Pricing Analytics for Reusable Resources

Yunjie Sun

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2019

ABSTRACT

Pricing Analytics for Reusable Resources

Yunjie Sun

First, we consider a fundamental pricing model for a single type of reusable resource in which a fixed number of units are used to serve stochastically arriving customers. Customers choose to purchase the resource based on their willingness-to-pay and the current price. If purchased, occupy one unit of the reusable resources for a random amount of time. The firm seeks to maximize a weighted combination of profit, market share, and service level. We establish a series of theoretical results that characterize the strong universal performance of static pricing in such an environment.

Second, we describe a comprehensive approach to pricing analytics for reusable resources in the context of rotable spare parts with an industrial partner. We discuss the process of instilling a new pricing culture and developing a scalable new pricing methodology at a major aircraft manufacturer. We develop a novel pricing analytics approach for all rotable spare parts. The new approach tackles the challenges of limited data availability, minimal demand information, and complex inventory dynamics. We also present a successful large-scale implementation of our approach which led to significant profit gains.

Third, we extend the pricing model for reusable resources to the setting of multiple customer classes. We describe two types of heuristics for this class of problem with accompanying numerical experiments. In addition, we provide a universal performance guarantee for a special case. We also discuss the role of substitution effects between different classes of customers.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgements

First and foremost, I must acknowledge my advisor, Adam Elmachtoub. Adam is an amazing person to be around, both as an advisor, and as a friend. Every time I got stuck in my research, Adam has always been able to provide me with good advice with optimism and encourage me to go further. Besides academic support, Adam is also a good source to talk to about personal life. We chat about NBA games, fitness, and even relationship status. He really takes good care of me and wants me to be successful both in research and in my personal life. His passion, enthusiasm, and optimism truly inspire me to be a better researcher and a better person.

I would also like to thank my collaborator, Omar Besbes. Omar has been a great coauthor and always encourages me to think in a thorough manner and adhere to high standards in research. I would also like to thank Karl Sigman, Van-Anh Troung, and Daniel Guetta for spending time serving in my committee.

My grateful thanks also go to Donald Goldfarb, Cliff Stein, Jose Blanchet, David Yao, Yuri Faenza, and Awi Federgruen for being great teachers in various fundamental domains of operations research. I would also like to express my deep appreciation to Arkadi Nemirovski, Pinar Keskinocak, and Andy Sun from Georgia Tech for encouraging and recommending me to pursue the doctoral degree.

I would also like to thank the staff in the IEOR department for planning many fun events and making the department feel like home. A special thanks to Lizbeth Morales who has helped me a lot on various things and made this process as smooth as possible.

During my time at Columbia, I have met some wonderful people and established good friendships with many of them. Micheal Hamilton, Ryan McNellis, and Lei Xiao are very good labmates who have helped me to improve my presentations every time before a conference. Fengpei Li, Shuoguang Yang, Fan Zhang, and Xiaopei Zhang are among the first few people I met at Columbia and we spent much time studying together for the core courses in the first year. I would like to thank Jinglei Zhang, with whom I worked on the third chapter of my thesis.

I specially thank my girlfriend Yeqing Zhou, for making this journey so meaningful and learning to tolerate me over the years. Yeqing has made my whole life bright and colorful. In addition, Yeqing has been always willing to listen to my thoughts, discuss research/proof ideas, and help me refine my work.

Finally, I would like to thank my parents Pin Cheng and Dongcheng Sun. My mother, Pin, has always been the greatest source of support and love to me. She has done everything she can to help me realize my dream. I could not imagine how could I be the person I am now without her selfless support. My father, Dongcheng, has been providing me with valuable high-level guidance over the years. I truly love you all!

To my mom.

THIS PAGE INTENTIONALLY LEFT BLANK

Introduction

In this thesis, we study pricing problems for reusable resources. A reusable resource is when the same resource to serve one customer can be used to serve future customers after service completion. For example, hotel rooms and airplane seats are one kind of reusable resource as they can be used to serve other customers after customers check out or the flight has arrived at the destination. The cars and bicycles in the recently booming ride-hailing and bicycle-sharing businesses are also good examples of reusable resources. Another example of a reusable resource is cloud computing servers, which can be reused sequentially by different customers to complete jobs. Moreover, reusable resources also play an very important role in the repair industry for large machinery such as aircrafts and trains. In this industry, a considerable amount of spare parts are known as rotable, meaning that the part can be repaired and used to serve future customers. Other than rotable spare parts, the hangar space at service centers is another example of reusable resources in the repair industry.

Pricing decisions are a critical task for a firm, not only for profitability but also for other targets such as market share and service level. Because reusable resources have their own special system dynamics, such dynamics must be taken into account in the pricing model to obtain a good pricing policy. In this thesis, we propose a pricing model that explicitly captures the special system dynamics as well as market competition in order to achieve various objectives of a firm. In the model, a fixed number of units of a reusable resource are used to serve customers. Price-sensitive customers arrive to the system according to a stochastic process. The usage duration is stochastic for customers who purchase the service and the firm may incur a cost to serve the customer.

In the first part of the thesis, we analyze the pricing model when the firm attempts to maximize a weighted combination of three central metrics: profit, market share, and service level. Under the assumptions of Poisson arrivals, exponential service time, and a concave revenue function, we prove that a static pricing policy simultaneously achieves at least 78.9% of the three metrics from the optimal policy. This near-optimal property of the static pricing policy holds for any parameter regime. In addition, we prove that a static pricing policy guarantees 95.5% of the optimal profit in the special case where there are two units of the reusable resource and the induced demand is linear in price. This work is detailed in Chapter 1 and Besbes, Elmachtoub, and Sun (2019b).

In the second part of the thesis, we describe a comprehensive approach to pricing analytics for rotable spare parts at a major aircraft manufacturer. Based on the pricing model for reusable resources, we develop a novel pricing analytics approach that tackles unique challenges such as limited data availability, minimal demand information, and complex inventory dynamics. We also discuss a large-scale implementation of our approach for all rotable spare parts with our industrial partner, which led to an improvement in profits of over 3.9% over a ten month period. This work is detailed in Chapter 2 and Besbes, Elmachtoub, and Sun (2019a).

In the third part of the thesis, we consider the pricing model for reusable resources with multiple customer classes. We propose two types of pricing heuristics to deal with the curse of dimentionality. The first is to construct simple pricing policies from the optimal dynamic pricing policy. The second is to find the optimal splitting of resources to each class and then construct static policies for each part of the split system. We conduct numerical studies showing the effectiveness of the two heuristics in such an environment. In addition, we prove that, under a special case of two units and two classes of customers, a static pricing policy that assigns one price to each class of customers guarantees 80% of the profit from the optimal policy. We also discuss the role of substitution effects between different classes of customers. This work is detailed in Chapter 3.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Static Pricing: Universal Guarantees for Reusable Resources

We consider a fundamental pricing model in which a fixed number of units of a reusable resource are used to serve customers. Customers arrive to the system according to a stochastic process and upon arrival decide whether or not to purchase the service, depending on their willingness-to-pay and the current price. The service time during which the resource is used by the customer is stochastic and the firm may incur a service cost. This model represents various markets for reusable resources such as cloud computing, shared vehicles, rotable parts, and hotel rooms. In this chapter, we analyze this pricing problem when the firm attempts to maximize a weighted combination of three central metrics: profit, market share, and service level. Under Poisson arrivals, exponential service times, and standard assumptions on the willingness-to-pay distribution, we establish a series of results that characterize the performance of static pricing in such environments. In particular, while an optimal policy is fully dynamic in such a context, we prove that a static pricing policy simultaneously guarantees 78.9% of the profit, market share, and service level from the optimal policy. Notably, this result holds for any service rate and number of units the firm operates. In the special case where there are two units and the induced demand is linear, we also prove that the static policy guarantees 95.5% of the profit from the optimal policy. Our numerical findings on a large testbed of instances suggest that the latter result is quite indicative of the profit obtained by the static pricing policy across all parameters.

1.1 Introduction

In many service industries, the same resource to serve one customer can be used to serve future customers once the initial service is completed. This type of resource is commonly referred to as a *reusable resource.* For instance, in the hotel or car rental industry, a fixed number of rooms or vehicles are available to accommodate customers. Each customer uses one of these resources for some number of days until check out or return, after which it is free to be used by another customer. In a related example, many offices, campuses, or apartment buildings offer a pool of bicycles or vehicles to be rented or shared by their members, and units are always returned to their origin after being used by a member. Another example of a reusable resource is cloud computing servers, which can be used by customers to complete jobs after which they become available for processing new jobs. Finally, another interesting example of a reusable resource arises in the repair industry for aircraft, trains, and other large machinery. Specifically, there is a class of spare parts that are known as *rotable*, meaning that when they break, the customer exchanges the broken part for a working part with the repair agent. When the repair agent receives the broken part, it is "utilized" for some time as it is being repaired, after which it becomes available again to service potential future customers.

All of the examples above share several important features which we shall incorporate into our model. First, the number of units available of each resource is fixed (over appropriate time horizons), as acquiring more capacity or units involves significant investments. Second, when a resource is used, the service time is generally stochastic and varies across customers. Third, customers are price-sensitive and in turn the demand rate in each of these applications can be controlled by the price (which can be a one-time fee or an hourly/daily fee to the customer). Fourth, there is a cost incurred by the service provider associated with the usage of a unit (e.g., in terms of cleaning, maintenance, or repair). Finally, in each of these settings it is highly unusual for a customer to wait for service. That is, if all units of the resource are occupied, the customer is typically lost.

In all of the settings above, the seller may have multiple objectives. The *profit rate* is clearly a fundamental objective for any service provider, but typically such providers also focus significantly on their *market share* and *service level*, i.e., the probability that an arriving customer finds a resource available. The latter two metrics are driven by the long term objectives of maintaining a prominent position in the market and ensuring that consumers find the service reliable.

In such environments, an optimal policy will be highly dynamic in general, adjusting its prices often, as a function of the supply conditions. The question this chapter aims to address is the following. *What is the performance of a simple static pricing (one price) policy in such environments?* This question has dual practical and theoretical motivations. On the one hand, in practice, dynamic pricing may not be feasible when prices need to be published in a catalog upfront or may be undesirable due to the negative perception by customers. On the other hand, the existing literature in dynamic pricing has argued for particular objectives that static pricing yields near-optimal performance in large-scale systems (see literature review). How robust is such an insight for a combination of objectives and for arbitrary scales? (While the scale for cloud computing may be large, it is often small as well, e.g., rotable spare parts.) In particular, in this chapter, we aim to derive *universal performance guarantees* for static pricing with respect to the profit, market share, and service level objectives, with an optimal dynamic pricing strategy serving as the benchmark. In particular, we aim to provide results on the strength of static pricing that hold across all possible parameter regimes and scales.

To that end, we anchor our analysis around the following prototypical model. A service provider manages a pool of a single type of reusable resource. The firm uses the reusable resources to deliver service to customers over an infinite horizon. Customers arrive according to a Poisson process in which the rate depends on the price set by the firm. We make the standard assumption that the revenue rate is concave in the arrival rate. Upon arrival, a customer seizes one unit of the resources for an exponentially distributed random amount of time and pays a fee (which could depend on the realized duration of usage or be fixed in advance). The unit of resource occupied by a customer becomes available to others after service completion. The firm may also incur some cost of service. The goal of the firm is to decide on the optimal pricing policy to maximize a combination of three different objectives: profit rate, market share, and service level.

The main contributions of this chapter lie in deriving universal performance guarantees for static pricing and can be summarized as follows.

- We establish that for any combination of the three objectives – profit rate, market share, and service level – there exists a static pricing policy which can achieve at least 78.9% of the value of each objective under the optimal dynamic pricing policy. This result holds for *any capacity size*, *market size*, and *service rate*. Our proof relies on constructing an explicit static policy, characterizing a lower bound in terms of the stock-in probabilities of our policy and the optimal policy, and analyzing this ratio using a change of variables.

- We consider a special case where the service provider is a profit maximizer, there are two units of the reusable resource, and the demand rate is linear in the price. We prove in this case that the static policy achieves at least 95.5% of the optimal profit from dynamic pricing. This result holds for *any market size* and *service rate*. Moreover, this exact scenario arises frequently in practice in the context of providing (very expensive) rotable spare parts (Chapter 2).
- We complement the theoretical results above with numerical experiments over a broad test bed. These illustrate that the performance of static pricing is in general even better. Furthermore, for profit maximization, we find the performance of static pricing is always above 97.5% of that obtained by an optimal dynamic pricing policy, indicating the robustness of the insights derived beyond the exact conditions assumed in the theorems.

To the best of our knowledge, these are the *first* universal guarantees derived for static pricing for this class of problems. Furthermore, the bounds derived highlight the very high performance of static pricing.

1.1.1 Literature review

We next provide an overview of the literature on the effectiveness of static pricing policies in the context of perishable inventory, queuing systems, and reusable resources. We note that although a server in a queueing system is indeed a reusable resource, these systems typically allow for customers to wait for service. In contrast, the reusable resources literature assumes that customers are immediately lost if no units are available.

The dynamic pricing literature has had an extensive focus on the context of perishable resources, where there is a finite time horizon to consume a finite number of units of one or more resources (see den Boer (2015) for a recent survey). The seminal work of Gallego and Van Ryzin (1994) shows that if the revenue function is concave, a static pricing policy loses at most $1/(2\sqrt{\min\{C, \lambda^* t\}})$, where C is the number of units and $\lambda^* t$ represents the expected number of sales under the myopic price. The authors also show a universal guarantee of $1 - 1/e$ for any parameter regime, with both results relying on a concavity assumption on the revenue rate (see also Ma et al. (2018)). Ma et al. (2018) recently generalize these results for the same model without the concavity assumption, and also provide non-adaptive pricing policies for assortment optimization and non-stationary demand settings with constant factor performance guarantees. Chen et al. (2018) showed that the $1-1/e$ guarantee and asymptotic optimality for static pricing also holds in the presence of strategic customers. Gallego et al. (2008) establish conditions for when static pricing is optimal in the presence of strategic customers. The value of static over dynamic pricing policies has also been considered in models with inventory cost and replenishment considerations, such as those in Federgruen and Heching (1999), Chen et al. (2006), Yin and Rajaram (2007), Chen et al. (2010). Related to static pricing policies are policies with limited price changes, such as those considered in Feng and Gallego (1995), Bitran and Mondschein (1997), Netessine (2006), Çelik et al. (2009), Chen et al. (2015), Cheung et al. (2017). Note that in our model there are no inventory costs, and inventory can be repeatedly reused over an infinite time horizon.

There is also an extensive literature on dynamic pricing in queues. Paschalidis and Tsitsiklis (2000) provide numerical results showing the promise of static pricing in multi-class systems. Ata and Shneorson (2006) studied the dynamic pricing of an M/M/1 service system where the objective is welfare maximization, and numerically show it can have significant gains over static pricing. Maglaras and Zeevi (2005) showed in a service system, the revenue generated by the fluid-optimal prices are near optimal when the capacity and market potential are both large. In a related model where the objective is revenue maximization, with observable queues, Kim and Randhawa (2017) quantify more precisely the asymptotic value of dynamic pricing in large systems and prove conditions under which a two price policy is almost as good as a dynamic pricing policy. Banerjee et al. (2015) provide a queueing analysis of a ride-share platform where the customers are modeled as servers, and show that a static price is asymptotically equal to a dynamic price policy for large-scale systems, although dynamic pricing is more robust to modeling error.

Closest to our formulation is the work of Gans and Savin (2007) who study dynamic pricing to maximize the expected profit for rentals. Their model considers discounted rewards with a discrete price ladder, although with multiple customer types. They show the near-optimality of static pricing in highly utilized rental systems where both the offered load and system capacity are large. To the best of our knowledge, all of the previously mentioned results quantifying the gap between static pricing and dynamic pricing hold asymptotically when the scale of the system is large. In contrast, our results provide universal guarantees that do not rely on the scale of the system. Recently, Banerjee et al. (2016) consider a general network of a single type of resource where prices control the rates between nodes, and prove a guarantee of $C/(C+n-1)$ for possibly multiple objectives but zero service times, where n is the number of nodes and C is the number of units. With non-zero service times, as we consider in this chapter, a looser guarantee can be provided only when C is large enough. In this chapter, we provides a guarantee for any number of units, but does not consider a general network.

Various related studies focus on dynamic heuristics, multiple types of reusable resources, and other levers beyond pricing. Lei and Jasin (2018) study the dynamic pricing problem in a setting with deterministic service times and describe policies that are asymptotically optimal in the regime where demand and resource capacity are both large. Doan, Lei, and Shen (2019) study the pricing problem of reusable resources under ambiguous distributions of demand and service time and use robust deterministic approximation models to construct asymptotic optimal fixed-price policies.

Variants of the assortment optimization problem have been considered in Rusmevichientong, Sumida, and Topaloglu (2017), Owen and Simchi-Levi (2018) and Gong et al. (2019) with various universal approximation guarantees. The results in the first two papers can be extended to allow for dynamic pricing with discrete price points. Iyengar, Sigman et al. (2004) and Levi and Radovanović (2010) use linear programming approaches to design admission control policies for such systems, which is a special case of dynamic pricing where a resource is either priced at a nominal price or at infinity. Their admission control policies are asymptotically optimal, and Levi and Radovanović (2010) also provides a universal guarantee of 1/2. Chen, Levi, and Shi (2017) and Chen and Shi (2018) consider generalizations of this model that permit advanced reservations and strategic customers, respectively.

1.1.2 Organization

The chapter is organized as follows. In Section 1.2, we describe the model along with structural properties of the optimal policy. In Section 1.3, we prove the 78.9% performance guarantee of static pricing under our multi-objective setting for any parameter range. We then refine our guarantee to 95.5% in Section 1.4 for the special case of profit maximization with two units under linear demand. We conduct numerical experiments in Section 1.5 that show that the actual performance of static pricing is even stronger than our guarantee, and that such performance still holds when the exact assumptions of the theoretical results do not hold.

1.2 Model and Preliminaries

In this section, we first describe a general model for pricing a reusable resource. We then describe the various performance objectives the service provider may use, followed by several important properties of the optimal dynamic pricing policy.

1.2.1 Pricing model for a reusable resource

We consider a model in which a service provider has a fixed number of identical, non-perishable units of a reusable resource that are sold to price-sensitive customers. The total number of units of the reusable resource that the provider has is C. At any point in time, each unit of the resource is either available for sale or occupied. Note that an occupied unit can be interpreted as a customer using the unit in the cloud computing and ride sharing examples, or being repaired in the rotable spare parts example from Section 1.1.

Customers arrive to the system over time according to a Poisson process with rate $\Lambda > 0$. Each customer has an i.i.d. willingness-to-pay drawn from a valuation distribution F. When a customer arrives, the provider offers a unit at some price p, and a customer decides to purchase usage of the resource if their willingness to pay exceeds p. We denote by $\lambda(p) := \Lambda\bar{F}(p)$ the effective arrival rate at price p. When a customer decides to purchase usage, one unit is then occupied for a random amount of time that follows an exponential distribution with mean $1/\mu$. We assume that the usage times are i.i.d. across customers and independent of the customer valuations.

While a unit is being occupied, the firm cannot sell that unit until it is returned to the system, i.e., a customer finishes using the unit or the provider finishes repairing the unit. The firm incurs a cost c to service one customer, which may be a cleaning, maintenance, or repair cost. Any customer that arrives when all units are occupied is lost, regardless of the current price being offered. This assumption is largely motivated by the fact that in most of our applications the customers are seeking immediate service, and would naturally seek out a competitor if the provider has no units available.

We assume that there is a one-to-one correspondence between prices and effective arrival rates so that $\lambda(p)$ has a unique inverse, denoted by $p(\lambda)$. Therefore, one can view the effective arrival rate λ as the decision variable. The firm dynamically determines a target effective arrival rate λ which can be realized with the corresponding price $p(\lambda)$. From an analysis perspective, the effective arrival rate is more convenient to work with. We shall make the standard assumption in the revenue management literature that the profit rate function $\lambda(p(\lambda) - c)$ is concave in λ.

The set of admissible policies, $\mathbf{\Pi}$, is the set of non-anticipating policies, i.e., policies such that the effective arrival rate at time t, $\lambda(t)$, may only depend on events up to t^-. We shall also be interested in the class of static policies, $\mathbf{\Pi}^s \subset \mathbf{\Pi}$, that simply fix a single arrival rate λ (price) at every time t.

1.2.2 Performance metrics

One natural metric when selling the reusable resources is the expected profit rate. Fix a pricing policy π and let $\lambda(t)$ denote the corresponding effective arrival rate at time t. Let $N^\pi(t)$ denote the corresponding arrival process. Note that the latter is a non-stationary Poisson process with intensity $\lambda(t)$. Let $Q^\pi(t)$ denote the number of on-hand units at time t. The long-run average profit rate $\mathcal{P}^\pi$ is given by

$$\mathcal{P}^\pi = \liminf_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\int_0^T \mathbf{1}\{Q^\pi(t) > 0\}(p(\lambda(t)) - c)dN^\pi(t)\right]. \tag{1.1}$$

For simplicity in the exposition of this chapter, we assume $p(\lambda(t))$ is a one-time fee a user pays for the service. Note that the analysis presented easily generalizes to the case when a user's payment depends on usage time, i.e., it is equivalent to charge a one-time price that is simply the price per time unit multiplied by the expected usage time.

While the firm wants to maximize its profit, it may also want to keep a certain level of market share, i.e., the expected number of units sold, as well as a certain service level. The market share objective $\mathcal{MS}^\pi$ is directly aligned with maximizing sales, while the service level objective $\mathcal{SL}^\pi$ is measured by the fraction of time at least one unit is available. These two objectives can be represented as

$$\mathcal{MS}^{\pi} = \liminf_{T\to\infty} \frac{1}{T}\mathbb{E}\left[N^{\pi}(T)\right]$$

and

$$\mathcal{SL}^{\pi} = \liminf_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\int_0^T \mathbf{1}\{Q^{\pi}(t) > 0\}dt\right].$$

Note that there is a trade-off between the various metrics; the optimal solution for one objective will generally be sub-optimal for another. For instance, maximizing the service level corresponds to setting a static price as large as possible, while maximizing market share corresponds to setting a static price of zero. Clearly neither price will result in any profit at all.

In order to take the different objectives into account simultaneously, we assume the firm maximizes a weighted combination of the objectives,

$$\alpha_1\mathcal{P}^{\pi} + \alpha_2\mathcal{MS}^{\pi} + \alpha_3\mathcal{SL}^{\pi}, \tag{1.2}$$

where $\alpha_1, \alpha_2, \alpha_3 \geq 0$ are the weights placed on each objective by the service provider. Without loss of generality, we assume $\alpha_1 + \alpha_2 + \alpha_3 = 1$. We let V^* denote the long-run value under the optimal policy, and is thus defined by

$$V^* \ := \ \sup_{\pi\in\mathbf{\Pi}} \left\{\alpha_1\mathcal{P}^{\pi} + \alpha_2\mathcal{MS}^{\pi} + \alpha_3\mathcal{SL}^{\pi}\right\}. \tag{1.3}$$

We denote by π^* an optimal policy. Similarly, we let V^s denote the long-run value under the optimal static policy, and is thus defined by

$$V^s := \sup_{\pi \in \mathbf{\Pi}^s} \left\{ \alpha_1 \mathcal{P}^\pi + \alpha_2 \mathcal{MS}^\pi + \alpha_3 \mathcal{SL}^\pi \right\}. \tag{1.4}$$

In this chapter, we focus on universal performance guarantees for static pricing. In particular, we shall focus on the worst-case performance of the optimal static pricing policy in comparison to the optimal dynamic policy. That is, we seek to characterize the maximum possible loss over all possible instances of our model. Formally, we let Ω denote the family of instances

$$\Omega := \{(C, \mu, p(\cdot), c, \alpha_1, \alpha_2, \alpha_3) : \\ C \in \mathbb{N}^+, c, \mu > 0, \alpha_1 + \alpha_2 + \alpha_3 = 1, \alpha_i \geq 0, \lambda(p(\lambda) - c) \text{ concave in } \lambda\}.$$

In turn, we aim to provide a universal lower bound on

$$\inf_{\Omega} \frac{V^s}{V^*},$$

which is the ratio between the objectives under the optimal static and dynamic pricing policies. In fact, we shall show that our bound applies to the corresponding ratios of each of the three objectives.

1.2.3 Analysis of the benchmark V^*

We shall now characterize the structure of an optimal solution to the dynamic pricing problem stated in Equation (1.3). Given the Poisson assumption on arrivals and the exponential assumption on service times, without loss of optimality, one may focus on stationary policies that update the price only at changes in the number of units on-hand. The memoryless property of the exponential distribution allows us to fully characterize the system (a continuous-time Markov chain) by the number of units on-hand. As we shall see, this allows us to provide closed-form expressions for the steady state distribution and objectives under a particular policy.

An admissible policy π may be represented by C arrival rates $\lambda_1, \ldots \lambda_C$. When the provider has only i units available, the price is set to $p(\lambda_i)$. Note that the static policy is a special case where $\lambda_1 = \ldots = \lambda_C$. Furthermore, the system can now be modeled as a birth-death process where each state represents the number of units available. The transition rate from state i to $i+1$ is $(C-i)\mu$ for $i = 0, \ldots, C-1$. The transition rate from i to $i-1$ is λ_i for $i = 1, \ldots, C$. A standard calculation for computing the steady state probabilities, $\mathbb{P}_i(\pi)$ yields that

$$\mathbb{P}_i(\pi) = \frac{C!}{(C-i)!} \frac{\Pi_{j=i+1}^{C} \frac{\lambda_j}{\mu}}{\sum_{k=0}^{C} \frac{C!}{(C-k)!} \Pi_{j=k+1}^{C} \frac{\lambda_j}{\mu}}, \qquad i = 0, \ldots, C.$$

Using the steady state probabilities, we may express our three objectives simply as

$$\mathcal{P}^{\pi} = \sum_{i=1}^{C} \lambda_i (p(\lambda_i) - c) \mathbb{P}_i(\pi) \tag{1.5}$$

$$\mathcal{MS}^{\pi} = \sum_{i=1}^{C} \lambda_i \mathbb{P}_i(\pi), \tag{1.6}$$

$$\mathcal{SL}^{\pi} = \sum_{i=1}^{C} \mathbb{P}_i(\pi) = 1 - \mathbb{P}_0(\pi). \tag{1.7}$$

Let us denote by λ_i^* the effective arrival rate in state i under the optimal policy, and by $\mathbb{P}_i^*$ the steady-state probabilities of being in state i under the optimal policy. In Lemma 1.2.1, we show a fundamental property that effective arrival rates are decreasing as the number of units available increases. Moreover, all such arrival rates do not exceed the myopic rate $\bar{\lambda}$ (the rate only maximizes the immediate reward without considering the future), which yields the highest possible instantaneous objective rate.

Lemma 1.2.1. *Let λ_i^* be the optimal arrival rate when the on-hand inventory level is i. Let $\bar{\lambda}$ denote the myopic arrival rate where $\bar{\lambda} = \arg\max_{\lambda} \lambda(\alpha_1(p(\lambda) - c) + \alpha_2)$. Then*

$$\bar{\lambda} \;\geq\; \lambda_C^* \;\geq\; \cdots \;\geq\; \lambda_1^*. \tag{1.8}$$

The proof of Lemma 1.2.1 can be found in Section 1.6. Notice that the result presented in Lemma 1.2.1 shares the same structural property as presented in Theorem 1 in Gans and Savin (2007) where the objective is only profit maximization in a discounted reward setting. We extend the analysis to a long-run average reward setting with multiple objectives and prove that monotonicity of optimal prices (and rates) still holds. We will make use of this property in the subsequent analysis.

1.3 Static Pricing Guarantee for Multi-Objective Optimization

We next investigate the performance of static pricing and present our first main result.

Theorem 1.3.1. *There exists a static pricing policy π^s that guarantees at least $\frac{15}{19}$ of the profit rate, market share, and service level of the optimal dynamic pricing policy. Equivalently,*

$$\inf_{\Omega} \min \left\{ \frac{\mathcal{P}^{\pi^s}}{\mathcal{P}^{\pi^*}}, \frac{\mathcal{MS}^{\pi^s}}{\mathcal{MS}^{\pi^*}}, \frac{\mathcal{SL}^{\pi^s}}{\mathcal{SL}^{\pi^*}} \right\} \geq \frac{15}{19}.$$

Theorem 1.3.1 provides a strong guarantee: there exists a static price that nearly approximates the performance of an optimal dynamic pricing policy. Specifically, this price guarantees that the profit rate, market share, and service level are at least $\frac{15}{19} \approx .789$ of the corresponding values under the dynamic pricing policy. Of course, a direct consequence of Theorem 1.3.1 is that the optimal single price will have an overall objective of at least 0.789 of the objective under the optimal dynamic pricing policy as well. It is important to note that our result makes no assumption on the number of units in the system, demand rate, or service rate. This is in stark contrast to the previous literature which require the system usage and capacity to be large to provide theoretical guarantees.

It is worthwhile to note that our proof is constructive and exhibits a particular static price that yields such performance. The static price behind our major finding is constructed using the optimal policy, which we denote by π^*. Recall that λ_i^* are the arrival rates under the optimal policy and $\mathbb{P}_i^*$ are the steady-state probabilities. The single price is simply chosen so that the corresponding arrival rate, $\tilde{\lambda}$, is the same as the expected arrival rate under the optimal policy when units are available. More specifically, the static arrival rate $\tilde{\lambda}$ is selected so that

$$\tilde{\lambda} = \frac{\sum_{i=1}^{C} \lambda_i^* \mathbb{P}_i^*}{\sum_{i=1}^{C} \mathbb{P}_i^*} = \frac{\sum_{i=1}^{C} \lambda_i^* \mathbb{P}_i^*}{1 - \mathbb{P}_0^*}. \tag{1.9}$$

Our proof, that we detail in the next subsection exploits this structure, together with the structure of the underlying birth and death process, to derive a universal guarantee.

