Dynamic Learning and Optimization for Operations Management Problems

by

He Wang

B.S., Tsinghua University (2011)
S.M., Massachusetts Institute of Technology (2013)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2016

Signature redacted

Author .....
Sloan School of Management
April 27, 2016

Signature redacted

Certified by.....
David Simchi-Levi
Professor of Engineering Systems
Professor of Civil and Environmental Engineering
Thesis Supervisor

Signature redacted

Accepted by .....
Dimitris Bertsimas
Boeing Professor of Operations Research
Co-director, Operations Research Center

Dynamic Learning and Optimization for Operations Management Problems

by

He Wang

Submitted to the Sloan School of Management
on April 27, 2016, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

With the advances in information technology and the increased availability of data, new approaches that integrate learning and decision making have emerged in operations management. The learning-and-optimizing approaches can be used when the decision maker is faced with incomplete information in a dynamic environment.

We first consider a network revenue management problem where a retailer aims to maximize revenue from multiple products with limited inventory constraints. The retailer does not know the exact demand distribution at each price and must learn the distribution from sales data. We propose a dynamic learning and pricing algorithm, which builds upon the Thompson sampling algorithm used for multi-armed bandit problems by incorporating inventory constraints. Our algorithm proves to have both strong theoretical performance guarantees as well as promising numerical performance results when compared to other algorithms developed for similar settings.

We next consider a dynamic pricing problem for a single product where the demand curve is not known a priori. Motivated by business constraints that prevent sellers from conducting extensive price experimentation, we assume a model where the seller is allowed to make a bounded number of price changes during the selling period. We propose a pricing policy that incurs the smallest possible regret up to a constant factor. In addition to the theoretical results, we describe an implementation at Groupon, a large e-commerce marketplace for daily deals. The field study shows significant impact on revenue and bookings.

Finally, we study a supply chain risk management problem. We propose a hybrid strategy that uses both process flexibility and inventory to mitigate risks. The interplay between process flexibility and inventory is modeled as a two-stage robust optimization problem: In the first stage, the firm allocates inventory, and in the second stage, after disruption strikes, the firm schedules its production using process flexibility to minimize demand shortage. By taking advantage of the structure of the second stage problem, we develop a delayed constraint generation algorithm that can efficiently solve the two-stage robust optimization problem. Our analysis of this model provides important insights regarding the impact of process flexibility on total inventory level and inventory allocation pattern.

Thesis Supervisor: David Simchi-Levi
Title: Professor of Engineering Systems
Professor of Civil and Environmental Engineering

Acknowledgments

This thesis is a reflection of my rewarding journey at MIT. I had the privilege to spend five years here among many incredible people.

First, I would like to thank my advisor, David Simchi-Levi, who has been a great mentor. I met David when I was an undergraduate at Tsinghua, and have worked with him first as a master student at CEE and later as a doctoral student at ORC. During the five years, I received not only great research guidance from David, but also many invaluable suggestions beyond research. I cannot image achieving all the accomplishment today without David's guidance and support.

Next, I would like to thank my thesis committee members, John Tsitsiklis and Georgia Perakis, for their time and effort. I have learned a lot from John since my first year Probability class, and have always admired him as a scholar. I have also been very fortunate to interact with Georgia in many occasions. Both of them have gave me valuable advice. In addition, I would like to thank them for serving on my General Exam committee, and for helping me during my academic job search.

I really appreciate the support from my home departments, CEE and ORC. I would like to thank the ORC co-directors, Dimitris Bertsimas and Patrick Jaillet, for their great job creating a collaborative and intellectually exhilarating environment. I am indebted to many CEE and ORC staff, especially Janet Kerrigan, Andrew Carvalho, and Laura Rose. Moreover, I would like to thank a few other MIT faculty – Karen Zheng, Nigel Wilson, Amedeo Odoni, Steve Graves, Jim Orlin – for their guidance and advice.

This thesis is a result of collaboration with several of my fellow ORC students. Chapter 2 is joint work with Kris Ferreira. Chapter 3 is joint work with Wang Chi Cheung and Alex Weinstein. Chapter 4 is joint work Yehua Wei. All of them are talented researchers and good friends, and I am very fortunate to work with them. I also received numerous help from Kris and Yehua about almost all aspects of graduate life, who have turned out to be my "unofficial" mentors at MIT.

I would also like to thank the research support from industry partners. I am very grateful to the research funding from Groupon, and would like to thank a few people at Groupon that I have worked with – Gaston L'Huillier, Francisco Larrain, Kamson Lai, Shi Zhao, Latife Genc-Kaya. Chapter 2 of this thesis comes from a direct collaboration with this team. I

would also like to thank my managers at IBM during my summer internship – Pavithra Harsha, Shivaram Subramanian, and Markus Ettl.

I am so lucky to have made many amazing friends at MIT: Yehua, Kris, Alex, Wang-Chi, Rong, Yaron, Nataly, Swati, Adam, Joline, Mila, Will, Peng, Michael, Andrew, Zach, Clark, Louis, Lu, Xiang, Dave, and many others. I would like to thank all of them.

Finally, I owe my deepest gratitude to my family. I would especially like to thank Yue, who has been my closest friend and caring partner during this journey.

– H.W.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Operations management refers to the administration of business practices to convert resources into goods and services. It studies a broad range of operational decisions including the design and management of products, processes, services and supply chains.[1] These operational decisions are often made in an uncertain environment. The uncertainty could rise from a variety of sources: randomness in production yield, disruptions in supply chains, and fluctuations in customer demand, etc.

In classical operations management literature, uncertainties are dealt with in two separate stages. In the first stage, the firm estimates some probability distribution from historical data using statistical tools, and use it to model uncertainty. In the second stage, the firm incorporates the probability distribution into some decision models, and optimizes its decisions given that probability distribution. Therefore, under this framework, the estimation process is separated from the decision process. Indeed, most operations management literature assumes the probability distribution as given.

Over the past few years, there has been growing interest in combining estimation processes and decision processes in operations management. This new trend is driven by two forces: one is the advances in information technology, which enable the firm to quickly collect and process large amounts of data, so that the estimation process can be completed in real time. Another driving force is the popular business practice of reducing product life cycle in order to introduce new products more frequently. The short product life cycle means that there is less time to complete the estimation process and the decision process separately.

[1]This definition is paraphrased from the "operations management" page on www.investopedia.com.

Therefore, many firms in fashion and online retail industries have adopted new approaches that integrate statistical learning into their decision processes.

To give an example, let us consider Groupon, a large e-commerce marketplace and a collaborator in this research. Groupon is a website where customers can purchase discount deals from local merchants. Every day, thousands of new deals are launched on Groupon's website. Deals are only available for limited time, ranging from several weeks to several months. Due to this business model, Groupon is faced with high level of demand uncertainty, mainly because there is no previous sales data for newly launched deals. This challenge presents an opportunity for Groupon to learn about customer demand using real time sales data after deals have been launched in order to obtain more accurate demand estimation.

Motivated by Groupon, this thesis considers several fundamental problems in operations management with dynamic learning and optimization. The key challenge of dynamic learning, or online learning, is to address the exploration-exploitation tradeoff. Generally speaking, exploitation means optimizing the system using the current available data and greedy solutions. Exploration means collecting more data by deviating from the current greedy optimal decision in order to improve future decision.

In the following, we will first give a brief review of the general exploration-exploration theory, and then present its applications in two operations management areas: revenue management and supply chain management.

1.1 Background

1.1.1 Exploration-Exploitation Tradeoff

Multi-armed bandit is a classical problem that models the essence of exploration-exploitation tradeoff. It was first proposed by Thompson (1933), who is motivated by clinical trials, and formally formulated by Robbins (1952). The basic problem is as follows: there are multiple slot machines in a casino, and there is a gambler who has finite number of tokens. The name "one-armed bandit" refers to the colloquial term of a slot machine, hence the model setting is called a "multi-armed bandit." With each token, the gambler can play any arm. The reward from each arm is i.i.d., but their reward distribution is unknown to the gambler. Therefore, the gambler needs to sequentially allocates tokens to different slot machines in order to learn their distributions. The goal is to maximize the total expected reward.

The multi-armed bandit problem has numerous variants. For example, one variant is called "continuous bandit" or "continuum bandit": Instead of finitely many arms, there are infinitely many arms indexed by a continuous interval, see Kleinberg and Leighton (2003), Mersereau et al. (2009), Rusmevichientong and Tsitsiklis (2010). Another variant is known as "contextual bandit," where the decision maker receives some external information that can help predict the reward distribution at the beginning of each round.

We will further discuss the multi-armed bandit literature in the subsequent chapters.

1.1.2 Dynamic Pricing and Online Demand Learning

The multi-armed bandit model can be used for dynamic pricing in the setting of incomplete demand information. In this setting, the firm has to learn about the demand model while changing price in real time. One can view each price as an "arm" of the bandit, and the revenue under that price as the "reward" associate with that "arm." This analogy builds a direct connection between multi-armed bandit problems and dynamic pricing problems.

Joint learning-and-pricing problems have received extensive research attention over the last decade. Recent surveys by Aviv and Vulcano (2012) and den Boer (2015) provide a comprehensive overview of this area. Recent revenue management papers that consider price experimentation for learning demand curves include Besbes and Zeevi (2009), Boyacı and Özer (2010), Wang et al. (2014) and Besbes and Zeevi (2015). Another stream of papers focuses on semi-myopic pricing policies using various learning methods. Examples include maximum likelihood estimation (Broder and Rusmevichientong 2012), Bayesian methods (Harrison et al. 2012), maximum quasi-likelihood estimation (den Boer and Zwart 2014, den Boer 2014) and iterative least-squares estimation (Keskin and Zeevi 2014).

In most of these papers, the dynamic pricing models deviate from the classical multi-armed bandit model because the seller can choose price from a continuous interval (and hence a "continuum bandit"). Another practical issue is that the classical multi-armed bandit model does not include the inventory constraint, while many sellers is faced with limited inventory. These issues will be further discussed in Chapter 2 and 3.

1.1.3 Adaptive Supply Chain Risk Mitigation

A key idea in the multi-armed bandit problem is to make decisions sequentially so that future decisions can be adaptive to newly revealed information. We further exploit this

idea in the setting of supply chain risk mitigation. More specifically, we consider a hybrid strategy using both ex-ante decisions (i.e., inventory) and ex-post decisions (i.e., flexibility). The ex-post decisions of this strategy are made adaptively after disruptions happen.

There is a rich literature on risk mitigation using inventory. Many of the earlier papers (e.g., Meyer et al. 1979, Song and Zipkin 1996, Arreola-Risa and DeCroix 1998) studied inventory risk mitigation in a single product setting. More recently, inventory mitigation strategies under multi-period, multi-echelon settings have also been considered (e.g., Bollapragada et al. 2004, DeCroix 2013). However, a main drawback of using inventory alone is that it may require too much inventory to achieve a satisfiable service level.

Process flexibility, also referred to as "mix flexibility" or "product flexibility," has also been observed as potential risk mitigation tool. Tomlin and Wang (2005) considers a risk mitigation strategy that uses a combination of mix-flexibility and dual sourcing. Tang and Tomlin (2008) suggests process flexibility as one of the five types of flexibility strategies that can be used to mitigate supply chain disruptions. And finally, Sodhi and Tang (2012) lists flexible manufacturing processes as one of the eleven robust supply chain strategies.

In this thesis, we propose a hybrid approach to study the risk mitigation strategy by combining both flexibility and inventory. This idea is partially studied by Gürler and Parlar (1997), Tomlin (2006) in the dual sourcing setting. The hybrid strategy requires the firm to allocate inventory before the uncertainties (*ex-ante*), and to adjust its production level after uncertainties are realized (*ex-post*). We will further discuss this problem in Chapter 4.

1.2 Overview

The remaining parts of this thesis are organized as follows.

In Chapter 2, we consider a network revenue management problem where an online retailer aims to maximize revenue from multiple products with limited inventory constraints. We propose a dynamic learning and pricing algorithm, which builds upon the Thompson sampling algorithm used for multi-armed bandit problems by incorporating inventory constraints.

In Chapter 3, we consider a dynamic pricing problem for a single product where the demand curve is not known a priori. We explicitly consider a business constraint, which prevents the seller from conducting extensive price experimentation. Our analysis provides

important structural insights into the optimal pricing strategies. In addition to the theoretical results, we will describe an implementation at Groupon, a large e-commerce marketplace for daily deals.

In Chapter 4, we study a supply chain risk management problem by considering a hybrid strategy that uses both (process) flexibility and inventory. The interplay between process flexibility and inventory is modeled as a two-stage robust optimization problem. We develop a delayed constraint generation algorithm that can efficiently solve the two-stage robust optimization problem.

Finally, we provide some concluding remarks in Chapter 5. The technical proofs for each chapter are included in the appendices.

Chapter 2

Online Network Revenue Management with Thompson Sampling

In this chapter, we focus on a classical revenue management problem: A retailer is given an initial inventory of resources and a finite selling season. Inventory cannot be replenished throughout the season. The firm must choose prices for a set of products to maximize revenue over the course of the season, where each product consumes certain amount of resource inventory. The retailer has the ability to observe consumer purchase decisions in real-time and can dynamically adjust the price at negligible cost.

For historical reason, this problem is known as the *network revenue management* problem, because it was first proposed by Gallego and Van Ryzin (1997) for the airline network yield management problem. In the airline setting, each "product" is an itinerary path from an origin to a destination, and each "resource" is a single flight leg in the network.

The network revenue management problem has been well-studied in the academic literature under the additional assumption that the mean demand rate associated with each price is known to the retailer prior to the selling season. In practice, many retailers do not know the exact mean demand rates; thus, we focus on the network revenue management problem with unknown demand.

Given unknown mean demand rates, the retailer faces a tradeoff commonly referred to as the *exploration-exploitation tradeoff* (see Section 1.1.1). In the network revenue management

setting, the retailer is constrained by limited inventory and thus faces an additional tradeoff. Specifically, pursuing the exploration objective comes at the cost of diminishing valuable inventory. Simply put, if inventory is depleted while exploring different prices, there is no inventory left to exploit the knowledge gained.

We develop an algorithm for the network revenue management setting with unknown mean demand rates which balances the exploration-exploitation tradeoff while also incorporating inventory constraints. Our algorithm is based on the Thompson sampling algorithm for the stochastic multi-armed bandit problem, where we add a linear program (LP) subroutine to incorporate inventory constraints. The proposed algorithm is easy to implement and also has strong performance guarantees. The flexibility of Thompson sampling allows our algorithm to be generalized to various extensions. More broadly, our algorithm can be viewed as a way to solve multi-armed bandit problems with resource constraints. Such problems have wide applications in dynamic pricing, dynamic online advertising, and crowdsourcing Badanidiyuru et al. (2013).

2.1 Literature Review

Due to the increased availability of real-time demand data, there is growing academic interest on dynamic pricing problems using a demand learning approach. Section 1.1.2 gave a general overview of this research area. The review below is focused on existing literature that considers inventory constraints. In Table 2.1, we list research papers that address limited inventory constraints and classify their models based on the number of products, allowable price sets being discrete or continuous, and whether the demand model is parametric or non-parametric.

As described earlier, our work generalizes to the network revenue management setting, thus allowing for multiple products, whereas much of the literature is for the single product setting.

