &= e^{-\int_{x=t}^{t+h}\Lambda(x)} e^{-\int_{x=0}^{t}\Lambda(x)}\left(\int_{x=t}^{t+h}\Lambda(x)\right) \\ &= e^{-\int_{x=0}^{t+h}\Lambda(x)}\Lambda(t)h + o(h).\end{aligned}$$

Therefore, the density of W_1 is

$$\begin{aligned}f_{W_1}(t) &= \lim_{h\to 0}\frac{\mathbb{P}(\text{First arrival } \in [t,t+h))}{h} \\ &= \lim_{h\to 0}\frac{e^{-\int_{x=0}^{t+h}\Lambda(x)}\Lambda(t)h + o(h)}{h} \\ &= e^{-\int_{x=0}^{t}\Lambda(x)}\Lambda(t).\end{aligned}$$

□