1.3.1 Proof of Theorem 1.3.1

The proof is organized around two main steps. In the first step, we exploit the concavity of the revenue rate function (in the quantity space) to establish that for each of the three objectives, the ratio of the performances under the static and optimal policies is at least the ratio of the corresponding service levels. The second step bounds the ratio of the service levels by 15/19 by enumerating several cases, with each case proven using elementary calculus. A key component of this second step is a change of variables from demand rates to the product of demand rates. Both steps fundamentally exploit the explicit construction of $\tilde{\lambda}$ in Eq. (1.9). With some abuse of notation, we index quantities with $\tilde{\lambda}$ to denote these under the static policy induced by this static rate.

Step 1. In the first step, we lower bound each of $\frac{\mathcal{P}^{\tilde{\lambda}}}{\mathcal{P}^{\pi^*}}$, $\frac{\mathcal{MS}^{\tilde{\lambda}}}{\mathcal{MS}^{\pi^*}}$, and $\frac{\mathcal{SL}^{\tilde{\lambda}}}{\mathcal{SL}^{\pi^*}}$ by $\frac{1-\mathbb{P}_0(\tilde{\lambda})}{1-\mathbb{P}_0^*}$. Note that by Eq. (1.7), the lower bound is exact for the service level ratio, i.e.,

$$\frac{\mathcal{SL}^{\tilde{\lambda}}}{\mathcal{SL}^{\pi^*}} = \frac{1 - \mathbb{P}_0(\tilde{\lambda})}{1 - \mathbb{P}_0^*}. \tag{1.10}$$

The lower bound is also exact for the market share ratio. Using Eqs. (1.6) and (1.9), we have that

$$\frac{\mathcal{MS}^{\tilde{\lambda}}}{\mathcal{MS}^{\pi^*}} = \frac{\tilde{\lambda}(1-\mathbb{P}_0(\tilde{\lambda}))}{\sum_{i=1}^{C}\lambda_i^*\mathbb{P}_i^*} = \frac{\frac{\sum_{i=1}^{C}\lambda_i^*\mathbb{P}_i^*}{1-\mathbb{P}_0^*}(1-\mathbb{P}_0(\tilde{\lambda}))}{\sum_{i=1}^{C}\lambda_i^*\mathbb{P}_i^*} = \frac{1-\mathbb{P}_0(\tilde{\lambda})}{1-\mathbb{P}_0^*}. \tag{1.11}$$

For the profit ratio by the ratio, we have

$$\begin{aligned}
\frac{\mathcal{P}^{\tilde{\lambda}}}{\mathcal{P}^{\pi^*}} &= \frac{\tilde{\lambda}(p(\tilde{\lambda})-c)(1-\mathbb{P}_0(\tilde{\lambda}))}{\sum_{i=1}^{C}\lambda_i^*(p(\lambda_i^*)-c)\mathbb{P}_i^*} \\
&= \frac{\tilde{\lambda}(p(\tilde{\lambda})-c)}{\sum_{i=1}^{C}\lambda_i^*(p(\lambda_i^*)-c)\frac{\mathbb{P}_i^*}{1-\mathbb{P}_0^*}} \cdot \frac{1-\mathbb{P}_0(\tilde{\lambda})}{1-\mathbb{P}_0^*} \\
&\geq \frac{\tilde{\lambda}(p(\tilde{\lambda})-c)}{\tilde{\lambda}(p(\tilde{\lambda})-c)} \cdot \frac{1-\mathbb{P}_0(\tilde{\lambda})}{1-\mathbb{P}_0^*} \\
&= \frac{1-\mathbb{P}_0(\tilde{\lambda})}{1-\mathbb{P}_0^*}.
\end{aligned} \tag{1.12}$$

The first equality follow from Eq. (1.5). The inequality follows from the fact that the function $\lambda(p(\lambda)-c)$ is concave in λ and applying Jensen's inequality to a random variable that takes value λ_i^* with probability $\frac{\mathbb{P}_i^*}{1-\mathbb{P}_0^*}$ for $i=1,\ldots,C$. Note that the expected value of this random variable is exactly $\tilde{\lambda}$ by Eq. (1.9). We next characterize the stock-in probabilities, and the remainder of the proof, in terms of the new z variables. This variable transformation unlocks the ability to apply (many) basic calculus ideas to prove our lower bound.

Step 2. To find the lower bound of the ratio of stock-in probabilities, we define a set of auxiliary notation which will be useful in our subsequent analysis. We define $a_i := \frac{C!}{(C-i)!}$ for $i=0,1,\ldots,C$ and $z_i := \Pi_{j=i}^{C}\frac{\lambda_j^*}{\mu}$ for $i=1,\ldots,C+1$. For clarity, note that $z_{C+1}=1$. We also define $x := \sum_{k=1}^{C} a_k z_{k+1}$ and $y := \sum_{k=2}^{C} a_k z_k$.

Using the steady-state probabilities derived in Section 1.2.3 and the definition of $\tilde{\lambda}$, the service levels of the static and dynamic policies can be written as

$$1 - \mathbb{P}_0^* = \frac{\sum_{i=1}^C a_i z_{i+1}}{\sum_{i=0}^C a_i z_{i+1}}$$
$$1 - \mathbb{P}_0(\tilde{\lambda}) = \frac{\sum_{i=1}^C a_i [(\sum_{k=1}^C a_k z_k)/(\sum_{k=1}^C a_k z_{k+1})]^{C-i}}{\sum_{i=0}^C a_i [(\sum_{k=1}^C a_k z_k)/(\sum_{k=1}^C a_k z_{k+1})]^{C-i}}.$$

From the above, it is clear that the ratio of the service levels may be written as a function of $z_1, \ldots, z_C$. We call this function $R(z_1, \ldots, z_C)$. Formally,

$$R(z_1, \cdots, z_C) := \frac{1 - \mathbb{P}_0(\tilde{\lambda})}{1 - \mathbb{P}_0^*} = \frac{(\sum_{k=0}^C a_k z_{k+1})(\sum_{i=1}^C a_i [(\sum_{k=1}^C a_k z_k)/(\sum_{k=1}^C a_k z_{k+1})]^{C-i})}{(\sum_{k=1}^C a_k z_{k+1})(\sum_{i=0}^C a_i [(\sum_{k=1}^C a_k z_k)/(\sum_{k=1}^C a_k z_{k+1})]^{C-i})}.$$

We next develop a uniform lower bound on $R(z_1, \cdots, z_C)$ by developing separate bounds for the cases where C is small ($C \leq 3$) or large ($C \geq 4$).

Step 2a. We prove the lower bound for the cases where C is at most 3. When $C = 1$, then $\tilde{\lambda} = \lambda_1^*$ and therefore $R(z_1) = 1$. When $C = 2$, we have

$$R(z_1, z_2) = \frac{z_1^2 + 4z_1z_2 + 3z_1 + 4z_2^2 + 6z_2 + 2}{z_1^2 + 4z_1z_2 + 2z_1 + 5z_2^2 + 6z_2 + 2} \geq \frac{4}{5},$$

where the inequality follows by matching terms and looking at the minimum ratio. When $C = 3$, the numerator of $R(z_1, z_2, z_3)$ is

$$48 + 192z_3 + 120z_2 + 32z_1 + 264z_3^2 + 336z_2z_3 + 96z_1z_3 + 108z_2^2 + 64z_1z_2 + 10z_1^2 + 120z_3^3$$
$$+ 228z_2z_3^2 + 68z_1z_3^2 + 144z_2^2z_3 + 88z_1z_2z_3 + 14z_1^2z_3 + 30z_2^3 + 28z_1z_2^2 + 9z_1^2z_2 + z_1^3$$

while the denominator of $R(z_1, z_2, z_3)$ is

$$48 + 192z_3 + 120z_2 + 24z_1 + 264z_3^2 + 336z_2z_3 + 72z_1z_3 + 108z_2^2 + 48z_1z_2 + 6z_1^2 + 128z_3^3$$
$$+ 252z_2z_3^2 + 60z_1z_3^2 + 168z_2^2z_3 + 84z_1z_2z_3 + 12z_1^2z_3 + 38z_2^3 + 30z_1z_2^{*2} + 9z_1^2z_2 + z_1^3.$$

By matching terms in the numerator and denominator, it is then clear that

$$R(z_1, z_2, z_3) \geq \frac{30}{38} = \frac{15}{19}.$$

Step 2b. Next, we consider the case where $C \geq 4$. While the function ratio $R(\cdot)$ is difficult to analyze directly, we will derive a lower bound on R, which we denote by $\tilde{R}(\cdot)$, which will be amenable to analysis. The bound can be derived simply by observing that

$$\begin{aligned}
R(z_1, \ldots, z_C) &= \frac{(\sum_{k=0}^C a_k z_{k+1})(\sum_{i=1}^C a_i[(\sum_{k=1}^C a_k z_k)/(\sum_{k=1}^C a_k z_{k+1})]^{C-i})}{(\sum_{k=1}^C a_k z_{k+1})(\sum_{i=0}^C a_i[(\sum_{k=1}^C a_k z_k)/(\sum_{k=1}^C a_k z_{k+1})]^{C-i})} \\
&= \frac{(\sum_{k=0}^C a_k z_{k+1})[\sum_{i=1}^C a_i(\sum_{k=1}^C a_k z_k)^{C-i}(\sum_{k=1}^C a_k z_{k+1})^{i-1}]}{[\sum_{i=0}^C a_i(\sum_{k=1}^C a_k z_k)^{C-i}(\sum_{k=1}^C a_k z_{k+1})^{i}]} \\
&\geq \frac{(\sum_{k=0}^C a_k z_{k+1})[\sum_{i=1}^4 a_i(\sum_{k=1}^C a_k z_k)^{C-i}(\sum_{k=1}^C a_k z_{k+1})^{i-1}]}{[\sum_{i=0}^4 a_i(\sum_{k=1}^C a_k z_k)^{C-i}(\sum_{k=1}^C a_k z_{k+1})^{i}]} \\
&= \frac{(\sum_{k=0}^C a_k z_{k+1})[\sum_{i=1}^4 a_i(\sum_{k=1}^C a_k z_k)^{4-i}(\sum_{k=1}^C a_k z_{k+1})^{i-1}]}{[\sum_{i=0}^4 a_i(\sum_{k=1}^C a_k z_k)^{4-i}(\sum_{k=1}^C a_k z_{k+1})^{i}]} \\
&=: \tilde{R}(z_1, \ldots, z_C).
\end{aligned}$$

Next, we derive a lower bound on $\tilde{R}(\cdot)$ though two subcases, depending on the ratio of y to z_1. We first establish in Lemma 1.3.1 (proved in Section 1.6) that the partial derivative with respect to the first argument is non-negative as long as $y \geq a_1 z_1$.

Lemma 1.3.1. *Fix $C \geq 4$. Fix $z_1, z_2, ..., z_C \in [0, \infty)^C$ and suppose $y \geq a_1 z_1$, then*

$$\frac{\partial \tilde{R}}{\partial z_1} \geq 0.$$

When $y \geq a_1 z_1$, Lemma 1.3.1 implies that the worst case value of $\tilde{R}$ occurs when $z_1 = 0$. In turn, in Lemma 1.3.2 (proved in Section 1.6), we establish a uniform lower bound on $\tilde{R}(0, z_2, \ldots, z_C)$.

Lemma 1.3.2. *Fix $C \geq 4$. For all $z_2, ..., z_C \in [0, \infty)^{C-1}$, we have*

$$\tilde{R}(0, z_2, \ldots, z_C) \geq \frac{104}{131}.$$

From Lemmas 1.3.1 and 1.3.2, we can conclude that when $y \geq a_1 z_1$, then $\tilde{R}(z_1, z_2, \ldots, z_C) \geq \tilde{R}(0, z_2, \ldots, z_C) \geq \frac{104}{131}$.

If $y \leq a_1 z_1$, then there is no guarantee on the derivative, but one may directly establish a uniform lower bound on $\tilde{R}$ as articulated in Lemma 1.3.3 (proved in Section 1.6).

Lemma 1.3.3. *Fix $C \geq 4$ and suppose $y \leq a_1 z_1$, then*

$$\tilde{R}(z_1, z_2, \ldots, z_C) \geq \frac{6}{7}.$$

Combining both cases, We conclude that

$$R(z_1, z_2, \ldots, z_C) \geq \tilde{R}(z_1, z_2, \ldots, z_C) \geq \min\{\frac{104}{131}, \frac{6}{7}\} \geq \frac{15}{19}.$$

This completes the proof of Theorem 1.3.1.

1.3.2 Tightness of analysis

We present an example which shows that the lower bound of $\frac{15}{19}$ in Theorem 1.3.1 can be tight for a family of instances. That is, we shall describe instances in which the static policy we construct, $\tilde{\lambda}$, achieves exactly a fraction 15/19 of the optimal dynamic policy. Namely, we shall fix $C = 3$, $\alpha_1 = 0$, $\alpha_2 = 0$, $\alpha_3 = 1$, μ to be arbitrarily close to 0, and $p(\lambda) = \frac{1}{\lambda}$.

Since $\alpha_3 = 1$, then the objective is to maximize the service level, that is

$$\max_{\pi} \mathcal{SL}^{\pi} = 1 - \mathbb{P}(\pi).$$

The service level is always bounded above by 1, and hence it is clear that the policy $(\lambda_1^*, \lambda_2^*, \lambda_3^*) = (0, \Lambda, \Lambda)$ is optimal since

$$\mathcal{SL}^{(0,\Lambda,\Lambda)} = \frac{\sum_{i=1}^{3} \frac{6}{(3-i)!}\mu^i \Pi_{j=i+1}^{3} \lambda_j^*}{\sum_{i=0}^{3} \frac{6}{(3-i)!}\mu^i \Pi_{j=i+1}^{3} \lambda_j^*} = \frac{\sum_{i=1}^{3} \frac{6}{(3-i)!}\mu^i \Pi_{j=i+1}^{3} \lambda_j^*}{0 + \sum_{i=1}^{3} \frac{6}{(3-i)!}\mu^i \Pi_{j=i+1}^{3} \lambda_j^*} = 1.$$

Now let us consider the static policy $\tilde{\lambda}$ which we construct according to Eq. (1.9). Recall from Section 1.3.1 that the performance of the static pricing policy with respect to the service level and market share objectives is $R(z_1, z_2, z_3)$, where $z_i := \Pi_{j=i}^{C} \frac{\lambda_j^*}{\mu}$. In addition, the performance of the static pricing policy with respect to the profit rate is also $R(z_1, z_2, z_3)$ because $\lambda(p(\lambda) - c)$ is linear in λ if $p(\lambda) = \frac{1}{\lambda}$, which makes the Jensen's inequality tight in the derivation of Eq. (1.12). Since $z_1 = 0$, then the ratio becomes

$$R(z_1, z_2, z_3)$$
$$= \frac{48 + 192z_3 + 120z_2 + 264z_3^2 + 336z_2z_3 + 108z_2^2 + 120z_3^3 + 228z_2z_3^2 + 144z_2^2z_3 + 30z_2^3}{48 + 192z_3 + 120z_2 + 264z_3^2 + 336z_2z_3 + 108z_2^2 + 128z_3^3 + 252z_2z_3^2 + 168z_2^2z_3 + 38z_2^3}.$$

Since $z_2 = \frac{\Lambda^2}{\mu^2}$ and $z_3 = \frac{\Lambda}{\mu}$, we have $z_3 \to \infty$ and $z_3 = o(z_2)$ as $\mu \to 0$, and hence

$$\lim_{\mu \to 0} R(z_1, z_2, z_3) = \frac{30}{38} = \frac{15}{19}.$$

1.4 Sharpening the Bound for Profit Maximization

In this section, we seek to focus more deeply on the profit maximization objective corresponding to $\alpha_1 = 1$. This objective is central in the literature and we aim to understand to what extent can our 78.9% guarantee from Section 1.3 be improved.

Theorem 1.4.1. *Fix $C = 2$, and consider any rate μ and linear demand function $\lambda(\cdot)$. Let π^* denote a revenue maximizing dynamic policy. Then there exists a static pricing policy π_s such that*

$$\frac{\mathcal{P}^{\pi^s}}{\mathcal{P}^{\pi^*}} \geq 0.955.$$

This result establishes that for profit maximization, a simple static pricing policy guarantees more than 95.5% of an optimal dynamic pricing policy. This is a much higher guarantee than for the general multi-objective case. In particular, for profit maximization, there is extremely limited value in dynamic pricing.

We note that due to the technical difficulty of the analysis, our result is limited to the case with only 2 units ($C = 2$), and when the demand is linear ($p(\cdot)$ and $\lambda(\cdot)$ are linear), a common assumption in both the literature and practice. However, in Section 1.5, we will see numerically that the level of guarantee above is valid beyond the case $C = 2$ and linear demand. In fact, our computational results shows that the 95.5% lower bound holds across all possible instances considered in this chapter.

The proof of Theorem 1.4.1 is again constructive in that it exhibits a particular static policy with such a guarantee. This policy is the same as the one presented in Eq (1.9). The proof relies on lower bounding the ratio of the service levels, which is indeed a lower bound on the profit ratio as seen in Eq. (1.12). Then, the first order conditions of the profit maximization objective are used to impose constraints on the worst-case arrival rates of an optimal policy, which allows us to find a tighter bound on the ratio of the service levels.

Proof. Proof of Theorem 1.4.1. Let λ_1^*, λ_2^* be the effective arrival rates under the optimal policy for profit maximization, and p_1^*, p_2^* be the corresponding optimal prices. Let $z_1 = \frac{\lambda_1^* \lambda_2^*}{\mu^2}$ and $z_2 = \frac{\lambda_2^*}{\mu}$. For the static policy, let $\tilde{\lambda}$ be defined according to (1.9). Since $C = 2$, by Eq. (1.12) we have

$$\frac{\mathcal{P}^{\tilde{\lambda}}}{\mathcal{P}^*} \geq \frac{1 - \mathbb{P}_0(\tilde{\lambda})}{1 - \mathbb{P}_0^*} = \frac{z_1^2 + 4z_1z_2 + 3z_1 + 4z_2^2 + 6z_2 + 2}{z_1^2 + 4z_1z_2 + 2z_1 + 5z_2^2 + 6z_2 + 2} := R(z_1, z_2).$$

Next, we show that $R(z_1, z_2)$ is increasing in z_1 and decreasing in z_2 by simply looking at the first partial derivatives. Taking derivatives of R w.r.t z_1 and z_2 gives

$$\begin{aligned}\frac{\partial R(z_1,z_2)}{\partial z_1} &= \frac{-z_1^2+2z_1z_2^2+4z_2^3+7z_2^2+6z_2+2}{(z_1^2+4z_1z_2+2z_1+5z_2^2+6z_2+2)^2} \geq 0\\ \frac{\partial R(z_1,z_2)}{\partial z_2} &= -\frac{2(z_1^2(z_2+2)+z_1(2z_2^2+7z_2+3)+z_2(3z_2+2))}{(z_1^2+4z_1z_2+2z_1+5z_2^2+6z_2+2)^2} \leq 0.\end{aligned}$$

To see that the partial derivative w.r.t. z_1 is non-negative, it is sufficient to show that $z_1 \leq z_2^2$, which follows from the fact that $\lambda_1^* \leq \lambda_2^*$, established in Lemma 1.2.1. To see that the partial derivative w.r.t. z_2 is non-positive, observe that all terms in the numerator are negative.

The remainder of the proof proceeds by dividing the analysis in two cases: if z_2 is above or below $\frac{\sqrt{7}-1}{3}$. When $z_2 \leq \frac{\sqrt{7}-1}{3}$, then in this case

$$R(z_1,z_2) \geq R(0,\frac{\sqrt{7}-1}{3}) \approx 0.9557$$

since $R(z_1,z_2)$ is increasing in z_1 and decreasing in z_2.

For the remainder of the proof we consider the case where $z_2 > \frac{\sqrt{7}-1}{3}$. In this case, we leverage the first-order optimality conditions of the problem to show in Lemma 1.4.1 that z_1 and z_2 must be within a provable quantity of one another. This constraint then allows us to tighten the lower bound on $R(\cdot)$. Denote $\gamma_i = -p'(\lambda_i)$. Notice that since the demand is linear, then $\gamma_1 = \gamma_2$. Define $\beta := \frac{p_1^*-c}{\gamma_1} \geq 0$, and now we are ready to state the bounds on z_1 and z_2 in Lemma 1.4.1 (proved in Section 1.6).

Lemma 1.4.1. *Let* $g(\beta, z_2) = \sqrt{(z_2+1)^2 + \beta z_2(z_2+2)} - (z_2+1)$. *Then*

$$z_1 \geq g(\beta, z_2) \tag{1.13}$$

$$z_2 \leq \sqrt{2\beta}. \tag{1.14}$$

By Lemma 1.4.1 and the fact that $R(z_1, z_2)$ is non-decreasing in z_1 we have that

$$\begin{aligned}
R(z_1, z_2) &\geq R(g(\beta, z_2), z_2) \\
&= \frac{(2+\beta)z_2^2 + (2\beta+3)z_2 + (2z_2+1)\sqrt{(1+\beta)z_2^2 + 2(1+\beta)z_2 + 1} + 1}{(3+\beta)z_2^2 + (2\beta+4)z_2 + 2z_2\sqrt{(1+\beta)z_2^2 + 2(1+\beta)z_2 + 1} + 2} \\
&= 1 - \frac{z_2^2 + z_2 + 1 - \sqrt{(1+\beta)z_2^2 + 2(1+\beta)z_2 + 1}}{(3+\beta)z_2^2 + (2\beta+4)z_2 + 2z_2\sqrt{(1+\beta)z_2^2 + 2(1+\beta)z_2 + 1} + 2} \\
&:= 1 - G(\beta, z_2). \qquad (1.15)
\end{aligned}$$

Therefore, minimizing $R(z_1, z_2)$ is equivalent to maximizing $G(\beta, z_2)$, for which we provide an upper bound in Lemma 1.4.2 (proved in Section 1.6).

Lemma 1.4.2. *If* $z_2 \geq \frac{\sqrt{7}-1}{3}$ *and* $\beta \geq 0$, *then* $G(\beta, z_2) \leq 0.0433$.

Therefore, in the case when $z_2 \geq \frac{\sqrt{7}-1}{3}$, Eq. (1.15) and Lemma 1.4.2 imply that

$$R(z_1, z_2) \geq 1 - G(\beta, z_2) \geq 1 - 0.0433 = 0.9567.$$

Combining both cases, we obtain the claimed result and the proof is complete. □

1.5 Numerical Experiments

In this section, we conduct a set of numerical experiments to test the performance of the static pricing policy. We consider three types of demand functions: linear, exponential, and logistic. For a linear demand curve, we assume it takes the form $\lambda = -ap + b$; for an exponential demand curve, we assume it follows $\lambda = be^{-ap}$; for the logistic demand curve, we assume it is $\lambda = \frac{b(1+e^{-ap^0})}{1+e^{a(p-p^0)}}$ where p^0 is the inflection point. Notice that in all three demand curves, the maximum demand rate is set to be b when the price is set to 0.

For each value of C, we randomly generate the mean usage time uniformly in $\frac{1}{\mu} \in [0.05, 50]$; a is randomly generated uniformly between 0.1 and 5; b is randomly generated uniformly between 0.5 and 10; p^0 is randomly generated uniformly between [0,20]. We assume that the average service cost is 0, i.e., $c = 0$. We generate 1,000 different instances of inputs and calculate the profit rate under the optimal dynamic pricing policy, the constructed static price policy $\tilde{\lambda}$ according to Eq. (1.9), and the best static price policy π^{s^*}. We report the *worst case* of $\frac{\mathcal{P}^{\tilde{\lambda}}}{\mathcal{P}^{\pi^*}}$ and $\frac{\mathcal{P}^{\pi^{s^*}}}{\mathcal{P}^{\pi^*}}$ for each capacity level C. The results are summarized in Table 1.1.

As one can observe, the performance of static pricing is generally higher than 97.5%. When $C = 2$, we observe the worst case to be 99.53% in the case of linear demand, which is even higher than the 95.5% guarantee proven in Theorem 1.4.1. We also note that this very high performance of static prices continues to hold when we depart from the exact assumptions of Theorem 1.4.1, for general values of C and for exponential and logistic demand curves.

In general, the worst case performance of static pricing (either the best static price or

	Linear		Exponential		Logistic	
C	$\frac{\mathcal{P}^{\tilde{\lambda}}}{\mathcal{P}^{\pi^*}}$	$\frac{\mathcal{P}^{\pi^{s^*}}}{\mathcal{P}^{\pi^*}}$	$\frac{\mathcal{P}^{\tilde{\lambda}}}{\mathcal{P}^{\pi^*}}$	$\frac{\mathcal{P}^{\pi^{s^*}}}{\mathcal{P}^{\pi^*}}$	$\frac{\mathcal{P}^{\tilde{\lambda}}}{\mathcal{P}^{\pi^*}}$	$\frac{\mathcal{P}^{\pi^{s^*}}}{\mathcal{P}^{\pi^*}}$
2	99.53%	99.54%	99.06%	99.07%	99.16%	99.18%
3	99.27%	99.28%	98.57%	98.60%	98.68%	98.72%
4	99.10%	99.12%	98.26%	98.31%	98.41%	98.46%
5	98.97%	99.00%	98.05%	98.11%	98.19%	98.28%
10	98.66%	98.71%	97.58%	97.70%	97.71%	97.84%
20	98.46%	98.55%	97.38%	97.56%	97.46%	97.70%
30	98.40%	98.51%	97.39%	97.57%	97.45%	97.68%
40	98.38%	98.51%	97.48%	97.62%	97.51%	97.72%
50	98.37%	98.51%	97.60%	97.69%	97.58%	97.79%

Table 1.1: Worst case profit ratio: static pricing policies vs. optimal dynamic pricing policy.

the price we construct) does not happen when $C = 2$. However, the ratio of the profit rate achieved by the static policy and the optimal profit rate appears to be relatively independent of the value of C. Of course, as C approaches infinity, the worst case ratio indeed converges to 1.

In addition, one may observe that the performance of the static price policy we constructed in the proofs is very close to the performance of the best static price for profit maximization. The difference of the worst case performance between the two static prices is usually less than 0.2%.

Using a similar testbed, we also conducted numerical experiments for the multi-objective case. We use the linear demand model in the numerical experiment and randomly generate the values of α_i's uniformly at random. The rest of the experiment settings are the same as described before. We calculate the worst case performance of our constructed static price compared to the total objective as well as for the three performance metrics. The results are presented in Table 1.2.

As one may notice, the lowest of the worst case performance ratio happens when $C = 3$

C	$\frac{V^{\bar{\lambda}}}{V^*}$	$\frac{\mathcal{P}^{\bar{\lambda}}}{\mathcal{P}^{\pi^*}}$	$\frac{\mathcal{MS}^{\bar{\lambda}}}{\mathcal{MS}^{\pi^*}}$	$\frac{\mathcal{SL}^{\bar{\lambda}}}{\mathcal{SL}^{\pi^*}}$
2	81.08%	84.70%	81.03%	81.03%
3	80.32%	83.85%	80.23%	80.23%
4	80.95%	84.45%	80.82%	80.82%
5	81.80%	85.27%	81.63%	81.63%
10	85.37%	88.67%	85.02%	85.02%
15	87.68%	90.79%	87.17%	87.17%
20	89.30%	92.22%	88.67%	88.67%

Table 1.2: Performance of static pricing with multiple objectives.

at 80.32% for the overall objectives, and 80.23% for the market share and service level objectives. For this worst case ratio, the values of α_i's are similar to the construction in our tightness example where α_3 is very close to 1 while α_1 and α_2 is close to 0. This finding is consistent with our tightness analysis.

1.6 Additional proofs

Proof of Lemma 1.2.1. We prove this lemma by transforming the continuous time Markov Decision Process (MDP) to a discrete time MDP and showing that the value iteration operator preserve concavity and monotonicity.

Using standard techniques (see, e.g., Bertsekas (2012)), the continuous time MDP associated to Equation (1.3) can be transformed into a discrete time MDP, through uniformization, and solved efficiently using value iteration. Let γ be given by

$$\gamma = \frac{1}{1 + \Lambda + C\mu},$$

where Λ is the maximum demand rate. Note that $1 + \Lambda + C\mu$ upper bounds the transition rates from any state in the Markov Chain.

Let $h(i)$ denote the relative, long-run expected reward associated with having i units available and η be the optimal average profit. The value iteration operator, $\mathcal{T}$, takes the following form,

$$\begin{aligned}\mathcal{T}h(i) = \max_{\lambda\in[0,\Lambda]}\{&\alpha_1\lambda(p(\lambda)-c)+\alpha_2\lambda+\alpha_3-\eta+\\ &\gamma\lambda h(i-1)+\gamma\mu(C-i)h(i+1)+(1-\gamma(\lambda+\mu(C-i))h(i))\}\ \forall i \quad (1.16)\end{aligned}$$

where

$$h(0)=0.$$

Letting $h^*(i)$ denote the relative optimal expected reward of having i units available, then $h^*(i)=\lim_{n\to\infty}\mathcal{T}^n h(i)$. We next prove the fact that $h^*(i)$ is nondecreasing and concave by showing $\mathcal{T}h(i)$ is nondecreasing and concave if $h(i)$ has the same properties.