The papers included in Table 2.1 propose pricing algorithms that can be roughly divided into three groups. The first group consider joint learning and pricing problem using dynamic programing (DP) Aviv and Pazgal (2005a,b), Bertsimas and Perakis (2006), Araman and Caldentey (2009), Farias and Van Roy (2010). The resulting DP is usually intractable due to high dimensions, so heuristics are often used in these papers. Since the additional inventory

	# products		set of prices		demand model	
	single	multiple	discrete	continuous	param	nonparam
Aviv and Pazgal (2005a)	X			X	X	
Aviv and Pazgal (2005b)	X			X	X	
Bertsimas and Perakis (2006)	X			X	X	
Araman and Caldentey (2009)	X			X	X	
Besbes and Zeevi (2009)	X			X	X	X
Farias and Van Roy (2010)	X			X	X	
Besbes and Zeevi (2012)		X	X	X		X
Badanidiyuru et al. (2013)		X	X	X		X
Wang et al. (2014)	X			X		X
Lei et al. (2014)	X			X		X
Chen et al. (2014)		X		X	X	X
This paper		X	X			X

Table 2.1: Literature on Dynamic Learning and Pricing with Limited Inventory

constraints add to the dimension of DPs, papers in this group almost exclusively consider single product settings in order to limit the complexity of the DPs.

The second group applies a simple strategy that separates the time horizon into a demand learning phase (exploration objective) and a revenue maximization phase (exploitation objective) Besbes and Zeevi (2009, 2012), Chen et al. (2014). Recently, Wang et al. (2014), Lei et al. (2014) show that when there is a single product, pricing algorithms can be improved by mixing the exploration and exploitation phases. However, their methods cannot be generalized beyond the single-product/continuous-price setting.[1]

The third group of papers builds on the classical multi-armed bandit algorithms (Badanidiyuru et al. (2013) and this paper) or the stochastic approximation methods Besbes and Zeevi (2009), Lei et al. (2014). The multi-armed bandit problem is often used to model the exploration-exploitation tradeoff in the dynamic learning and pricing model *without* limited inventory constraints; see Gittins et al. (2011), and Bubeck and Cesa-Bianchi (2012) for an overview of this problem. Thompson sampling is a powerful algorithm used for the multi-armed bandit problem, and it is a key building block of the algorithm that we propose.

Thompson sampling. In one of the earliest papers on the multi-armed bandit problem, Thompson (1933) proposed a randomized Bayesian algorithm, which was later referred to as *Thompson sampling*. The basic idea of Thompson sampling is that at each time period,

[1]Lei et al. (2014) also consider a special case of multi-product setting where there is no cross-elasticity in product demands.

random numbers are sampled according to the posterior probability distributions of the reward of each arm, and then the arm with the highest sampled reward is chosen. The algorithm is also known as *probability matching* since the probability of an arm being chosen matches the posterior probability that this arm has the highest expected reward. Compared to other popular multi-armed bandit algorithms such as those in the Upper Confidence Bound (UCB) family Lai and Robbins (1985), Auer et al. (2002a), Garivier and Cappé (2011), Thompson sampling enjoys comparable theoretical performance guarantees Agrawal and Goyal (2011), Kaufmann et al. (2012), Agrawal and Goyal (2013) and better empirical performance Chapelle and Li (2011). In addition, the Thompson sampling algorithm shows great flexibility and has been adapted to various model settings Russo and Van Roy (2014). We believe that a salient feature of this algorithm's success is its ability to update mean demand estimation in every period and then exploit this knowledge in subsequent periods. As you will see, we take advantage of this property in our development of a new algorithm for the network revenue management problem when demand distribution is unknown.

2.2 Model

We consider a retailer who sells N products, indexed by $i = 1, \ldots, N$, over a finite selling season. These products consume M limited resources, indexed by $j = 1, \ldots, M$. Specifically, one unit of product i consumes a_{ij} units of resource j, which has I_j units of initial inventory. There is no replenishment during the selling season.

The selling season is divided into T periods. In each period $t = 1, \ldots, T$, the retailer offers a price vector $P(t) = [P_1(t), \ldots, P_N(t)]$, where $P_i(t)$ is the price for product i at period t. We use $\mathcal{P}$ to denote the admissible price set. After the retailer chooses $P(t) \in \mathcal{P}$, customers observe the price vector chosen and make purchase decisions. We use vector $D(t) = [D_1(t), \ldots, D_N(t)]$ to denote the realized demand at period t.

1. If there is inventory available to satisfy demand for all products, then the retailer receives revenue $\sum_{i=1}^{N} D_i(t)P_i(t)$. Inventory is depleted by $\sum_{i=1}^{N} D_i(t)a_{ij}$ for each resource $j = 1, \ldots, M$.

2. If there is not enough resource available to satisfy demand for some products, demand for those products is lost. In our analysis, we assume that the selling season immediately ends when there is a lost demand. Since we are aiming at a lower bound of the

retailer's expected revenue, this simplification does not change the result.

We assume that demand is only affected by the current price, and is not affected by previous or future prices. The joint demand distribution per period under price $p \in \mathcal{P}$ is denoted by $F(p, \theta)$, where $\theta \in \Theta$ is some parameter. The mean demand under distribution $F(p, \theta)$ is denoted by $d(p, \theta) = [d_1(p, \theta), \ldots, d_N(p, \theta)]$. By assuming the joint distribution $F(\cdot, \theta)$, demand for different products or demand under different prices can be correlated. The true parameter θ is unknown the retailer, and the estimates of θ are updated in a Bayesian fashion. We denote by $H(t)$ the history $(P(1), D(1), \ldots, P(t), D(t))$. The posterior distribution of θ given the history is $\mathbb{P}(\theta \in \cdot \mid H(t))$. The retailer's goal is to choose price vector $P(t)$ sequentially to maximize revenue over the course of the selling season.

In the following, we present several special cases based on demand models that are widely used in practice.

2.2.1 Discrete Price Case

In many practical cases, the price set consists of finite number of price vectors: $\mathcal{P} = \{p_1, \ldots, p_K\}$. Here, each $p_k \in \mathcal{P}$ is a N-dimensional vector, specifying the price for each of the N products.

Nonparametric Demand Model

We assume an independent prior: $F(p_k, \theta) = \Pi_i F_i(p_k, \theta_{ik})$. Based on this prior, we can update the demand for each product under each price separately.

Note the assumption of the independent prior that does not restrict the true demand distribution to be independent across different products and different prices. In fact, the true demand distribution can have arbitrary correlation and dependence between prices and products. Imposing an independent prior simply means that we update *the marginal demand* for each product under each price separately. Therefore, this approach is "nonparametric" in that it does not impose any restrictive assumptions on the demand correlation.

Multi-armed Bandit with Global Constraints

The discrete price model can also be used to model a multi-armed bandit problem with global resource constraints. The problem is also known as "bandits with knapsacks" in

Badanidiyuru et al. (2013).

The problem is the following: there are multiple arms ($k = 1, \ldots, K$) and multiple resources ($j = 1, \ldots, M$). At each time period $t = 1, \ldots, T$, the decision maker chooses to pull one of the arms. If arm k is pulled, it generates a Bernoulli variable with mean θ_k, which is unknown to the decision maker. If the generated value is one, the decision maker consumes b_{jk} units of resource j; in addition, the decision maker receives r_k units of reward. If the generated value is zero, no resource is consumed and no reward is received. (More generally, if resource consumptions and rewards in each period are not binary but have bounded probability distributions, we can reduce the problem to the one with binary distributions using the re-sampling trick described in Section 2.3.1.) At any given time, if there exists $j = 1, \ldots, M$ such that the total consumption of resource j up to this time is greater than a fixed constant I_j, the decision process immediately stops and no future rewards will be received.

To see show that this problem is a special case of our model, we can consider a set of products indexed by $k = 1, \ldots, K$. The price set is discrete: $\mathcal{P} = \{p_1, \ldots, p_K\}$, where choosing price p_k means only offering product k at price r_k, while making other products unavailable. The mean demand for product k is θ_k, and the resource consumption coefficient is $a_{kj} = b_{kj}$.

In line with standard terminology in the multi-armed bandit literature, we will refer to "pulling arm k" as the retailer "offering price p_k", and "arm k" will be used interchangeably with "price p_k".

The presence of inventory constraints significantly complicates the problem, even for the special case of a single product. In the classical multi-armed bandit setting, if success probability of each arm is known, the optimal strategy is to choose the arm with the highest mean reward. But in the presence of limited inventory, a mixed strategy that chooses multiple prices over the selling season may achieve higher revenue than any single price strategy. Therefore, a reasonable strategy for the multi-armed bandit model with global constraint should not converge to the optimal single price, but to the optimal distribution of (possibly) multiple prices. Another challenging task is to estimate the time when inventory runs out and the selling season ends early, which is itself a random variable depending on the chosen strategy. Such estimation is necessary for computing the expected reward. This is opposed to classical multi-armed bandit problems for which algorithms always end at a

fixed period.

Display Advertising Example. To show an application of the multi-armed bandit model with global constraint, we consider an example of placing online ads. Suppose there is an online ad platform (e.g. Google) that uses the pay per click system. For each user logging on to a third-party website, Google may display a banner ad on the website. If the user clicks the ad, the advertiser sponsoring the ad pays some amount of money that is split between Google and the website hosting the banner ad (known as the *publisher*). If the user does not click the ad, no payment is made. Assuming that click rates for ads are unknown, Google faces the problem of allocating ads to publishers to maximize its revenue, while satisfying advertisers' budgets.

This problem fits into the limited resource multi-armed bandit model as follows: each arm is an ad, and each resource corresponds to an advertiser's budget. If ad k is clicked, the advertiser j sponsoring ad k pays b_{jk} units from its budget, of which Google gets a split of r_k.

Note that this model is only a simplified version of the real problem. In practice, Google also obtains some data about the user and the website (e.g. the user's location or the website's keywords) and is able to use this information to improve its ad display strategy. We consider such an extension with contextual information in Section 2.6.

2.3 Thompson Sampling Algorithm with Limited Inventory

In this section, we propose an algorithm called "Thompson Sampling with Limited Inventory" that builds off the original Thompson sampling algorithm Thompson (1933) to incorporate inventory constraints.

For convenience, we first define a "dummy price", p_∞, where $d_i(p_\infty, \theta) = 0$ for all $i = 1, \ldots, N$ and $\theta \in \Theta$. We define $\mathcal{X}$ as the set of all probability distribution over $\mathcal{P} \cup \{p_\infty\}$. For any $x \in \mathcal{X}$, which is a probability measure over $\mathcal{P} \cup \{p_\infty\}$, we let $\tilde{x}$ be the corresponding normalized probability measure over $\mathcal{P}$. That is, for any subset $\mathcal{P}' \subset \mathcal{P}$, we have $\tilde{x}(\mathcal{P}') = x(\mathcal{P}')/x(\mathcal{P})$.

We present the Thompson Sampling with Limited Inventory algorithm in Algorithm 1.

Algorithm 1 Thompson Sampling with Inventory (TS-general)

The retailer starts with inventory level $I_j(0) = I_j$ for all $j = 1, \ldots, M$.
Repeat the following steps for all $t = 1, ..., T$:

1. Sample Demand: sample θ_t from $\mathbb{P}(\theta \in \cdot \mid H(t-1))$.

2. Optimize: Solve the following optimization problem:

$$f(\theta(t)) = \max_{x \in \mathcal{X}} \mathbb{E}_{p \sim x}[\sum_{i=1}^{N} p_i d_i(p, \theta(t))]$$

$$\text{subject to} \quad \mathbb{E}_{p \sim x}[\sum_{i=1}^{N} a_{ij} d_i(p, \theta(t))] \leq I_j/T \quad \text{for all } j = 1, \ldots, M$$

   Let $x(t)$ be its optimal solution, and $\tilde{x}(t)$ be the corresponding normalized distribution over $\mathcal{P}$.

3. Offer Price: Retailer chooses price vector $P(t) \in \mathcal{P}$ according to probability distribution $\tilde{x}(t)$.

4. Update: Customer's purchase decisions, $D(t)$, are revealed to the retailer. The retailer updates the history $H(t) = H(t-1) \cup \{P(t), D(t)\}$ and the posterior distribution of θ, $\mathbb{P}(\theta \in \cdot \mid H(t))$. The retailer also updates inventory level $I_j(t) = I_j(t-1) - \sum_{i=1}^{N} D_i(t) a_{ij}$ for all $j = 1, \ldots, M$.

5. Check Inventory Level: If $I_j(t) \leq 0$ for any resource j, the algorithm terminates.

2.3.1 Special Cases of the Algorithm

Discrete Price, Bernoulli Demand We consider a special case where there are finitely many prices: $\mathcal{P} = \{p_1, \ldots, p_K\}$. Demands for each product are Bernoulli random variables. Let $N_k(t-1)$ be the number of time periods that the retailer has offered price p_k in the first $t-1$ periods, and let $W_{ik}(t-1)$ be the number of periods that product i is purchased under price p_k during these periods. Define $I_j(t-1)$ as the remaining inventory of resource j at the beginning of the t^{th} time period. Define constants $c_j = I_j/T$ for resource j, where $I_j = I_j(0)$ is the initial inventory level.

The model defined in Section 2.2.1 allows for any joint demand distribution $F(p, \theta)$. However, to avoid misspecification of the demand model, we consider a Bayesian updating process which only updates the marginal demand distribution under the chosen price. We present this process in Algorithm 2.

In Algorithm 2, steps 1 and 4 are based on the Thompson sampling algorithm for the classical multi-armed bandit setting. In particular, in step 1, the retailer randomly samples product demands according to demand posterior distribution. In step 4, upon observing customer purchase decisions, the retailer updates the posterior distributions under the chosen price. We use Beta posterior distributions in the algorithm—a common choice in Thompson sampling—because the Beta distribution is conjugate to Bernoulli random variables. The posterior distributions of the unchosen price vectors are not changed.

The algorithm differs from the ordinary Thompson sampling algorithm in steps 2 and 3. In step 2, instead of choosing the price with the highest reward using sampled demand, the retailer first solves a linear program (LP) which identifies the optimal mixed price strategy that maximizes expected revenue given sampled demand. The first constraint specifies that the average resource consumption per time period cannot exceed the initial inventory divided by length of the time horizon. The second constraint specifies that the sum of the probabilities of choosing all price vectors cannot exceed one. In step 3, the retailer randomly offers one of the K price vectors according to probabilities specified by the LP's optimal solution. Note that if all resources have positive inventory levels, we have $\sum_{k=1}^{K} x_k(t) > 0$ in the optimal solution to the LP, so the probabilities are well-defined.

In the remainder of the paper, we use TS-fixed as an abbreviation for Algorithm 2. The term "fixed" refers to the fact that the algorithm uses a fixed inventory to time ratio

Algorithm 2 Thompson Sampling with Inventory (TS-fixed)

Repeat the following steps for all $t = 1, ..., T$:

1. Sample Demand: For each price k and each product i, sample $\theta_{ik}(t)$ from a $Beta(W_{ik}(t-1)+1, N_k(t-1) - W_{ik}(t-1)+1)$ distribution.

2. Optimize: Solve the following linear program, $OPT(\theta)$

$$
\begin{aligned}
f(\theta) = \max_{x_k} & \sum_{k=1}^{K}(\sum_{i=1}^{N} p_{ik}\theta_{ik}(t))x_k \\
\text{subject to } & \sum_{k=1}^{K}(\sum_{i=1}^{N} a_{ij}\theta_{ik}(t))x_k \leq c_j \quad \text{for all } j = 1, \ldots, M \\
& \sum_{k=1}^{K} x_k \leq 1 \\
& x_k \geq 0, \text{ for all } k = 1, \ldots, K
\end{aligned}
$$

   Let $x(t) = (x_1(t), \ldots, x_K(t))$ be the optimal solution to $OPT(\theta)$ at time t.

3. Offer Price: Retailer chooses price vector $P(t) = p_k$ with probability

$$x_k(t) / \sum_{k=1}^{K} x_k(t).$$

4. Update: Customer's purchase decisions, $D(t)$, are revealed to the retailer. The retailer sets $N_k(t) = N_k(t-1) + 1$, $W_{ik}(t) = W_{ik}(t-1) + D_i(t)$ for all $i = 1, \ldots, N$, and $I_j(t) = I_j(t-1) - \sum_{i=1}^{N} D_i(t)a_{ij}$ for all $j = 1, \ldots, M$.