For any state i, we can rewrite the value iteration presented in Equation (1.16) as follows:

$$\mathcal{T}h(i)=A(i)+B(i)$$

where

$$\begin{aligned}A(i)&=\max_{\lambda\in[0,\Lambda]}\left[\alpha_1\lambda(p(\lambda)-c)+\alpha_2\lambda+\gamma\lambda h(i-1)+\gamma(1+\Lambda-\lambda)h(i)\right],\\ B(i)&=\gamma\mu\left[(C-i)h(i+1)+ih(i)\right]-\eta+\alpha_3.\end{aligned}$$

Denote

$$\lambda_i = \arg\max A(i)$$

In order to show $\mathcal{T}h(i)$ is nondecreasing and concave, we will show both $A(i)$ and $B(i)$ are nondecreasing and concave.

To show that $A(i)$ is nondecreasing in i, observe that

$$\begin{aligned}
A(i) - A(i-1) &= A(i)|_{\lambda_i} - A(i-1)|_{\lambda_{i-1}} \\
&\geq A(i)|_{\lambda_{i-1}} - A(i-1)|_{\lambda_{i-1}} \\
&= \gamma\lambda_{i-1}\left[h(i-1) - h(i-2)\right] + \gamma(1 + \Lambda - \lambda_{i-1})\left[h(i) - h(i-1)\right] \\
&\geq 0.
\end{aligned}$$

The first inequality comes from the fact that λ_i is the maximizer of $A(i)$. The last inequality comes from the assumption that $h(\cdot)$ is nondecreasing.

To show that $B(i)$ in nondecreasing in i, observe that

$$\begin{aligned}
B(i) - B(i-1) &= \gamma\mu\left((C-i)h(i+1) + ih(i) - (C-i+1)h(i) - (i-1)h(i-1)\right) \\
&= \gamma\mu\left((C-i)\left[h(i+1) - h(i)\right] + (i-1)\left[h(i) - h(i-1)\right]\right) \\
&\geq 0,
\end{aligned}$$

since $h(\cdot)$ is nondecreasing and both γ and μ are positive.

To establish the concavity of $A(i)$, observe that

$$\begin{aligned}
&A(i-1)+A(i+1)-2A(i)\\
&=A(i-1)|_{\lambda_{i-1}}+A(i+1)|_{\lambda_{i+1}}-2A(i)|_{\lambda_i}\\
&\leq A(i-1)|_{\lambda_{i-1}}+A(i+1)|_{\lambda_{i+1}}-A(i)|_{\lambda_{i-1}}-A(i)|_{\lambda_{i+1}}\\
&=\gamma\lambda(\lambda_{i-1})h(i-2)+\gamma(1+\Lambda-\lambda_{i-1})h(i-1)+\lambda_{i+1}h(i)+\gamma(1+\Lambda-\lambda_{i+1})h(i+1)\\
&\quad-\lambda_{i-1}h(i-1)+\gamma(1+\Lambda-\lambda_{i-1})h(i)-\lambda_{i+1}h(i-1)+\gamma(1+\Lambda-\lambda_{i+1})h(i)\\
&=\gamma\lambda_{i-1}\left[h(i-2)+h(i)-2h(i-1)\right]+\gamma(1+\Lambda)\left[h(i-1)+h(i+1)-2h(i)\right]\\
&\quad+\gamma\lambda_{i+1}\left[2h(i)-h(i-1)-h(i+1)\right]\\
&=\gamma\lambda_{i-1}\left[h(i-2)+h(i)-2h(i-1)\right]+\gamma(1+\Lambda-\lambda_{i+1})\left[h(i-1)+h(i+1)-2h(i)\right]\\
&\leq 0.
\end{aligned}$$

The first inequality follows from the fact that λ_i is the maximizer of $A(i)$. Since Λ is the maximum rate of customer arrivals, then $1+\Lambda-\lambda_{i+1}$ is positive and the last inequality follows from the concavity of $h(\cdot)$.

To establish the concavity of $B(i)$, observe that

$$\begin{aligned}
B(i-1)+B(i+1)-2B(i)&=\gamma\mu\left[(C-(i-1))h(i)+(i-1)h(i-1)\right]\\
&\quad+\gamma\mu\left[(C-(i+1))h(i+2)+(i+1)h(i+1)\right]\\
&\quad-\gamma\mu\left[2(C-i)h(i+1)-2ih(i)\right]\\
&=\gamma\mu\left[(i-1)\left[h(i-1)+h(i+1)-2h(i)\right]\right]\\
&\quad+\gamma\mu\left[(C-i-1)\left[h(i+2)+h(i)-2h(i+1)\right]\right]
\end{aligned}$$

$$\leq 0.$$

The last inequality follows from the assumption that $h(\cdot)$ is concave and the fact that both γ and μ are positive.

Recall from Equation (1.16), the optimal prices can be solved using the following equation,

$$\lambda_i^* = \arg\max_{\lambda} \ \lambda[\alpha_1(p(\lambda) - c) + \alpha_2 - \gamma(h^*(i) - h^*(i-1)].$$

Given the nondecreasing and concave properties of $h^*(\cdot)$, we can conclude the desired property of the optimal policy. $\square$

Proof of Lemma 1.3.1. The proof follows by simply showing that $\frac{\partial \tilde{R}(z_1,\ldots,z_C)}{\partial z_1} \geq 0$, which is equivalent to showing that the numerator of $\frac{\partial \tilde{R}(z_1,\ldots,z_C)}{\partial z_1}$ is non-negative. To do this, we first establish a few facts.

Since λ_i^* is non-decreasing in i from Lemma 1.2.1, then for $k = 1, \ldots, C$ we have that

$$z_1 z_{k+1} = \Pi_{i=1}^{C} \frac{\lambda_i^*}{\mu} \Pi_{j=k+1}^{C} \frac{\lambda_j^*}{\mu} \leq \Pi_{i=2}^{C} \frac{\lambda_i^*}{\mu} \Pi_{j=k}^{C} \frac{\lambda^* h_j}{\mu} = z_2 z_k. \tag{1.17}$$

Therefore,

$$y(a_1 z_1 + y) = \left(\sum_{j=2}^{C} a_j z_j\right)\left(\sum_{i=1}^{C} a_i z_i\right) \geq a_2 z_2 \left(\sum_{i=1}^{C} a_i z_i\right) \geq a_2 \left(\sum_{i=1}^{C} a_i z_1 z_{i+1}\right) = a_2 z_1 x. \tag{1.18}$$

where the second inequality follows from Eq. (1.17).

Under the assumption of $y \geq a_1 z_1$ and the fact that $y \leq (C-1)x$, we also have that

$$x \geq z_1, \tag{1.19}$$

$$x \geq \frac{a_1 z_1 + y}{2(C-1)}. \tag{1.20}$$

Using the definitions of x and y, we may rewrite $\tilde{R}(\cdot)$ as

$$\tilde{R}(z_1, \ldots, z_C) = \frac{(a_0 z_1 + x)[a_1(a_1 z_1 + y)^3 + a_2(a_1 z_1 + y)^2 x + a_3(a_1 z_1 + y)x^2 + a_4 x^3]}{a_0(a_1 z_1 + y)^4 + a_1(a_1 z_1 + y)^3 x + a_2(a_1 z_1 + y)^2 x^2 + a_3(a_1 z_1 + y)x^3 + a_4 x^4}.$$

The derivative of the numerator of $\tilde{R}(z_1, \ldots, z_C)$is

$$\begin{aligned}
&[(a_1 z_1 + y)^4 + a_1(a_1 z_1 + y)^3 x + a_2(a_1 z_1 + y)^2 x^2 + a_3(a_1 z_1 + y)x^3 + a_4 x^4]\times \\
&[3a_1^2(a_1 z_1 + y)^2(z_1 + x) + 2a_1 a_2(a_1 z_1 + y)(z_1 + x)x + a_1 a_3(z_1 + x)x^2 \\
&\quad + a_1(a_1 z_1 + y)^3 + a_2(a_1 z_1 + y)^2 x + a_3(a_1 z_1 + y)x^2 + a_4 x^3] \\
&- [a_1(a_1 z_1 + y)^3(z_1 + x) + a_2(a_1 z_1 + y)^2(z_1 + x)x \\
&\quad + a_3(a_1 z_1 + y)(z_1 + x)x^2 + a_4(z_1 + x)x^3]\times \\
&[4a_1(a_1 z_1 + y)^3 + 3a_1^2(a_1 z_1 + y)^2 x + 2a_1 a_2(a_1 z_1 + y)x^2 + a_1 a_3 x^3]
\end{aligned}$$

$$\begin{aligned}
=&3a_1^2(a_1 z_1 + y)^6(z_1 + x) + 2a_1 a_2(a_1 z_1 + y)^5(z_1 + x)x + a_1 a_3(a_1 z_1 + y)^4(z_1 + x)x^2 \\
&\quad + a_1(a_1 z_1 + y)^7 + a_2(a_1 z_1 + y)^6 x + a_3(a_1 z_1 + y)^5 x^2 + a_4(a_1 z_1 + y)^4 x^3 \\
&+ 3a_1^3(a_1 z_1 + y)^5(z_1 + x)x + 2a_1^2 a_2(a_1 z_1 + y)^4(z_1 + x)x^2 + a_1^2 a_3(a_1 z_1 + y)^3(z_1 + x)x^3 \\
&\quad + a_1^2(a_1 z_1 + y)^6 x + a_1 a_2(a_1 z_1 + y)^5 x^2 + a_1 a_3(a_1 z_1 + y)^4 x^3 + a_1 a_4(a_1 z_1 + y)^3 x^4
\end{aligned}$$

$$+ 3a_1^2a_2(a_1z_1 + y)^4(z_1 + x)x^2 + 2a_1a_2^2(a_1z_1 + y)^3(z_1 + x)x^3 + a_1a_2a_3(a_1z_1 + y)^2(z_1 + x)x^4$$

$$+ a_1a_2(a_1z_1 + y)^5x^2 + a_2^2(a_1z_1 + y)^4x^3 + a_2a_3(a_1z_1 + y)^3x^4 + a_2a_4(a_1z_1 + y)^2x^5$$

$$+ 3a_1^2a_3(a_1z_1 + y)^3(z_1 + x)x^3 + 2a_1a_2a_3(a_1z_1 + y)^2(z_1 + x)x^4 + a_1a_3^2(a_1z_1 + y)(z_1 + x)x^5$$

$$+ a_1a_3(a_1z_1 + y)^4x^3 + a_2a_3(a_1z_1 + y)^3x^4 + a_3^2(a_1z_1 + y)^2x^5 + a_3a_4(a_1z_1 + y)x^6$$

$$+ 3a_1^2a_4(a_1z_1 + y)^2(z_1 + x)x^4 + 2a_1a_2a_4(a_1z_1 + y)(z_1 + x)x^5 + a_1a_3a_4(z_1 + x)x^6$$

$$+ a_1a_4(a_1z_1 + y)^3x^4 + a_2a_4(a_1z_1 + y)^2x^5 + a_3a_4(a_1z_1 + y)x^6 + a_4^2x^7$$

$$- 4a_1^2(a_1z_1 + y)^6(z_1 + x) - 3a_1^3(a_1z_1 + y)^5(z_1 + x)x - 2a_1^2a_2(a_1z_1 + y)^4(z_1 + x)x^2$$

$$- a_1^2a_3(a_1z_1 + y)^3(z_1 + x)x^3 - 4a_1a_2(a_1z_1 + y)^5(z_1 + x)x - 3a_1^2a_2(a_1z_1 + y)^4(z_1 + x)x^2$$

$$- 2a_1a_2^2(a_1z_1 + y)^3(z_1 + x)x^3 - a_1a_2a_3(a_1z_1 + y)^2(z_1 + x)x^4 - 4a_1a_3(a_1z_1 + y)^4(z_1 + x)x^2$$

$$- 3a_1^2a_3(a_1z_1 + y)^3(z_1 + x)x^3 - 2a_1a_2a_3(a_1z_1 + y)^2(z_1 + x)x^4 - a_1a_3^2(a_1z_1 + y)(z_1 + x)x^5$$

$$- 4a_1a_4(a_1z_1 + y)^3(z_1 + x)x^3 - 3a_1^2a_4(a_1z_1 + y)^2(z_1 + x)x^4$$

$$- 2a_1a_2a_4(a_1z_1 + y)(z_1 + x)x^5 - a_1a_3a_4(z_1 + x)x^6$$

$$=3a_1^2(a_1z_1 + y)^6(z_1 + x) + [2a_1a_2 + 3a_1^3](a_1z_1 + y)^5(z_1 + x)x$$

$$+ [a_1a_3 + 5a_1^2a_2](a_1z_1 + y)^4(z_1 + x)x^2 + a_1(a_1z_1 + y)^7 + [a_2 + a_1^2](a_1z_1 + y)^6x$$

$$+ [a_3 + 2a_1a_2](a_1z_1 + y)^5x^2 + [a_4 + a_2^2 + 2a_1a_3](a_1z_1 + y)^4x^3$$

$$+ [4a_1^2a_3 + 2a_1a_2^2](a_1z_1 + y)^3(z_1 + x)x^3 + [2a_1a_4 + 2a_2a_3](a_1z_1 + y)^3x^4$$

$$+ [3a_1a_2a_3 + 3a_1^2a_4](a_1z_1 + y)^2(z_1 + x)x^4 + [2a_2a_4 + a_3^2](a_1z_1 + y)^2x^5$$

$$+ [a_1a_3^2 + 2a_1a_2a_4](a_1z_1 + y)(z_1 + x)x^5 + 2a_3a_4(a_1z_1 + y)x^6 + a_1a_3a_4(z_1 + x)x^6 + a_4^2x^7$$

$$- 4a_1^2(a_1z_1 + y)^6(z_1 + x) - [3a_1^3 + 4a_1a_2](a_1z_1 + y)^5(z_1 + x)x$$

$$- [5a_1^2a_2 + 4a_1a_3](a_1z_1 + y)^4(z_1 + x)x^2 - [4a_1^2a_3 + 2a_1a_2^2 + 4a_1a_4](a_1z_1 + y)^3(z_1 + x)x^3$$

$$- [3a_1a_2a_3 + 3a_1^2a_4](a_1z_1 + y)^2(z_1 + x)x^4 - [a_1a_3^2 + 2a_1a_2a_4](a_1z_1 + y)(z_1 + x)x^5$$

$$- a_1a_3a_4(z_1 + x)x^6$$

$$=a_1(a_1z_1 + y)^7 + [a_2 + a_1^2](a_1z_1 + y)^6x + [a_3 + 2a_1a_2](a_1z_1 + y)^5x^2$$

$$+ [a_4 + a_2^2 + 2a_1a_3](a_1z_1 + y)^4x^3 + [2a_1a_4 + 2a_2a_3](a_1z_1 + y)^3x^4$$

$$+ [2a_2a_4 + a_3^2](a_1z_1 + y)^2x^5 + 2a_3a_4(a_1z_1 + y)x^6 + a_4^2x^7$$

$$- a_1^2(a_1z_1 + y)^6(z_1 + x) - 2a_1a_2(a_1z_1 + y)^5(z_1 + x)x - 3a_1a_3(a_1z_1 + y)^4(z_1 + x)x^2$$

$$- 4a_1a_4(a_1z_1 + y)^3(z_1 + x)x^3$$

$$=a_1(a_1z_1 + y)^7 + a_2(a_1z_1 + y)^6x + a_3(a_1z_1 + y)^5x^2 + [a_4 + a_2^2 - a_1a_3](a_1z_1 + y)^4x^3$$

$$+ [2a_2a_3 - 2a_1a_4](a_1z_1 + y)^3x^4 + [2a_2a_4 + a_3^2](a_1z_1 + y)^2x^5 + 2a_3a_4(a_1z_1 + y)x^6 + a_4^2x^7$$

$$- a_1^2(a_1z_1 + y)^6z_1 - 2a_1a_2(a_1z_1 + y)^5z_1x - 3a_1a_3(a_1z_1 + y)^4z_1x^2 - 4a_1a_4(a_1z_1 + y)^3z_1x^3$$

$$=a_1y(a_1z_1 + y)^6 + a_2y(a_1z_1 + y)^5x + a_3y(a_1z_1 + y)^4x^2 + [a_4 + a_2^2 - a_1a_3]y(a_1z_1 + y)^3x^3$$

$$+ [2a_2a_3 - 2a_1a_4](a_1z_1 + y)^3x^4 + [2a_2a_4 + a_3^2](a_1z_1 + y)^2x^5 + 2a_3a_4(a_1z_1 + y)x^6 + a_4^2x^7$$

$$- a_1a_2(a_1z_1 + y)^5z_1x - 2a_1a_3(a_1z_1 + y)^4z_1x^2 - [3a_1a_4 - a_1a_2^2 + a_1^2a_3](a_1z_1 + y)^3z_1x^3$$

$$\geq a_1a_2(a_1z_1 + y)^5z_1x + a_2^2(a_1z_1 + y)^4z_1x^2 + a_1a_3(a_1z_1 + y)^4z_1x^2$$

$$+ [a_1a_4 + a_1a_2^2 - a_1^2a_3](a_1z_1 + y)^3z_1x^3 + [2a_2a_3 - 2a_1a_4](a_1z_1 + y)^3z_1x^3$$

$$
\begin{aligned}
&+\frac{2a_2a_4+a_3^2}{2(C-1)}(a_1z_1+y)^3z_1x^3+\frac{a_3a_4}{2(C-1)^2}(a_1z_1+y)^3z_1x^3+a_4^2x^7\\
&-a_1a_2(a_1z_1+y)^5z_1x-2a_1a_3(a_1z_1+y)^4z_1x^2-[3a_1a_4-a_1a_2^2+a_1^2a_3](a_1z_1+y)^3z_1x^3
\end{aligned}
$$

$$\geq 0.$$

The first equality follows from expanding the products completely. The second equality follows from combining positive terms, and then the negative terms. The third equality follows from canceling terms out. The fourth equality follows from expanding (z_1+x) terms and simplifying. The fifth equality follows form expanding (a_1z_1+y) in some of the positive terms and simplifying. The first inequality follows from lower bounding some terms using $y \geq a_1z_1$, Eq. (1.18), Eq. (1.19), or Eq. (1.20). The second inequality follows since

$$a_2^2 = C^2(C-1)^2 \geq C^2(C-1)(C-2) = a_1a_3$$

and

$$
\begin{aligned}
&a_1a_4+a_1a_2^2-a_1^2a_3+2a_2a_3-2a_1a_4+\frac{2a_2a_4+a_3^2}{2(C-1)}+\frac{a_3a_4}{2(C-1)^2}-[3a_1a_4-a_1a_2^2+a_1^2a_3]\\
=&2a_1a_2^2-2a_1^2a_3-4a_1a_4+2a_2a_3+\frac{2a_2a_4+a_3^2}{2(C-1)}+\frac{a_3a_4}{2(C-1)^2}\\
=&C^2(6+C(6C-13))\\
>&0
\end{aligned}
$$

when $C \geq 4$. $\square$

Proof of Lemma 1.3.2. Recall that

$$\tilde{R}(0, z_2, \ldots, z_C) = \frac{\sum_{i=1}^{4} a_i y^{4-i} x^i}{\sum_{i=0}^{4} a_i y^{4-i} x^i}$$

The main idea in proving this lemma is to compare the ratio of the coefficient of every term in $\tilde{R}(0, z_2, \ldots, z_C)$. First, we restrict our focus only to the ratio of the coefficients for the terms when y^4 is expanded, since the ratio of the coefficients terms not in y^4 is 1. To see the fact that every term not in y^4 has the same value of coefficient in both the numerator and denominator, we can rewrite $\tilde{R}(0, z_2, \ldots, z_C)$ as

$$\tilde{R}(0, z_2, \ldots, z_C) = \frac{\sum_{i=1}^{4} a_i y^{4-i} x^i}{y^4 + \sum_{i=1}^{4} a_i y^{4-i} x^i}$$

Since every term not in y^4 must be in $\sum_{i=1}^{4} a_i y^{4-i} x^i$, and $\sum_{i=1}^{4} a_i y^{4-i} x^i$ appears in both numerator and denominator, then the ratio of the coefficient of the terms not in y^4 must be 1. Since $\tilde{R}(0, z_2, \ldots, z_C) \leq 1$ by definition, we only need to focus on the ratio of the coefficient of the terms in y^4 to find the lower bound of $\tilde{R}(0, z_2, \ldots, z_C)$.

We now calculate a lower bound on the ratio of the coefficients for the y^4 terms. By the definition of $y = \sum_{k=2}^{C} a_k z_k$, every term in y^4 takes the form: $z_2^{k_2} \cdots z_C^{k_C}$ where $\sum_{i=2}^{C} k_i = 4, k_i \in \mathbb{N}$. Therefore, a combination $(k_2, \ldots, k_C)$ uniquely defines a term in y^4. For $i = 1, \ldots, 4$, we use the set $\mathcal{S}_i$ to select possible ways of choosing terms from y and x, and is defined as

$$\mathcal{S}_i = \{k', k'' : \sum_{j=2}^{C} k'_j = 4 - i,$$

$$\sum_{j=2}^{C} k''_j = i,$$

$$k'_j + k''_j = k_j,\ j = 2, \ldots, C$$

$$k'_j, k''_j \in \mathbb{N}\}.$$

Now let $A(k_2, \ldots, k_C)$ denote the coefficient of the term defined by $(k_2, \ldots, k_C)$ in the numerator and $B(k_2, \ldots, k_C)$ denote the coefficient of that term in the denominator. Plugging in $y = \sum_{k=2}^{C} a_k z_k$ and $x = \sum_{k=1}^{C} a_k z_{k+1}$ into $\tilde{R}(0, z_2, \ldots, z_C)$, we have that

$$\begin{aligned} A(k_2, \ldots, k_C) &= \sum_{i=1}^{4} a_i \Bigg[\sum_{k', k'' \in \mathcal{S}_i} \frac{(4-i)!}{k'_2! \cdots k'_C!} \Pi_{j=2}^{C} \left(\frac{C!}{(C-j)!} \right)^{k'_j} \\ &\qquad \cdot \frac{i!}{k''_2! \cdots k''_C!} \Pi_{j=2}^{C} \left(\frac{C!}{(C-j+1)!} \right)^{k''_j} \Bigg] \\ B(k_2, \ldots, k_C) &= \frac{4!}{k_2! \cdots k_C!} \Pi_{j=2}^{C} \left(\frac{C!}{(C-j)!} \right)^{k_j} + A(k_2, \ldots, k_C). \end{aligned}$$

Notice that $A(k_2, \ldots, k_C)$ is from $\sum_{i=1}^{4} a_i y^{4-i} x^i$, and $\frac{4!}{k_2! \cdots k_C!} \Pi_{j=2}^{C} \left(\frac{C!}{(C-j)!} \right)^{k_j}$ is from y^4.

Therefore,

$$\begin{aligned} \tilde{R}(z_2, \ldots, z_C) &\geq \min \frac{A(k_2, \ldots, k_C)}{B(k_2, \ldots, k_C)} \\ &= \min \frac{A(k_2, \ldots, k_C)}{\frac{4!}{k_2! \cdots k_C!} \Pi_{j=2}^{C} \left(\frac{C!}{(C-j)!} \right)^{k_j} + A(k_2, \ldots, k_C)} \\ &= \min \frac{1}{\frac{\frac{4!}{k_2! \cdots k_C!} \Pi_{j=2}^{C} \left(\frac{C!}{(C-j)!} \right)^{k_j}}{A(k_2, \ldots, k_C)} + 1} \\ &= \min \frac{1}{F(k_2, \ldots, k_C) + 1} \end{aligned}$$

where

$$F(k_2,\dots,k_C) = \frac{\frac{4!}{k_2!\cdots k_C!}\Pi_{j=2}^{C}\left(\frac{C!}{(C-j)!}\right)^{k_j}}{A(k_2,\dots,k_C)}.$$

To find the minimum of $\frac{A(k_2,\dots,k_N)}{B(k_2,\dots,k_N)}$ is equivalent to finding the maximum of $F(k_2,\dots,k_C)$.

We next show that for $C \geq 4$, $F(k_2,\dots,k_C)$ is upper bounded by $\frac{27}{104}$. First, we show $F(k_2,\dots,k_C)$ is maximized when $k_2 = 4$ and $k_i = 0, \forall i = 3,\dots,C$. This corresponds to the term z_2^4. To see this, observe that

$$\begin{aligned}
F(k_2,\dots,k_C) &= \frac{\frac{4!}{k_2!\cdots k_C!}\Pi_{j=2}^{C}\left(\frac{C!}{(C-j)!}\right)^{k_j}}{\sum_{i=1}^{4} a_i\left[\sum_{k',k''\in\mathcal{S}_i}\frac{(4-i)!}{k_2'!\cdots k_C'!}\Pi_{j=2}^{C}\left(\frac{C!}{(C-j)!}\right)^{k_j'}\frac{i!}{k_2''!\cdots k_C''!}\Pi_{j=2}^{C}\left(\frac{C!}{(C-j+1)!}\right)^{k_j''}\right]}\\
&\leq \frac{\frac{4!}{k_2!\cdots k_C!}}{\sum_{i=1}^{4}\frac{C!}{(C-i)!}\frac{1}{(C-1)^i}\sum_{k',k''\in\mathcal{S}_i}\frac{(4-i)!}{k_2'!\cdots k_C'!}\frac{i!}{k_2''!\cdots k_C''!}}\\
&= \frac{\frac{4!}{k_2!\cdots k_C!}(C-1)^4}{\sum_{i=1}^{4}\frac{C!}{(C-i)!}(C-1)^{4-i}\sum_{k',k''\in\mathcal{S}_i}\frac{(4-i)!}{k_2'!\cdots k_C'!}\frac{i!}{k_2''!\cdots k_C''!}}\\
&= \frac{(C-1)^4}{\sum_{i=1}^{4}\frac{C!}{(C-i)!}(C-1)^{4-i}\frac{\sum_{k',k''\in\mathcal{S}_i}\frac{(4-i)!}{k_2'!\cdots k_C'!}\frac{i!}{k_2''!\cdots k_C''!}}{\frac{4!}{k_2!\cdots k_C!}}}\\
&= \frac{(C-1)^4}{\sum_{i=1}^{4}\frac{C!}{(C-i)!}(C-1)^{4-i}}\\
&:= H(C).
\end{aligned}$$

The first equality is by the definition of $F(k_2,\dots,k_C)$. The first inequality holds since for any i, the maximum possible ratio of the product terms in the numerator and the denominator is $(C-1)^i$. The second equality follows by multiplying the numerator and denominator by $(C-1)^4$. The third equality follows by dividing the numerator and denominator by $\frac{4!}{k_2!\cdots k_C!}$.

The last equality holds because $\sum_{k',k''\in\mathcal{S}_i} \frac{(4-i)!}{k_2'!\cdots k_C'!}\frac{i!}{k_2''!\cdots k_C''!}$ and $\frac{4!}{k_2!\cdots k_C!}$ equivalent calculations of the same multinomial coefficient.

Next, we show that $H(C)$ is decreasing in C for $C \geq 4$. Notice that

$$\begin{aligned}
H(C+1) - H(C) &= \frac{C^4}{\sum_{i=4}^{4} \frac{(C+1)!}{(C+1-i)!} C^{4-i}} - \frac{(C-1)^4}{\sum_{i=1}^{4} \frac{C!}{(C-i)!}(C-1)^{4-i}} \\
&= \frac{C^4 \left(\sum_{i=1}^{4} \frac{C!}{(C-i)!}(C-1)^{4-i}\right) - (C-1)^4 \left(\sum_{i=4}^{4} \frac{(C+1)!}{(C+1-i)!} C^{4-i}\right)}{\left(\sum_{i=4}^{4} \frac{(C+1)!}{(C+1-i)!} C^{4-i}\right)\left(\sum_{i=1}^{4} \frac{C!}{(C-i)!}(C-1)^{4-i}\right)} \\
&= \frac{-2C(C-1)[2C(C-1)(C-2)-1]}{\left(\sum_{i=4}^{4} \frac{(C+1)!}{(C+1-i)!} C^{4-i}\right)\left(\sum_{i=1}^{4} \frac{C!}{(C-i)!}(C-1)^{4-i}\right)} \\
&\leq 0 \text{ for } C \geq 4.
\end{aligned}$$

So,

$$F(k_2, \ldots, k_N) \leq H(C) \leq H(4) = \frac{27}{104}.$$

Therefore, we have the lower bound of $\tilde{R}(0, z_2, \ldots, z_C)$ as

$$\begin{aligned}
\tilde{R}(0, z_2, \ldots, z_C) &\geq \min \frac{T(k_2, \ldots, k_C)}{B(k_2, \ldots, k_C)} \\
&= \frac{1}{\max F(k_2, \ldots, k_C) + 1} \\
&\geq \frac{1}{H(4) + 1} \\
&= \frac{1}{\frac{27}{104} + 1} \\
&= \frac{104}{131}.
\end{aligned}$$

□

Proof of Lemma 1.3.3. We directly calculate the lower bound of $\tilde{R}(z_1, \dots, z_C)$. In this case, one can show that $z_C \geq C-1$ and $z_1 \geq (C-1)^{C-1}$, so the z_1^C dominates the rest of terms in $\tilde{R}(z_1, \dots, z_C)$. Since the coefficient of z_1^C is the same in the numerator and denominator, one can expect, in this case, $\tilde{R}(z_1, \dots, z_C)$ to be close to 1.

First note that we have

$$y = \sum_{k=2}^{C} a_k z_k \leq (C-1) \sum_{k=2}^{C} a_{k-1} z_k \leq (C-1)x.$$

Then

$$\begin{aligned}
&\tilde{R}(z_1, \dots, z_C) \\
&= \frac{a_1(a_1z_1+y)^2(a_0z_1+x) + a_2(a_1z_1+y)(a_0z_1+x)x + a_3(a_0z_1+x)x^2}{a_0(a_1z_1+y)^3 + a_1(a_1z_1+y)^2x + a_2(a_1z_1+y)x^2 + a_3x^3} \\
&= \frac{(Cz_1+y)^2(Cz_1+Cx) + (C-1)(Cz_1+y)(Cz_1+Cx)x + (C-1)(C-2)(Cz_1+Cx)x^2}{(Cz_1+y)^3 + C(Cz_1+y)^2y + C(C-1)(Cz_1+y)x^2 + C(C-1)(C-2)x^3} \\
&\geq \frac{A + (2C^3-C^2)z_1x^2 + (3C^2-C)z_1xy + C(C-1)(C-2)z_1x^2}{A + y^3 + C^3z_1^2x + 2C^2z_1xy + C^2z_1^2y + 2Cz_1y^2} \\
&\geq \frac{7y^3 + (2C^3-C^2)z_1^2x + (3C^2-C)z_1xy + [C^2(C-1) + C(C-1)(C-2)]z_1x^2}{8y^3 + (2C^3-C^2)z_1^2x + (3C^2-C)z_1xy + [C^2(C-1) + C(C-1)^2]z_1x^2} \\
&\geq \min\{\frac{7}{8}, \frac{2C-2}{2C-1}\} \\
&= \frac{6}{7}
\end{aligned}$$

where

$$A = Cz_1(Cz_1+y)^2 + C(C-1)(Cz_1+y)x^2 + Cxy^2.$$

The first inequality comes from dropping $C(C-1)(C-2)x^3$ in both the numerator and denominator. The second inequality follows from the facts that $A \geq 7y^3$ and $C^2z_1^2y + 2Cz_1y^2 \leq C^2(C-1)z_1^2x + C(C-1)z_1xy + C(C-1)^2z_1x^2$, since $y \leq a_1z_1 = Cz_1$ and $y \leq (C-1)x \leq Cx$. The last inequality follows by the assumption that $C \geq 4$. $\square$

Proof of Lemma 1.4.1. We derive the optimal condition of the objective function to bound λ_1 and λ_2. Recall that our objective function in the case of $C = 2$ is

$$\begin{aligned} &\max\ \lambda_1(p(\lambda_1)-c)\mathbb{P}_1 + \lambda_2(p(\lambda_2)-c)\mathbb{P}_2 \\ =&\max\ \frac{2\mu\lambda_1\lambda_2(p(\lambda_1)-c) + 2\mu^2\lambda_2(p(\lambda_2)-c)}{\lambda_1\lambda_2 + 2\mu\lambda_2 + 2\mu^2} := f(\lambda_1,\lambda_2) \end{aligned} \tag{1.21}$$

Denote $\gamma_i = -p'(\lambda_i)$. Notice that if $\lambda = -ap + b$ is linear, then $\gamma_1 = \gamma_2 = \frac{1}{a}$. Taking derivative of $f(\lambda_1, \lambda_2)$ w.r.t λ_1, λ_2 and set those to zero yields

$$\begin{aligned} \frac{\partial f}{\partial \lambda_1} &= 0 \Rightarrow 2\gamma_1\frac{\lambda_2}{\mu}\left(\frac{\lambda_1}{\mu}\right)^2 + 2\gamma_1(2\frac{\lambda_2}{\mu}+2)\frac{\lambda_1}{\mu} - (2(p_1-c)(2\frac{\lambda_2}{\mu}+2) - 2\frac{\lambda_2}{\mu}(p(\lambda_2)-c)) = 0 \\ \frac{\partial f}{\partial \lambda_2} &= 0 \Rightarrow 2\gamma_2(\frac{\lambda_1}{\mu}+2)\left(\frac{\lambda_2}{\mu}\right)^2 + 4\gamma_2\frac{\lambda_2}{\mu} - 2(2(p(\lambda_1)-c)\frac{\lambda_1}{\mu} + 2(p(\lambda_2)-c)) = 0. \end{aligned}$$

Therefore, the optimal $\frac{\lambda_i^*}{\mu}$'s take the form of

$$\frac{\lambda_1^*}{\mu} = \frac{\sqrt{[\gamma_1(\frac{\lambda_2^*}{\mu}+1)]^2 + \gamma_1\frac{\lambda_2^*}{\mu}((p(\lambda_1^*)-c)(2\frac{\lambda_2^*}{\mu}+2) - \frac{\lambda_2^*}{\mu}(p(\lambda_2^*)-c))} - \gamma_1(\frac{\lambda_2^*}{\mu}+1)}{\gamma_1\frac{\lambda_2^*}{\mu}} \tag{1.22}$$

$$\frac{\lambda_2^*}{\mu} = \frac{\sqrt{4\gamma_2^2 + 4\gamma_2(\frac{\lambda_1^*}{\mu}+2)(2(p(\lambda_1^*)-c)\frac{\lambda_1^*}{\mu} + 2(p(\lambda_2^*)-c))} - 2\gamma_2}{2\gamma_2(\frac{\lambda_1^*}{\mu}+2)} \tag{1.23}$$

Notice that by definition, $z_1 = \frac{\lambda_1^*}{\mu}\frac{\lambda_2^*}{\mu}$, $z_2 = \frac{\lambda_2^*}{\mu}$, and $\beta = \frac{p(\lambda_1^*)-c}{\gamma_1} = \frac{p(\lambda_1^*)-c}{\gamma_2}$. Therefore, we have

$$\begin{aligned}
z_1 &= \frac{\sqrt{[\gamma_1(z_2+1)]^2 + \gamma_1 z_2((p(\lambda_1^*)-c)(2z_2+2) - z_2(p(\lambda_2^*)-c))} - \gamma_1(z_2+1)}{\gamma_1} \\
&\geq \frac{\sqrt{[\gamma_1(z_2+1)]^2 + \gamma_1 z_2(p(\lambda_1^*)-c)(z_2+2)} - \gamma_1(z_2+1)}{\gamma_1} \\
&= \sqrt{(z_2+1)^2 + \beta z_2(z_2+2)} - (z_2+1)
\end{aligned}$$

and

$$\begin{aligned}
z_2 &= \frac{\sqrt{4\gamma_2^2 + 4\gamma_2(\frac{\lambda_1^*}{\mu}+2)(2(p(\lambda_1^*)-c)\frac{\lambda_1^*}{\mu} + 2(p(\lambda_2^*)-c))} - 2\gamma_2}{2\gamma_2(\frac{\lambda_1^*}{\mu}+2)} \\
&\leq \frac{\sqrt{4\gamma_2(\frac{\lambda_1^*}{\mu}+2)(2(p(\lambda_1^*)-c)\frac{\lambda_1^*}{\mu} + 2(p(\lambda_2^*)-c))}}{2\gamma_2(\frac{\lambda_1^*}{\mu}+2)} \\
&= \sqrt{\frac{2(p(\lambda_1^*)-c)\frac{\lambda_1^*}{\mu} + 2(p(\lambda_2^*)-c)}{\gamma_2(\frac{\lambda_1^*}{\mu}+2)}} \\
&\leq \sqrt{\frac{2(p(\lambda_1^*)-c)(\frac{\lambda_1^*}{\mu}+1)}{\gamma_2(\frac{\lambda_1^*}{\mu}+2)}} \leq \sqrt{\frac{2(p(\lambda_1^*)-c)}{\gamma_2}} = \sqrt{2\beta}.
\end{aligned}$$

$\square$

Proof of Lemma 1.4.2. We show in this case $G(\beta, z_2)$ is nondecreasing in z_2 so that we can plug in the upper bound of z_2 to find the maximum of $G(\beta, z_2)$. Letting $A := \sqrt{(1+\beta)z_2^2 + 2(1+\beta)z_2 + 1}$, then the numerator of $\frac{\partial G(\beta,z_2)}{\partial z_2}$ equals

$$\frac{z_2^3(3+4\beta+\beta^2) + z_2^2(5+3\beta^2+3A+3\beta(2+A)) + 2z_2(1+\beta^2+A+\beta A) - 2\beta A}{A}. \quad (1.24)$$

In the case that $z_2 \geq \frac{\sqrt{7}-1}{3}$, Equation (1.24) is guaranteed to be non-negative since the coefficient of βA equals $3z_2^2 + 2z_2 - 2$ which is non-negative. Therefore, we can plug in the upper bound of z_2 to maximize $G(\beta, z_2)$. By Equation (1.14) in Lemma 1.4.1, we have

$$z_2 \leq \sqrt{2\beta}.$$

Therefore,

$$G(\beta, z_2) \leq \frac{2\beta + \sqrt{2\beta} + 1 - \sqrt{(1+\beta)2\beta + 2(1+\beta)\sqrt{2\beta} + 1}}{(3+\beta)2\beta + (2\beta+4)\sqrt{2\beta} + 2\sqrt{2\beta}\sqrt{(1+\beta)2\beta + 2(1+\beta)\sqrt{2\beta} + 1} + 2} := h(\beta).$$

Next, we find the maximum value of $h(\beta)$ by looking at the first order condition. Setting $h'(\beta) = 0$ yields the following equation,

$$5\sqrt{2\beta} + 3\sqrt{2}\beta^{5/2} + 4\beta^3 - 4\beta^2 B + \beta(2 - 4B) + 2(1+B) - \sqrt{2}\beta^{3/2}(2+3B) = 0$$

where

$$B = \sqrt{1 + 2(\sqrt{2} + \sqrt{\beta})\sqrt{\beta}(1+\beta)}.$$

Let $\theta = \sqrt{\beta}$, then we have to solve the following,

$$4\theta^6 + 3\sqrt{2}\theta^5 - 2\sqrt{2}\theta^3 + 2\theta^2 + 5\sqrt{2}\theta + 2 = (4\theta^4 + 4\theta^2 + 3\sqrt{2}\theta - 2)\sqrt{1 + 2(\sqrt{2}+\theta)\theta(1+\theta^2)}. \quad (1.25)$$

Squaring both sides gives the following polynomial,

$$16\theta^{12} + 56\sqrt{2}\theta^{11} + 210\theta^{10} + 244\sqrt{2}\theta^9 + 316\theta^8 + 96\sqrt{2}\theta^7$$

$$- 18\theta^6 - 36\sqrt{2}\theta^5 - 36\theta^4 - 48\sqrt{2}\theta^3 - 66\theta^2 - 12\sqrt{2}\theta = 0.$$

The twelve roots to above equation are

$$\begin{aligned}\theta =&\{-1.59237, -0.951779 \pm 0.164422i, -0.750502 \pm 1.74268i,\\ &- 0.547073 \pm 0.940637i, -0.401417, 0, 0.356881 \pm 0.649577i, 0.768987\}.\end{aligned}$$

Since $\beta \geq 0$, then $\theta = \sqrt{\beta} \geq 0$. Therefore, only $\theta = 0$ and $\theta = 0.768987$ can be the only real valued solutions. Notice that $\theta = 0$ is not the solution to Equation (1.25), therefore $\theta^* = 0.768987$ is the unique real solution to Equation (1.25). The corresponding $\beta^* \approx 0.591341$.

Since $h'(0.1) \approx 0.13 > 0$ and $h'(1) \approx -0.007 < 0$, then $h(\beta)$ is increasing in $[0, \beta^*]$ and decreasing in $[\beta^*, \infty]$, therefore, $\beta^* \approx 0.59341$ maximizes $h(\beta)$ with the maximum value approximately 0.0433. $\square$

Chapter 2

Pricing Analytics for Rotable Spare Parts

In this chapter, we describe a comprehensive approach to pricing analytics for reusable resources in the context of rotable spare parts, which are parts that can be repeatedly repaired and resold. Working in collaboration with a major aircraft manufacturer, we aim to instill a new pricing culture and develop a scalable new pricing methodology. Pricing rotable spare parts presents unique challenges ranging from limited data availability, minimal demand information, and complex inventory dynamics. We develop a novel pricing analytics approach that tackles all of these challenges and that can be applied across all rotable spare parts. We then describe a large-scale implementation of our approach with our industrial partner, which led to an improvement in profits of over 3.9% over a ten month period.

2.1 Introduction

In this work, we focus on the engineering and implementation of a systematic pricing approach for thousands of rotable spare parts at a major aircraft original equipment manufacturer (OEM). (For confidentiality reasons, we shall strictly refer to our industrial partner as 'the OEM'.) In addition to manufacturing, a large part of the business is ensuring a high quality of customer service, which includes the ability to provide spare parts for all operating aircraft. (Note that aircraft is both singular and plural.) The management of spare parts is increasingly critical as the age of the fleet increases and the number of aircraft no longer under warranty increases. Due to the increasing opportunities for spare part sales, there has been increased availability of spare parts by competitor service providers. Thus, pricing spare parts optimally, while maintaining a high level of service, has become progressively more challenging and paramount over the years.

The OEM faces the challenge of pricing thousands of different spare parts. Such pricing decisions present unique challenges, especially for an important subclass of parts called *rotable spare parts*, for which the selling process is quite different than the regular spare parts. Rotable spare parts, which is the subject of this work, represent a majority of spare parts sales, with price tags of up to hundreds of thousands of dollars per unit. When purchasing a rotable spare part, a customer will give their broken unit to the OEM in exchange for a functional unit. This swap of a broken part for the functional part is known as an *exchange sale*. The OEM will then send the broken unit to a repair facility. When the broken part is repaired and sent back, it is then placed back into the OEM's inventory and can be sold again. We remark that rotable spare parts are an example of a reusable resource, which are a subject of increased attention due to the increasing popularity of ride sharing and cloud computing systems.

One key characteristic in selling rotable spare parts in that the total number of units is fixed for the OEM. These units are divided into two groups: the on-hand units which are available for sale and the in-repair units which are not available. Moreover, since there are many competitors in the market, when a customer asks for a rotable spare part but the OEM has no available units, then the sale will almost certainly be lost to a competitor. Note that substituting one part number for another is not physically possible. Although dynamic pricing policies are natural to consider and were our original intent, i.e., a policy where the price changes depending on the number of available units, we actually utilize a static pricing policy as it is near-optimal empirically and theoretically. This stems from the work we have established in Chapter 1.

The objective of this project is to redesign and improve the pricing process while utilizing all available data, and ultimately provide a systematic and scalable approach to pricing in this context. In collaboration with experts from the aircraft OEM, we developed a novel and systematic approach that leverages existing data and captures the special features of rotable spare parts to derive the best pricing strategy to maximize profit. The existing approach, prior to our work, to pricing rotable spare parts was driven only by the repair cost, and did not factor in repair time, competition, and the special inventory dynamics. More broadly, the project is also one that aims at changing the conversation about pricing within the organization, potentially uncovering scope for systematic approaches to pricing of other offerings by the firm.

2.1.1 Unique challenges

Our engineering approach for the price optimization tool we developed dealt with the following challenges unique to rotable spare parts:

1. *Modeling the special inventory dynamics and market competition.*

   The rotable selling process has its own unique inventory dynamics, where the inventory constraints, stochasticity in repair time, as well as market competition, all play critical roles. Capturing this information and synthesizing how it should impact price decisions is key for a successful model.

2. *No knowledge on price-demand relationship.*

   The OEM, in accordance with common practice in the aircraft industry, does not often change the prices of rotable spare parts. This, in turn, provides very little information on the price sensitivity of customers and more broadly the relationship between expected demand and price.

3. *Limited data for each rotable spare part.*

   Although in the aggregated level, the amount of available data is reasonably large, the data associated to each rotable spare part is very limited due to the slow moving nature of the system. Most rotable spare parts are highly expensive parts with lifetimes of many years. A typical rotable spare part may be purchased only two or three times per year, and thus parameter estimates are inherently noisy.

2.1.2 Timeline and general approach

This project started in the Summer of 2016, culminating with a large-scale controlled controlled experiment ending in Spring 2019. Figure 2.1 depicts a timeline of the project, highlighting some key steps.

It is important to note that an initial pilot program was conducted to obtain buy-in within the organization on the potential for adjusting prices, and specifically ensuring that the market is indeed price-sensitive. With this successful pilot and early buy-in on the potential, significant effort was put in developing a tailored mathematical model that addressed all the business needs. In turn, the challenge resided in estimating input for the model when possible

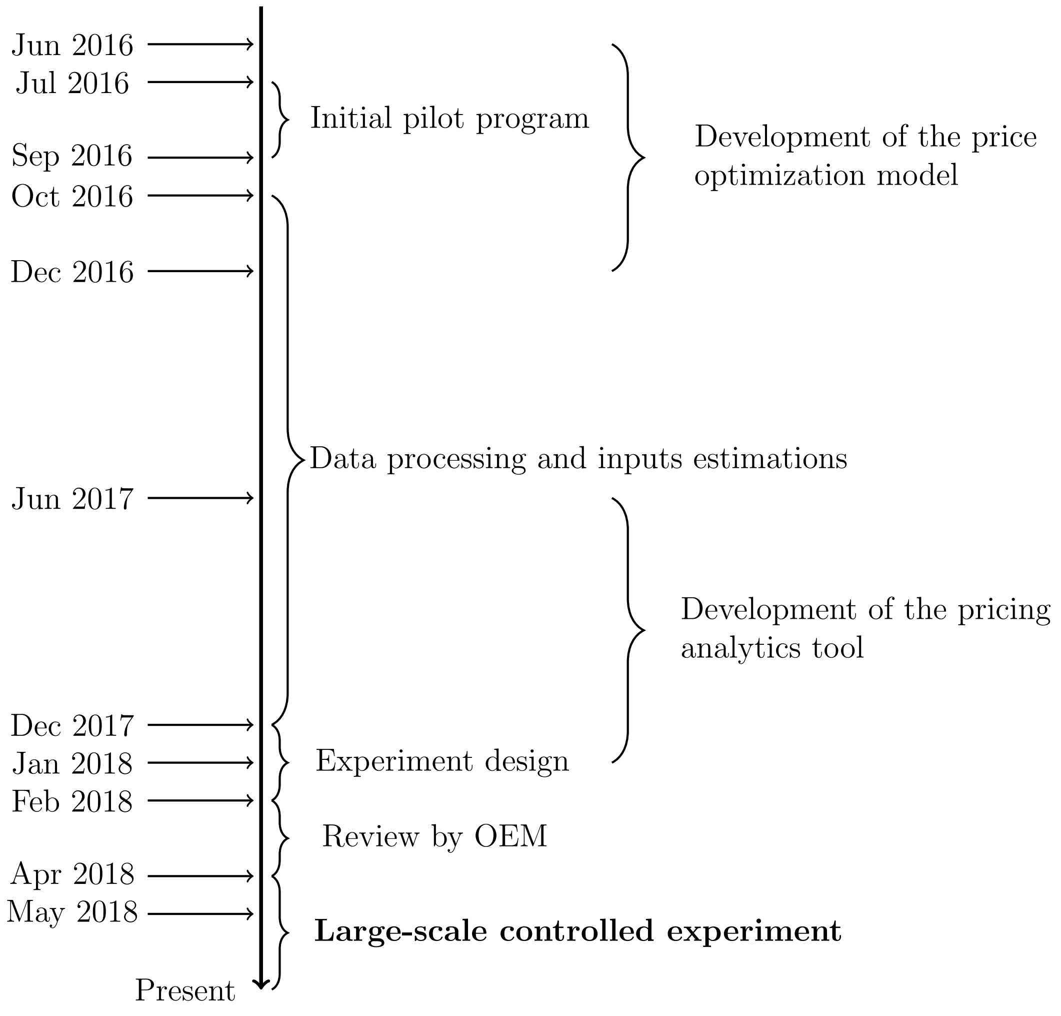

Jun 2016
Jul 2016
Sep 2016
Oct 2016
Dec 2016
Jun 2017
Dec 2017
Jan 2018
Feb 2018
Apr 2018
May 2018
Present
Initial pilot program
Development of the price optimization model
Data processing and inputs estimations
Development of the pricing analytics tool
Experiment design
Review by OEM
Large-scale controlled experiment

Figure 2.1: Timeline of the project

or proxies when it was not directly possible. This involved a big effort in aggregating various disparate data sources living on different systems, cleaning the data, and quantifying the residual uncertainty on the inputs needed. Finally, we developed a pricing analytics tool that allowed users to get default price suggestions based on the available data. In Figure 2.2, we present an overview of the data-driven systematic approach we developed in the project.

We developed a model for rotable spare parts that can be used to understand the profit rate as a function of the total number of units, demand rate, repair time, and prices. To find the optimal prices, we estimate the inputs of the model using various datasets from the

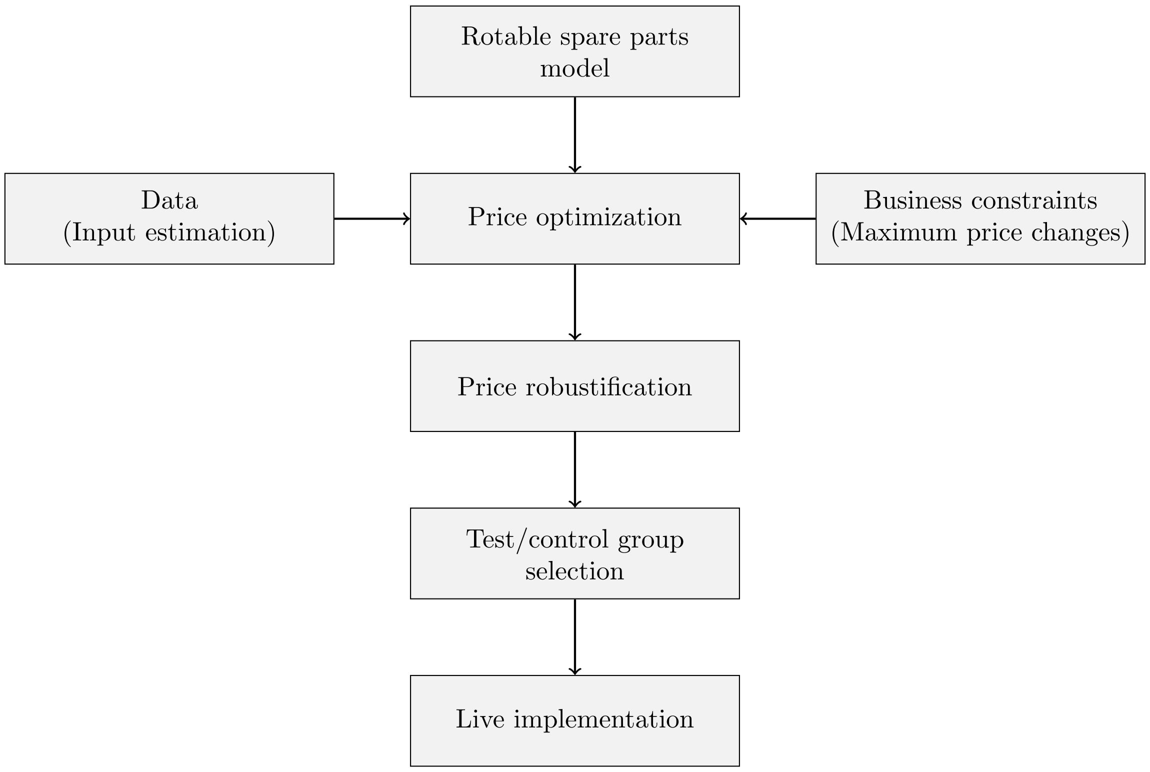

Rotable spare parts model
Data (Input estimation)
Price optimization
Business constraints (Maximum price changes)
Price robustification
Test/control group selection
Live implementation

Figure 2.2: Data-driven pricing approach

OEM and factor in other business constraints on the maximum price changes. Once the optimal prices are found on the estimated model, the prices are then robustified in the sense that we potentially change the price so that we ensure that it is robust to the uncertainty in all the parameter estimates. We highlight the importance of the robust approach to ensure that prices are not overly sensitive to our assumptions and the high level of uncertainty in some inputs. A key feature of the decision support tool we developed is that the data and the uncertainty associated with some inputs (and their implications) can be overridden by the users, allowing users to challenge their own assumptions or potentially refine the inputs with knowledge not codified in the existing databases.

After review and testing of the tool, together with the OEM, we decided to launch a large-scale controlled experiment to test the suggested price changes. We use the robustified price to guide how we split the rotable spare parts into the control and test groups for the implementation. Based on 10 months, we evaluated that the new system lead to an increase of 3.9% in profits under a difference-in-difference analysis. Equally importantly, this project has led to multiple other initiatives around pricing analytics within the organization.

2.1.3 Related literature

There has been a long history in research on inventory management of rotable spare parts dating back to Allen and D'Esopo (1968). Since then, many studies have been conducted in inventory management of repairable parts such as Graves (1985) and Cohen et al. (1989) who studied the problem of determining optimal inventory levels. Guide Jr and Srivastava (1997) provided a review of models and applications of repairable inventory. More recently, several studies focused on aircraft spare parts. Simao and Powell (2009) used approximate dynamic programming to determine the inventory level of aircraft spare parts at each warehouse while Aisyati et al. (2013) studied the inventory policy using a continuous review model. Muckstadt (2004) provides a comprehensive overview of modeling approaches and solution methodologies for addressing service parts inventory problems.

At a high level, the system dynamics of rotable spare parts could be considered as a closed-loop supply chain. There are many works have been done in this broad domain such as Fleischmann et al. (2003), Savaskan et al. (2004), Guide Jr and Van Wassenhove (2009), Calmon and Graves (2017). Another stream of works focuses on allocation and overhaul planning of rotable spare parts such as Tedone (1989), Arts and Flapper (2015), Erkoc and Ertogral (2016).

Relatively fewer works focus on finding best pricing strategies in selling rotable spare parts, typically in the context of reusable resources. Gans and Savin (2007) study dynamic pricing to maximize the expected profit for rentals. Their model considers discounted rewards with a discrete price ladder, although with multiple customer types. They show the near-optimality of static pricing in highly utilized rental systems where both the offered load and system capacity are large. In Chapter 1, we showed a static pricing policy is provably near-optimal in all parameter regimes and such results hold even when the number of units is small, which is the case for rotable spare parts that we examine. Lei and Jasin (2018) studied a related pricing problem of reusable resources where the service time is deterministic.

We remark that there has been a limited but steady stream of literature on successful pricing model implementations such as Smith and Achabal (1998), Natter et al. (2007), Caro and Gallien (2012), Ferreira et al. (2015), Simchi-Levi and Wu (2018). These implementations are typically for in fast-moving industries such as fashion or online retail where there is a wealth of data. However, to the best of our knowledge, this is the first work on the implementation of a pricing model in a slow-moving environment, and the first such work in the context of reusable resources.

2.2 Rotable Spare Parts Pricing

2.2.1 Model and assumptions

We begin by describing a model that captures the expected profit rate of rotable spare part as a function of its price and inventory dynamics. We note that this same model shall be applied to each (of the thousands) rotable spare part separately. For each rotable spare part, the total number of units is fixed. We let C denote the total number of units, which is also referred to as the pool size. Customer requests are assumed to arrive to the OEM according to a Poisson process with rate $\Lambda > 0$. Given the current price, p, of the rotable spare part, a customer decides to purchase the rotable spare part if their willingness-to-pay exceeds p. We denote by $\lambda(p)$ the effective arrival rate at price p, which we shall later fit to be a decreasing, linear function. When a customer decides to purchase a unit of the rotable spare part, they will give their broken unit back to the OEM in the exchange sale. The OEM will then send the part for repair, and incur an associated expected repair cost c. The repair time (including travel time) of the rotable spare part is assumed to be exponentially distributed with mean μ^{-1} periods. We note that both interarrival times and repair times are each generated from independent and identically distributed processes.

Since both the interarrival and repair times are exponentially distributed, the rotable selling process can be modeled as a Markov decision process (MDP) where the state is one-dimensional. Specifically, the state is one of $\{0, 1, \ldots, C\}$, which represents the number of unit that the OEM currently has on-hand (available for sale). The transition rate of having i units on-hand to $i + 1$ units on-hand thus is $(C - i)\mu$ because there are $(C - i)$ units repair, each with an i.i.d. repair time with mean $\frac{1}{\mu}$. On the other hand, the transition rate of having i units on-hand to $i - 1$ units on-hand is simply $\lambda(p)$. Figure 2.3 illustrates the Markov chain embedding of our model. The numbers in the circle represent the number of on-hand units currently in the system.

The goal is to find the optimal price of the rotable spare part to maximize the expected profit rate, i.e., the product of the arrival rate, profit per unit sold, and availability (steady-

$$\begin{tikzcd}
0 \arrow[r, bend left, "C\mu"] & 1 \arrow[l, bend left, "\lambda(p)"] \arrow[r, bend left, "(C-1)\mu"] & 2 \arrow[l, bend left, "\lambda(p)"] \arrow[r, bend left, "(C-2)\mu"] & 3 \arrow[l, bend left, "\lambda(p)"] \arrow[r, bend left, "(C-3)\mu"] & \cdots \arrow[l, bend left, "\lambda(p)"] \arrow[r, bend left, "\mu"] & C \arrow[l, bend left, "\lambda(p)"]
\end{tikzcd}$$

Figure 2.3: Underlying Markov chain

state probability of having at least one unit to sell). Letting $\mathbb{P}_0(p)$ denote the steady state probability of having zero units available (stock out probability), then our objective to maximize the expected profit rate can be written as

$$\max_{p} \ \lambda(p)(p-c)(1-\mathbb{P}_0(p)), \tag{2.1}$$

where the stock-out probability $\mathbb{P}_0(p)$ can be expressed as

$$\mathbb{P}_0(p) = \frac{(\frac{\lambda(p)}{\mu})^C}{\sum_{i=0}^{C} \frac{C!}{(C-i)!}(\frac{\lambda(p)}{\mu})^{C-i}}.$$

Since the demand rate $\lambda(p)$ is assumed to be decreasing in p, the profit of selling one unit, $p-c$, is increasing in p, and the stock out probability $\mathbb{P}_0(p)$ is decreasing in p, there is a non-trivial trade off between making more profit per sale, the rate of selling, and the availability.