5. Check Inventory Level: If $I_j(t) \leq 0$ for any resource j, the algorithm terminates.

$c_j = I_j/T$ in the LP for every period.

Inventory Updating Intuitively, improvements can be made to the TS-fixed algorithm by incorporating the real time inventory information. In particular, we can change $c_j = I_j/T$ to $c_j(t) = I_j(t-1)/(T-t+1)$ in the LP in step 2. This change is shown in Algorithm 3. We refer to this modified algorithm as the "Thompson Sampling with Inventory Rate Updating algorithm" (TS-update, for short).

Algorithm 3 Thompson Sampling with Inventory Rate Updating (TS-update)

Repeat the following steps for all $t = 1, ..., T$:

- Perform step 1 in Algorithm 1.
- Optimize: Solve the following linear program, $OPT(\theta)$

$$
\begin{aligned}
f(\theta) = \max_{x_k} \; & \sum_{k=1}^{K} (\sum_{i=1}^{N} p_{ik}\theta_{ik}(t))x_k \\
\text{subject to } & \sum_{k=1}^{K} (\sum_{i=1}^{N} a_{ij}\theta_{ik}(t))x_k \leq c_j(t) \quad \text{for all } j = 1, \ldots, M \\
& \sum_{k=1}^{K} x_k \leq 1 \\
& x_k \geq 0, \text{ for all } k = 1, \ldots, K
\end{aligned}
$$

- Perform steps 3–5 in Algorithm 1.

In the revenue management literature, the idea of using updated inventory rates, $c_j(t)$, has been studied under the assumption that the demand distribution is known Secomandi (2008), Jasin and Kumar (2012). Recently, Chen et al. (2014) consider a pricing policy using updated inventory rates in the unknown demand setting. In practice, using updated inventory rates usually improves revenue compared to using fixed inventory rates c_j Talluri and van Ryzin (2005). We also show in Section 2.5 that TS-update outperforms TS-fixed in simulations. Unfortunately, since TS-update involves a randomized demand sampling step, theoretical performance analysis of this algorithm appears to be much more challenging than TS-fixed and other algorithms in the existing literature.

General Demand Distributions with Bounded Support If $d_i(p_k)$ is randomly distributed with bounded support $[\underline{d}_{ik}, \bar{d}_{ik}]$, we can reduce the general distribution to a two-

point distribution, and update it with Beta priors as in Algorithm 2. Suppose the retailer observes random demand $D_i(t) \in [\underline{d}_{ik}, \bar{d}_{ik}]$. We then re-sample a new random number, which equals to $\underline{d}_{ik}$ with probability $(\bar{d}_{ik} - D_i(t))/(\bar{d}_{ik} - \underline{d}_{ik})$ and equals to $\bar{d}_{ik}$ with probability $(D_i(t) - \underline{d}_{ik})/(\bar{d}_{ik} - \underline{d}_{ik})$. It is easily verifiable that the re-sampled demand has the same mean as the original demand.[2] By using re-sampling, the theoretical results in Section 2.4 also hold for the bounded demand setting.

As a special case, if demands have multinomial distributions for all $i = 1, \ldots, N$ and $k = 1, \ldots, K$, we can use Beta parameters similarly as in Algorithm 2 without resorting to re-sampling. This is because the Beta distribution is conjugate to the multinomial distribution.

Poisson Demand Distribution If the demand follows a Poisson distribution, we can use the Gamma distribution as the conjugate prior; see Algorithm 4. We use $Gamma(\alpha, \lambda)$ to represent a Gamma distribution with shape parameter α and rate parameter λ.

Algorithm 4 Thompson Sampling with Poisson Demand

Repeat the following steps for all $t = 1, ..., T$:

- Sample Demand: For each price k and each product i, sample $\theta_{ik}(t)$ from a $Gamma(W_{ik}(t-1) + 1, N_k(t-1) + 1)$ distribution.
- Optimize: Solve the linear program, $OPT(\theta)$, used in either Algorithm 2 or Algorithm 3.
- Perform steps 3-5 in Algorithm 1.

Note that Poisson demand cannot be reduced to the Bernoulli demand setting by the re-sampling method described above because Poisson demand has unbounded support. Note that Besbes and Zeevi (2012) assume that customers arrive according to a Poisson distribution, so when we compare our algorithm with theirs in Section 2.5, we use this variant of our algorithm.

2.4 Theoretical Analysis

In this section, we present a theoretical analysis of the TS-fixed algorithm. We consider a scaling regime where the initial inventory I_j increases linearly with the time horizon T for all resources $j = 1, \ldots, M$. Under this scaling regime, the average inventory per time period

[2]This re-sampling trick is also mentioned by Agrawal and Goyal (2013).

$c_j = I_j/T$ remains constant. This scaling regime is widely used in revenue management literature.

2.4.1 Benchmark and Linear Programming Relaxation

To evaluate the retailer's strategy, we compare the retailer's revenue with a benchmark where the true demand distribution is known a priori. We define the retailer's regret as

$$Regret(T) = E[R^*(T)] - E[R(T)],$$

where $R^*(T)$ is the optimal revenue if the demand distribution is known a priori, and $R(T)$ is the revenue when the demand distribution is unknown. In words, the regret is a non-negative quantity measuring the retailer's revenue loss due to not knowing the latent demand.

Because evaluating the expected optimal revenue with known demand requires solving a dynamic programming problem, it is difficult to compute the optimal revenue exactly even for moderate problem sizes. Gallego and Van Ryzin (1997) show that the expected optimal revenue can be approximated by the following upper bound. Let x_k be the fraction of periods that the retailer chooses price p_k for $k = 1, \ldots, K$. The upper bound is given by the following deterministic LP, denoted by OPT_{UB}:

$$\begin{aligned} f^* = \max & \sum_{k=1}^{K} (\sum_{i=1}^{N} p_{ik} d_i(p_k)) x_k \\ \text{subject to} & \sum_{k=1}^{K} (\sum_{i=1}^{N} a_{ij} d_i(p_k)) x_k \leq c_j \quad \text{for all } j = 1, \ldots, M \\ & \sum_{k=1}^{K} x_k \leq 1 \\ & x_k \geq 0, \text{ for all } k = 1, \ldots, K. \end{aligned}$$

Recall that $d_i(p_k)$ is the expected demand of product i under price p_k. Problem OPT_{UB} is almost identical to the LP used in step 2 of TS-fixed, except that it uses the true mean demand instead of sampled demand from posterior distributions. We denote the optimal value of OPT_{UB} as f^* and the optimal solution as $(x_1^*, \ldots, x_n^*)$. A well-known result in

Gallego and Van Ryzin (1997) shows that

$$E[R^*(T)] \leq f^*T.$$

In addition, if the retailer chooses prices according to probability distribution $(x_1^*, \ldots, x_n^*)$ at each time period, the expected revenue is within $O(\sqrt{T})$ of the upper bound f^*T.

2.4.2 Analysis of Thompson Sampling with Inventory Algorithm

We now prove the following regret bound for TS-fixed.

Theorem 2.1. *Suppose that the optimal solution(s) of OPT_{UB} are non-degenerate. If the demand distribution has bounded support, the regret of* TS-fixed *is bounded by*

$$Regret(T) \leq O(\sqrt{T} \log T \log \log T).$$

Proof Sketch. The complete proof of Theorem 2.1 can be found in Appendix A. We start with the case where there is a unique optimal solution, and the proof has three main parts. Suppose X^* is the optimal basis of OPT_{UB}. The first part of the proof shows that TS-fixed chooses arms that do not belong to X^* for no more than $O(\sqrt{T} \log T)$ times. Then in the second part, assuming there is unlimited inventory, we bound the revenue when only arms in X^* are chosen. In particular, we take advantage of the fact that TS-fixed performs continuous exploration and exploitation, so the LP solution of the algorithm converges to the optimal solution of OPT_{UB}. In the third part, we bound the expected lost sales due to having limited inventory, which should be subtracted from the revenue calculated in the second part. The case of multiple optimal solutions is proved along the same line. □

Note that the non-degeneracy assumption of Theorem 2.1 only applies to the LP with the true mean demand, OPT_{UB}. The theorem does not require that optimal solutions to $OPT(\theta)$, the LP that the retailer solves at each step, to be non-degenerate. Moreover, the simulation experiments in Section 2.5 show that TS-fixed performs well even if the non-degeneracy assumption does not hold.

It is useful to compare the results in Theorem 2.1 to the regret bounds in Besbes and Zeevi (2012) and Badanidiyuru et al. (2013), since our model settings are essentially the same. Our regret bound improves upon the $O(T^{\frac{2}{3}})$ bound proved in Besbes and Zeevi (2012) and

matches (omitting log factors) the $O(\sqrt{T})$ bound in Badanidiyuru et al. (2013). We believe that the reason why our algorithm and the one proposed in Badanidiyuru et al. (2013) have stronger regret bounds is that they both perform continuous exploration and exploitation, whereas the algorithm in Besbes and Zeevi (2012) separates periods of exploration and exploitation.

However, we should note that the regret bound in Theorem 2.1 is *problem-dependent*, whereas the bounds in Besbes and Zeevi (2012) and Badanidiyuru et al. (2013) are *problem-independent*. More specifically, we show that TS-fixed or TS-update guarantees $Regret(T) \leq C\sqrt{T}\log T \log\log T$, where the constant C is a function of the demand data. As a result, the retailer cannot compute the constant C a priori since the mean demand is unknown. In contrast, the bounds proved in Besbes and Zeevi (2012) and Badanidiyuru et al. (2013) are independent of the demand data and only depend on parameters such as the number of price vectors or the number of resource constraints, which are known to the retailer. Moreover, the regret bounds in Besbes and Zeevi (2012) and Badanidiyuru et al. (2013) do not require the non-degeneracy assumption.

It is well-known that the *problem-independent lower bound* for the multi-armed bandit problem is $Regret(T) \geq \Omega(\sqrt{T})$ Auer et al. (2002b). Since the multi-armed bandit problem can be viewed as a special case of our setting where inventory is unlimited, the algorithm in Badanidiyuru et al. (2013) has the best possible problem-independent bound (omitting log factors). The $\Omega(\sqrt{T})$ lower bound is also proved separately by Besbes and Zeevi (2012) and Badanidiyuru et al. (2013).[3] On the other hand, it is not clear what the *problem-dependent lower bound* is for our setting, so we do not know for sure if our $O(\sqrt{T})$ problem-dependent bound (omitting log factors) can be improved.

2.5 Numerical Results

In this section, we first provide an illustration of the TS-fixed and TS-update algorithms for the setting where a single product is sold throughout the course of the selling season, and we compare these results to other proposed algorithms in the literature. Then we present results for a multi-product example; for consistency, the example we chose to use is identical

[3]Badanidiyuru et al. (2013) proves a more general lower bound where the initial inventory is not required to scale linearly with time T. However, one can show that their bound becomes $\Omega(\sqrt{T})$ under the additional assumption that inventory is linear in T.

to the one presented in Section 3.4 of Besbes and Zeevi (2012).

2.5.1 Single Product Example

Consider a retailer who sells a single product ($N = 1$) throughout a finite selling season. Without loss of generality, we can assume that the product is itself the resource ($M = 1$) which has limited inventory. The set of feasible prices is $\{\$29.90, \$34.90, \$39.90, \$44.90\}$, and the mean demand is given by $d(\$29.90) = 0.8, d(\$34.90) = 0.6, d(\$39.90) = 0.3$, and $d(\$44.90) = 0.1$. As aligned with our theoretical results, we show numerical results when inventory is scaled linearly with time, i.e. initial inventory $I = \alpha T$, for $\alpha = 0.25$ and 0.5.

We evaluate and compare the performance of the following five dynamic pricing algorithms which have been proposed for our setting:

- TS-update: intuitively, this algorithm outperforms TS-fixed and is thus what we suggest for retailers to use in practice.
- TS-fixed: this is the algorithm we have proposed with strong regret bounds as shown in Section 2.4.
- The algorithm proposed in Besbes and Zeevi (2012): we implemented the algorithm with $\tau = T^{2/3}$ as suggested in their paper, and we used the actual remaining inventory after the exploration phase as an input to the optimization problem which sets prices for the exploitation phase.
- The PD-BwK algorithm proposed in Badanidiyuru et al. (2013): this algorithm is based on the primal and the dual of OPT_{UB}. For each period, it estimates upper bounds of revenue, lower bounds of resource consumption, and the dual price of each resource, and then selects the arm with the highest revenue-to-resource-cost ratio.
- Thompson sampling (TS): this is the algorithm described in Thompson (1933) which has been proposed for use as a dynamic pricing algorithm but does *not* consider inventory constraints.

We measure performance as the average percent of "optimal revenue" achieved over 500 simulations. By "optimal revenue", we are referring to the upper bound on optimal revenue where the retailer knows the mean demand at each price prior to the selling season; this

upper bound is the solution to OPT_{UB}, f^*T, described in Section 2.4.1. Thus, the percent of optimal revenue achieved is at least as high as the numbers shown. Figure 2-1 shows performance results for the five algorithms outlined above.

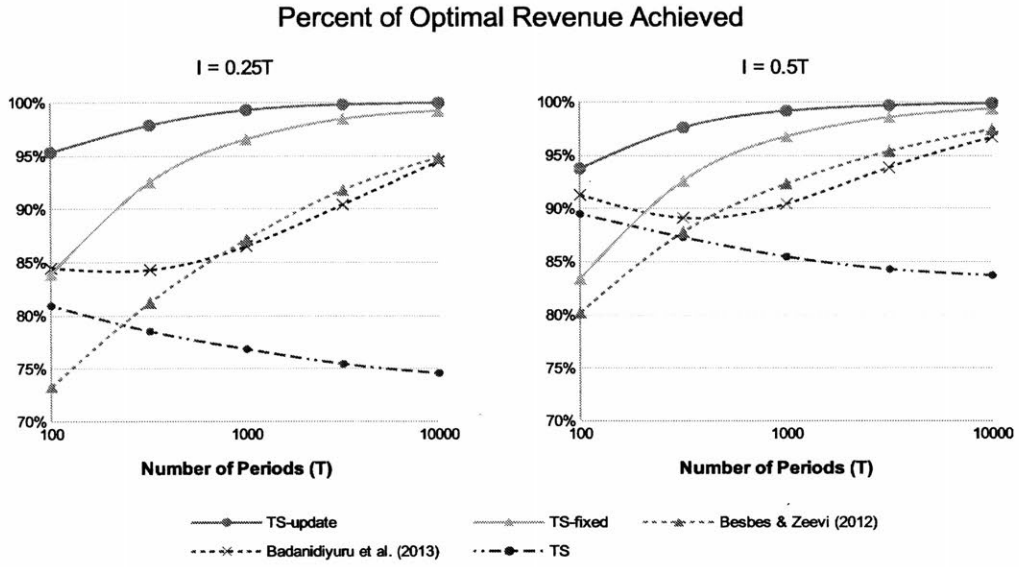


Figure 2-1: Performance Comparison of Dynamic Pricing Algorithms – Single Product Example

The first thing to notice is that all four algorithms that incorporate inventory constraints converge to the optimal revenue as the length of the selling season increases. The TS algorithm, which does not incorporate inventory constraints, does not converge to the optimal revenue. This is because in each of the examples shown, the optimal pricing strategy is a mixed strategy where two prices are offered throughout the selling season as opposed to a single price being offered to all customers. The optimal strategy when $I = 0.25T$ is to offer the product at \$39.90 to $\frac{3}{4}$ of the customers and \$44.90 to the remaining $\frac{1}{4}$ of the customers. The optimal strategy when $I = 0.5T$ is to offer the product at \$34.90 to $\frac{2}{3}$ of the customers and \$39.90 to the remaining $\frac{1}{3}$ of the customers. In both cases, TS converges to the suboptimal price \$29.90 offered to all the customers since this is the price that maximizes expected revenue given unlimited inventory. This really highlights the necessity of incorporating inventory constraints when developing dynamic pricing algorithms.

Overall, TS-update outperforms all of the other algorithms in both examples. Interestingly, when considering only those algorithms that incorporate inventory constraints, the gap

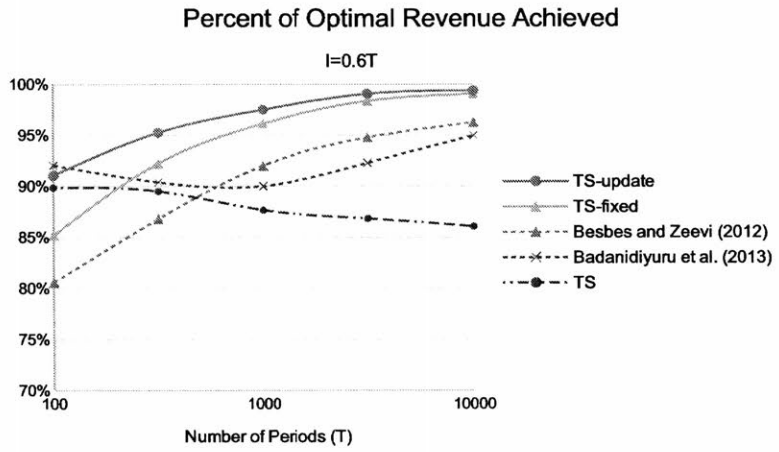


Figure 2-2: Performance Comparison of Dynamic Pricing Algorithms – Single Product with Degenerate Optimal Solution

between TS-update and the others generally shrinks when (i) the length of the selling season increases, and (ii) the ratio I/T increases. This is consistent with many other examples that we have tested and suggests that our algorithm is particularly powerful (as compared to the others) when inventory is very limited and the selling season is short. In other words, our algorithm is able to more quickly learn mean demand and identify the optimal pricing strategy, which is particularly useful for low inventory settings.

We then perform another experiment when the optimal solution to OPT_{UB} is degenerate. We assume the initial inventory is $I = 0.6T$, so the degenerate optimal solution is to offer the product at \$34.90 to all customers. Note that the degenerate case is not covered in the result of Theorem 2.1. Despite the lack of theoretical support, Figure 2-2 shows that TS-fixed and TS-update still perform well.

2.5.2 Multi-Product Example

Now we consider the example presented in Section 3.4 of Besbes and Zeevi (2012) where a retailer sells two products ($N = 2$) using three resources ($M = 3$). Selling one unit of product $i = 1$ consumes 1 unit of resource $j = 1$, 3 units of resource $j = 2$, and no units of resource $j = 3$. Selling one unit of product $i = 2$ consumes 1 unit of resource 1, 1 unit of resource 2, and 5 units of resource 3. The set of feasible prices is $(p_1, p_2) \in \{(1, 1.5), (1, 2), (2, 3), (4, 4), (4, 6.5)\}$. Besbes and Zeevi (2012) assume customers arrive according to a multivariate Poisson process. We would like to compare performance

using a variety of potential underlying functions that specify mean demand, so we consider the following three possibilities for mean demand of each product as a function of the price vector:

1. *Linear:* $\mu(p_1, p_2) = (8 - 1.5p_1, 9 - 3p_2)$,
2. *Exponential:* $\mu(p_1, p_2) = (5e^{-0.5p_1}, 9e^{-p_2})$, and
3. *Logit:* $\mu(p_1, p_2) = \left(\frac{10e^{-p_1}}{1+e^{-p_1}+e^{-p_2}}, \frac{10e^{-p_2}}{1+e^{-p_1}+e^{-p_2}}\right)$.

Since customers arrive according to a Poisson process, we must evaluate the variant of TS-fixed and TS-update described in Section 2.3.1 that allows for such arrivals. Since the PD-BwK algorithm proposed in Badanidiyuru et al. (2013) does not apply to the setting where customers arrive according to a Poisson process, we cannot include this algorithm in our comparison.

We again measure performance as the average percent of "optimal revenue" achieved, where optimal revenue refers to the upper bound on optimal revenue when the retailer knows the mean demand at each price prior to the selling season, f^*T. Thus, the percent of optimal revenue achieved is at least as high as the numbers shown. Figure 2-3 shows average performance results over 500 simulations for each of the three underlying demand functions; we show results when inventory is scaled linearly with time, i.e. initial inventory $I = \alpha T$, for $\alpha = (3, 5, 7)$ and $\alpha = (15, 12, 30)$.

As in the single product example, each algorithm converges to the optimal revenue as the length of the selling season increases. In most cases, the TS-update algorithm outperforms the algorithm proposed in Besbes and Zeevi (2012). The TS-fixed algorithm has slightly worse performance than TS-update as expected, but in several cases the difference between the two algorithms is almost indistinguishable. For each set of parameters and when $T = 10,000$, TS-update and TS-fixed achieve 99–100% of the optimal revenue whereas the Besbes and Zeevi (2012) algorithm achieves 92–98% of the optimal revenue. As we saw in the single product example, TS-update performs particularly well when inventory is very limited ($I = (3, 5, 7)T$); it is able to more quickly learn mean demand and identify the optimal pricing strategy. TS-update and TS-fixed also seem to perform particularly well when mean demand is linear. Finally, note that the algorithm's performance appears to be fairly consistent across the three demand models tested; this suggests that the retailer can confidently use our algorithm even when the underlying demand function is unknown.

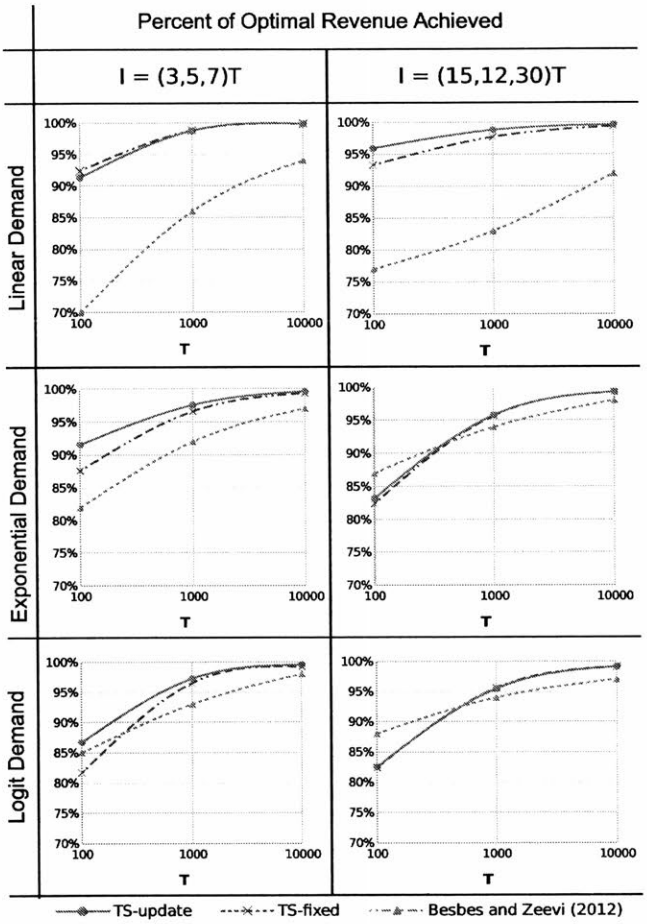


Figure 2-3: Performance Comparison of Dynamic Pricing Algorithms – Multi-Product Example

2.6 Extension: Demand with Contextual Information

In the model described in Section 2.2, we assume that the mean demand rates are unknown to the retailer but fixed over time. However, there are various factors that may cause the demand function to change over time, including seasonality, markdowns of competitors' prices, and shifts of customer preferences. In this section, we adapt the algorithm to changing demands.

Several papers in the revenue management literature have studied dynamic pricing problems with unknown and changing demand Aviv and Pazgal (2005b), Besbes and Zeevi (2011), Besbes and Sauré (2014). These papers usually assume that only historical sales data can be used to detect demand changes. Unlike these papers, we assume that at each period,

the retailer receives some *contextual information* (also known as *context*, *feature*, or *side information*) that can be used to predict demand changes. For example, the contextual information may contain competitors' prices or seasonality factors predicted from previous selling seasons. As another example, in the customized pricing/ad-display problem, each period models one customer arrival, so the contextual information may include personal features of arriving customers Chen et al. (2015b). For more examples on the contextual multi-armed bandit problem, we refer the readers to Chapter 4 of Bubeck and Cesa-Bianchi (2012).

2.6.1 Model and Algorithm

Suppose that at the beginning of each period t, the retailer observes some context $\xi(t)$, where $\xi(t) \in \mathcal{X} \subset \mathbb{R}^d$ is a d-dimensional vector. Given context $\xi(t)$, the mean demand of product i under price p_k is a Bernoulli variable with mean $d_i(\xi(t), p_k)$, where $d_i(\cdot, p_k) : \mathcal{X} \to [0, 1]$ is a fixed function for all $i = 1, \ldots, N$ and $k = 1, \ldots, K$. Similar to the original model, we assume that function $d_i(\cdot, p_k)$ is unknown to the retailer.

For this model, we propose a modification of TS-fixed and TS-update to incorporate the contextual information (see Algorithm 5). We name the modified algorithm Thompson Sampling with Contextual Information (TS-contextual, for short). The main changes are in steps 1, 2 and 5. In step 1, given contextual information, $\xi(t)$, the retailer predicts the mean demand for each product i and price k, denoted by $h_i(\xi(t), p_k)$. Then the retailer samples demand from Beta distributions in step 2. Note that the predicted demand, $h_i(\xi(t), p_k)$, replaces the simple average of historical demand, $W_{ik}(t-1)/N_k(t-1)$, used in step 1 of TS-fixed and TS-update. Steps 3 and 4 remain the same. In step 5, the retailer observes customer purchase decisions and updates its predictions for the chosen price, p_k. The update requires a regression over all pairs of contextual information and purchase decisions in the historical data for which price p_k is offered:

$$\{(\xi(s), z_i(s)) \mid 1 \le s \le t, p(s) = p_k\}.$$

For example, if the retailer uses logistic regression, the predicted mean demand has the following form:

$$h_i(\xi, p_k) = \frac{1}{1 + e^{-\beta_0(p_k) - \beta_i^T(p_k)\xi}},$$

where parameters $\beta_0(p_k) \in \mathbb{R}, \beta_i(p_k) \in \mathbb{R}^d$, for all $i = 1, \ldots, N$ and $k = 1, \ldots, K$, are the estimated coefficients of the logistic function. The retailer has the freedom of choosing other prediction methods in TS-contextual.

Algorithm 5 Thompson Sampling with Contextual Information (TS-contextual)

Suppose the retailer starts with $h_i(\cdot, p_k)$, an estimation of the true demand function $d_i(\cdot, p_k)$, for all $i = 1, \ldots, N$ and $k = 1, \ldots, K$.
Repeat the following steps for all $t = 1, ..., T$:

1. Observe Contextual Information $\xi(t)$: Compute demand prediction $h_i(\xi(t), p_k)$ for all $i = 1, \ldots, N$ and $k = 1, \ldots, K$.

2. Sample Demand: For each price k and each product i, sample $\theta_{ik}(t)$ from a $Beta(h_i(\xi(t), p_k)N_k(t-1) + 1, N_k(t) - h_i(\xi(t), p_k)N_k(t-1) + 1)$ distribution.

3. Optimize: Solve the following linear program, $OPT(\theta)$

$$
\begin{aligned}
f(\theta) = \max_{x_k} & \sum_{k=1}^{K} (\sum_{i=1}^{N} p_{ik}\theta_{ik}(t)) x_k \\
\text{subject to } & \sum_{k=1}^{K} (\sum_{i=1}^{N} a_{ij}\theta_{ik}(t)) x_k \leq c_j \text{ (or } c_j(t)) \quad \text{for all } j = 1, \ldots, M \\
& \sum_{k=1}^{K} x_k \leq 1 \\
& x_k \geq 0, \text{ for all } k = 1, \ldots, K
\end{aligned}
$$

   Let $(x_1(t), \ldots, x_K(t))$ be the optimal solution to the LP.

4. Offer Price: Retailer chooses price vector $p(t) = p_k$ with probability $x_k(t) / \sum_{k=1}^{K} x_k(t)$.

5. Update: Customer purchase decisions, $z_i(t)$, are revealed to the retailer. For each product $i = 1, \ldots, N$, the retailer performs a regression (e.g. logistic regression) on the historical data:

$$\{(\xi(s), z_i(s)) \mid 1 \leq s \leq t, p(s) = p_k\},$$

   and updates $h_i(\cdot, p_k)$—the predicted demand function associated with product i and price p_k. The retailer also sets $N_k(t) = N_k(t-1)+1$ and $I_j(t) = I_j(t-1) - \sum_{i=1}^{N} z_i(t) a_{ij}$ for all $j = 1, \ldots, M$.

6. Check Inventory Level: If $I_j(t+1) \leq 0$ for any resource j, the algorithm terminates.

2.6.2 Upper Bound Benchmark

In the case where the space of contextual information $\mathcal{X}$ is finite and context ξ are generated i.i.d., we can upper bound the expected optimal revenue when the demand functions and

the distribution of ξ are known. The upper bound is a deterministic linear programming relaxation, similar to the one presented in Section 2.4.1. Suppose that $q(\xi)$ is the probability mass function of context ξ. The upper bound is given by

$$
\begin{aligned}
f^* = \max \; & \sum_{\xi \in \mathcal{X}} \sum_{k=1}^{K} \Big(\sum_{i=1}^{N} p_{ik} d_i(\xi, p_k) \Big) x_k(\xi) q(\xi) \\
\text{subject to } & \sum_{\xi \in \mathcal{X}} \sum_{k=1}^{K} \Big(\sum_{i=1}^{N} a_{ij} d_i(\xi, p_k) \Big) x_k(\xi) q(\xi) \leq c_j \quad \text{for all } j = 1, \ldots, M \\
& \sum_{k=1}^{K} x_k(\xi) \leq 1 \quad \text{for all } \xi \in \mathcal{X} \\
& x_k(\xi) \geq 0, \text{ for all } k = 1, \ldots, K, \xi \in \mathcal{X}.
\end{aligned}
$$

To prove the upper bound, we can view different context as different "products". In particular, we can consider a new model with $N \times |\mathcal{X}|$ "products" without contextual information, and show that the new model is equivalent to the contextual model described in Section 2.6.1. Suppose demand for product $(i, \xi) \in N \times |\mathcal{X}|$ has mean $d_i(\xi, p_k) q(\xi)$. Product (i, ξ) in the new model has the same feasible price set and resource consumption rate as product i in the contextual model. Then, there is an obvious equivalence between the retailer's pricing strategy in the contextual model, determined by $x_k(\xi)$, and the pricing strategy of product (i, ξ) for all $i = 1, \ldots, N$ in the new model. Due to this equivalence, the upper bound above is an immediate result of the upper bound in Section 2.4.1.

2.6.3 Numerical Example

Consider an example where the retailer sells a single product (i.e. $N = M = 1$) with initial inventory $I = 0.6T$. At each period, the retailer observes an exogenous context $\xi \in [0, 1]$ and chooses between two prices $\{\$9.99, \$19.99\}$. We assume that ξ represents the competitor's pricing effect normalized between 0 and 1; small ξ means the competitor offers a higher price, while larger ξ means the competitor offers a lower price. We tested two scenarios where the context ξ is either discrete or continuous: 1) ξ is generated i.i.d. from a Uniform$(0, 1)$ distribution, and 2) ξ is generated i.i.d. from a Bernoulli(0.5) distribution.