In this model, we capture the competition in the market through the term $\lambda(\cdot)$, while the inventory dynamics including the total number of units and repair time of a rotable spare part is captured in the stock out probability $\mathbb{P}_0(\cdot)$. These key inputs were not captured by the previous 'cost+margin' method employed in the past.

Although in principle a dynamic pricing policy is optimal for this MDP, and was the original intent of the project, we instead relied on a static pricing policy for two reasons. First, from a performance perspective, numerical tests showed that the best static price loses at most 2.5% compared to dynamic pricing and theoretically Chapter 1 proved the near-optimality of a static price in such systems. From the data, 2.5% is generally smaller than the average error of our estimated input parameters. Second, from a practical perspective, a static pricing policy allows the OEM to keep its current practice of publishing a catalog of prices for rotable spare parts at the beginning of each year and maintaining the same price throughout the year. There will be no need to develop a new system for deploying the new pricing algorithm.

2.2.2 Justification of assumptions.

In deriving the price optimization model, we made two key assumptions which we seek to justify: 1) customers arrive according to a Poisson process and 2) repair times are random, and specifically exponentially distributed. Throughout the rest of the chapter, we will use two generic examples of rotable spare parts, a sensor and a jack, to illustrate our ideas.

Since the system of selling rotable spare parts is very slow-moving, many rotable spare parts have very limited sales during the year. Figure 2.4 depicts the average sales per year of each rotable spare part from 2010 to 2017. One observes that there are significant differences across spare parts in terms of volumes but also one notes here the very slow-moving nature of the environment. The majority of rotable spare parts have less than 3 units sold per year, in which case estimating the repair and interarrival times is naturally very noisy.

To gain some intuition on the distribution of interarrival and repair times, we focus

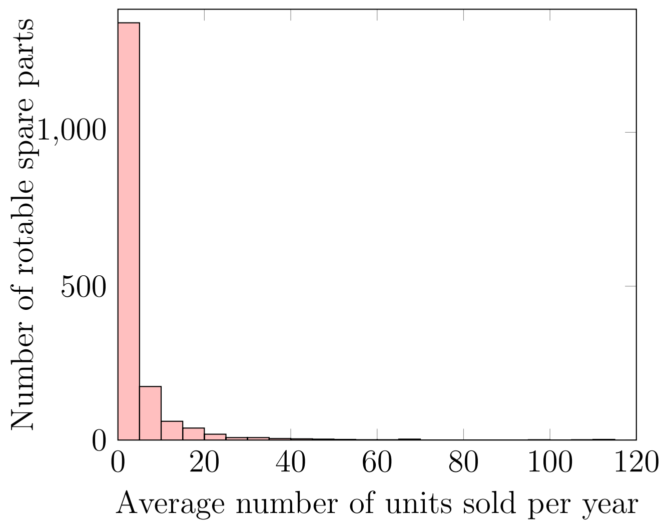
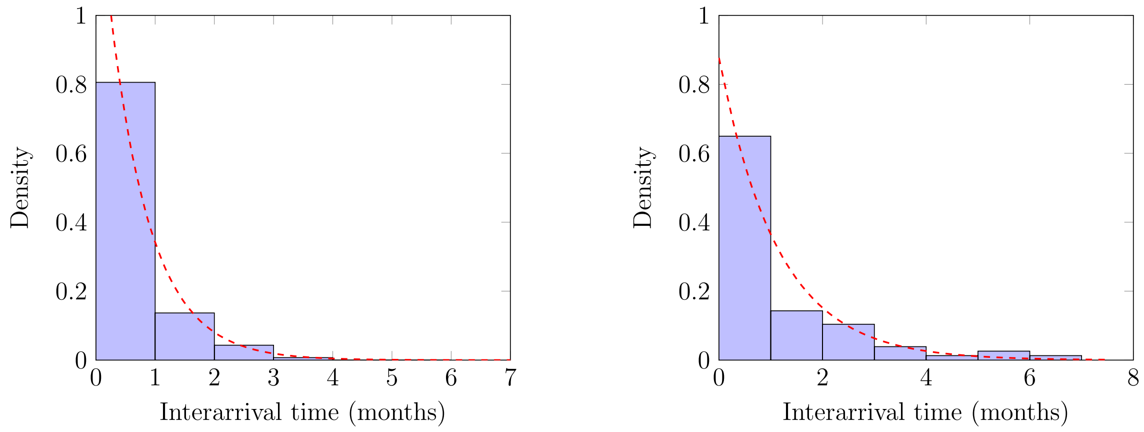


Figure 2.4: Histogram of sales volume

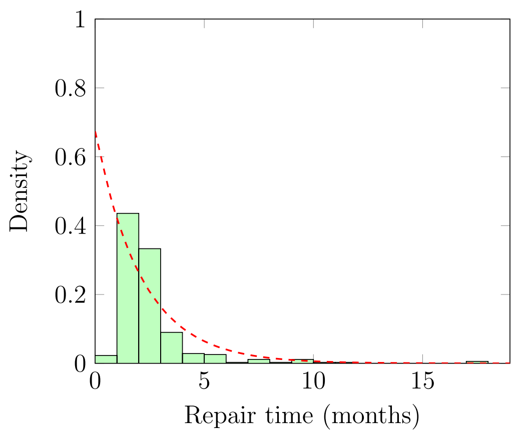
on rotable spare parts with higher volumes. For instance, Figures 2.5 shows the empirical distributions for the sensor and jack. Similarly, we look at the empirical distributions of the repair times for the sensor and jack in Figure 2.6. Though the price of the parts may be changed by the OEM at the beginning of the year by at most 3% due to inflation, we ignore such changes when calculating the interarrival time.



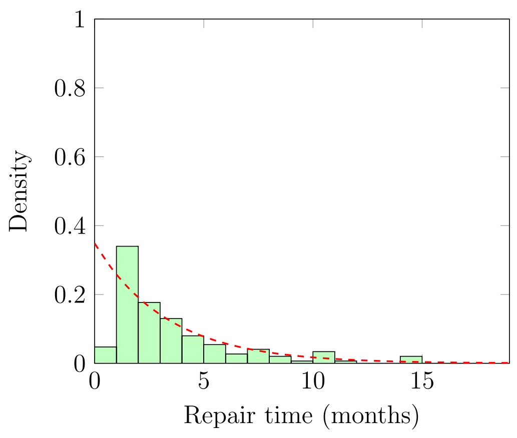
(a) Sensor (b) Jack

Figure 2.5: Typical histograms of interarrival times

In Figures 2.5 and 2.6, we also depict the best-fit exponential curves (red dashed line)



(a) Sensor



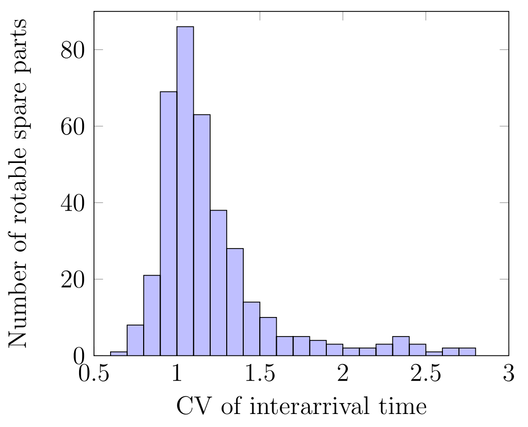
(b) Jack

Figure 2.6: Typical histograms of repair times

to the data. The data suggests that both interarrival and repair times are inherently quite random, and exponential fits appear to be reasonable. Admittedly, for the repair time a better fit might be a deterministic time plus an exponential time, although this type of random variable would impose significant technical challenges. The exponential fit, while not perfect, captures most of the shape of the empirical distribution.

Moreover, the behavior above is fairly typical across other rotable spare parts with volume of at least 3 units per year. To demonstrate this, we compute the coefficient of variation of the interarrival and repair times for each rotable spare part. In Figure 2.7, we report the empirical coefficient of variations (CV) of interarrival and repair times for all spare parts with more than 3 units sold per year.

A large proportion of the parts have coefficients of variation around 1 for both interarrival and repair times, in line with the assumed variation based on the exponential assumption. Moreover, the minimum CV we find is significantly far from zero, motivating the use of random times in our model. We shall later account for imperfections in the model by adding a robust component to our pricing algorithm. While the data is not perfectly in line with



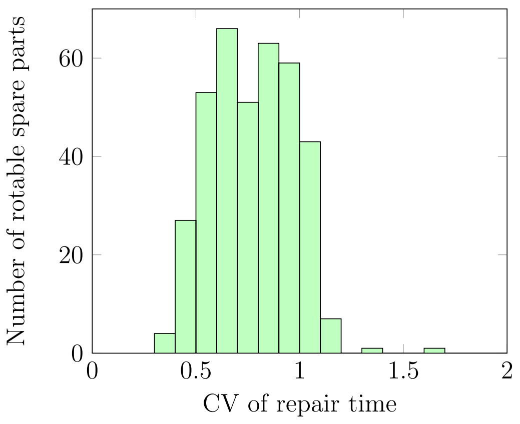


Figure 2.7: Histograms of the coefficient of variation of the interarrival and repair times

the exponential assumption (which would lead to an exact CV value of 1), the exponential assumption captures the randomness associated with repair times while allowing us to have a tractable function to optimize.

2.2.3 Price sensitivity of customers

One question initially raised by our industry partner was how price-sensitive customers were, and whether there was significant potential for price optimization. To test this, and before engaging fully in a revamp of the pricing process, we conducted a simple pilot program on a limited group of rotable spare parts in the summer of 2016. The approach in the pilot program was to discount parts with low sales and low risk of backorders so as to not to disturb the existing supply chain. To do so, we selected the parts and discount levels according to a decision tree as shown in Figure 2.8.

We changed prices of rotable spare parts in three leaves of the decision tree. We started the pilot program on July 12th, 2016 by reducing the price on 182 parts while another 95 parts belonging to the same leaves of the decision tree were in the control group. We

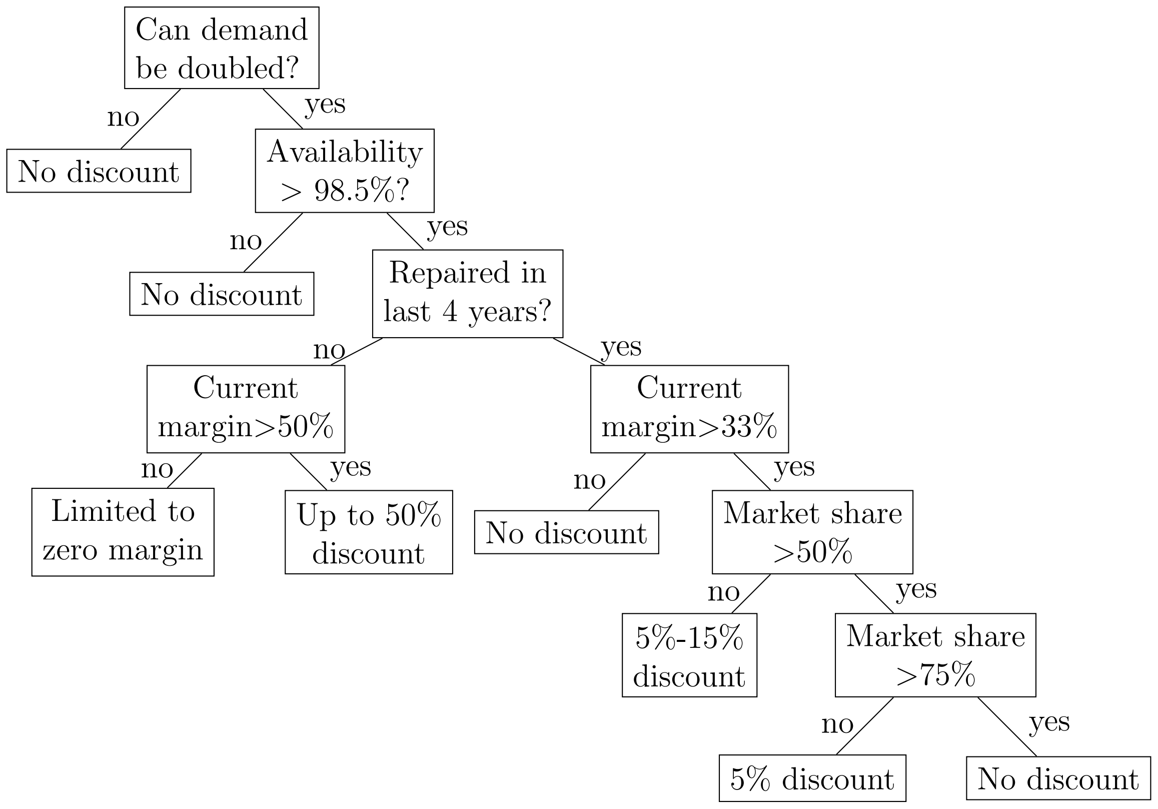


Figure 2.8: Decision tree of selecting discounts for rotable spare parts.

considered 77 days before and after the start date of the pilot program and performed a difference-in-difference (DiD) analysis. Our analysis suggested an overall increase in profit of 17% and an estimated increase in sales volume of over 44%.

Although the pilot was relatively short and the parts included in the pilot program represent only a small portion of rotable spare parts, mainly the parts with very limited sales, the results of the pilot program indicated that there is significant potential for optimizing prices. This provided some evidence that customers are indeed conscious of price changes in the rotable selling market and that that this is an environment where data-driven pricing can be leveraged to optimize prices. Furthermore, the pilot program convinced our industry partner of the need to systematize their pricing approach, and they decided to launch our pricing analytics approach across the entire rotable spare parts supply chain. Next, we detail how key inputs were estimated for our model.

2.3 Input estimation

The estimation of inputs required extracting, aggregating, unifying, and cleaning data across multiple databases at the OEM. Moreover, estimating the demand function $\lambda(p)$ was a particularly challenging task due to the fact that the OEM rarely changes prices of their parts beyond inflation adjustments.

2.3.1 Pool size

The OEM identifies rotable parts at two levels, *individual* and *family.* An individual part has its unique part number while a family may contain several individual part numbers if they have different generations or are symmetric parts (one is applied to the left of the airplane and the other to the right). Given various business constraints at the family level, and the need to aggregate data for very slow moving parts, our analysis is performed at the family level. Aggregating all data sources, we constructed the mapping of individual part numbers to their family part number, applicability (models of airplanes that the family can be used), and the description of each family. Aggregating data from on-hand and in-repair inventory for each individual part, and leveraging the family-individual mapping above, we obtain the total number of units in each family. This number corresponds to C in our price optimization model. For our two running examples, we have 10 units of the sensor and 11 units of the jack.

2.3.2 Repair time and cost

We collected the records of repair orders for the last 10 years. Using the mapping of part number to family number, we calculated the average repair time (also called Total Turn Around Time (TTAT)) and repair cost for each rotable spare part family. These estimates correspond to the parameters $1/\mu$ and c in our price optimization model. In addition, we also calculated the standard deviation of these two quantities which is used later when we robustify our optimized prices. The average repair time of the sensor is 2.88 months with a standard deviation of 2.92 months, while the average repair time of the jack is 3.79 months with a standard deviation of about 3.34 months. The costs for these examples are not reported for confidentiality reasons.

2.3.3 Price-demand relationship

The estimation of inputs above is fairly straightforward and the main difficulty is dealing with missing data, outliers, and accounting for the remaining uncertainty (that we deal with by "robustifying" our prices later on). However, a key input for which no estimate is available in the data is the relationship between price and demand, i.e., the function $\lambda(p)$. Estimating a demand curve is challenging in general, but even more so in a setting with almost no price experimentation. (A price experiment would take years due to the slow-moving nature of rotable sales).

The given data only provides one point on the demand curve, which is the current price and the corresponding current demand rate, For example, for the sensor, the current demand rate is about 1.5 sales per month at its current price, and for the jack, the current price has a demand rate of about 0.8 units per month. Quite notably, there is no counterfactual understanding of price changes. This presents a unique challenge. *Can one still approach pricing in a systematic fashion without a demand-curve?* Is there a proxy for such a demand curve despite the structure above? We develop below a systematic approach in this environment. This should be seen as providing a starting point in estimating the demand curve and we will discuss how we deal with the remaining uncertainty later when we "robustify" the suggested prices.

The approach we take is one of assuming a linear structure of demand and attempting to obtain a proxy for demand at an alternate point, a hypothetical price where the OEM would be able to capture full market share Λ (or close to it). Using the current price point and this alternate point, we simply fit a line between these two points to generate $\lambda(p)$. Figure 2.9 illustrates the proposed demand model. With such an approach, the question becomes one of 1) estimating a price at which one would capture full market share and 2) estimating the total demand, Λ in the market for any part. After discussions across the firm with experts, we make the assumption that the firm can get full market share if the price is set to the average repair cost or half of the current price, whichever is higher.

Since we have already estimated the current demand rate of each rotable part, we next need to estimate the current market share so that we can calculate the full demand rate. Note that if $\tilde{p}$ is the current price and MS is the market share, then the estimate of the full demand rate Λ is $\lambda(\tilde{p})/MS$. Although the OEM may have knowledge on the market share of its rotable sellings business at an aggregate level, however, the market share at each rotable part level is unknown.

It is important to note that in theory, there are many ways to obtain the market share

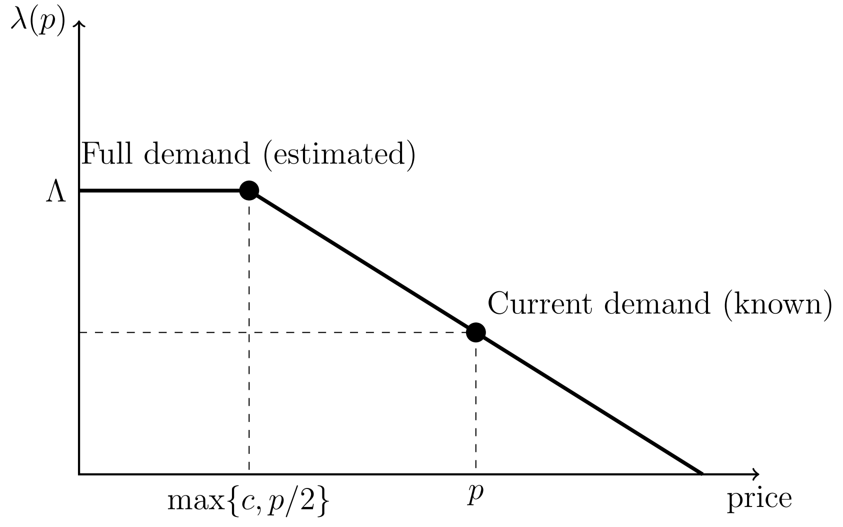


Figure 2.9: Demand model

at the part level and we investigated many of those. Repair logs of airplanes are one source, however, only around 70% customers use a common software, and in a lot of cases, the removal of a rotable spare part is only for testing or is a precedent step of fixing another part. This data source, while effective in theory, proved to be highly incomplete in practice. We also investigated the data in user manuals to try to infer frequencies of replacements needed, but these lead to inaccurate estimates of demand as the service manuals are more geared toward inspection than replacements. We also analyzed the rotable purchase history from warranty customers and the OEM owned aircraft, of which all the purchases of rotable spare parts can be safely assumed to be from the OEM. However, the purchase frequency of warranty customers and the OEM owned aircraft is not representative of the entire fleet because the models of their aircraft are limited and the age of their aircraft are relatively new.

Given the limitations above with all available data sources, we decided to adopt a simple but robust way to estimate the current market share. Rather than focusing on parts, we focus on customers and airplanes. We assume, according to common practice in the aircraft industry, that each rotable spare part needs to be replaced at least once every ten years. For any rotable spare part i, we can calculate the total number of customers who should have bought the part over 10 years, T_i, by simply counting the number of airplanes that the rotable spare parts can be applied on. On the other hand, from the sales data, we can extract the number of distinct customers who bought the rotable spare part i, denoted by B_i. In turn, the market share of rotable spare part i is estimated by $MS_i = \frac{B_i}{T_i}$. Notice that in this approach, we underweighted the customers who regularly shop from the OEM and overweighted the customers who seldom purchase from the OEM.

As an example, we estimated that the market share for the sensor to be 30% while the market share of the jack to be 22%. We notice that most of the estimates may have some error associated with them due to the limited amount of recorded data for each rotable spare part. We directly address this in the development of the pricing analytics tool by providing estimated bounds on the market share and generating prices that are robust to changes in the inputs.

2.4 Price Optimization and Robustification

A natural approach would be to simply compute the optimal price for each spare part under the estimated inputs by solving (2.1). However, as stated earlier, there are multiple sources of potential errors in the estimation of the inputs given the unique environment we operate in. We develop a robust pricing approach to account for such potential errors in calculating the suggested price for each rotable spare part. The main idea in calculating the suggested price is to treat the estimate inputs as ranges of possible values rather than just a fixed value since the initial estimates may not be accurate due to lack of data. We select the price that works the best on a variety of scenarios, which would make the suggested price to be robust to estimation error. We developed the following procedure to calculate the suggested price for a given rotable spare part.

Rotable Spare Part Pricing Algorithm

Step 1. Find the candidate prices

- Estimate inputs (C, $\frac{1}{\mu}$, c, market share(MS)).
- Calculate the base optimal price, p_{opt} by solving (2.1).
- Compute the minimum and maximum prices that can achieve at least 95% of optimality under estimated inputs, denoted by p_{min} and p_{max}.

Step 2. Select a robust price

- Generate $1,000$ inputs $(\frac{1}{\mu'}, c', MS')$ according to

$$\frac{1}{\mu'} \sim \begin{cases} Normal(\frac{1}{\mu}, std_{\frac{1}{\mu}}), \text{if } std_{\frac{1}{\mu}} > 0 \text{ and \# records} \geq 5 \\ Uniform(0.8\frac{1}{\mu}, 1.2\frac{1}{\mu}), \text{o.w.} \end{cases}$$

$$c' \sim \begin{cases} Normal(c, std_c), \text{if } std_c > 0 \text{ and \# records} \geq 5 \\ Uniform(0.8c, 1.2c), \text{o.w.} \end{cases}$$

$$MS' \sim \begin{cases} Uniform(MS_{min}, MS), \text{w.p. } \frac{1}{2} \\ Uniform(MS, MS_{max}), \text{w.p. } \frac{1}{2} \end{cases}$$

where

$$MS_{min} = \max\{\min\{0.8MS, MS - 0.05\}, 0.01\}$$

$$MS_{max} = \min\{\max\{1.2MS, MS + 0.05\}, 0.99\}.$$

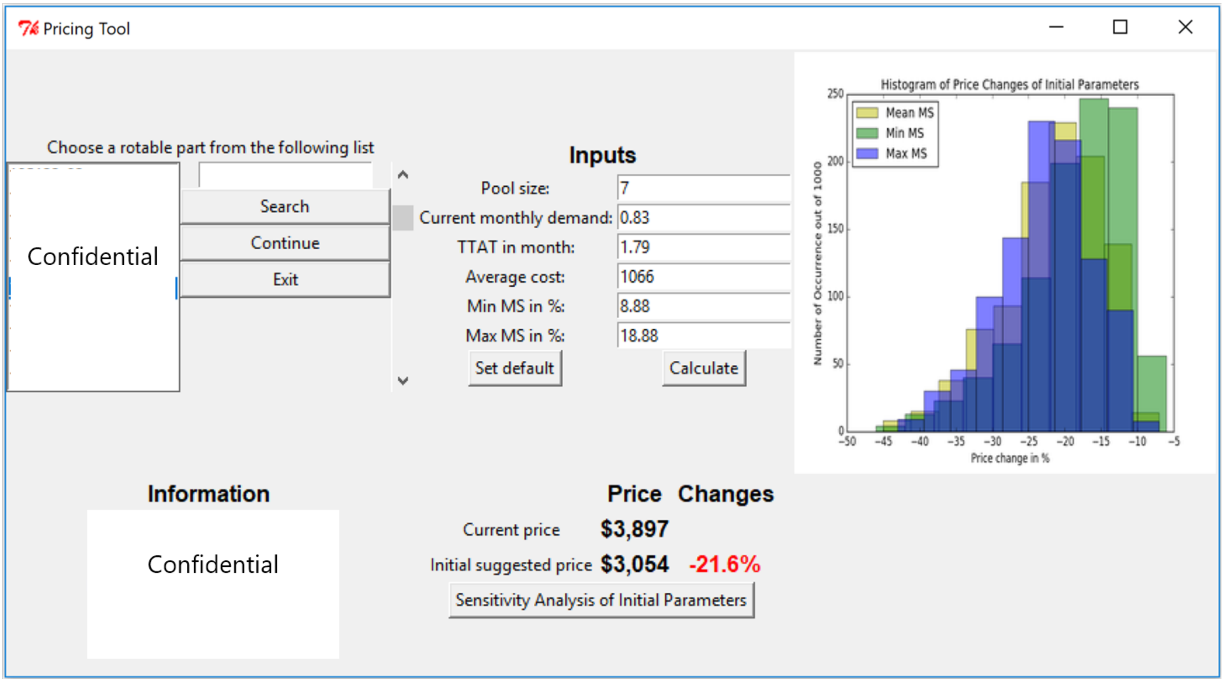
- Evaluate the average profit rate of p_{opt}, p_{min}, and p_{max} under each set of generated inputs $(C, \frac{1}{\mu'}, c', MS')$ on the objective function in (2.1).
- Return the price with the highest average profit value.

In our running examples of the sensor and the pump, the optimal price for the sensor to reduced the price by about 11%, while price reductions associated to the minimum and maximum prices of 95% optimality are 18% and 4%. After evaluating the three candidates, the final suggested price is chosen to be the optimal price from the model which is about a 11% decrease of the original price. In the jack example, the price reductions of our candidates are 24%, 16%, and 10%. After evaluating on randomly generated inputs, the highest price candidate is the selected suggested price and this corresponds to a 10% price decrease. The main reason we chose the maximum candidate as the final suggested price for the jack is due to the high variance in the repair cost, in which case the algorithm tends to be more conservative.

2.5 Implementation

In this section, we discuss the visualization and decision support tool we developed for helping the OEM in the implementation as well as the controlled experiment conducted at the OEM for 1,702 rotable spare parts.

2.5.1 Visualization and Decision Support Tool


Pricing Tool
Choose a rotable part from the following list
Search
Continue
Exit
Inputs
Set default
Calculate
Information
Price Changes
Sensitivity Analysis of Initial Parameters
Histogram of Price Changes of Initial Parameters
Mean MS
Min MS
Max MS
Number of Occurrence out of 1000
Price change in %

Confidential

Pool size: 7
Current monthly demand: 0.83
TTAT in month: 1.79
Average cost: 1066
Min MS in %: 8.88
Max MS in %: 18.88

Confidential

Current price $3,897
Initial suggested price $3,054 -21.6%

Figure 2.10: Visualization and Decision Support Tool.

In Figure 2.10, we show a screenshot of the Visualization and Decision Support Tool. This tool is implemented in Python using the Tkinter package for the GUI. On the top left of the tool, there is a panel that allows the users to search and choose the rotable spare part (by family number) they want to analyze. After the user selects a specific rotable part, the basic information such as part number, usage, applicability (which models it can be applied on), and past sales of that rotable part are displayed in the lower left corner. Next, the estimated inputs of the price optimization model are displayed in the top middle region of the tool. Instead of displaying the estimated market share, we choose to display the range of the market share. The minimum and maximum market share levels are calculated using the formula described in the algorithm above.

We designed the entries of inputs to be editable so that if the users do not agree with our estimation, they can manually overwrite the value. Note that if the minimum and maximum market shares are changed, then the estimated market share is updated to be the average of the minimum and maximum. Another advantage of making the entries editable is that it allows the users to do what-if analyses, which may help them in other part of the operations. For example, they may want to understand whether to increase or decrease the pool size, and how to negotiate with repair agents on repair time and cost.

When the users agree on the value of inputs and click calculate, the suggested price as well as the percentage change will be displayed at the lower middle part of the tool. This price is the output returned by the algorithm described above. In addition, since the estimation of market share contains the most uncertainty, we built a sensitivity analysis function in the tool to show how different market share level will impact the suggestion. This corresponds to the histogram on the right side of Figure 2.10. To create the histogram, we generate $(\frac{1}{\mu'}, c')$ 1,000 times using the same procedure described in Step 2 of the algorithm. Then, we calculate the optimal prices under the different sets of inputs $(C, \frac{1}{\mu'}, c', MS)$, $(C, \frac{1}{\mu'}, c', MS_{min})$, and $(C, \frac{1}{\mu'}, c', MS_{max})$ according to different levels of market share. Finally, the histograms of percentage of changes associated to the optimal prices is plotted on the visualization and decision support tool. The sensitivity analysis aims to provide guidance to the users on the direction of the price change (whether to increase or decrease the price). As long as the users believe the range of the market share is correct, the sensitivity analysis will provide the frequency of optimal price changes. Even if the users choose not to follow the suggested price, the sensitivity analysis may inform the users which direction the price should go in order to maximize the profit rate. The users can leverage this information as well as other business constraints to make a final decision.

Overall, the tool we developed serves as a decision support tool which gives a suggested price which is robust to input estimation errors for a given rotable spare part. This tools helps the users in making pricing decisions and understand the implications of the assumptions they have about market shares, pool size, repair time, and repair cost.

2.5.2 Experiment design

We now describe the controlled experiment we conducted to test the effectiveness of our pricing analytics tool. To ensure the control and test groups are comparable, for each rotable part selected in the test group, one needs a 'similar' rotable part in the control group. One natural way is to look at the estimated inputs of the price optimization model of each rotable spare part. If every estimated input is similar between two rotable spare parts, then one could claim these two parts are similar, and more importantly, the suggested changes of these two parts would be similar as well. However, this approach did not work because there are barely two rotable spare parts of which all estimated inputs are similar. To overcome this difficulty, we propose the following procedure in selecting parts into the control and test groups.