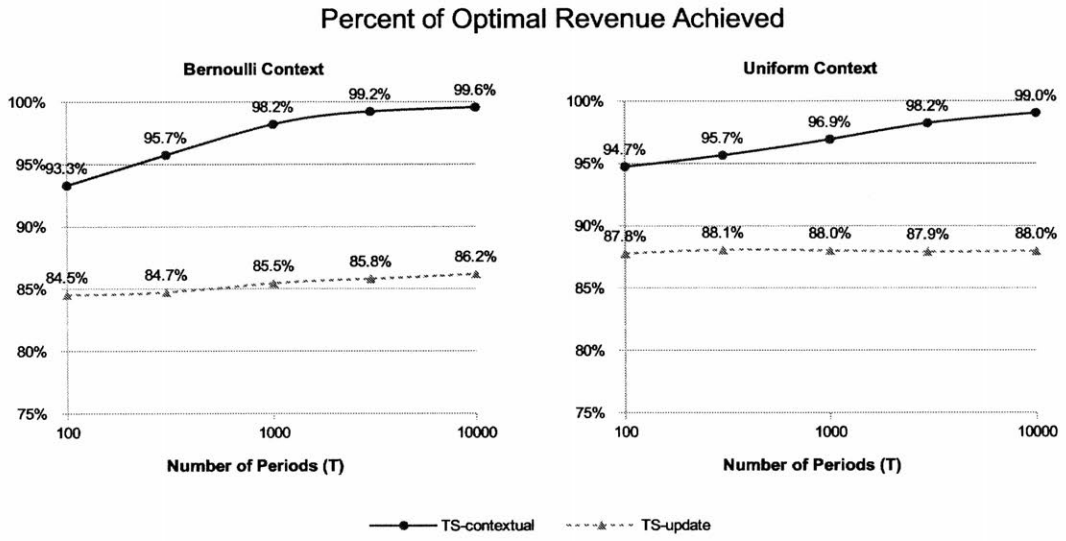


Figure 2-4: Performance Comparison of Dynamic Pricing Algorithms – Contextual Example

The mean demand (unknown to the retailer) as a function of ξ is

$$d(\xi, \$9.99) = 0.7e^{-0.2\xi}, \quad d(\xi, \$19.99) = 0.5e^{-1.0\xi}.$$

In particular, we assume the mean demand at the higher price ($19.99) decreases faster as ξ increases, because customer demand at a high price may be more sensitive when the competitor offers a price discount.

We compare the performance of the following two pricing algorithms:

- TS-contextual: we use logistic regression $h(\xi, \cdot) = 1/(1 + e^{-\beta_0 - \beta_1 \xi})$ to estimate the demand function $d(\xi, \cdot)$. We also use the updated inventory rate $c_j(t)$ in the LP (similar to TS-update) instead of the fixed inventory rate c_j.

- TS-update: the retailer ignores the contextual information. Because we assume that ξ is i.i.d., ignoring the contextual information reduces the problem to the simple case where demands are i.i.d. at each price, so TS-update can be applied to this example.

Note that in both algorithms, the retailer knows neither the demand functions nor the distribution of ξ. We compare the two algorithms to the LP upper bound when the demand functions and the distribution of ξ are known. (In the case where $\xi \sim$ Uniform$(0, 1)$, the upper bound is calculated by approximating the uniform distribution with a discrete uniform

distribution.) Figure 2-4 shows average performance results over 500 simulations for each distribution of ξ. Clearly, the revenue is significantly improved when the retailer incorporates contextual information in the pricing strategy. When $T = 10,000$, TS-contextual achieves 99% of the optimal revenue (technically, the LP upper bound) whereas TS-update achieves only 85%–88% of the optimal revenue. Over all instances, TS-contextual increases revenue by 8%–16% compared to TS-update.

Figure 2-4 also shows that TS-contextual converges faster to optimality when $\xi \sim \text{Bernoulli}(0.5)$ compared to when $\xi \sim \text{Uniform}(0, 1)$. We suspect that the faster convergence can be explained by two reasons. First, it requires fewer data points to learn demand with Bernoulli context, since there are only two context types. Second, our algorithm incorrectly specifies the demand functions as logistic functions, while the true demand functions are exponential functions of the contextual information. Misspecification may hurt the algorithm's performance for the uniform context, but not for the Bernoulli context. In the latter case, TS-contextual only needs demand predictions for $\xi = 0$ and $\xi = 1$, so misspecification for $\xi \in (0, 1)$ does not matter. Since the retailer has freedom to choose the regression method in TS-contextual, other regression methods may be used to reduce misspecification error.

Finally, we note that even without model misspecification, the expected revenue of TS-contextual may not converge to the LP upper bound. This is because in TS-contextual, we define an LP constraint that bounds the resource consumption rate for *any* given context, while the upper bound LP bounds the resource consumption rate averaged over *all* context instances. Of course, since the constraint in TS-contextual is more conservative, the algorithm can be applied to cases where the contextual information is not i.i.d., whereas the LP upper bound requires the i.i.d. assumption.

Chapter 3

Dynamic Pricing and Demand Learning with Limited Price Experimentation

Groupon is a large e-commerce marketplace where customers can purchase discount deals from local merchants such as restaurants, spas and house cleaning services. Every day, thousands of new deals are launched on Groupon's website. Due to its business model, Groupon is faced with high level of demand uncertainty, mainly because there is no previous sales for the newly launched deals. This challenge presents an opportunity for Groupon to learn about customer demand using real time sales data after deals have been launched so as to obtain more accurate demand estimation and adjust prices.

Generally, in revenue management, when the underlying relationship between demand and price is unknown a priori, the seller can use price experimentation for demand learning. In this chapter, we consider a dynamic pricing model where the exact demand function is unknown but belongs to a finite set of possible demand functions, or demand hypotheses. The seller faces an exploration-exploitation tradeoff between actively adjusting price to gather demand information and optimizing price for revenue maximization.

Dynamic pricing under a finite set of demand hypotheses has previously been considered by Rothschild (1974) and Harrison et al. (2012). But unlike this work, both of these papers focus on customized pricing, where price is changed for every arriving customer. For example, the motivation of Harrison et al. (2012) is pricing for financial services such as consumer

and auto loans, where sellers can quote a different interest rate for each customer.

However, for many e-commerce companies like Groupon, charging a different price for each arriving customer is impossible, either because of implementation constraints, or for fear of confusing customers and receiving negative customer response. In our collaboration, Groupon stipulated as a rule that the number of price changes has to be as few as possible for each deal, so that the customers would not observe frequent price changes. Motivated by this practical business constraint on price experimentation, the model in this chapter includes an explicit constraint on the number of price changes during the sales horizon.

We quantify the impact of this constraint on the seller's revenue using *regret*, defined as the gap between the revenue of a clairvoyant who has full information on the demand function and the revenue achieved by a seller facing unknown demand. Our main finding is a sharp characterization of regret as a function of the number of price changes allowed. When there are T periods in the sales horizon, we propose a pricing policy with at most m price changes, whose regret is bounded by $O(\log^{(m)} T)$, or m iterations of the logarithm. Furthermore, we prove that the regret of any *non-anticipating* pricing policy, i.e., a policy where the current price does not depend on unrealized future demands, is lower bounded by $\Omega(\log^{(m)} T)$. Thus, the regret bound achieved by our proposed policy is tight up to a constant factor.

A natural question is how frequently one needs to change price to achieve a constant regret. Harrison et al. (2012) shows that a semi-myopic policy can achieve a constant regret, but the policy requires changing price for every time period. To answer this question, we show that a modified version of our algorithm with no more than $O(\log^* T)$ price changes, where $\log^* T$ is the smallest number m such that $\log^{(m)} T \leq 1$, achieves a constant regret.

This characterization of the regret bound has two important implications. First, imposing a price change constraint always incurs a cost on revenue, since the seller cannot achieve a constant regret using any *finite* number of price changes. Second, the incremental effect of price changes decreases quickly. The first price change reduces regret from $O(T)$ to $O(\log T)$; each additional price change thereafter compounds a logarithm to the order of regret. As a result, the first few price changes generate most of the benefit of dynamic pricing. Interestingly, while the value $\log^* T$ is unbounded when T increases, its growth rate is extremely slow. For example, if the number of time periods T is less than 3 million, we still have $\log^* T$ no greater than 3.

Motivated by these results, we implemented a pricing strategy at Groupon where each deal can have at most one price change. The result of a field experiment shows significant improvement in both revenue and deal bookings at Groupon.

3.1 Related Literature

Joint learning-and-pricing problems have received extensive research attention over the last decade; readers are referred to the survey in Chapter 1. However, all the literature mentioned previously does not assume any constraint on price experimentation. In the dynamic pricing literature with known demand distribution, several papers consider limited price changes; examples include Feng and Gallego (1995), Bitran and Mondschein (1997), Netessine (2006) and Chen et al. (2015a). Caro and Gallien (2012) reports that the fashion retailer Zara uses a clearance pricing policy with a pre-determined price ladder, which essentially allows for only a limited number of mark-down prices. Zbaracki et al. (2004) provide empirical results on the cost of price changes.

To the best of our knowledge, the only work that considers price-changing constraints in an unknown demand setting is Broder (2011). The author assumes that the demand function belongs to a known parametric family, e.g. linear family, but has unknown parameters. He shows that in order to achieve the optimal regret, a pricing policy needs at least $\Theta(\log T)$ price changes. However, the result only applies to a restricted class of policies where the seller cannot use any knowledge of T.

Our model is different from the model by Broder (2011) in the following aspects. First, we assume a finite number of demand hypotheses, while Broder (2011) assumes a parametric family of demand functions. This is a fundamental difference because the optimal regret in Broder's case is $\Theta(\sqrt{T})$, while in our case the regret can be bounded by a constant. Second, we do not assume a restricted class of policies as in Broder (2011), and our results hold for any pricing policies. Last but not least, unlike Broder (2011) where the number of price changes is an output from the model, we design a pricing algorithm that accepts the number of price changes as an input constraint, and achieves the best possible regret bound under that constraint.

3.2 Model Formulation

We consider a seller offering a single product with unlimited supply for T periods. The set of allowable prices is denoted by $\mathcal{P}$. For example, $\mathcal{P}$ can either be an interval $[\underline{p}, \overline{p}]$ or a finite set $\{p_1, \ldots, p_k\}$, although no restriction on $\mathcal{P}$ is assumed here. In the t^{th} period $(t = 1, \ldots, T)$, the seller offers a unit price $P_t \in \mathcal{P}$, and observes a random customer demand X_t, i.e. the number of units purchased by customers. Given $P_t = p$, the distribution of X_t is only determined by price p, and is independent of previous prices and demands $\{P_1, X_1, \ldots, P_{t-1}, X_{t-1}\}$. We use $D(p) \sim X_t$ to denote a random variable distributed as X_t given $P_t = p$. The corresponding *mean demand function* $d : \mathcal{P} \to \mathbb{R}_+$ is defined as $d(p) = \mathbb{E}[D(p)]$.

The distribution of $D(p)$ is unknown to the seller. However, the seller knows that the distribution belongs to a finite set of *demand models*, or demand distributions as a function of p. The demand models are indexed by $i = 1, \ldots, K$. We use $\mathbb{P}_i(\cdot)$ and $\mathbb{E}_i(\cdot)$ to denote the probability measure and expectation under demand model i. In particular, the mean demand function $d(p)$ belongs to a finite set of K demand functions, denoted by $\Phi = \{d_1(p), \ldots, d_K(p)\}$, where $d_i(p) = \mathbb{E}_i[D(p)]$. For each demand function $d_i \in \Phi$, $(i = 1, \ldots, K)$, the expected revenue per period is $r_i(p) = pd_i(p)$. We also denote the optimal revenue for demand function d_i by $r_i^* = \max_{p \in \mathcal{P}} r_i(p)$ and an optimal price by $p_i^* \in \arg\max_{p \in \mathcal{P}} r_i(p)$. The seller does not necessarily know the distribution of demand model i apart from the mean $d_i(p)$.

For all $p \in \mathcal{P}$ and $i = 1, \ldots, K$, the probability distribution of $D(p)$ is assumed to be *light-tailed* with parameters (σ, b), where $\sigma, b > 0$. That is, we have $\mathbb{E}_i[e^{\lambda(D(p) - d_i(p))}] \leq \exp(\lambda^2\sigma^2/2)$ for all $|\lambda| < 1/b$. Note that the class of light-tailed distributions includes all sub-Gaussian distributions. Some common light-tailed distributions include normal, Poisson and Gamma distributions, as well as all distributions with bounded support, such as binomial and uniform distributions.

3.2.1 Pricing Policies

We say that π is a non-anticipating pricing policy if the price P_t offered at period t is determined by the realized demand $(X_1, \ldots, X_{t-1})$ and previous prices $(P_1, \ldots, P_{t-1})$, but does not depend on future demand. For $i = 1, \ldots, K$, let $\mathbb{P}_i^\pi(\cdot)$ and $\mathbb{E}_i^\pi(\cdot)$ be the probability measure and expectation induced by policy π if the underlying demand model is i. In this

case, the seller's expected revenue in T periods under policy π is given by

$$R_i^\pi(T) = \mathbb{E}_i^\pi \left[\sum_{t=1}^{T} P_t X_t \right] = \mathbb{E}_i^\pi \left[\sum_{t=1}^{T} P_t \mathbb{E}_i^\pi [X_t \mid P_t] \right] = \mathbb{E}_i^\pi \left[\sum_{t=1}^{T} r_i(P_t) \right]. \tag{3.1}$$

As motivated earlier, in many revenue management applications, the seller faces a constraint on the number of price changes. In the model, we assume that the seller can make at most m changes to the price over the course of the sales event, where m is a fixed integer. A feasible policy π should therefore satisfy the following condition:

$$\mathbb{P}_i^\pi \left(\sum_{t=2}^{T} I(P_t \neq P_{t-1}) \leq m \right) = 1, \quad \forall i = 1, \ldots, K,$$

where $I(\cdot)$ is the indicator function. We refer to a policy with at most m price changes as an *m-change policy.*

The performance of a pricing policy is measured against the optimal policy in the full information case. If the true demand is d_i, then a clairvoyant with full knowledge of the demand function would offer price p_i^* and obtain expected revenue r_i^* for every period. The *regret* with respect to demand d_i is defined as the gap between the expected revenue achieved by the clairvoyant and the one achieved by policy π, namely

$$\text{Regret}_i^\pi(T) = T r_i^* - R_i^\pi(T) = \mathbb{E}_i^\pi \left[\sum_{t=1}^{T} (r_i^* - r_i(P_t)) \right]. \tag{3.2}$$

Finally, we define the minimax regret for the demand set, $\Phi = \{d_1, \ldots, d_K\}$, as

$$\text{Regret}_\Phi^\pi(T) = \max_{i=1,\ldots,K} \text{Regret}_i^\pi(T).$$

When there is no ambiguity of which policy we are referring to, we suppress the superscript "π" in the notation for clarity: $\mathbb{E}_1 := \mathbb{E}_1^\pi$, $\mathbb{P}_1 := \mathbb{P}_1^\pi$.

3.2.2 Notations

We use $\log^{(m)} T$ to represent m iterations of the logarithm, $\log(\log(\ldots \log(T)))$, where m is the number of price changes. For convenience, we let $\log(x) = 0$ for all $0 \leq x < 1$, so the function $\log^{(m)} T$ is defined for all $T \geq 1$. Similarly, we define $e^{(0)} := 1$ and $e^{(\ell)} := \exp(e^{(\ell-1)})$

for $\ell \geq 1$. As mentioned earlier, function $\log^* T$ denotes the smallest nonnegative integer m such that $\log^{(m)} T \leq 1$. For any real number x, $\lceil x \rceil$ denotes the minimum integer greater than or equal to x. For any finite set S, $|S|$ is the cardinality of S. We occasionally use notations $a \vee b = \max\{a, b\}$, $a \wedge b = \min\{a, b\}$.

3.3 Main Results: Upper and Lower Bounds on Regret

In this section we prove the main results: an upper bound and a lower bound on regret as a function of the number of price changes. We first design a non-anticipating pricing policy that changes price no more than m times and achieves a regret of $O(\log^{(m)} T)$. Then, we show that the regret of any non-anticipating policy with at most m price changes is at least $\Omega(\log^{(m)} T)$. Thus, our proposed pricing policy achieves the optimal regret bounds up to a constant factor.

3.3.1 Upper Bound

We propose a policy mPC (which stands for "m-price change") that achieves a regret of $O(\log^{(m)} T)$ with at most m price changes. An important feature of policy mPC is that it applies a *discriminative* price for every period. A price p is *discriminative* if demands $d_1(p), \cdots, d_K(p)$ are mutually distinct.

We make the following assumption on the set of demand functions Φ:

Assumption 3.1. *For all $d_i \in \Phi = \{d_1, \cdots, d_K\}$, there exists a corresponding revenue-optimal price $p_i^* \in argmax_{p \in \mathcal{P}} r_i(p)$ such that p_i^* is a discriminative price for Φ, that is, $d_1(p_i^*), \cdots, d_K(p_i^*)$ are distinct. Moreover, such price p_i^* can be efficiently computed.*

Assumption 3.1 ensures that the seller is able to learn the underlying demand curve while maximizing its revenue for any given demand function $d_i \in \Phi$. In fact, we will show in Section 3.3.4 that this condition is both sufficient and necessary for achieving a regret bound better than $o(\log T)$.

Algorithm 6 describes the mPC policy. The policy partitions the finite time horizon $1, \cdots, T$ into $m + 1$ phases. For each $0 \leq \ell \leq m$, a single price P_ℓ^* is offered through Phase ℓ, which starts at period $\tau_\ell + 1$ and ends at $\tau_{\ell+1}$. Phase 0 to Phase $m - 1$ are called the *learning phases*, and Phase m is referred to as the *earning phase*. Except for a constant

Algorithm 6 m-change policy mPC

1: INPUT:

- A set of demand functions $\Phi = \{d_1, \cdots, d_K\}$.
- A discriminative price P_0^*.

2: (Learning) Set $\tau_0 = 0$.

3: **for** $\ell = 0, \cdots, m-1$ **do**

4: **if** $\log^{(m-\ell)} T = 0$ **then**

5: Set $\tau_{l+1} = 0$ and $P_{\ell+1}^* = P_\ell^*$.

6: **else**

7: From period $\tau_\ell + 1$ to $\tau_{\ell+1} := \tau_\ell + \left\lceil M_\Phi(P_\ell^*) \log^{(m-\ell)} T \right\rceil$, set the offered price as P_ℓ^*.

8: At the end of period $\tau_{\ell+1}$, compute the sample mean $\bar{X}^\ell$ from period $\tau_\ell + 1$ to $\tau_{\ell+1}$:

$$\bar{X}^\ell := \frac{\sum_{j=\tau_\ell+1}^{\tau_{\ell+1}} X_j}{\tau_{\ell+1} - \tau_\ell}, \text{ where } X_j = \text{Number of items sold in period } j.$$

9: Choose an index $i_\ell \in \{1, \cdots, K\}$ which solves

$$\min_{i \in \{1,\cdots,K\}} \left| \bar{X}^\ell - d_i(P_\ell^*) \right|.$$

10: Set the next offered price as $P_{\ell+1}^* = p_{i_\ell}^*$, where $p_{i_\ell}^*$ is the optimal price for demand d_{i_ℓ}.

11: **end if**

12: **end for**

13: (Earning) From period $\tau_m + 1$ to period $\tau_{m+1} = T$, set the selling price as P_m.

factor $M_\Phi(P_\ell^*)$, which is to be defined later, the lengths of phases are iterated-exponentially (tetrationally) increasing, which ensures an optimal balance between exploration and exploitation.

At the end of learning phase ℓ, policy mPC computes the sample mean $\bar{X}^\ell$ of the sales under price P_ℓ^* (in line 8 of the algorithm). Since price P_ℓ^* is discriminative, the seller gains new information about the underlying demand in this learning phase. She then updates her belief on the true demand distribution to be $d_{i_{\ell+1}}$ (in line 9), and sets the offered price $P_{\ell+1}^*$ to be $p_{i_{\ell+1}}^*$ in the next phase. In going through all the learning phases, the seller progressively refines her estimate on the optimal price, which enables her to establish the choice of optimal price in the earning phase.

The function $M_\Phi(P)$ in line (7) of the mPC algorithm is defined as follows.

Definition 3.2. *Let $p \in \mathcal{P}$ be a discriminative price. We define $M_\Phi(p)$ as*

$$M_\Phi(p) := \frac{16\sigma^2}{\min_{i\neq j}\left(d_i(p) - d_j(p)\right)^2} \vee \frac{8b}{\min_{i\neq j}|d_i(p) - d_j(p)|}, \tag{3.3}$$

where the minimum is taken over distinct pairs of indices $i, j \in \{1, \cdots, K\}$.

Since we assume p to be discriminative, $M_\Phi(p)$ is well defined. The function $M_\Phi(p)$ measures the distinguishability of the demand functions $d_1, \cdots, d_k$ under the discriminative price p. We explain the definition of $M_\Phi(p)$ further in the analysis of mPC.

Define $M_\Phi^* = \max_{i\in\{1,\cdots,K\}} M_\Phi(p_i^*)$ and $r^* = \max_{i\in\{1,\cdots,K\}} r_i^*$. The following result shows that the regret of mPC is bounded by $O(\log^{(m)} T)$.

Theorem 3.3. *Suppose the demand set Φ satisfies Assumption 3.1. For all $T \geq 1$, the regret of mPC is bounded by*

$$\mathrm{Regret}_\Phi^{\mathsf{mPC}}(T) \leq C_\Phi(P_0^*)\max\{\log^{(m)} T, 1\} + 4(M_\Phi^* + 1)r^*,$$

where $C_\Phi(P_0^*) = \max_{i\in\{1,\cdots,K\}}\{M_\Phi(P_0^*)(r_i^* - r_i(P_0^*))\}$.

Proof Idea of Theorem 3.3. In the proof, we establish that the regret incurred in Phase 0 is $O(\log^{(m)} T)$, and the cumulative regret incurred in the remaining phases is $O(1)$. At the beginning of Phase 0, which is also the beginning of the sale horizon, the seller has no information on the optimal price. Thus, the regret during Phase 0 is proportional to

the length of Phase 0. However, in each of the subsequent phases, the seller can choose a price based on the previous sale history. By choosing the lengths of the subsequent phases appropriately, we ensure that the total regret in these phases is $O(1)$.

Remark 3.4. *In Phase 0, the discriminative price P_0^* is given as a input. One can further reduce the regret bound by choosing a discriminative price P_0^* which minimizes the regret during Phase 0, namely $C_\Phi(P_0^*)$.*

Remark 3.5. *In line* (9) *of Algorithm 6, the test to select a demand function i_ℓ is a simple comparison between the sample mean $\bar{X}^\ell$ and the mean demand function value $d_i(P_\ell^*)$. Therefore, the algorithm does not require the seller to know the demand distributions for each demand model. Nevertheless, if the seller does know the demand distribution, line* (9) *can be replaced by other selection criteria, such as a likelihood ratio test, to improve the learning accuracy.*

Remark 3.6. *The proof shows that in the special case of $m = 1$, Assumption 3.1 is not required for Theorem 3.3. We only require that the initial price P_0^* is discriminative.*

3.3.2 Lower Bound

We show next that for a family of problem instances, any m-change policy incurs a regret of $\Omega(\log^{(m)} T)$. Thus, the regret achieved by the m-change policy mPC is optimal up to a constant factor.

Consider a problem instance (Γ) that satisfies the following conditions:

1. There exists a constant $Q_\Gamma > 0$, such that $\sum_{i=1}^{K}(r_i^* - r_i(p)) \geq Q_\Gamma$ for all $p \in \mathcal{P}$.
2. The demand $D(p) \in \mathbb{N}$ for any price $p \in \mathcal{P}$.
3. Given $p \in \mathcal{P}$, there exists a subset $\mathcal{B}_p \subset \mathbb{N}$, such that for all i, $\mathbb{P}_i(D(p) = d) > 0$ if and only if $d \in \mathcal{B}_p$.
4. There exists a constant $0 < \kappa_\Gamma < 1$, such that $\mathbb{P}_i(D(p) = d)/\mathbb{P}_j(D(p) = d) \geq \kappa_\Gamma$ for all $i, j \in \{1, \ldots, K\}$, $p \in \mathcal{P}, d \in \mathcal{B}_p$.

The first condition states that there is no price $p \in \mathcal{P}$ that simultaneously maximizes the revenue of all demand functions in Φ. This ensures that the problem instance is nontrivial and a learning process is necessary for maximizing the revenue when the demand function is

unknown. The second condition is that demand must be integers. The third condition states that all demand functions have the same support for a given price. The fourth condition states that the ratios of probability mass functions of different demand models are bounded.

The key step in the proof of the lower bound theorem is to quantify the performance of a pricing policy under different demand functions. This is made precise by following lemma.

Lemma 3.7 (Change-of-Measure Lemma). *Let $H_t = (P_1, X_1, \cdots, P_t, X_t)$ be the history observed by the end of period t, and let h_t be a realization of H_t. For any non-anticipating pricing policy π, we have*

$$\mathbb{P}_i^{\pi}(H_t = h_t) \geq \kappa_{\Gamma}^{t} \mathbb{P}_{i'}^{\pi}(H_t = h_t),$$

for all $i, i' \in \{1, \ldots, K\}$. The constant κ_{Γ} is defined in the condition (Γ).

The regret lower bound of any m-change policy is formally stated in the following.

Theorem 3.8 (Lower Bound Theorem). *For any m-change policy π on problem instance Γ, there exists a constant $\theta_m > 0$ such that for any $T > \theta_m$, we have*

$$\mathrm{Regret}_{\Phi}^{\pi}(T) \geq \frac{1}{K} C_{\Gamma} Q_{\Gamma} \log^{(m)} T,$$

where $C_{\Gamma} := (-8 \log \kappa_{\Gamma})^{-1} \wedge 1$ and Q_{Γ} is given by the first condition of (Γ).

We acknowledge here that the main idea of the proof of Theorem 3.8 is due to Wang Chi Cheung. The complete proof can be found in Cheung et al. (2015).

Taken together, the proofs of the upper and lower bounds provide important insights into the structure of any optimal m-change policy. With high probability, an optimal m-change policy has $m - 1$ learning phases of lengths $\Theta(\log^{(m)} T), \cdots, \Theta(\log T)$. They are followed by the last phase, which is the earning phase on the last $T - \Theta(\log T)$ time periods.

The lengths of the learning phases are set in a way to ensure an optimal balance between learning and earning. If any of the learning phases is shortened significantly, such lack of learning will incur a large regret in the subsequent phases. In general, for each $\ell \in \{1, \cdots, m\}$, if the ℓ^{th} learning phase is of length $o(\log^{(m-\ell+1)} T)$, then a regret of $\Omega(\log^{(m-\ell)} T)$ is incurred in the subsequent phases. This quantifies the value of learning in any m-change policy.

3.3.3 Unbounded but Infrequent Price Experiments

Policy mPC defines m learning phases with iterated-exponentially (tetrationally) increasing lengths. This motivates us to consider a modification of mPC, which improves the regret bound to a constant. We call this modified policy uPC (which standing for "unbounded price changes"), see Algorithm 7. Although the number of price changes under this policy is not bounded by any finite number as T increases, it grows extremely slowly with order $O(\log^* T)$, where $\log^* T = \min\{m \in \mathbb{Z}^+ : \log^{(m)} T \leq 1\}$. For example, for $T \leq 3,000,000$, we have $\log^* T \leq 3$. According to the Lower Bound Theorem in Section 3.3.2, this is the minimum growth rate possible.

Algorithm 7 Policy uPC

1: INPUT:
- A set of demand functions $\Phi = \{d_1, \cdots, d_K\}$.
- A discriminative price P_0^*.

2: Set $\tau_0 = 0$.
3: **for** $\ell = 0, 1, \cdots$ **do**
4: From period $\tau_\ell + 1$ to $\tau_{\ell+1} := \tau_\ell + \left\lceil M_\Phi(P_\ell^*) e^{(\ell)} \right\rceil$, set the offered price as P_ℓ^*.
5: **if** $T \leq \tau_{\ell+1}$ **then** stop the algorithm at period T.
6: **else**
7: At the end of period $\tau_{\ell+1}$, compute the sample mean $\bar{X}^\ell$ from period $\tau_\ell + 1$ to $\tau_{\ell+1}$:

$$\bar{X}^\ell := \frac{\sum_{j=\tau_\ell+1}^{\tau_{\ell+1}} X_j}{\tau_{\ell+1} - \tau_\ell}, \text{ where } X_j = \text{Number of items sold in period } j.$$

8: Choose an index $i_\ell \in \{1, \cdots, K\}$, which solves

$$\min_{i \in \{1, \cdots, K\}} \left| \bar{X}^\ell - d_i(P_\ell^*) \right|.$$

9: Set the next offered price as $P_{\ell+1}^* = p_{i_\ell}^*$, where $p_{i_\ell}^*$ is an optimal price for demand d_{i_ℓ}.
10: **end if**
11: **end for**

Proposition 3.9. *Suppose Assumption 3.1 holds. For all $T \geq 1$, the pricing policy* uPC *has regret*

$$\text{Regret}^{\text{uPC}}(T) \leq C_\Phi(P_0^*) + 2(M_\Phi^* + 1) r^*,$$

where $C_\Phi(P_0^*) = \max_{i \in \{1, \cdots, K\}} \{M_\Phi(P_0^*)(r_i^* - r_i(P_0^*))\}$.

Furthermore, uPC is an *anytime* policy, meaning that the seller can apply uPC algorithm

without any knowledge of T. Anytime policies can be used for customized pricing. In customized pricing, each customer arrival is modeled as a single time period, so T is the total number of customers arrivals Harrison et al. (2012), Broder and Rusmevichientong (2012). Since uPC is an anytime policy, the seller is not required to know the total number of customers arrivals.

In comparison, if the seller is only allowed to change price m times, it is impossible to achieve the optimal regret bound $O(\log^{(m)} T)$ if the seller does not know T. The Lower Bound Theorem shows that in order to achieve the optimal regret bound, the ℓ^{th} price change must occur at $\Theta(log^{(m-\ell+1)}T)$, so it is impossible to determine when to change price without knowing T.

3.3.4 Discussion on the Discriminative Price Assumption

The $O(\log^{(m)} T)$ regret of mPC and the $O(1)$ regret of uPC hold under the assumption that there exists an optimal discriminative price for each demand function (Assumption 3.1). In fact, one can show that this assumption is necessary for any non-anticipating policy to achieve a regret better than $o(\log T)$.

Proposition 3.10. *If Assumption 3.1 is violated, then there exists a price set $\mathcal{P}$ and a demand set Φ such that any non-anticipating pricing policy incurs a regret of $\Omega(\log T)$, even if that policy is allowed to change price for infinitely many times.*

Proposition 3.10 implies that the best possible regret bound without Assumption 3.1 is $O(\log T)$.

The remaining question is what the best regret bound is when Assumption 3.1 does not hold. We show next that for any set of K demand functions, policy kPC (see Algorithm 8) achieves regret bound of $O(\log T)$ with at most $K-1$ price changes.