Part Selection Procedure

Step 1. Group rotable spare parts by their usage category, e.g. valve, pump, etc.

Step 2. In each usage group, sort rotable spare parts in ascending order based on their suggested price changes.

Step 3. In each usage group with size at least two, select the rotable spare parts according the following rule. For $i = 1, 3, 5, 7, ...$, randomly assign one of i and $i + 1$ into the test group and the other into the control group.

If the size of the usage group is odd, randomly assign the last part into test or control.

Step 4. Combine usage groups of size one, repeat Step 2 and Step 3 for the combined group.

The key idea behind this selection procedure is that the usage of each rotable spare parts is a natural classifier of different rotable spare parts. Instead of focusing on the similarity of all estimated inputs between two different rotable spare parts, which is a high dimensional clustering problem, we just focus on the outcome, the suggested price changes, which is the result of running the price optimization algorithm described above. This leads to a one-dimensional problem, thus the selection method is easy to explain internally in the organization.

Using the above procedure, 852 out of 1,702 rotable spare parts are selected into the test group and the remaining 850 rotable spare parts are in the control group. Within the test group, 744 rotable spare parts are selected to receive price decrease while the prices of 108 rotable spare parts are set increase. In the control group, we decrease the prices on 743 rotable spare parts and increase the prices on 107 rotable spare parts. We handed the visualization and decision support tool as well as the lists of the control and test groups to the team at the OEM in the Spring of 2018. The OEM's team used the tool while updating input estimates on 498 rotable spare parts and reviewed the suggested changes for each rotable spare part, and decided on the best price while using tool.

2.5.3 Implementation results

The OEM changed the price of the rotable spare parts in the test group on May 4th, 2018 and sent a notification to its customers. The notification sent by the OEM did not include the part numbers of the rotable spare parts that received price change so that we can isolate the effect of the price change from the marketing effort.

We collected the sales data of the 1,702 rotable spare parts from July 7th, 2017 to March 5th, 2019. This data represents the sales of rotable spare parts for 208 working days before and after the implementation of price changes. We perform a difference-in-difference (DiD) analysis in the aggregate level to measure the effect of our price optimization model as well as the decision support tool.

Since the objective of the price optimization model is to maximize the expected profit rate, therefore, in the DiD comparison, we focus on the profit changes in two groups before and after the implementation. From the DiD analysis, we see an estimated impact of 3.9% *increase* in profit from our pricing analytics tool. We omit the details of the analysis due to confidentiality reasons.

To ensure the increase in profit is not contributed by some days with very good (or poor) sales records, we provide a confidence interval of the DiD estimate of the profit increase. The interval is generated by generating 10,000 randomly generated bootstrapped datasets consisting of 208 working days with replacement in both before and after periods. For each dataset, we compute the DiD estimate and find that the confidence interval of the estimate to be [3.61%,3.95%]. Conservatively, the new pricing algorithm may add over millions of dollars of profit per year in selling rotable spare parts for the OEM.

2.6 Conclusion

In this collaboration project with a major aircraft OEM, we investigated the problem of setting appropriate prices in selling rotable spare parts. We adopted a data-driven price optimization approach to maximize the expected profit rate in rotable selling, which can also be used for a broader class of problems concerning reusable resources. This approach captures the special system dynamics such as fixed pool size, random repair times, exchange sales, and market competition, most of which are not taken into account in the legacy approach. In addition, the algorithm proposed and the tools developed allow users to understand the implications of input errors and to be robust against these.

We conducted a large-scale controlled experiment on 1,702 different rotable spare parts at the OEM starting in the beginning of May 2018 and received encouraging implementation results. The successful implementation demonstrates the power of a structured and systematic pricing analytics approach, even when facing a slow-moving environment.

As of today, the OEM is engaged in expanding and building an internal version of the visualization and decision support tool so that it can seamlessly integrates with internal data flows. This methodology and its refinements will be a core part of pricing rotable spare parts. In addition, this project opened up the discussion of a systematic review of pricing processes across the organization, which has lead to multiple new projects including rental tool pricing and service center quote estimation.

Acknowledgement

We would like to thank our industrial partner for their valuable discussions and help throughout the project.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

Pricing Reusable Resources with Multiple Customer Classes

We consider the problem of pricing reusable resources when there are multiple classes of customers. This is an extension to the single class problem analyzed in Chapter 1. To deal with the curse of dimentionality of the problem, we propose two types of heuristics. The first is based on the construction of simple pricing policies directly from the optimal dynamic pricing policy. The second is based on the construction of static pricing policies in a split system where units are dedicated to each class. We provide numerical experiments showing the performance of the proposed heuristics. In addition, we present a performance guarantee of the static pricing policy on the simplest version of the problem where there are only two units and two classes of customers. We also discuss the role of substitution effects between different classes of customers.

3.1 Introduction

In some applications of selling reusable resources, customers may be able to be distinguished into different classes based on their service time, sensitivities to the price changes, and market sizes. Recall an example of a reusable resource is a rotable spare part. An original equipment manufacturer (OEM) may use the same rotable spare part to serve two classes of customers, those who want to rent and those who want to purchase (exchange sale). If a customer rents the rotable spare part, then the rental time would be the service time. If a customer chooses to do an exchange sale, then the customer gives the broken part to the OEM in exchange for a functional one. The time to repair the broken part would be the service time. In this example, rental and exchange sale customers are two different classes of customers since the market size, price sensitivity, and service time of each class is very different.

Another example of managing reusable resources facing multiple classes of customers happens in the service centers of an aircraft OEM. A service center manages a certain number of hangar spaces, which is a kind of reusable resource, to perform maintenance tasks for its customers. Naturally, there are several levels of maintenance tasks which are based on the total usage/condition (e.g. 10,000 miles vs 50,000 miles maintenance for cars, 1,200 hours vs 3,600 hours maintenance for planes). Though each maintenance task only requires one hangar space, different levels of maintenance tasks have different prices and service times. In such a setting, one cannot simply treat all customers to be in the same class and thus multiple classes are required.

To this end, we want to study the problem of pricing for reusable resources when facing multiple classes of customers. We focus our analysis on a model where a service provider manages a fixed pool of a single type of reusable resource. The firm uses these units to deliver service to its customers over an infinite time horizon. Each class of customers arrive to the system according to a Poisson process where the rate depends on the price set for its class and this rate may or may not be depended on the prices for other classes. We make a standard assumption that the revenue rate for each class is concave in its arrival rate. Upon arrival, customer uses one unit of the reusable resource for an exponentially distributed amount of time which depends on their class. The unit being occupied by the customer cannot be acquired by other customers before service completion. The firm may occur a service cost which also depends on the customer's class. The goal of the firm is to set a simple pricing policy which maximizes its long-run expected profit rate through selling the reusable resources.

3.1.1 Related literature

There have been many studies on reusable resources and our work is an extension of the work presented in Chapter 1, where we prove a universal guarantee for static pricing when only one class of customers present. There are some studies on managing reusable resources with multiple customer classes. For example, Gans and Savin (2007) analyzed the rental system facing multiple contract and walk-in classes and showed that static policies are asymptotically optimal. Levi and Radovanović (2010) formed a linear program to devise a class selection policy for revenue management in systems with reusable resources and prices are fixed. They prove that their class selection policy achieves at least half of the optimal expected long-run revenue rate for their model with a single resource. For more related research on reusable resources, as well as the comparison between dynamic and static pricing policies, refer to Chapter 1 and the reference therein.

Aside from the studies of reusable resources, our model for pricing reusable resources facing multiple classes of customers is related to research on revenue management of multiclass queues. Maglaras (2006) studied the problem of revenue management of an $M_n/M/1$ queue using a fluid analysis and proposed a simple pricing heuristic which has near-optimal performance numerically. Çelik and Maglaras (2008) studied the problem of maximizing profit at a make-to-order manufacturer which provides multiple products. They derived near-optimal dynamic pricing, lead-time quotation, sequencing, and expediting policies based on an approximation diffusion control problem. In Afèche (2013), the author considered the problem of designing a price/lead-time menu and scheduling policy to maximize revenues from two types of heterogeneous time-sensitive customers. The author showed that a strategic delay policy, which prioritizes impatient customers, but artificially inflates the lead times of patient customers, may be optimal. Our work is also related to research on flexibility in queueing systems (e.g., Sheikhzadeh et al. (1998) and Gurumurthi and Benjaafar (2004)) since we consider a heuristic where splitting resources is required.

Our study on comparing a system with and without substitution effects in demand is related to research on substitutable products. For example, Ceryan et al. (2013) studied a joint implementation of price- and capacity-based substitution mechanisms where a firm produces substitutable products via a capacity portfolio consisting of both product-dedicated and flexible resources. They characterized the structure of the optimal production and pricing decisions and explored the effects of changing various problem parameters to the optimal policy structure. Other studies on substitutable products include pricing and quantity decision of a single period problem at a retailer (Tang and Yin 2007), demand management and inventory control (Song and Xue 2007), and joint ordering and pricing strategy (Ye 2008). Most of these papers assumed infinite production capacity, which is different from our setting where we only manages a fixed number of units and our goal is to find price decisions to maximize profit rate.

3.1.2 Organization

The rest of the chapter is organized in the following way. In Section 3.2, we present the multi-class pricing model for reusable resources. We also discuss the way we model demand with multiple customer classes. In Section 3.3, we propose two types of simple pricing policies for our problem and provide numerical studies of the heuristics. In Section 3.4, we show a performance guarantee of the static pricing policy on a special case where there are only two units and two classes of customers. We then compare the case when substitution effects in demand exists between different classes of customer to the non-substitution case in Section 3.5.

3.2 Model

In this section, we first present the pricing model for reusable resources with multiple customer classes, which is an extension to the general model introduced in Chapter 1. We then discuss the performance metric we focus on and basic properties. Finally, we describe the demand model we consider to capture inter-class effects.

3.2.1 Multi-classes pricing model for reusable resources

We consider the problem of price optimization for reusable resources when there are multiple classes of customers. The firm manages C units of a single type of reusable resource to serve M classes of price-sensitive customers. These units are identical and non-perishable. At any point in time, each unit of the resource is either available for sale or occupied by a customer (from any class). For example, an occupied unit can be one hangar space used for a maintenance task or a rotable spare part on a rental customer's plane.

We assume customers arrive to seek one unit of a reusable resource corresponding to a Poisson process at rate $\Lambda > 0$. Each class has different willingness-to-pay function $F_j(p_j; \mathbf{p})$ ($\mathbf{p}$ is the vector of prices) and choose to purchase one unit if their valuations exceed the price set for its class, p_j. Therefore, we denote $\lambda_j(\mathbf{p}) := \Lambda \bar{F}_j(p_j; \mathbf{p})$ the effective arrival rate (demand rate) for class j. After a customer of class j purchases, one unit of the reusable resource is then occupied for a random amount of time which follows an exponential distribution with mean $1/\mu_j$. We assume that the usage times are i.i.d. across the customers in the same class and independent of the customer's valuation.

While a unit is being occupied, the firm cannot sell that unit until it is returned to the system, i.e., the hangar space is released after finishing the maintenance or a rental customer returns the rotable spare part. The firm incurs a cost c_j to serve a class j customer. We assume that if all units are occupied when any customer arrives, that customer will be lost regardless of the price being offered.

3.2.2 Performance metric

The objective of the firm in selling the reusable resources is to maximize the expected profit rate. Because of the assumptions of Poisson arrivals and exponential usage duration for every class of customers, the system of selling reusable resources can be modeled as a Markov Decision Process (MDP). Let n_j denote the number of class j customers in the system, then the state space $\mathcal{S}$ can be expressed by

$$\mathcal{S} = \{(n_1, \ldots, n_M) : \sum_{j=1}^{M} n_j \leq C,\ n_j \geq 0,\ n_j \in \mathbb{Z},\ \forall j = 1, \ldots, M\}.$$

Given a state $s \in \mathcal{S}$, the firm's action is to set the price for each class of customers, denoted by the vector of prices by $\mathbf{p}^s$. Let λ_{js} denote the demand rate of class j customers for state s. The transition probability between states can be expressed as

$$(n_1, \ldots, n_k, \ldots, n_M) \rightarrow (n_1, \ldots, n_k + 1, \ldots, n_M) = \begin{cases} \frac{\lambda_{ks}}{\sum_{j=1}^{M} \lambda_{js} + n_j \mu_j}, \text{ if } \sum_{j=1}^{M} n_j \leq C - 1 \\ \\ 0, \text{otherwise.} \end{cases}$$

$$(n_1, \ldots, n_k, \ldots, n_M) \rightarrow (n_1, \ldots, n_k - 1, \ldots, n_M) = \begin{cases} \frac{n_k \mu_k}{\sum_{j=1}^{M} \lambda_{js} + n_j \mu_j}, \text{ if } n_k \geq 1 \\ \\ 0, \text{ otherwise.} \end{cases}$$

Finally, the immediate reward of serving a class j customer is $p_j - c_j$.

Let $\mathbf{P}$ be the price decisions for the system, $\mathbf{P}$ is a $M \times |\mathcal{S}|$ matrix. Note that $\mathbf{p}^s$ is a column in $\mathbf{P}$. Denote $\mathbb{P}_s(\mathbf{P})$ the steady-state probability of the system being in state s under price decisions $\mathbf{P}$. We express the objective function of maximizing the expected profit rate is

$$\max_{\mathbf{P}} \sum_{s \in \mathcal{S}} \left(\sum_{j=1}^{M} \lambda_{js}(p_{js} - c_j) \right) \mathbb{P}_s(\mathbf{P}) \tag{3.1}$$

where λ_{js} and p_{js} are the demand rate and associated price for class j customer in state s under price decision $\mathbf{P}$. One may use standard techniques such as value iteration and policy iteration to solve this MDP. The main difficulty in solving the MDP is the curse of dimentionality since $|\mathcal{S}| = \mathcal{O}(C^M)$.

3.2.3 Demand model

In our model, we consider two types of demand relations across different classes. One is the non-substitution case where the price for each class has no effect on the demand rate of other classes. Managing the hangar spaces at service centers described in Section 3.1 is an example of this type because the level of maintenance tasks is mandatory based on the usage of the planes, one cannot choose to do a lower level maintenance even if the price is lower. In general, in such application, the demand rate for a certain level of maintenance tasks would only be affected by the price of that level and would not be affected by the prices of other levels. On the other hand, we also consider the substitution case where the demand rate for a certain class is not only determined by the price for that class, but also affected by the prices of other classes. The rotable spare parts serving both rental and exchange sale customers is a good example for this case. When seeking a functional rotable spare part, some customers are in-between renting or buying a rotable spare part. Their decisions depend on the list prices of the two options and they will choose the one which results in a lower total cost based on their preferences.

To this end, we assume that the demand rate of one type of customer may or may not be affected by the prices for the other type of customers. We shall assume for each class j, the demand function $\lambda_j(\mathbf{p})$ has the following properties: i) $\lambda_j(\mathbf{p})$ is decreasing in p_j ii) $\lambda_j(\mathbf{p})$ is non-decreasing in p_k, for all $k \neq j$. The first condition ensures that for any class, the price set for that class has a negative effect on its demand rate while the second condition represents the substitution effects brought by the prices of other classes.

An example of demand functions satisfying the above two properties are linear functions with the form

$$\boldsymbol{\lambda} = A\mathbf{p} + \mathbf{b}, \tag{3.2}$$

where A is an $M \times M$ matrix in which the diagonal entries $a_{jj} < 0$ and other entries $a_{jk} \geq 0$, for all $j \neq k$. The diagonal entries represent the main price-demand relationship for each class of customers, and thus these entries are assumed to be negative. The off-diagonal entries represent the effects of price of other classes, and we assume the customers from one class may shift to another class if the price is low enough. Notice that in the case where $a_{jk} = 0$, it represents the condition that the demand rate for class j is not affected by other classes. The vector $\mathbf{b}$ represents the demand rate when prices for all classes are set to zero. Notice that the linear demand function presented in Eq. (3.2) is used in the numerical experiments presented later in this chapter.

3.3 Heuristics

One may use value iteration to find the optimal policy to the MDP presented in Section 3.2. However, the optimal solution to the general multi-classes customers pricing model suffers from the drawback of the curse of dimensionality since the size of state space $\mathcal{S}$ is on the order of C^M. Therefore, to apply the optimal policy in such a system may be impractical. Inspired by the performance guarantee shown in Chapter 1, we aim to understand how a simple pricing policy performs in the multiple class setting. In this section, we propose two different pricing heuristics where there is no substitution effects between different classes of customers.

3.3.1 Construction from the optimal dynamic policy

In this subsection, we first describe heuristics to the multi-classes system based on the optimal dynamic pricing policy, then we provide the numerical experiments showing these simple pricing policies have good performance under various input parameters.

The first idea we investigate is to directly construct the static pricing policies from the optimal policy, which follows the method described in Chapter 1 since such construction is shown to be near-optimal in the single class case under all parameter regimes. The algorithm to construct the Static Pricing Heuristic is the following. We construct a single static price for every class.

Static Pricing Heuristic

Step 1. Solve the MDP optimally using value iteration, let λ^*_{js} denote the optimal demand rate of class j customers in state $s = (n_1, \dots, n_M)$, and $\mathbb{P}^*_s$ be the corresponding steady-state probability under the optimal policy.

Step 2. For each class $j = 1, \dots, M$ customers, calculate the static price as

$$\tilde{\lambda}_j = \frac{\sum_{s:n_1+\cdots+n_M<C} \lambda^*_{js}\mathbb{P}^*_s}{\sum_{s:n_1+\cdots+n_M<C} \mathbb{P}^*_s}. \tag{3.3}$$

Step 3. Form the pricing heuristic leading to arrival rates $(\tilde{\lambda}_1, \dots, \tilde{\lambda}_M)$.

Notice that in the construction of the Static Pricing Heuristic, for each class, we make the expected demand rate under the static pricing policy equal to that under the optimal policy when the system is not fully occupied (customers can be accepted).

The Static Pricing Heuristic reduces the number of prices of the system from $\mathcal{O}(MC^M)$ to M. One may wonder whether using a static pricing policy is too restrictive since it ignores inventory levels. We next propose another simple pricing heuristic, the Level-dependent Pricing Heuristic, which has more flexibility than the Static Pricing Heuristic and still maintains a simple structure. The algorithm is as follows.

Level-dependent Pricing Heuristic

Step 1. Solve the MDP optimally using value iteration, let λ^*_{js} denote the optimal demand rate of class j customers of the state $s = (n_1, \ldots, n_M)$, and $\mathbb{P}^*_s$ be the corresponding steady-state probability under the optimal policy.

Step 2. For each system occupancy level, $l = 0, \ldots, C-1$, calculate the demand rate for class $j = 1, \ldots, M$ customers as

$$\hat{\lambda}^l_j = \frac{\sum_{s:n_1+\cdots+n_M=l} \lambda^*_{js}\mathbb{P}^*_s}{\sum_{s:n_1+\cdots+n_M=l} \mathbb{P}^*_s} \tag{3.4}$$

Step 3. Form the pricing heuristic based on the occupancy level $l = 0, \ldots, C-1$ as $(\hat{\lambda}^l_1, \ldots, \hat{\lambda}^l_M)$, and $\hat{\lambda}^C_j = 0, \ \forall\ j$.

Note that Eq. (3.4) ensures the expected demand rate for each class of customers at any occupancy level is the same as in the optimal policy under the same occupancy level. The Level-dependent Pricing Heuristic contains CM prices. Notice that the construction of the

Static Pricing Heuristic presented earlier in Eq. (3.3) can be viewed as a weighted average of the Level-dependent Pricing Heuristic (Eq. (3.4)).

We next discuss the numerical experiments conducted to demonstrate the performance of the two proposed heuristics. In the numerical experiments, we focus on the setting where $M = 2$ (two classes of customers) due to computational limits. As stated in Section 3.2, the demand functions used in the numerical studies are linear functions (see Eq. (3.2)). In the two classes setting without substitution, we have

$$A = \begin{bmatrix} a_{11} & 0 \\ 0 & a_{22} \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}.$$

Let μ_1, μ_2 denote the usage rate of class 1 and 2 customers, respectively. In the numerical experiment, we vary C from 2 to 8. In each level of C, we calculate the optimal profit rate and the profit rate under two proposed pricing heuristics using all combinations of a_{11}, a_{22}, b_1, b_2, μ_1, and μ_2 where

$$\begin{aligned}
&a_{11} \in \{-0.1, -0.2, -0.3, -0.4, -0.5\} \\
&a_{22} \in \{-0.05, -0.1, -0.15, -0.2, -0.25\} \\
&b_1 \in \{3, 5\} \\
&b_2 \in \{1, 2\} \\
&\mu_1 \in \{0.1, 0.2\} \\
&\mu_2 \in \{0.5, 0.8\}.
\end{aligned}$$

In the experiment, we set the cost of serving both classes to be zero. The results of the numerical experiment are summarized in Table 3.1.

	C	2	3	4	5	6	7	8
Level Policy	Average	99.9%	99.9%	99.9%	99.9%	99.9%	99.9%	99.9%
	Worst	98.9%	99.0%	99.2%	99.4%	99.5%	99.6%	99.6%
Static Policy	Average	99.2%	98.7%	98.3%	98.2%	98.1%	98.1%	98.2%
	Worst	94.7%	94.3%	94.0%	94.2%	94.5%	95.0%	95.4%

Table 3.1: Comparison of simple pricing heuristics and the optimal policy, $M = 2$

As one may observe, the performance of the Level-dependent Pricing Heuristic is near optimal for all levels of C which indicates that the loss caused by reducing $\mathcal{O}(MC^M)$ prices to MC prices is minimal. Moreover, even when only M prices are used in the Static Pricing Heuristic, the average loss of the optimality is less than 2% in the experiments and the worst case loss is less than 6%. In addition, one may notice that under any level of C, the performance of the level-dependent policy dominates that of the static policy. Due to the curse of dimentionality, e.g., for the case of $C = 8$, it takes around 1 minute to compute the result for 1 instance, we did not run the experiment on larger C or more input sets. We believe the results in Table 3.1 still demonstrate the good performance of simple pricing policies in such an environment when facing multiple classes customers and is an interesting future direction for theoretical results.

In addition, we numerically search for the combination of inputs for the worst case performance of the static pricing heuristic. We discover that the performance of the Static Pricing Heuristic becomes worse in the case where $a_{11} \ll a_{22}$, $b_1 \gg b_2$, and $\mu_1 \ll \mu_2$. For example, when $a_{11} = -10$, $a_{22} = -0.001$, $b_1 = 50$, $b_2 = 0.1$, $\mu_1 = 0.01$, $\mu_2 = 10$, comparisons between the performance of the Static Pricing Heuristic and the optimal expected profit rate under dynamic pricing policy is summarized in Table 3.2. One may observe that the worst

C	2	3	4	5	6	7	8
Static Policy	80.6%	79.6%	80.2%	80.9%	81.7%	82.4%	83.0%

Table 3.2: Worst case performance of Static Pricing Heuristic, $M = 2$

case performance of the static pricing heuristic is at least 79.6% for all levels of C. We will provide a theoretical guarantee later in Section 3.4 for the performance of the static pricing heuristic when $C = 2$.

3.3.2 Splitting resources

We next examine another heuristic which is to transform the shared system (all units can serve all classes) to a split system where a certain class of customers are served by a subset of units. Such applications may appear in service centers where the hangars are the reusable resources to be used for different maintenance tasks. One may prefer a dedicated assignment when certain level of maintenance tasks requires some unique and heavy equipment that are inconvenient to be transported between hangars. We want to investigate, under such an environment, how the split system performs compared to the shared system.

Let $\mathcal{C}$ be the set of all possible splittings of C units of resources, then $\mathcal{C}$ is defined as

$$\mathcal{C} = \{(C_1, \ldots, C_M) : \sum_{j=1}^{M} C_j = C,\ C_j \geq 0,\ C_j \in \mathbb{Z}\ \forall j = 1, \ldots, M\}.$$

Let $\mathcal{P}(C_1, \ldots, C_M)$ denote the optimal profit rate under splitting $(C_1, \ldots, C_M) \in \mathcal{C}$. Then we have

$$\mathcal{P}(C_1,\ldots,C_M)=\sum_{j=1}^{M}\mathcal{P}_j(C_j),$$

where $\mathcal{P}_j(C_j)$ represents the optimal profit rate (allowing for dynamic pricing) that can be achieved when assigning C_j units to class j customers exclusively.

We now describe the method of splitting resources. We want to find the optimal splitting such that the total expected profit rate in the split system is maximized. Notice this splitting method has the advantage that solving the MDP described in Eq (3.1) is not required. This would be helpful when one faces a system with many units and classes. We now describe the algorithm to form the static pricing heuristic (Optimal Splitting Heuristic) through splitting the resources.

Optimal Splitting Heuristic

Step 1. Evaluate $\mathcal{P}(C_1,\ldots,C_M)$ for all possible splitting $(C_1,\ldots,C_M)\in\mathcal{C}$.

Step 2. Select the splitting $(C_1^*,\ldots,C_M^*)$ which gives the highest expected profit rate, which is $(C_1^*,\ldots,C_M^*)=\arg\max_{(C_1,\ldots,C_M)\in\mathcal{C}}\mathcal{P}(C_1,\ldots,C_M)$.

Step 3. Construct the static policy for each class j using the same method as in Chapter 1 using C_j^* units and the optimal prices from $\mathcal{P}_j(C_j^*)$.

We next provide numerical results to show the performance of the Optimal Splitting Heuristic in the split system. The parameters in the numerical experiments are the same as described in Section 3.3.1. First, we compare the expected profit rate in the split system using static policies (Optimal Splitting Heuristic) to the optimal expected profit rate in the shared system. In addition, since Optimal Splitting Heuristic also provides a way of constructing of static prices, we want to examine the performance of such a static pricing policy in the shared system. We report the average and worst ratios for both cases based on different levels of C. The results are summarized in Table 3.3.

	C	2	3	4	5	6	7	8
Split system	Average	95.4%	94.5%	93.8%	93.4%	93.3%	93.3%	93.4%
(no sharing)	Worst	83.2%	86.5%	86.8%	87.4%	88.3%	88.7%	89.5%
Shared system	Average	96.4%	95.2%	94.5%	94.5%	95.5%	96.3%	96.8%
	Worst	81.2%	66.9%	66.7%	69.2%	75.0%	79.6%	83.2%

Table 3.3: Performance of Optimal Splitting Heuristic, $M = 2$

As one may observe, on average, the loss due to splitting resources is less than 7%, and the worst case loss is less than 18%. This result suggests adopting a split system should also be acceptable option to the firm when there exist some soft constraints in managing reusable resources. Moreover, though the average performance of the static pricing policy from Optimal Splitting Heuristic in the shared system is higher, one would better stick on the split system because such static pricing policy has a more robust performance in the split system. For example, when $C = 4$, we can get at least 86.8% of the optimal profit rate in the split system while applying static policy from Optimal Splitting Heuristic to the shared system only obtains 66.7% of the optimal profit rate in the worst case.

3.4 Performance Guarantee on Two Units Two Classes System

In this section, we provide a performance guarantee of the static pricing policy on the simplest multi-classes system where the firm manages only two units of the reusable resource to serve two classes of customers. In this section, we make the following change to the notation where $\lambda_{js} = \lambda_j s$. The states in the MDP of such system are $(0, 0)$, $(0, 1)$, $(1, 0)$, $(0, 2)$, $(1, 1)$, and $(2, 0)$. Figure 3.1 illustrates the transitions between the states under a static pricing policy.