For this purpose, we need the following definition:

Definition 3.11. *For any nonempty subset of demand functions $A \subset \{d_1, \ldots, d_K\}$, let*

$$\tilde{p}_A \in \arg\max_{p \in \mathcal{P}} |\{d_i(p) \mid d_i \in A\}|.$$

In other words, $\tilde{p}_A$ is a price that maximizes the number of distinct values of $d_i(p)$ for all $d_i \in A$.

Furthermore, define

$$\tilde{M}_A(p) := \frac{8\sigma^2}{\min_{(i,j):d_i(p)\neq d_j(p)} (d_i(p) - d_j(p))^2} \vee \frac{4b}{\min_{(i,j):d_i(p)\neq d_j(p)} |d_i(p) - d_j(p)|}, \tag{3.4}$$

where the minimum is taken over all pairs of demand functions $d_i, d_j \in A$ such that $d_i(p) \neq d_j(p)$.

Note that if $|A| \geq 2$, for any pair of demand functions d_i and d_j in A, we can always find a price p such that $d_i(p) \neq d_j(p)$, because otherwise the two demand functions are identical. So the value $\tilde{M}_A(\tilde{p}_A)$ in line (4) of Algorithm 8 is well defined for any $|A| \geq 2$.

Algorithm 8 Policy kPC.

1: INPUT: A set of demand functions $\Phi = \{d_1, \cdots, d_K\}$.
2: (Learning) Set $A \leftarrow \Phi$. Set $\ell = 0$, $\tau_0 = 0$.
3: **while** $|A| \neq 1$ **do**:
4: Set the price as $P_\ell^* = \tilde{p}_A$ from period $\tau_\ell + 1$ to $\tau_{\ell+1} := \tau_\ell + \left\lceil \tilde{M}_A(P_\ell^*) \log T \right\rceil$.
5: At the end of period $\tau_{\ell+1}$, compute the sample mean $\bar{X}^\ell$ from period $\tau_\ell + 1$ to $\tau_{\ell+1}$:

$$\bar{X}^\ell := \frac{\sum_{j=\tau_\ell+1}^{\tau_{\ell+1}} X_j}{\tau_{\ell+1} - \tau_\ell}, \text{ where } X_j = \text{Number of items sold in period } j.$$

6: Update A: keep all d_i in set A if it is a minimizer of $\min_{d_i \in A} \left|\bar{X}^\ell - d_i(P_\ell^*)\right|$. Eliminate other demand functions from A. If there are two minimizers d_i and d_j such that $d_i(P_\ell^*) < \bar{X}^\ell < d_j(P_\ell^*)$, remove d_j and only keep d_i.
7: Set $\ell \leftarrow \ell + 1$.
8: **end while**
9: (Earning) Suppose $A = \{d_i\}$. From period $\tau_\ell + 1$ to period $\tau_{\ell+1} = T$, set the selling price as $P_\ell^* = p_i^*$.

Proposition 3.12. *For all $T \geq 1$, the regret of kPC is bounded by*

$$\text{Regret}_\Phi^{kPC}(T) \leq (K-1)(\tilde{M}_\Phi r^* \log T + 3r^*),$$

where $\tilde{M}_\Phi = \max_{A \subset \{1,\ldots,K\}} \tilde{M}_A(\tilde{p}_A)$.

Proof Idea of Proposition 3.12. In each of the learning phases, the definition of the algorithm (line 6) guarantees that at least one demand function is eliminated. The number of iterations in the while loop is at most $K - 1$, and thus the regret of the learning phases is $O((K-1)\log T)$. We then show that with high probability, the single demand function remained in the earning phase is the true demand function.

Figure 3-1: Screenshot of A Restaurant Deal on Groupon's Website

3.4 Field Experiment at Groupon

We collaborated with Groupon, a large e-commerce marketplace for daily deals, to implement the pricing algorithm presented in Section 3.3. Groupon offers discount deals from local merchants to subscribed customers. By the second quarter of 2015, Groupon served more than 500 cities worldwide, had nearly 49 million active customers and featured more than 510,000 active deals.

To illustrate Groupon's business model, consider the Japanese restaurant deal taken from Groupon's website and depicted in Figure 3-1. The deal can be purchased through Groupon at $17 and redeemed at the Japanese restaurant for $30. The amount paid by a customer ($17) is called "booking". The booking is then split between Groupon and the local merchant. For example, an agreement may allow the local merchant to receive $12 and Groupon to keep $5 as its net revenue. In most cases, a deal is only available for a limited time, ranging from several weeks to several months.

Prior to our collaboration, Groupon applied a fixed price strategy for each deal. Our initial analysis suggested that Groupon could benefit from the dynamic pricing algorithm proposed in Section 3.3 for the following reasons:

- A majority of Groupon's deals are offered on its website for the first time, and there is not enough historical data to predict demand before new deals are launched. Thus, there is an opportunity for Groupon to learn from real time sales data after deals are launched so as to improve its demand forecast and pricing strategies.

- Deals are offered for a limited time, so there is a time tradeoff between price experimentation and revenue maximization, which is a tradeoff addressed by our pricing algorithm.

- Groupon managers prefer using as few price changes as possible for a number of reasons. First, they are concerned that frequent price changes may confuse customers. Second, it is easy to communicate and explain a simple dynamic pricing algorithm with minimal price changes to merchants.

- Since Groupon mainly offers coupons instead of physical products, we ignored the inventory constraint in this implementation. Technically, each deal has a cap that specifies the maximum quantity that can be sold, but historical data show that only a small fraction of deals have reached their caps. Moreover, once a deal has reached its cap, Groupon can renegotiate with the merchant to increase the cap. So the unlimited inventory assumption is a reasonable approximation of reality.

The pricing algorithm proposed in Section 3.3 requires a set of possible demand functions as an input. In the rest of this section, we propose a method to generate demand function sets based on clustering. We then describe implementation details and state additional business constraints specified by Groupon. Finally, we present the implementation results and the analysis.

3.4.1 Generating the Demand Function Set

Recall that we assume a finite demand function set, Φ, in the model assumption. In reality, it is unlikely that the true demand function will belong to the finite set that we estimated. However, our goal is to find a set Φ, such that the true demand function can be well approximated by at least one function in the set. To this end, we propose the following three step process to generate a finite set of linear demand functions.[1]

– Step 1: We first collect data on historical deals that have been tested for dynamic pricing. Given a new deal, we select a subset of historical deals with similar features (e.g. deal category, price range, discount rate). Since deals in

[1]We use linear demand functions to approximate local price elasticity, but our method can also be adapted for other forms of parametric demand families such as Cobb-Douglas functions, as long as the demand functions can be specified by finitely many parameters.

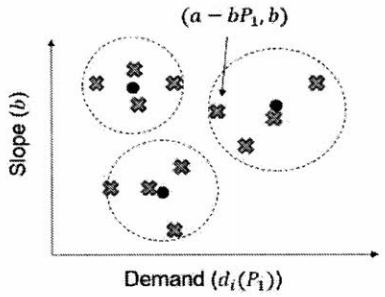


Figure 3-2: Applying K-means Clustering to Generate K Linear Demand Functions

this subset were tested for dynamic pricing, they have been offered under at least two different prices, so we can fit a linear demand function for *each* historical deal.

– Step 2: The linear demand functions are then mapped into points on a 2-dimensional plane according the following rule: the y-coordinate of the plane represents the negative slope of linear functions, and the x-coordinate represents the mean demand valued at the initial price of the new deal. For example, suppose we fit a demand function $d_i(p) = a - bp$ for a historical deal, and the new deal has an initial price P_1, then this demand function is mapped to the point $(a - bP_1, b)$. Using this mapping, each deal in the subset can be represented by a point on the plane, shown as an 'x' in Figure 3-2. Moreover, the mapping is bijective, so there is a one-to-one correspondence between any point on the plane and a linear demand function.

– Step 3: We apply K-means clustering to group the points into K clusters. For example, Figure 3-2 shows the clustering result for $K = 3$. Note that the centroids of the clusters are points on the plane, so according to the bijective mapping defined in Step 2, they also represent linear demand functions. In particular, if a centroid is located at (x, y), it corresponds to the linear function $d(p) = x + (P_1 - p)y$. Collectively, the centroids of K clusters represent K linear demand functions, which form the demand function set.

Typically, Step 1 would produce hundreds to thousands of historical deals. If we omit Step 3 and simply use linear functions generated in Step 2 as input to our dynamic pricing algorithm, it would be difficult for the learning algorithm to correctly identify the true

demand function among thousands of functions within a short period of time. Therefore, we use clustering in Step 3 to limit the number of demand functions to be learned.

To minimize the demand prediction error, we need to select an appropriate number K to balance the bias-variance tradeoff. When K is large, we have a large set of demand functions that can better approximate the true demand function (small bias), but learning about the correct demand function is hard (large variance), since the constant M_{Φ}^{*} in the regret bound of Theorem 3.3 is large. When K is small, the demand function set has a large approximation error relative to the true demand function (large bias), but it is easier to identify the best function in this set (small variance).

To determine the best value of K, we apply cross-validation: The historical deals are randomly split into training and testing sets. Each deal in the testing set had been offered under two prices (p_1, p_2) and is treated as a new deal with initial price (p_1). We then generate demand functions from the training set following the three-step process described above. After that, we select one function among the K functions whose value at p_1 is the closest to the actual demand of the new deal at p_1, since this is the function that would have been chosen by our learning algorithm. Next, we compare the realized demand under price p_2 to the mean demand predicted by the selected function at p_2. The difference between these two values can be interpreted as the prediction error of our learning algorithm. We repeat this process for different values of K, and choose K that minimizes prediction error.

In Figure 3-3, we plot the mean squared error (MSE) of demand prediction for different values of K. The dark curve is the MSE of purchase quantity per impression (i.e., per customer visit), and the gray curve is the MSE of booking per impression. The figure shows that the prediction error is large for small values of K, and then decreases as K increases. This implies that for small values of K, none of the demand functions in the demand set is close to the true demand function (i.e., large bias), so the prediction error is large. For very few demand functions ($2 \leq K \leq 10$), the bias is so large that the prediction error is even worse than that of fixed pricing ($K = 1$). Therefore, it is important to choose a large enough K so that at least one of the demand function in the demand set is close to the true demand function. We select K to be around 100 in our final implementation at Groupon. Once K is chosen, the demand functions are generated according to Step 3 in the aforementioned process. Notice that the prediction error will eventually go up due to over-fitting when we select a K that is sufficiently large (i.e., large variance). This trend is

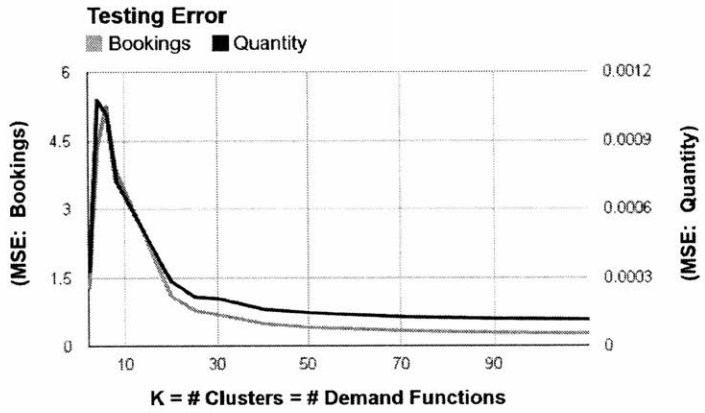


Figure 3-3: Mean Squared Error of Demand Prediction for Different Values of K

not shown in Figure 3-3, because we didn't have enough historical deals in this example to demonstrate over-fitting.

3.4.2 Implementation Details

Compared with the model defined in Section 3.2, we face some additional constraints in practice due to Groupon's business rules, so our pricing algorithm must be adjusted to include these implementation details.

For each deal, we suppose that the allowable price set is a continuous interval $\mathcal{P} = [\underline{p}, \overline{p}]$. Each time period is defined as one day, and prices of deals are changed at midnight local time. Since customer traffic for Groupon is not necessarily time homogeneous, in the data pre-processing step, the sales data have been normalized to remove the time effect. More specifically, the normalized sales quantity is the original sales quantity multiplied by a normalization factor, which only depends on time (e.g., number of days after launch, holiday/weekend, etc.) and is specific for each deal category. After this normalization step, we can treat demand in different time periods as stationary.

The main result in Section 3.3 shows that the first price change captures most of the benefit of dynamic pricing, which reduces regret from $\Theta(T)$ to $\Theta(\log T)$. So we decided to apply the single price change policy at Groupon, i.e., the mPC policy defined in Section 3.3.1 with $m = 1$.

In the Groupon implementation, the initial price of a deal is negotiated by the local merchant and Groupon, so we treat the initial price as a fixed input — this is the same price

Table 3.1: Deals Selected in the Field Experiment

	Beauty/Health	Food/Drink	Leisure/Activities	Services	Shopping
# deals	591	111	259	274	60
Avg bks/day	35.1	88.0	37.6	27.2	9.3

that Groupon would have used under its existing fixed price policy. Since a finite set of linear demand functions has only finite non-discriminative prices in the price interval $[\underline{p}, \overline{p}]$, it is unlikely that the initial price is non-discriminative. In fact, in all the examples that we tested, the initial price P_1 is discriminative with respect to the demand set.

When changing price, Groupon decided to allow for price decrease only. The main reason for this decision is that many merchants use Groupon as a marketing channel to attract new customers, so Groupon is unwilling to reduce sales quantity by increasing price, even if it may potentially increase revenue. Groupon further imposes a constraint that price can only be decreased between 5% and 30%. Therefore, if the pricing algorithm recommends either a price decrease of less than 5% or a price increase, then no price change is made. If the algorithm recommends a price decrease of more than 30%, then price is decreased by only 30%.

Moreover, the agreement between Groupon and the local merchant specifies that the merchant's share of revenue is not affected after price decrease. For example, suppose a deal has an initial price of \$20, and both Groupon and the merchant receive \$10 from each deal purchased under the initial price. If the pricing algorithm reduces the price to, say, \$15, then Groupon would receive \$5 and the merchant would still receive \$10.[2] This agreement guarantees that merchants always benefit from price decrease, and hopefully this would make merchants more willing to accept the dynamic pricing policy, which is designed to be a win-win strategy for both Groupon and local merchants. In practice, Groupon always give the merchants two options: they can either use the existing fixed price policy, or they can choose to use the dynamic pricing algorithm.

3.4.3 Field Experiment Results

The field experiment at Groupon consists of two stages. In the first stage we focused on fine-tuning the pricing algorithm, while the second stage was used as a final evaluation of

[2]In our pricing algorithm, we can easily include merchant's revenue share by redefining the optimal price as $p_i^* \in \arg\max_{p \in \mathcal{P}} (p - c) d_i(p)$, where c is the merchant's fixed revenue split. All the results in Section 4 go through with this modification.

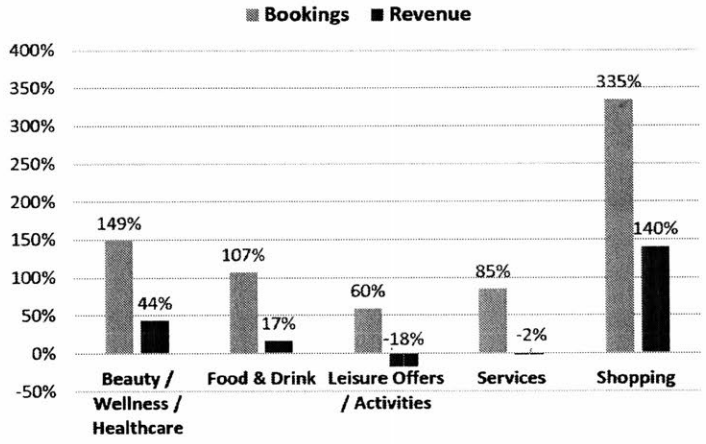


Figure 3-4: Bookings and Revenue Increase by Deal Category

the algorithm performance.