We adopt the same approach as in Theorem 1.3.1 in Chapter 1 to get a lower bound on the performance of the static policy in such system.

$$\begin{tikzcd}
(0,0) \arrow[r, bend left, "\lambda_2"] \arrow[d, bend right, "\lambda_1"'] & (0,1) \arrow[r, bend left, "\lambda_2"] \arrow[l, bend left, "\mu_2"] \arrow[d, bend left, "\lambda_1"'] & (0,2) \arrow[l, bend left, "2\mu_2"'] \\
(1,0) \arrow[u, bend right, "\mu_1"'] \arrow[r, bend right, "\lambda_2"'] \arrow[d, bend right, "\lambda_1"'] & (1,1) \arrow[u, bend right, "\mu_1"'] \arrow[l, bend right, "\mu_2"] & \\
(2,0) \arrow[u, bend right, "2\mu_1"'] & &
\end{tikzcd}$$

Figure 3.1: State transitions of $C = 2$, $M = 2$ system for static policy

Theorem 3.4.1. *When $C = M = 2$, if the revenue function of each class is concave, the best static pricing policy can achieve at least 80% of the optimal expected profit rate from the dynamic pricing policy.*

Proof. Let $\lambda_1^*(i, j)$ and $\lambda_2^*(i, j)$ denote the optimal arrival rate for class 1 and 2 customers when we have i class 1 customers and j class 2 customers in the system. Denote $\mathbb{P}^*(i,j)$ the stationary probability of having i class 1 customers and j class 2 customers. We construct the static pricing policy ($\tilde{\lambda}_1$,$\tilde{\lambda}_2$) as follows

$$\tilde{\lambda}_1 = \frac{\sum_{i+j<2} \lambda_1^*(i,j)\mathbb{P}^*(i,j)}{1 - \sum_{i+j=2} \mathbb{P}^*(i,j)} \tag{3.5}$$

$$\tilde{\lambda}_2 = \frac{\sum_{i+j<2} \lambda_2^*(i,j)\mathbb{P}^*(i,j)}{1 - \sum_{i+j=2} \mathbb{P}^*(i,j)} \tag{3.6}$$

Let $\mathcal{P}^*$ and $\mathcal{P}^{\tilde{\lambda}_1,\tilde{\lambda}_2}$ denote the expected profit rate under optimal policy and the constructed static policy, respectively. Assume that the revenue rates for both classes of customers are concave in λ, we have

$$\begin{aligned}
\frac{\mathcal{P}^{\tilde{\lambda}_1,\tilde{\lambda}_2}}{\mathcal{P}^*} &= \frac{[\tilde{\lambda}_1(p(\tilde{\lambda}_1) - c_1) + \tilde{\lambda}_2(p(\tilde{\lambda}_2) - c_2)][1 - \sum_{i+j=2} \mathbb{P}^{\tilde{\lambda}_1,\tilde{\lambda}_2}(i,j)]}{\sum_{i+j<2}[\lambda_1^*(i,j)(p(\lambda_1^*(i,j)) - c_1) + \lambda_2^*(i,j)(p(\lambda_2^*(i,j)) - c_2)]\mathbb{P}^*(i,j)} \\
&= \frac{\tilde{\lambda}_1(p(\tilde{\lambda}_1) - c_1) + \tilde{\lambda}_2(p(\tilde{\lambda}_2) - c_2)}{\sum_{i+j<2}[\lambda_1^*(i,j)(p(\lambda_1^*(i,j)) - c_1) + \lambda_2^*(i,j)(p(\lambda_2^*(i,j)) - c_2)]\frac{\mathbb{P}^*(i,j)}{1-\sum_{i+j=2}\mathbb{P}^*(i,j)}} \\
&\quad \cdot \frac{1 - \sum_{i+j=2} \mathbb{P}^{\tilde{\lambda}_1,\tilde{\lambda}_2}(i,j)}{1 - \sum_{i+j=2} \mathbb{P}^*(i,j)} \\
&\geq \frac{\tilde{\lambda}_1(p(\tilde{\lambda}_1) - c_1) + \tilde{\lambda}_2(p(\tilde{\lambda}_2) - c_2)}{\tilde{\lambda}_1(p(\tilde{\lambda}_1) - c_1) + \tilde{\lambda}_2(p(\tilde{\lambda}_2) - c_2)} \cdot \frac{1 - \sum_{i+j=2} \mathbb{P}^{\tilde{\lambda}_1,\tilde{\lambda}_2}(i,j)}{1 - \sum_{i+j=2} \mathbb{P}^*(i,j)} \\
&= \frac{1 - \sum_{i+j=2} \mathbb{P}^{\tilde{\lambda}_1,\tilde{\lambda}_2}(i,j)}{1 - \sum_{i+j=2} \mathbb{P}^*(i,j)} \\
&= \frac{\mathbb{P}^{\tilde{\lambda}_1,\tilde{\lambda}_2}(0,0) + \mathbb{P}^{\tilde{\lambda}_1,\tilde{\lambda}_2}(0,1) + \mathbb{P}^{\tilde{\lambda}_1,\tilde{\lambda}_2}(1,0)}{\mathbb{P}^*(0,0) + \mathbb{P}^*(0,1) + \mathbb{P}^*(1,0)}
\end{aligned} \tag{3.7}$$

The inequality follows from applying Jensen's inequality to concave functions. Again, we can bound the ratio of expected profit rate under the constructed static policy and the optimal policy by the ratio of stock-in probabilities. Note that such relation holds for arbitrary C

and M as long as the profit rate function for each class is concave.

From the balance equations, we can calculate the steady state probabilities of states under which customers can be accepted as follows

$$\begin{bmatrix} \mathbb{P}^*(0,0) \\ \mathbb{P}^*(1,0) \\ \mathbb{P}^*(0,1) \end{bmatrix} = \frac{1}{D} \cdot \begin{bmatrix} 2(\mu_1)^2(\mu_2)^2(\lambda_1^*(0,1) + \lambda_2^*(1,0) + \mu_1 + \mu_2) \\ 2\mu_1(\mu_2)^2(\lambda_1^*(0,0)\lambda_1^*(0,1) + \lambda_2^*(0,0)\lambda_1^*(0,1) + \lambda_1^*(0,0)(\mu_1 + \mu_2)) \\ 2(\mu_1)^2\mu_2(\lambda_1^*(0,0)\lambda_2^*(1,0) + \lambda_2^*(0,0)\lambda_2^*(1,0) + \lambda_2^*(0,0)(\mu_1 + \mu_2)) \end{bmatrix} \tag{3.8}$$

where

$$\begin{aligned} D =& \lambda_1^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)(\mu_2)^2 + 2\lambda_1^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1\mu_2 + 2\lambda_1^*(0,0)\lambda_1^*(0,1)\mu_1(\mu_2)^2 \\ &+ \lambda_1^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)(\mu_1)^2 + \lambda_1^*(0,0)\lambda_1^*(1,0)\mu_1(\mu_2)^2 + \lambda_1^*(0,0)\lambda_1^*(1,0)(\mu_2)^3 \\ &+ 2\lambda_1^*(0,0)\lambda_2^*(1,0)(\mu_1)^2\mu_2 + 2\lambda_1^*(0,0)\lambda_2^*(1,0)\mu_1(\mu_2)^2 + 2\lambda_1^*(0,0)(\mu_1)^2(\mu_2)^2 \\ &+ 2\lambda_1^*(0,0)\mu_1(\mu_2)^3 + \lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)(\mu_2)^2 + 2\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1\mu_2 \\ &+ 2\lambda_2^*(0,0)\lambda_1^*(0,1)(\mu_1)^2\mu_2 + 2\lambda_2^*(0,0)\lambda_1^*(0,1)\mu_1(\mu_2)^2 + \lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)(\mu_1)^2 \\ &+ \lambda_2^*(0,0)\lambda_2^*(0,1)(\mu_1)^3 + \lambda_2^*(0,0)\lambda_2^*(0,1)(\mu_1)^2\mu_2 + 2\lambda_2^*(0,0)\lambda_2^*(1,0)(\mu_1)^2\mu_2 \\ &+ 2\lambda_2^*(0,0)(\mu_1)^3\mu_2 + 2\lambda_2^*(0,0)(\mu_1)^2(\mu_2)^2 + 2\lambda_1^*(0,1)(\mu_1)^2(\mu_2)^2 \\ &+ 2\lambda_2^*(1,0)(\mu_1)^2(\mu_2)^2 + 2(\mu_1)^3(\mu_2)^2 + 2(\mu_1)^2(\mu_2)^3. \end{aligned}$$

Constructing the static pricing policy based on Eqs. (3.5) and (3.6) using Eq. (3.8), we then express Eq. (3.7) in terms of $\lambda_j^*(0,0)$, $\lambda_j^*(1,0), \lambda_j^*(0,1), \mu_j, \forall j = 1,2$. The resulting equation has 219 terms in both numerator and denominator and the full expression can be found in Section 3.7. Finding the minimum coefficient of each term in numerator and denominator yields the 80% result. □

3.5 Substitutable Demand

In some applications, there may exist substitution effects in demand between different classes of customers where increasing the price for one class may shift the customer from that class to other classes. One example of such a situation in the rotable selling business is where some rotable spare parts can be both rented and sold. Customers choose to rent the rotable spare part if they have an outside contractor that can repair their broken part and the total price (rental price from the OEM plus the repair price from the contractor) is less than the exchange sale price from the OEM. In such a setting, increasing the rental price may push the customer from renting to buying from the OEM. Also, exchange sale customers may choose to rent if the exchange sale price increases or rental price decreases. In this section, we analyze and compare the system with substitution to the system without substitution and show that if all parameters stay the same as in the system without substitution, introducing substitution effect will improve the expected profit rate in selling the reusable resources.

Let $\lambda_j^0(\mathbf{p})$ denote the demand function for every class j in the system where there are no substitution effects in demand of different customer classes. Denote $\lambda_j^1(\mathbf{p})$ the demand function for class j customers in the system with substitution effects. We make following assumptions to the demand functions.

Assumption 3.5.1. *Let $f(x_1, \ldots, x_m)$ be an m-dimensional non-decreasing function, the demand functions $\lambda_j^0(\mathbf{p})$ and $\lambda_j^1(\mathbf{p})$ have the following properties:*

- $\lambda_j^0(\mathbf{p}) = f(0, \ldots, a_{jj}p_j, \ldots, 0)$ *where* $a_{jj} < 0$.

- $\lambda_j^1(\mathbf{p}) = f(a_{j1}p_1, \ldots, a_{jj}p_j, \ldots, a_{jm}p_m)$ *where* $a_{jk} \geq 0$ *for all* $k \neq j$.

In the example of linear demand functions of the form of $\boldsymbol{\lambda} = A\mathbf{p} + \mathbf{b}$, the coefficient matrix A^0 of a non-substitution system has the property that the diagonal entries $a_{jj} < 0$ and other entries $a^0_{jk} = 0$ for all $j \neq k$. While in the substitution case, the coefficient matrix A^1 has the same diagonal entries as A^0 and at least one of the off-diagonal entries $a_{jk} > 0$.

Theorem 3.5.1. *Under Assumption 3.5.1, for any* C*,* M*,* $\mu_j, b_j,\ j = 1, \ldots, M$*, let* $\mathcal{P}^*_{sub}$ *and* $\mathcal{P}^*_{reg}$ *denote the optimal expected profit rate with and without substitution effects, respectively. We have*

$$\mathcal{P}^*_{sub} \geq \mathcal{P}^*_{reg}.$$

Proof. For any state s, let $\lambda^{0^*}_{js}$ denote the optimal demand rate for class j customers under $\lambda^0_j(\mathbf{p})$ (no substitution), and denote $p^{0^*}_{js}$ the associated optimal price and $\mathbb{P}^{0^*}_s$ the steady-state probability of state s under the optimal policy.

Let $\mathcal{P}^{\lambda^{0^*}_{js}}_{\text{sub}}$ denote applying the optimal demand rate $\lambda^{0^*}_{js}$ under $\lambda^0_j(\mathbf{p})$ to the system of substitution effects $\lambda^1_j(\mathbf{p})$, clearly we have

$$\mathcal{P}^*_{\text{sub}} \geq \mathcal{P}^{\lambda^{0^*}_{js}}_{\text{sub}}$$

because λ^*_{js} is a feasible policy to the system under $\lambda^1_j(\mathbf{p})$.

Furthermore, for any state s, the steady-state probability in $\mathcal{P}^{\lambda^*_{js}}_{\text{sub}}$ is exactly the same as that in $\mathcal{P}^*_{\text{reg}}$, which equals $\mathbb{P}^*_s$. This is because the transition rate for any state s is the same in two systems.

Let p_{js}^1 be the price to achieve $\lambda_{js}^{0^*}$ in state s for class j in the system with substitution effects. Recall in $\lambda_j^1(\mathbf{p})$, the demand rate is increasing in at least one p_k. Therefore, $p_{js}^1 \geq p_{js}^{0^*}$.

Hence, we have

$$\mathcal{P}_{\text{sub}}^* \geq \mathcal{P}_{\text{sub}}^{\lambda_{js}^{0^*}} = \sum_{s\in\mathcal{S}} \lambda_{js}^{0^*}(p_{js}^1 - c_j)\mathbb{P}_s^* \geq \sum_{s\in\mathcal{S}} \lambda_{js}^{0^*}(p_{js}^{0^*} - c_j)\mathbb{P}_s^* = \mathcal{P}_{\text{reg}}^*.$$

□

This theorem establish the fact that the presence of the substitution effects is beneficial to the firm in maximizing the profit rate in selling reusable resources.

Next, we provide the numerical experiment showing the extra profit rate potential in a substitution system. The inputs for the system with no substitution effects are generated as the same in previous sections. For the substitution system, we additionally set $a_{12} = 0.02$ and $a_{21} = 0.01$. We keep the cross-class effects (a_{12} and a_{21}) to be the same in the numerical experiment. Since we are varying a_{11} and a_{22}, keeping a_{12} and a_{21} the same indeed reflects the different levels of substitution effect. We calculate the improvement by introducing substitution effects, i.e., $\frac{\mathcal{P}_{\text{sub}}^* - \mathcal{P}_{\text{reg}}^*}{\mathcal{P}_{\text{reg}}^*}$, and the results of the numerical experiment are summarized in Table 3.4. As one may observe, the average and maximum improvements

C	2	3	4	5	6	7	8
Minimum	2.3%	2.3%	2.4%	2.4%	2.4%	2.4%	2.4%
Average	10.0%	9.3%	8.8%	8.4%	8.2%	8.0%	7.9%
Maximum	51.1%	45.1%	39.5%	35.6%	32.8%	30.7%	29.1%

Table 3.4: Improvement by introducing substitution effects, $M = 2$

by introducing the substitution effects diminish as the number of units the firm manages increases while the minimum improvement almost stays flat. However, on average, the substitution effects can still bring a significant amount of additional profit rate in selling reusable resources.

In addition, we evaluate the performance of both the level-dependent policy and the static policy (introduced in Section 3.3.1) in the system with substitution and the results can be found in Table 3.5. Notice that the performance of both policies on the system with

	C	2	3	4	5	6	7	8
Level Policy	Average	99.9%	99.9%	99.8%	99.8%	99.9%	99.9%	99.9%
	Worst	98.8%	99.1%	99.2%	99.4%	99.5%	99.5%	99.6%
Static Policy	Average	99.2%	98.6%	98.3%	98.1%	98.1%	98.1%	98.1%
	Worst	95.2%	94.7%	94.4%	94.3%	94.5%	94.8%	95.2%

Table 3.5: Performance of simple pricing heuristics in the system with substitution effects, $M = 2$

substitution effects mimics that in the non substitution system as the optimality gaps are almost identical for each level of C. This suggests that even when facing a system with substitution effects, one can still adopt simple pricing heuristics from Section 3.3.1 which will provide the majority of the profit rate gained under the optimal dynamic pricing policy.

3.6 Future Directions

We believe this work has provided promising evidence that a static pricing policy is near optimal in selling reusable resources even facing multiple classes of customers. Consequently, there are many research directions to consider. For example, one can try to generalize the bound in Theorem 3.4.1 to arbitrary C and M. In addition, it would be interesting to characterize the performance of the split system, this could relate to the literature on dedicated versus flexible systems. Finally, it would be of value to further consider the substitution effects setting where introducing substitution will change the parameters in the original demand functions, as this is closer to the situation in many real world applications.

3.7 Proof of 80% Bound

The numerator of Eq. (3.7) after plugging in the constructed static policy is

$$\begin{aligned}
&\lambda_1^*(0,0)^2\lambda_1^*(0,1)^2\lambda_1^*(1,0)^2\mu_2^4 + 4\lambda_1^*(0,0)^2\lambda_1^*(0,1)^2\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1\mu_2^3 \\
&+ 4\lambda_1^*(0,0)^2\lambda_1^*(0,1)^2\lambda_1^*(1,0)\mu_1\mu_2^4 + 4\lambda_1^*(0,0)^2\lambda_1^*(0,1)^2\lambda_2^*(1,0)^2\mu_1^2\mu_2^2 \\
&+ 8\lambda_1^*(0,0)^2\lambda_1^*(0,1)^2\lambda_2^*(1,0)\mu_1^2\mu_2^3 + 4\lambda_1^*(0,0)^2\lambda_1^*(0,1)^2\mu_1^2\mu_2^4 \\
&+ 2\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^2 + 4\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_2^*(1,0)^2\mu_1^3\mu_2 \\
&+ 4\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^3\mu_2^2 + 2\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_1^*(1,0)^2\mu_1\mu_2^4 \\
&+ 2\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_1^*(1,0)^2\mu_2^5 + 8\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^3 \\
&+ 8\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1\mu_2^4 + 8\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_1^*(1,0)\mu_1^2\mu_2^4 \\
&+ 8\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_1^*(1,0)\mu_1\mu_2^5 + 8\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(1,0)^2\mu_1^3\mu_2^2 \\
&+ 8\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(1,0)^2\mu_1^2\mu_2^3 + 16\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^3\mu_2^3 \\
&+ 16\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^2\mu_2^4 + 8\lambda_1^*(0,0)^2\lambda_1^*(0,1)\mu_1^3\mu_2^4 + 8\lambda_1^*(0,0)^2\lambda_1^*(0,1)\mu_1^2\mu_2^5 \\
&+ \lambda_1^*(0,0)^2\lambda_2^*(0,1)^2\lambda_2^*(1,0)^2\mu_1^4 + 2\lambda_1^*(0,0)^2\lambda_2^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^3\mu_2^2 \\
&+ 2\lambda_1^*(0,0)^2\lambda_2^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^3 + 4\lambda_1^*(0,0)^2\lambda_2^*(0,1)\lambda_2^*(1,0)^2\mu_1^4\mu_2 \\
&+ 4\lambda_1^*(0,0)^2\lambda_2^*(0,1)\lambda_2^*(1,0)^2\mu_1^3\mu_2^2 + 4\lambda_1^*(0,0)^2\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^2 \\
&+ 4\lambda_1^*(0,0)^2\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^3\mu_2^3 + \lambda_1^*(0,0)^2\lambda_1^*(1,0)^2\mu_1^2\mu_2^4 + 2\lambda_1^*(0,0)^2\lambda_1^*(1,0)^2\mu_1\mu_2^5 \\
&+ \lambda_1^*(0,0)^2\lambda_1^*(1,0)^2\mu_2^6 + 4\lambda_1^*(0,0)^2\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^3\mu_2^3 + 8\lambda_1^*(0,0)^2\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^4
\end{aligned}$$

$$+ 4\lambda_1^*(0,0)^2\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1\mu_2^5 + 4\lambda_1^*(0,0)^2\lambda_1^*(1,0)\mu_1^3\mu_2^4 + 8\lambda_1^*(0,0)^2\lambda_1^*(1,0)\mu_1^2\mu_2^5$$

$$+ 4\lambda_1^*(0,0)^2\lambda_1^*(1,0)\mu_1\mu_2^6 + 4\lambda_1^*(0,0)^2\lambda_2^*(1,0)^2\mu_1^4\mu_2^2 + 8\lambda_1^*(0,0)^2\lambda_2^*(1,0)^2\mu_1^3\mu_2^3$$

$$+ 4\lambda_1^*(0,0)^2\lambda_2^*(1,0)^2\mu_1^2\mu_2^4 + 8\lambda_1^*(0,0)^2\lambda_2^*(1,0)\mu_1^4\mu_2^3 + 16\lambda_1^*(0,0)^2\lambda_2^*(1,0)\mu_1^3\mu_2^4$$

$$+ 8\lambda_1^*(0,0)^2\lambda_2^*(1,0)\mu_1^2\mu_2^5 + 4\lambda_1^*(0,0)^2\mu_1^4\mu_2^4 + 8\lambda_1^*(0,0)^2\mu_1^3\mu_2^5 + 4\lambda_1^*(0,0)^2\mu_1^2\mu_2^6$$

$$+ 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)^2\lambda_1^*(1,0)^2\mu_2^4 + 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)^2\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1\mu_2^3$$

$$+ 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)^2\lambda_1^*(1,0)\mu_1^2\mu_2^3 + 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)^2\lambda_1^*(1,0)\mu_1\mu_2^4$$

$$+ 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)^2\lambda_2^*(1,0)^2\mu_1^2\mu_2^2 + 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)^2\lambda_2^*(1,0)\mu_1^3\mu_2^2$$

$$+ 16\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)^2\lambda_2^*(1,0)\mu_1^2\mu_2^3 + 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)^2\mu_1^3\mu_2^3$$

$$+ 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)^2\mu_1^2\mu_2^4 + 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^2$$

$$+ 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_1^*(1,0)\mu_1^3\mu_2^2 + 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_1^*(1,0)\mu_1^2\mu_2^3$$

$$+ 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_2^*(1,0)^2\mu_1^3\mu_2$$

$$+ 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2$$

$$+ 12\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^3\mu_2^2 + 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\mu_1^4\mu_2^2$$

$$+ 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\mu_1^3\mu_2^3 + 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)^2\mu_1\mu_2^4$$

$$+ 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)^2\mu_2^5 + 12\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^3$$

$$+ 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1\mu_2^4 + 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)\mu_1^3\mu_2^3$$

$$+ 16\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)\mu_1^2\mu_2^4 + 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)\mu_1\mu_2^5$$

$$+ 16\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)^2\mu_1^3\mu_2^2 + 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)^2\mu_1^2\mu_2^3$$

$$+ 16\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^2 + 40\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^3\mu_2^3$$

$$+ 16\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^2\mu_2^4 + 16\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\mu_1^4\mu_2^3$$

$$+ 24\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\mu_1^3\mu_2^4 + 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\mu_1^2\mu_2^5$$

$$+ 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)^2\lambda_2^*(1,0)^2\mu_1^4 + 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)^2\lambda_2^*(1,0)\mu_1^5$$

$$+ 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)^2\lambda_2^*(1,0)\mu_1^4\mu_2 + 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^3\mu_2^2$$

$$+ 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^3 + 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_1^*(1,0)\mu_1^4\mu_2^2$$

$$+ 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_1^*(1,0)\mu_1^3\mu_2^3 + 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_1^*(1,0)\mu_1^2\mu_2^4$$

$$+ 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)^2\mu_1^4\mu_2 + 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)^2\mu_1^3\mu_2^2$$

$$+ 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^5\mu_2 + 16\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^2$$

$$+ 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^3\mu_2^3 + 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\mu_1^5\mu_2^2$$

$$+ 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\mu_1^4\mu_2^3 + 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\mu_1^3\mu_2^4$$

$$+ 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^3\mu_2^3 + 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^4$$

$$+ 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(1,0)\mu_1^4\mu_2^3 + 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(1,0)\mu_1^3\mu_2^4$$

$$+ 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(1,0)\mu_1^2\mu_2^5 + 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(1,0)^2\mu_1^4\mu_2^2$$

$$+ 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(1,0)^2\mu_1^3\mu_2^3 + 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(1,0)\mu_1^5\mu_2^2$$

$$+ 24\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(1,0)\mu_1^4\mu_2^3 + 16\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(1,0)\mu_1^3\mu_2^4$$

$$+ 8\lambda_1^*(0,0)\lambda_2^*(0,0)\mu_1^5\mu_2^3 + 16\lambda_1^*(0,0)\lambda_2^*(0,0)\mu_1^4\mu_2^4 + 8\lambda_1^*(0,0)\lambda_2^*(0,0)\mu_1^3\mu_2^5$$

$$+ 3\lambda_1^*(0,0)\lambda_1^*(0,1)^2\lambda_1^*(1,0)\mu_1^2\mu_2^4 + 6\lambda_1^*(0,0)\lambda_1^*(0,1)^2\lambda_2^*(1,0)\mu_1^3\mu_2^3 + 6\lambda_1^*(0,0)\lambda_1^*(0,1)^2\mu_1^3\mu_2^4$$

$$+ 3\lambda_1^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^2 + 3\lambda_1^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^4$$

$$+ 6\lambda_1^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)\mu_1^3\mu_2^4 + 6\lambda_1^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)\mu_1^2\mu_2^5$$

$$+ 6\lambda_1^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)^2\mu_1^3\mu_2^3 + 12\lambda_1^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^3$$

$$+ 18\lambda_1^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^3\mu_2^4 + 12\lambda_1^*(0,0)\lambda_1^*(0,1)\mu_1^4\mu_2^4 + 12\lambda_1^*(0,0)\lambda_1^*(0,1)\mu_1^3\mu_2^5$$

$$+ 3\lambda_1^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)^2\mu_1^4\mu_2^2 + 3\lambda_1^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^5\mu_2^2$$

$$+ 3\lambda_1^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^3 + 3\lambda_1^*(0,0)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^3\mu_2^4$$

$$+ 3\lambda_1^*(0,0)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^5 + 3\lambda_1^*(0,0)\lambda_1^*(1,0)\mu_1^4\mu_2^4 + 6\lambda_1^*(0,0)\lambda_1^*(1,0)\mu_1^3\mu_2^5$$

$$+ 3\lambda_1^*(0,0)\lambda_1^*(1,0)\mu_1^2\mu_2^6 + 6\lambda_1^*(0,0)\lambda_2^*(1,0)^2\mu_1^4\mu_2^3 + 6\lambda_1^*(0,0)\lambda_2^*(1,0)^2\mu_1^3\mu_2^4$$

$$+ 6\lambda_1^*(0,0)\lambda_2^*(1,0)\mu_1^5\mu_2^3 + 18\lambda_1^*(0,0)\lambda_2^*(1,0)\mu_1^4\mu_2^4 + 12\lambda_1^*(0,0)\lambda_2^*(1,0)\mu_1^3\mu_2^5$$

$$+ 6\lambda_1^*(0,0)\mu_1^5\mu_2^4 + 12\lambda_1^*(0,0)\mu_1^4\mu_2^5 + 6\lambda_1^*(0,0)\mu_1^3\mu_2^6 + \lambda_2^*(0,0)^2\lambda_1^*(0,1)^2\lambda_1^*(1,0)^2\mu_2^4$$

$$+ 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)^2\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1\mu_2^3 + 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)^2\lambda_1^*(1,0)\mu_1^2\mu_2^3$$

$$+ 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)^2\lambda_1^*(1,0)\mu_1\mu_2^4 + 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)^2\lambda_2^*(1,0)^2\mu_1^2\mu_2^2$$

$$+ 8\lambda_2^*(0,0)^2\lambda_1^*(0,1)^2\lambda_2^*(1,0)\mu_1^3\mu_2^2 + 8\lambda_2^*(0,0)^2\lambda_1^*(0,1)^2\lambda_2^*(1,0)\mu_1^2\mu_2^3$$

$$+ 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)^2\mu_1^4\mu_2^2 + 8\lambda_2^*(0,0)^2\lambda_1^*(0,1)^2\mu_1^3\mu_2^3 + 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)^2\mu_1^2\mu_2^4$$

$$+ 2\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^2 + 2\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_1^*(1,0)\mu_1^3\mu_2^2$$

$$+ 2\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_1^*(1,0)\mu_1^2\mu_2^3 + 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_2^*(1,0)^2\mu_1^3\mu_2$$

$$+ 8\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2 + 8\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^3\mu_2^2$$

$$+ 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\mu_1^5\mu_2 + 8\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\mu_1^4\mu_2^2$$

$$+ 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\mu_1^3\mu_2^3 + 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^3$$

$$+ 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_1^*(1,0)\mu_1^3\mu_2^3 + 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_1^*(1,0)\mu_1^2\mu_2^4$$

$$+ 8\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(1,0)^2\mu_1^3\mu_2^2 + 16\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^2$$

$$+ 16\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^3\mu_2^3 + 8\lambda_2^*(0,0)^2\lambda_1^*(0,1)\mu_1^5\mu_2^2$$

$$+ 16\lambda_2^*(0,0)^2\lambda_1^*(0,1)\mu_1^4\mu_2^3 + 8\lambda_2^*(0,0)^2\lambda_1^*(0,1)\mu_1^3\mu_2^4 + \lambda_2^*(0,0)^2\lambda_2^*(0,1)^2\lambda_2^*(1,0)^2\mu_1^4$$

$$+ 2\lambda_2^*(0,0)^2\lambda_2^*(0,1)^2\lambda_2^*(1,0)\mu_1^5 + 2\lambda_2^*(0,0)^2\lambda_2^*(0,1)^2\lambda_2^*(1,0)\mu_1^4\mu_2 + \lambda_2^*(0,0)^2\lambda_2^*(0,1)^2\mu_1^6$$

$$+ 2\lambda_2^*(0,0)^2\lambda_2^*(0,1)^2\mu_1^5\mu_2 + \lambda_2^*(0,0)^2\lambda_2^*(0,1)^2\mu_1^4\mu_2^2 + 4\lambda_2^*(0,0)^2\lambda_2^*(0,1)\lambda_2^*(1,0)^2\mu_1^4\mu_2$$

$$+ 8\lambda_2^*(0,0)^2\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^5\mu_2 + 8\lambda_2^*(0,0)^2\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^2 + 4\lambda_2^*(0,0)^2\lambda_2^*(0,1)\mu_1^6\mu_2$$

$$+ 8\lambda_2^*(0,0)^2\lambda_2^*(0,1)\mu_1^5\mu_2^2 + 4\lambda_2^*(0,0)^2\lambda_2^*(0,1)\mu_1^4\mu_2^3 + 4\lambda_2^*(0,0)^2\lambda_2^*(1,0)^2\mu_1^4\mu_2^2$$

$$+ 8\lambda_2^*(0,0)^2\lambda_2^*(1,0)\mu_1^5\mu_2^2 + 8\lambda_2^*(0,0)^2\lambda_2^*(1,0)\mu_1^4\mu_2^3 + 4\lambda_2^*(0,0)^2\mu_1^6\mu_2^2 + 8\lambda_2^*(0,0)^2\mu_1^5\mu_2^3$$

$$+ 4\lambda_2^*(0,0)^2\mu_1^4\mu_2^4 + 3\lambda_2^*(0,0)\lambda_1^*(0,1)^2\lambda_1^*(1,0)\mu_1^2\mu_2^4 + 6\lambda_2^*(0,0)\lambda_1^*(0,1)^2\lambda_2^*(1,0)\mu_1^3\mu_2^3$$

$$+ 6\lambda_2^*(0,0)\lambda_1^*(0,1)^2\mu_1^4\mu_2^3 + 6\lambda_2^*(0,0)\lambda_1^*(0,1)^2\mu_1^3\mu_2^4 + 3\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^2$$

$$+ 3\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\mu_1^5\mu_2^2 + 3\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\mu_1^4\mu_2^3$$

$$+ 3\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^4 + 3\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)\mu_1^3\mu_2^4$$

$$+ 3\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)\mu_1^2\mu_2^5 + 6\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)^2\mu_1^3\mu_2^3$$

$$+ 18\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^3 + 12\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^3\mu_2^4 + 12\lambda_2^*(0,0)\lambda_1^*(0,1)\mu_1^5\mu_2^3$$

$$+ 18\lambda_2^*(0,0)\lambda_1^*(0,1)\mu_1^4\mu_2^4 + 6\lambda_2^*(0,0)\lambda_1^*(0,1)\mu_1^3\mu_2^5 + 3\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)^2\mu_1^4\mu_2^2$$

$$+ 6\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^5\mu_2^2 + 6\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^3 + 3\lambda_2^*(0,0)\lambda_2^*(0,1)\mu_1^6\mu_2^2$$

$$+ 6\lambda_2^*(0,0)\lambda_2^*(0,1)\mu_1^5\mu_2^3 + 3\lambda_2^*(0,0)\lambda_2^*(0,1)\mu_1^4\mu_2^4 + 6\lambda_2^*(0,0)\lambda_2^*(1,0)^2\mu_1^4\mu_2^3$$

$$+ 12\lambda_2^*(0,0)\lambda_2^*(1,0)\mu_1^5\mu_2^3 + 12\lambda_2^*(0,0)\lambda_2^*(1,0)\mu_1^4\mu_2^4 + 6\lambda_2^*(0,0)\mu_1^6\mu_2^3 + 12\lambda_2^*(0,0)\mu_1^5\mu_2^4$$

$$+ 6\lambda_2^*(0,0)\mu_1^4\mu_2^5 + 2\lambda_1^*(0,1)^2\mu_1^4\mu_2^4 + 4\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^4 + 4\lambda_1^*(0,1)\mu_1^5\mu_2^4 + 4\lambda_1^*(0,1)\mu_1^4\mu_2^5$$

$$+ 2\lambda_2^*(1,0)^2\mu_1^4\mu_2^4 + 4\lambda_2^*(1,0)\mu_1^5\mu_2^4 + 4\lambda_2^*(1,0)\mu_1^4\mu_2^5 + 2\mu_1^6\mu_2^4 + 4\mu_1^5\mu_2^5 + 2\mu_1^4\mu_2^6,$$

and the denominator is

$$\lambda_1^*(0,0)^2\lambda_1^*(0,1)^2\lambda_1^*(1,0)^2\mu_2^4 + 4\lambda_1^*(0,0)^2\lambda_1^*(0,1)^2\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1\mu_2^3$$

$$+ 4\lambda_1^*(0,0)^2\lambda_1^*(0,1)^2\lambda_1^*(1,0)\mu_1\mu_2^4 + 4\lambda_1^*(0,0)^2\lambda_1^*(0,1)^2\lambda_2^*(1,0)^2\mu_1^2\mu_2^2$$

$$+ 8\lambda_1^*(0,0)^2\lambda_1^*(0,1)^2\lambda_2^*(1,0)\mu_1^2\mu_2^3 + 5\lambda_1^*(0,0)^2\lambda_1^*(0,1)^2\mu_1^2\mu_2^4$$

$$+ 2\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^2 + 4\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_2^*(1,0)^2\mu_1^3\mu_2$$

$$+ 4\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^3\mu_2^2 + 2\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_1^*(1,0)^2\mu_1\mu_2^4$$

$$+ 2\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_1^*(1,0)^2\mu_2^5 + 8\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^3$$

$$+ 8\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1\mu_2^4 + 8\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_1^*(1,0)\mu_1^2\mu_2^4$$

$$+ 8\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_1^*(1,0)\mu_1\mu_2^5 + 8\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(1,0)^2\mu_1^3\mu_2^2$$

$$+ 8\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(1,0)^2\mu_1^2\mu_2^3 + 18\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^3\mu_2^3$$

$$+ 16\lambda_1^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^2\mu_2^4 + 10\lambda_1^*(0,0)^2\lambda_1^*(0,1)\mu_1^3\mu_2^4 + 10\lambda_1^*(0,0)^2\lambda_1^*(0,1)\mu_1^2\mu_2^5$$

$$+ \lambda_1^*(0,0)^2\lambda_2^*(0,1)^2\lambda_2^*(1,0)^2\mu_1^4 + 2\lambda_1^*(0,0)^2\lambda_2^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^3\mu_2^2$$

$$+ 2\lambda_1^*(0,0)^2\lambda_2^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^3 + 4\lambda_1^*(0,0)^2\lambda_2^*(0,1)\lambda_2^*(1,0)^2\mu_1^4\mu_2$$

$$+ 4\lambda_1^*(0,0)^2\lambda_2^*(0,1)\lambda_2^*(1,0)^2\mu_1^3\mu_2^2 + 4\lambda_1^*(0,0)^2\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^2$$

$$+ 4\lambda_1^*(0,0)^2\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^3\mu_2^3 + \lambda_1^*(0,0)^2\lambda_1^*(1,0)^2\mu_1^2\mu_2^4 + 2\lambda_1^*(0,0)^2\lambda_1^*(1,0)^2\mu_1\mu_2^5$$

$$+ \lambda_1^*(0,0)^2\lambda_1^*(1,0)^2\mu_2^6 + 4\lambda_1^*(0,0)^2\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^3\mu_2^3 + 8\lambda_1^*(0,0)^2\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^4$$

$$+ 4\lambda_1^*(0,0)^2\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1\mu_2^5 + 4\lambda_1^*(0,0)^2\lambda_1^*(1,0)\mu_1^3\mu_2^4 + 8\lambda_1^*(0,0)^2\lambda_1^*(1,0)\mu_1^2\mu_2^5$$

$$+ 4\lambda_1^*(0,0)^2\lambda_1^*(1,0)\mu_1\mu_2^6 + 5\lambda_1^*(0,0)^2\lambda_2^*(1,0)^2\mu_1^4\mu_2^2 + 8\lambda_1^*(0,0)^2\lambda_2^*(1,0)^2\mu_1^3\mu_2^3$$

$$+ 4\lambda_1^*(0,0)^2\lambda_2^*(1,0)^2\mu_1^2\mu_2^4 + 10\lambda_1^*(0,0)^2\lambda_2^*(1,0)\mu_1^4\mu_2^3 + 18\lambda_1^*(0,0)^2\lambda_2^*(1,0)\mu_1^3\mu_2^4$$

$$+ 8\lambda_1^*(0,0)^2\lambda_2^*(1,0)\mu_1^2\mu_2^5 + 5\lambda_1^*(0,0)^2\mu_1^4\mu_2^4 + 10\lambda_1^*(0,0)^2\mu_1^3\mu_2^5 + 5\lambda_1^*(0,0)^2\mu_1^2\mu_2^6$$

$$+ 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)^2\lambda_1^*(1,0)^2\mu_2^4 + 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)^2\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1\mu_2^3$$

$$+ 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)^2\lambda_1^*(1,0)\mu_1^2\mu_2^3 + 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)^2\lambda_1^*(1,0)\mu_1\mu_2^4$$

$$+ 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)^2\lambda_2^*(1,0)^2\mu_1^2\mu_2^2 + 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)^2\lambda_2^*(1,0)\mu_1^3\mu_2^2$$

$$+ 16\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)^2\lambda_2^*(1,0)\mu_1^2\mu_2^3 + 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)^2\mu_1^3\mu_2^3$$

$$+ 10\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)^2\mu_1^2\mu_2^4 + 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^2$$

$$+ 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_1^*(1,0)\mu_1^3\mu_2^2 + 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_1^*(1,0)\mu_1^2\mu_2^3$$

$$+ 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_2^*(1,0)^2\mu_1^3\mu_2$$

$$+ 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2$$

$$+ 12\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^3\mu_2^2 + 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\mu_1^4\mu_2^2$$

$$+ 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\mu_1^3\mu_2^3 + 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)^2\mu_1\mu_2^4$$

$$+ 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)^2\mu_2^5 + 12\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^3$$

$$+ 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1\mu_2^4 + 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)\mu_1^3\mu_2^3$$

$$+ 16\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)\mu_1^2\mu_2^4 + 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)\mu_1\mu_2^5$$

$$+ 16\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)^2\mu_1^3\mu_2^2 + 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)^2\mu_1^2\mu_2^3$$

$$+ 16\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^2 + 44\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^3\mu_2^3$$

$$+ 16\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^2\mu_2^4 + 18\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\mu_1^4\mu_2^3$$

$$+ 28\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\mu_1^3\mu_2^4 + 10\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(0,1)\mu_1^2\mu_2^5$$

$$+ 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)^2\lambda_2^*(1,0)^2\mu_1^4 + 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)^2\lambda_2^*(1,0)\mu_1^5$$

$$+ 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)^2\lambda_2^*(1,0)\mu_1^4\mu_2 + 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^3\mu_2^2$$

$$+ 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^3 + 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_1^*(1,0)\mu_1^4\mu_2^2$$

$$+ 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_1^*(1,0)\mu_1^3\mu_2^3 + 2\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_1^*(1,0)\mu_1^2\mu_2^4$$

$$+ 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)^2\mu_1^4\mu_2 + 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)^2\mu_1^3\mu_2^2$$

$$+ 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^5\mu_2 + 16\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^2$$

$$+ 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^3\mu_2^3 + 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\mu_1^5\mu_2^2$$

$$+ 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\mu_1^4\mu_2^3 + 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(0,1)\mu_1^3\mu_2^4$$

$$+ 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^3\mu_2^3 + 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^4$$

$$+ 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(1,0)\mu_1^4\mu_2^3 + 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(1,0)\mu_1^3\mu_2^4$$

$$+ 4\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_1^*(1,0)\mu_1^2\mu_2^5 + 10\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(1,0)^2\mu_1^4\mu_2^2$$

$$+ 8\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(1,0)^2\mu_1^3\mu_2^3 + 10\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(1,0)\mu_1^5\mu_2^2$$

$$+ 28\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(1,0)\mu_1^4\mu_2^3 + 18\lambda_1^*(0,0)\lambda_2^*(0,0)\lambda_2^*(1,0)\mu_1^3\mu_2^4$$

$$+ 10\lambda_1^*(0,0)\lambda_2^*(0,0)\mu_1^5\mu_2^3 + 20\lambda_1^*(0,0)\lambda_2^*(0,0)\mu_1^4\mu_2^4 + 10\lambda_1^*(0,0)\lambda_2^*(0,0)\mu_1^3\mu_2^5$$

$$+ 2\lambda_1^*(0,0)\lambda_1^*(0,1)^2\lambda_1^*(1,0)\mu_1^2\mu_2^4 + 4\lambda_1^*(0,0)\lambda_1^*(0,1)^2\lambda_2^*(1,0)\mu_1^3\mu_2^3 + 6\lambda_1^*(0,0)\lambda_1^*(0,1)^2\mu_1^3\mu_2^4$$

$$+ 2\lambda_1^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^2 + 2\lambda_1^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^4$$

$$+ 4\lambda_1^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)\mu_1^3\mu_2^4 + 4\lambda_1^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)\mu_1^2\mu_2^5$$

$$+ 4\lambda_1^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)^2\mu_1^3\mu_2^3 + 10\lambda_1^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^3$$

$$+ 14\lambda_1^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^3\mu_2^4 + 12\lambda_1^*(0,0)\lambda_1^*(0,1)\mu_1^4\mu_2^4 + 12\lambda_1^*(0,0)\lambda_1^*(0,1)\mu_1^3\mu_2^5$$

$$+ 2\lambda_1^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)^2\mu_1^4\mu_2^2 + 2\lambda_1^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^5\mu_2^2$$

$$+ 2\lambda_1^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^3 + 2\lambda_1^*(0,0)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^3\mu_2^4$$

$$+ 2\lambda_1^*(0,0)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^5 + 2\lambda_1^*(0,0)\lambda_1^*(1,0)\mu_1^4\mu_2^4 + 4\lambda_1^*(0,0)\lambda_1^*(1,0)\mu_1^3\mu_2^5$$

$$+ 2\lambda_1^*(0,0)\lambda_1^*(1,0)\mu_1^2\mu_2^6 + 6\lambda_1^*(0,0)\lambda_2^*(1,0)^2\mu_1^4\mu_2^3 + 4\lambda_1^*(0,0)\lambda_2^*(1,0)^2\mu_1^3\mu_2^4$$

$$+ 6\lambda_1^*(0,0)\lambda_2^*(1,0)\mu_1^5\mu_2^3 + 16\lambda_1^*(0,0)\lambda_2^*(1,0)\mu_1^4\mu_2^4 + 10\lambda_1^*(0,0)\lambda_2^*(1,0)\mu_1^3\mu_2^5$$

$$+ 6\lambda_1^*(0,0)\mu_1^5\mu_2^4 + 12\lambda_1^*(0,0)\mu_1^4\mu_2^5 + 6\lambda_1^*(0,0)\mu_1^3\mu_2^6 + \lambda_2^*(0,0)^2\lambda_1^*(0,1)^2\lambda_1^*(1,0)^2\mu_2^4$$

$$+ 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)^2\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1\mu_2^3 + 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)^2\lambda_1^*(1,0)\mu_1^2\mu_2^3$$

$$+ 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)^2\lambda_1^*(1,0)\mu_1\mu_2^4 + 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)^2\lambda_2^*(1,0)^2\mu_1^2\mu_2^2$$

$$+ 8\lambda_2^*(0,0)^2\lambda_1^*(0,1)^2\lambda_2^*(1,0)\mu_1^3\mu_2^2 + 8\lambda_2^*(0,0)^2\lambda_1^*(0,1)^2\lambda_2^*(1,0)\mu_1^2\mu_2^3$$

$$+ 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)^2\mu_1^4\mu_2^2 + 8\lambda_2^*(0,0)^2\lambda_1^*(0,1)^2\mu_1^3\mu_2^3 + 5\lambda_2^*(0,0)^2\lambda_1^*(0,1)^2\mu_1^2\mu_2^4$$

$$+ 2\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^2 + 2\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_1^*(1,0)\mu_1^3\mu_2^2$$

$$+ 2\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_1^*(1,0)\mu_1^2\mu_2^3 + 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_2^*(1,0)^2\mu_1^3\mu_2$$

$$+ 8\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2 + 8\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^3\mu_2^2$$

$$+ 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\mu_1^5\mu_2 + 8\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\mu_1^4\mu_2^2$$

$$+ 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(0,1)\mu_1^3\mu_2^3 + 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^3$$

$$+ 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_1^*(1,0)\mu_1^3\mu_2^3 + 4\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_1^*(1,0)\mu_1^2\mu_2^4$$

$$+ 8\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(1,0)^2\mu_1^3\mu_2^2 + 16\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^2$$

$$+ 18\lambda_2^*(0,0)^2\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^3\mu_2^3 + 8\lambda_2^*(0,0)^2\lambda_1^*(0,1)\mu_1^5\mu_2^2 + 18\lambda_2^*(0,0)^2\lambda_1^*(0,1)\mu_1^4\mu_2^3$$

$$+ 10\lambda_2^*(0,0)^2\lambda_1^*(0,1)\mu_1^3\mu_2^4 + \lambda_2^*(0,0)^2\lambda_2^*(0,1)^2\lambda_2^*(1,0)^2\mu_1^4 + 2\lambda_2^*(0,0)^2\lambda_2^*(0,1)^2\lambda_2^*(1,0)\mu_1^5$$

$$+ 2\lambda_2^*(0,0)^2\lambda_2^*(0,1)^2\lambda_2^*(1,0)\mu_1^4\mu_2 + \lambda_2^*(0,0)^2\lambda_2^*(0,1)^2\mu_1^6 + 2\lambda_2^*(0,0)^2\lambda_2^*(0,1)^2\mu_1^5\mu_2$$

$$+ \lambda_2^*(0,0)^2\lambda_2^*(0,1)^2\mu_1^4\mu_2^2 + 4\lambda_2^*(0,0)^2\lambda_2^*(0,1)\lambda_2^*(1,0)^2\mu_1^4\mu_2 + 8\lambda_2^*(0,0)^2\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^5\mu_2$$

$$+ 8\lambda_2^*(0,0)^2\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^2 + 4\lambda_2^*(0,0)^2\lambda_2^*(0,1)\mu_1^6\mu_2 + 8\lambda_2^*(0,0)^2\lambda_2^*(0,1)\mu_1^5\mu_2^2$$

$$+ 4\lambda_2^*(0,0)^2\lambda_2^*(0,1)\mu_1^4\mu_2^3 + 5\lambda_2^*(0,0)^2\lambda_2^*(1,0)^2\mu_1^4\mu_2^2 + 10\lambda_2^*(0,0)^2\lambda_2^*(1,0)\mu_1^5\mu_2^2$$

$$+ 10\lambda_2^*(0,0)^2\lambda_2^*(1,0)\mu_1^4\mu_2^3 + 5\lambda_2^*(0,0)^2\mu_1^6\mu_2^2 + 10\lambda_2^*(0,0)^2\mu_1^5\mu_2^3 + 5\lambda_2^*(0,0)^2\mu_1^4\mu_2^4$$

$$+ 2\lambda_2^*(0,0)\lambda_1^*(0,1)^2\lambda_1^*(1,0)\mu_1^2\mu_2^4 + 4\lambda_2^*(0,0)\lambda_1^*(0,1)^2\lambda_2^*(1,0)\mu_1^3\mu_2^3 + 4\lambda_2^*(0,0)\lambda_1^*(0,1)^2\mu_1^4\mu_2^3$$

$$+ 6\lambda_2^*(0,0)\lambda_1^*(0,1)^2\mu_1^3\mu_2^4 + 2\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^2$$

$$+ 2\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\mu_1^5\mu_2^2 + 2\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(0,1)\mu_1^4\mu_2^3$$

$$+ 2\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)\lambda_2^*(1,0)\mu_1^2\mu_2^4 + 2\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)\mu_1^3\mu_2^4$$

$$+ 2\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_1^*(1,0)\mu_1^2\mu_2^5 + 4\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)^2\mu_1^3\mu_2^3$$

$$+ 14\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^3 + 10\lambda_2^*(0,0)\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^3\mu_2^4 + 10\lambda_2^*(0,0)\lambda_1^*(0,1)\mu_1^5\mu_2^3$$

$$+ 16\lambda_2^*(0,0)\lambda_1^*(0,1)\mu_1^4\mu_2^4 + 6\lambda_2^*(0,0)\lambda_1^*(0,1)\mu_1^3\mu_2^5 + 2\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)^2\mu_1^4\mu_2^2$$

$$+ 4\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^5\mu_2^2 + 4\lambda_2^*(0,0)\lambda_2^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^3 + 2\lambda_2^*(0,0)\lambda_2^*(0,1)\mu_1^6\mu_2^2$$

$$+ 4\lambda_2^*(0,0)\lambda_2^*(0,1)\mu_1^5\mu_2^3 + 2\lambda_2^*(0,0)\lambda_2^*(0,1)\mu_1^4\mu_2^4 + 6\lambda_2^*(0,0)\lambda_2^*(1,0)^2\mu_1^4\mu_2^3$$

$$+ 12\lambda_2^*(0,0)\lambda_2^*(1,0)\mu_1^5\mu_2^3 + 12\lambda_2^*(0,0)\lambda_2^*(1,0)\mu_1^4\mu_2^4 + 6\lambda_2^*(0,0)\mu_1^6\mu_2^3 + 12\lambda_2^*(0,0)\mu_1^5\mu_2^4$$

$$+ 6\lambda_2^*(0,0)\mu_1^4\mu_2^5 + 2\lambda_1^*(0,1)^2\mu_1^4\mu_2^4 + 4\lambda_1^*(0,1)\lambda_2^*(1,0)\mu_1^4\mu_2^4 + 4\lambda_1^*(0,1)\mu_1^5\mu_2^4 + 4\lambda_1^*(0,1)\mu_1^4\mu_2^5$$

$$+ 2\lambda_2^*(1,0)^2\mu_1^4\mu_2^4 + 4\lambda_2^*(1,0)\mu_1^5\mu_2^4 + 4\lambda_2^*(1,0)\mu_1^4\mu_2^5 + 2\mu_1^6\mu_2^4 + 4\mu_1^5\mu_2^5 + 2\mu_1^4\mu_2^6.$$

THIS PAGE INTENTIONALLY LEFT BLANK

Bibliography

Afèche, Philipp. 2013. Incentive-compatible revenue management in queueing systems: optimal strategic delay. *Manufacturing & Service Operations Management* **15**(3) 423–443.

Aisyati, Azizah, Wakhid Ahmad Jauhari, Cucuk Nur Rosyidi. 2013. Determination inventory level for aircraft spare parts using continuous review model. *International Journal of Business Research and Management (IJBRM)* **4**(1) 1–12.

Allen, Stephen G, Donato A D'Esopo. 1968. An ordering policy for repairable stock items. *Operations Research* **16**(3) 669–674.

Arts, Joachim, Simme Douwe Flapper. 2015. Aggregate overhaul and supply chain planning for rotables. *Annals of Operations Research* **224**(1) 77–100.

Ata, Bariş, Shiri Shneorson. 2006. Dynamic control of an m/m/1 service system with adjustable arrival and service rates. *Management Science* **52**(11) 1778–1791.

Banerjee, Siddhartha, Daniel Freund, Thodoris Lykouris. 2016. Pricing and optimization in shared vehicle systems: An approximation framework. *arXiv preprint arXiv:1608.06819* .

Banerjee, Siddhartha, Ramesh Johari, Carlos Riquelme. 2015. Pricing in ride-sharing platforms: A queueing-theoretic approach. *Proceedings of the Sixteenth ACM Conference on Economics and Computation*. ACM, 639–639.

Bertsekas, Dimitri P. 2012. *Dynamic programming and optimal control*, vol. 2. Athena Scientific.

Besbes, Omar, Adam N Elmachtoub, Yunjie Sun. 2019a. Pricing analytics for rotable spare parts .

Besbes, Omar, Adam N Elmachtoub, Yunjie Sun. 2019b. Static pricing: Universal guarantees for reusable resources. *arXiv preprint arXiv:1905.00731* .

Bitran, Gabriel R, Susana V Mondschein. 1997. Periodic pricing of seasonal products in retailing. *Management science* **43**(1) 64–79.

Calmon, Andre P, Stephen C Graves. 2017. Inventory management in a consumer electronics closed-loop supply chain. *Manufacturing & Service Operations Management* **19**(4) 568–585.

Caro, Felipe, Jérémie Gallien. 2012. Clearance pricing optimization for a fast-fashion retailer. *Operations Research* **60**(6) 1404–1422.

Çelik, Sabri, Costis Maglaras. 2008. Dynamic pricing and lead-time quotation for a multiclass make-to-order queue. *Management Science* **54**(6) 1132–1146.

Çelik, Sabri, Alp Muharremoglu, Sergei Savin. 2009. Revenue management with costly price adjustments. *Operations research* **57**(5) 1206–1219.

Ceryan, O., O. Sahin, I. Duenyas. 2013. Dynamic pricing of substitutable products in the presence of capacity flexibility. *Manufacturing & Service Operations Management* **15**(1) 86–101.

Chen, Hong, Owen Q Wu, David D Yao. 2010. On the benefit of inventory-based dynamic pricing strategies. *Production and Operations Management* **19**(3) 249–260.

Chen, Qi, Stefanus Jasin, Izak Duenyas. 2015. Real-time dynamic pricing with minimal and flexible price adjustment. *Management Science* **62**(8) 2437–2455.

Chen, Yiwei, Vivek F Farias, Nikolaos K Trichakis. 2018. On the efficacy of static prices for revenue management in the face of strategic customers. *Management Science* .

Chen, Yiwei, Retsef Levi, Cong Shi. 2017. Revenue management of reusable resources with advanced reservations. *Production and Operations Management* **26**(5) 836–859.

Chen, Yiwei, Cong Shi. 2018. Near-optimal pricing policy for service systems with reusable resources and forward-looking customers .

Chen, Youhua, Saibal Ray, Yuyue Song. 2006. Optimal pricing and inventory control policy in periodic-review systems with fixed ordering cost and lost sales. *Naval Research Logistics (NRL)* **53**(2) 117–136.

Cheung, Wang Chi, David Simchi-Levi, He Wang. 2017. Dynamic pricing and demand learning with limited price experimentation. *Operations Research* **65**(6) 1722–1731.

Cohen, Morris A, Paul R Kleindorfer, Hau L Lee. 1989. Near-optimal service constrained stocking policies for spare parts. *Operations Research* **37**(1) 104–117.

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

Doan, Xuan Vinh, Xiao Lei, Siqian Shen. 2019. Pricing of reusable resources under ambiguous distributions of demand and service time with emerging applications. *European Journal of Operational Research* .

Erkoc, Murat, Kadir Ertogral. 2016. Overhaul planning and exchange scheduling for maintenance services with rotable inventory and limited processing capacity. *Computers & Industrial Engineering* **98** 30–39.

Federgruen, Awi, Aliza Heching. 1999. Combined pricing and inventory control under uncertainty. *Operations research* **47**(3) 454–475.

Feng, Youyi, Guillermo Gallego. 1995. Optimal starting times for end-of-season sales and optimal stopping times for promotional fares. *Management science* **41**(8) 1371–1391.

Ferreira, Kris Johnson, Bin Hong Alex Lee, David Simchi-Levi. 2015. Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management* **18**(1) 69–88.

Fleischmann, Moritz, Jo AEE Van Nunen, Ben Gräve. 2003. Integrating closed-loop supply chains and spare-parts management at ibm. *Interfaces* **33**(6) 44–56.

Gallego, Guillermo, Robert Phillips, Özge Şahin. 2008. Strategic management of distressed inventory. *Production and Operations Management* **17**(4) 402–415.

Gallego, Guillermo, Garrett Van Ryzin. 1994. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management science* **40**(8) 999–1020.

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

Gong, Xiao-Yue, Vineet Goyal, Garud Iyengar, David Simchi-Levi, Rajan Udwani, Shuangyu Wang. 2019. Online assortment optimization with reusable resources. *Available at SSRN 3334789* .

Graves, Stephen C. 1985. A multi-echelon inventory model for a repairable item with one-for-one replenishment. *Management science* **31**(10) 1247–1256.

Guide Jr, V Daniel R, Rajesh Srivastava. 1997. Repairable inventory theory: Models and applications. *European Journal of Operational Research* **102**(1) 1–20.

Guide Jr, V Daniel R, Luk N Van Wassenhove. 2009. Or forumthe evolution of closed-loop supply chain research. *Operations research* **57**(1) 10–18.

Gurumurthi, Suri, Saif Benjaafar. 2004. Modeling and analysis of flexible queueing systems. *Naval Research Logistics (NRL)* **51**(5) 755–782.

Iyengar, Garud, Karl Sigman, et al. 2004. Exponential penalty function control of loss networks. *The Annals of Applied Probability* **14**(4) 1698–1740.

Kim, Jeunghyun, Ramandeep S Randhawa. 2017. The value of dynamic pricing in large queueing systems. *Operations Research* .

Lei, Yanzhe, Stefanus Jasin. 2018. Real-time dynamic pricing for revenue management with reusable resources, advance reservation, and deterministic service time requirements .

Levi, Retsef, Ana Radovanović. 2010. Provably near-optimal lp-based policies for revenue management in systems with reusable resources. *Operations Research* **58**(2) 503–507.

Ma, Will, David Simchi-Levi, Jinglong Zhao. 2018. Dynamic pricing under a static calendar. *Available at SSRN 3251015* .

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

Maglaras, Constantinos, Assaf Zeevi. 2005. Pricing and design of differentiated services: Approximate analysis and structural insights. *Operations Research* **53**(2) 242–262.

Muckstadt, John A. 2004. *Analysis and algorithms for service parts supply chains*. Springer Science & Business Media.

Natter, Martin, Thomas Reutterer, Andreas Mild, Alfred Taudes. 2007. Practice prize reportan assortmentwide decision-support system for dynamic pricing and promotion planning in diy retailing. *Marketing Science* **26**(4) 576–583.

Netessine, Serguei. 2006. Dynamic pricing of inventory/capacity with infrequent price changes. *European Journal of Operational Research* **174**(1) 553–580.

Owen, Zachary, David Simchi-Levi. 2018. Price and assortment optimization for reusable resources. *Available at SSRN 3070625* .

Paschalidis, I Ch, John N Tsitsiklis. 2000. Congestion-dependent pricing of network services. *IEEE/ACM transactions on networking* **8**(2) 171–184.

Rusmevichientong, Paat, Mika Sumida, Huseyin Topaloglu. 2017. Dynamic assortment optimization for reusable products with random usage durations .

Savaskan, R Canan, Shantanu Bhattacharya, Luk N Van Wassenhove. 2004. Closed-loop supply chain models with product remanufacturing. *Management science* **50**(2) 239–252.

Sheikhzadeh, Mehdi, Saifallah Benjaafar, Diwakar Gupta. 1998. Machine sharing in manufacturing systems: Total flexibility versus chaining. *International Journal of Flexible Manufacturing Systems* **10**(4) 351–378.

Simao, Hugo, Warren Powell. 2009. Approximate dynamic programming for management of high-value spare parts. *Journal of Manufacturing Technology Management* **20**(2) 147–160.

Simchi-Levi, David, Michelle Xiao Wu. 2018. Powering retailers digitization through analytics and automation. *International Journal of Production Research* **56**(1-2) 809–816.

Smith, Stephen A, Dale D Achabal. 1998. Clearance pricing and inventory policies for retail chains. *Management Science* **44**(3) 285–300.

Song, J., Z. Xue. 2007. Demand management and inventory control for substitutable products. *Working paper* .

Tang, Christopher S, Rui Yin. 2007. Joint ordering and pricing strategies for managing substitutable products. *Production and Operations Management* **16**(1) 138–153.

Tedone, Mark J. 1989. Repairable part management. *Interfaces* **19**(4) 61–68.

Ye, Kelly Kelly Yunqing. 2008. Joint pricing and inventory decision for competitive products. Ph.D. thesis, Massachusetts Institute of Technology.

Yin, Rui, Kumar Rajaram. 2007. Joint pricing and inventory control with a markovian demand model. *European Journal of Operational Research* **182**(1) 113–126.