In the first stage, we tested different ways to generate demand function sets (Φ). The method presented in the previous subsection was the approach that we finally selected. We also used the first stage experiment to test different price switching time. In the definition of algorithm mPC for $m = 1$, the price is switched at period $\lceil M_\Phi(P_0) \log T \rceil$, where the constant $M_\Phi(P_0)$ is given by Equation (3.3). However, this constant is mainly designed for proving the theoretical regret bound. In practice, we tested several price switching times (between 1 to 7 days) through live experiment, and the switching time with the best performance was chosen.

The second (evaluation) stage of the field experiment lasted for several weeks. During this testing period, 1,295 deals were selected by our dynamic pricing algorithm for price decrease. These deals span five product categories: Beauty/Healthcare, Food/Drink, Leisure/Activities, Services, and Shopping. Table 3.1 provides information on the number of deals and the average daily bookings per category. We note that if a deal was tested for dynamic pricing but was not selected for price decrease, that deal is not included in this dataset.

In the field experiment we focused on two performance metrics. One is the total amount of money paid by customers to Groupon, referred to as *bookings*, and it is directly related to Groupon's market share. The other is the part of the revenue that Groupon keeps after paying local merchants, referred to simply as *revenue*. For each product category, we compare the average bookings and revenue per day pre and post price change. Since the initial

price is determined by Groupon's current fixed price policy, comparing the average daily bookings and daily revenue before and after the price change measures the revenue lift of our dynamic pricing policy over Groupon's existing fixed price policy. As we did in the data preprocessing step, the lift has been normalized to control for time-varying demand effect. Specifically, when generating Figure 3-4, we multiply the booking and revenue quantities per day by a normalization factor, which only depends on time (e.g., number of days after launch, holiday/weekend, etc.) and deal category.

Figure 3-4 shows the average increase in daily bookings and revenue after price decrease. Overall, daily bookings are increased by 116%, and daily revenue is increased by 21.7%. Among the five categories, Beauty/Healthcare, Food/Drink, and Shopping have significant increase in both revenue and bookings. Services category has almost no revenue change but significant bookings increase. Leisure/Activities category has a negative gain in revenue. For Groupon, Beauty/Healthcare and Food/Drink are the two leading categories in terms of revenue generation. Our pricing algorithm has solid revenue gain in these two categories.

Further analysis of the field experiment result shows that reducing price has a much bigger impact on deals that have fewer bookings per day, which holds across all categories. Overall, the average increase in daily revenue is 116% for deals with bookings per day below the median, while the increase is only 14% for deals with bookings per day above the median. This effect also explains the big increase in bookings and revenue for the Shopping category, the category with the smallest mean daily bookings among all five categories, see Table 3.1. Therefore, the increases on the Shopping deals are also the most significant.

Lastly, our pricing algorithm has a poor performance for Leisure/Activities category, despite the fact that this categories has almost the same level of average daily bookings as the Beauty category. We suspect the reason is that some features of customer demand for Leisure/Activities deals are not captured by our demand model. For example, it might be that the weekend/holiday effect is stronger for this category than we estimated. The data preprocessing step does include time normalization for weekend/holiday effect, but it is likely that customers purchase Leisure/Activities deals a few days before holidays, instead of during holidays. Therefore, further work is needed to improve the demand prediction method for the Leisure/Activities category.

Chapter 4

An Adaptive Robust Optimization Approach to Supply Chain Risk Mitigation

In the last few years we have seen significant increase in supply chain risk and as a results managing supply chain risks has emerged as one of the top business challenges. Indeed, the number of financially significant natural catastrophes have been steadily going up over the last decades Hedde (2014). As reported in a Forbes article, 80 percent of companies worldwide reported that better protection of supply chains is a priority Culp (2013).

To effectively manage supply chain risks, a firm needs to maintain a certain performance guarantee under various scenarios of supply and demand uncertainty. In this study, we focus on a hybrid of two popular risk mitigation strategies: holding additional inventory and employing (process) flexibility. To study the hybrid strategy, we introduce a two-stage robust optimization model where the inventory is (optimally) stored in the first stage, while in the second stage, after the capacity and demand uncertainties are realized, the firm would have a recourse decision of allocating its (process) flexible capacity to minimize its performance loss. To solve this model, we propose a constraint generation algorithm that takes advantage of the special network flow structure under general polyhedral uncertainty sets. Moreover, we analyze a special case of the model and derive interesting insights into the optimal inventory strategies under different degrees of process flexibility.

Next, we explain how process flexibility and inventory are applied as risk mitigation

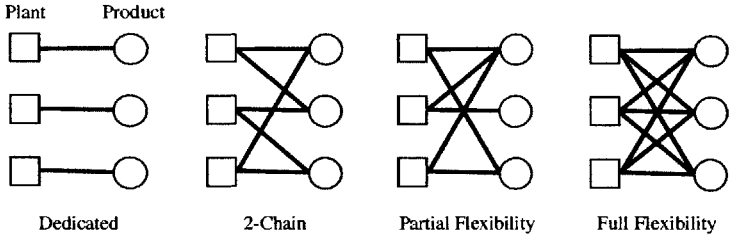
strategies.

Risk Mitigation Strategies Risk mitigation inventory, also known as protective inventory, has been identified in a number of papers as an important tool for dealing with supply chain risk (see literature review in Section 4.1.1). However, holding a large amount of inventory would incur high inventory holding cost. As a result, firms would typically hold the minimum amount of inventory subject to a given performance requirements.

Process flexibility is defined as the "ability to build different types of products in the same manufacturing plant or on the same production line at the same time" Jordan and Graves (1995). For example, under *full flexibility* design, each plant is capable of producing all products, while under *dedicated* design (i.e. no process flexibility), each plant is capable of producing just a single product, see Figure 4-1.

With process flexibility, the firm is capable of adjusting production under uncertainties in both demand and supply, and as a result, is in a better position to match its available capacity and realized demand. Unfortunately, implementing full flexibility can be very expensive since each plant needs to be capable of producing all products Simchi-Levi (2010). Therefore, *partial* flexibility designs are considered. Under such designs, each plant is capable of producing just a few products.

Figure 4-1: Process Flexibility Designs



It is important to note that inventory decisions and process flexibility are closely related. Consider the dedicated and full flexibility design in Figure 4-1. Suppose that the firm holds inventory for Product 2, and a disruption occurs at Plant 1. Under the dedicated design, the inventory of Product 2 would not be helpful, since the inventory cannot be used to satisfy any of the demand for Product 1. However, under the full flexibility design, inventory of Product 2 can be used to satisfy demand for product 2 while the flexible production capacity at Plant 2 can be allocated to produce Product 1. In other words, when a firm has both

inventory and process flexibility, inventory helps to free up excess flexible capacities during an unforeseen event, thus improving the firm's ability to mitigate risk. As a result, the existence of process flexibility can reduce the amount of inventory needed, and the synergy between inventory and process flexibility makes the hybrid strategy very compelling.

This paper is motivated by a collaboration with a large automotive manufacturing company that applies both process flexibility and inventory to manage risk in its supply chain. The manufacturer has many production facilities and some of the facilities are endowed with process flexibility, i.e., each of these facilities has the capability to process different components for different vehicle lines. At the same time, the manufacturer keeps inventories for different components to guard against potential risks. Of course, the manufacturer would like to hold as little inventory as possible, while using the inventory and process flexibility to maintain its service level under various unforeseen events.

Although it is well documented that both flexibility and inventory can improve supply chain resiliency (see literature review in Section 1.1.3 and Section 4.1.1), the synergy between inventory and process flexibility is not well understood. To the best of our knowledge, no existing literature has considered a hybrid strategy combining partial process flexibility designs and inventory as a way to mitigate against risks in a multi-product supply chain. Motivated by this observation, we introduce a two-stage robust optimization model that determines the optimal inventory allocation under general process flexibility designs. Next, we provide a summary of our findings.

4.1 Overview and Summary of Results

Robust Optimization Model for Inventory Allocation under (Process) Flexible Designs. In Section 4.2, we introduce a two-stage robust programming to model the firm's inventory decision under arbitrary (partial) process flexibility designs. In the first stage, given a process flexibility design, the firm optimizes its inventory levels for all products to ensure that demand shortage does not exceed a certain level subject to uncertainties in production capacity and demand. The uncertainties are modeled using polyhedral uncertainty sets. In the second stage, after the uncertainties are realized, the firm uses both inventory and process flexibility to minimize demand shortage.

Computational Algorithm. In Section 4.3, we reformulate the two-stage robust optimization problem as a linear program with exponentially many constraints. To solve the linear program, we propose a delayed constraint generation method. In the constraint generation, we show that our sub-problems can be reformulated as mixed integer linear programs (MILP) by applying the McCormick envelopes and taking advantage of the structure of the second stage recourse problems. In our computational experiments, we were able to solve the two-stage robust optimization problem for systems with up to 100 products. Moreover, we show that our algorithm will converge to the optimal solution in at most 2^N iterations, where N is the number of products of the system.

Total Inventory Levels. In Section 4.4.1, we provide a closed-form characterization of the exact total inventory levels for a family of flexibility designs known as the K-chains, when demand and capacity uncertainty sets are symmetric. Using this characterization, we analyze a family of different flexibility designs and show the impact of these designs on the firm's optimal inventory levels. In the presence of plant disruption, changing from a dedicated network to a 2-chain design provides a large portion of benefit. Surprisingly, significant benefit is also achieved by increasing the degree of flexibility beyond the 2-chain design, even when demand variability is relatively low. For example, the minimum inventory level associated with full flexibility can be often achieved by having limited degrees (4-chain or 5-chain) of flexibility.

Inventory Allocation Strategy. In Section 4.4.2, we study a specific class of uncertainty sets and find that the optimal inventory allocation pattern can be completely different under different flexibility designs. In particular, when the firm has a high degree of flexibility, it may be optimal to hold more inventory for products with lower demand variability, and to hold less inventory for products with higher demand variability. This observation is in contrast with classical inventory management theory, which states that higher demand variance implies that more inventory is needed in order to achieve the same service level. Indeed, we show that classical theory remains true when the firm has a low degree of flexibility. But when plants are highly flexible, it is more effective to satisfy high variability demand using plant capacity, and hence more inventory is allocated to low variability demand.

The insights described above are mostly obtained for systems where the number of plants

is equal to the number of products. Although such settings are not typical in real world systems, as Jordan and Graves (1995) argues, understanding these ideal settings provides insight into realistic scenarios. Indeed, in Section 4.5.2, we consider an example where the number of plants is different than the number of products. This example is based on a data set with 16 vehicle models and 8 assembly plants at General Motors presented in Jordan and Graves (1995). Using numerical experiments, we find that all main insights are carried over to this example.

4.1.1 Related Literature

There is rich literature on inventory for risk mitigation, see the review in Section 1.1.3. Researchers have also considered hybrid strategies where firms apply both dual sourcing and inventory to mitigate risks Gürler and Parlar (1997), Tomlin (2006).

Process flexibility, also referred to as "mix flexibility" or "product flexibility," has also been observed as potential risk mitigation tool. Tomlin and Wang (2005) considers a risk mitigation strategy that uses a combination of mix-flexibility and dual sourcing.

While many researchers investigated the effectiveness of fully flexible resources Fine and Freund (1990), Tomlin and Wang (2005), our model allows one to study arbitrary partial process flexibility designs in a multi-plant, multi-product system. This feature is motivated by the seminal work of Jordan and Graves (1995), who argue that fully flexible resources are often too expensive or impossible to implement, while a little bit of flexibility in the system can often provides the same benefit as full flexibility. More recently, the effectiveness of sparse flexibilities designs have been verified by a number of theoretical developments Chou et al. (2010, 2011), Simchi-Levi and Wei (2012), Wang and Zhang (2015).

In this paper, we propose a robust optimization approach to study the risk mitigation strategy of combining both partial process flexibility and inventory. The model is a two-stage robust optimization model where the hybrid strategy requires the firm to allocate inventory before the uncertainties are realized (*ex-ante*), and to schedule its flexible production after uncertainties are realized (*ex-post*). Our approach follows the recent approach in the robust optimization literature Ben-Tal et al. (2009), Bandi and Bertsimas (2012), which proposes to model probabilistic distributions with the worst-case scenario in an uncertainty set. It also is similar to the model studied by Graves and Willems (2000), who argued that in supply chain management, it is often easier for the firm to commits to service level guarantees under

a range of uncertainties, rather than probabilistic service guarantees.

We also survey some of the literature of two-stage robust optimization relating to our work. Atamtürk and Zhang (2007) studied various two-stage robust network flow problem, showing that while the problem is NP-hard in general, it can yield polynomial-timed solutions for a special class of networks. Computational methods for solving general two-stage robust linear programs include Thiele et al. (2010) who proposed a cutting-plane method, while Zeng and Zhao (2013) proposed a column-and-constraint generation algorithm. More recently, Ardestani-Jaafari and Delage (2014) used the Affinely Adjustable Robust Counterpart framework to study a two-stage robust optimization model for location transportation problems. For a more detailed overview of the recent advances and applications of two-stage robust optimization, we refer the readers to the recent surveys Bertsimas et al. (2011), Gabrel et al. (2014).

4.2 The Model

Suppose that a firm has M plants, denoted by $\mathcal{S}_i$ for $i = 1, \ldots, M$, and produces N products, denoted by $\mathcal{T}_j$ for $j = 1, \ldots, N$. A plant may produce one or multiple products, and the *flexibility design* specifies which products each plant can make. More formally, the flexibility design can be represented as a bipartite network, where a link $(\mathcal{S}_i, \mathcal{T}_j)$ exists if and only if plant i is able to produce product j. We refer to the set of such links as $\mathcal{F}$. For a given flexibility design $\mathcal{F}$ and a subset of product $A \subset \{\mathcal{T}_1, \ldots, \mathcal{T}_N\}$, we define $P_{\mathcal{F}}(A) = \{\mathcal{S}_i : (\mathcal{S}_i, \mathcal{T}_j) \in \mathcal{F}, \mathcal{T}_j \in A\}$ as the plants that can produce at least one of these products.

We assume that the plant capacities and product demands are uncertain quantities, which reflect risks such as plant disruptions and the fluctuation of market demand. The firm thus faces a two-stage decision problem. In the first stage, the firm determines the inventory level for each product. Such inventory is also referred to as *risk mitigation inventory*, and s_j is used to denote the inventory of product j. In the second stage, the firm observes realized product demands and available plant capacities, and adjusts its production schedule to minimize demand shortage. The interplay between risk mitigation inventory and flexibility is a key factor that we capture in our model.

The firm's decisions can be naturally modeled as a two-stage robust optimization prob-

lem. We first define the firm's second stage problem of minimizing shortage in Section 4.2.1, and then outline the complete two-stage problem in Section 4.2.2.

4.2.1 Shortage Function

Suppose that the firm observes available capacity $\mathbf{c} = (c_1, c_2, \ldots, c_M)$ for each plant and realized demand $\mathbf{d} = (d_1, d_2, \ldots, d_N)$ for each product. We define $\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d})$ to be the minimum demand shortage, given flexibility design $\mathcal{F}$, inventory vector (decision from the first stage) $\mathbf{s} = (s_1, s_2, \ldots, s_N)$, capacities $\mathbf{c}$ and demands $\mathbf{d}$.

Let x_{ij} be the units of product j produced by plant i, and l_j be the shortage of product j. $\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d})$ is expressed as the following optimization problem.

$$\begin{aligned}
\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d}) = \min \quad & \sum_{j=1}^{N} l_j \\
\text{s.t.} \quad & \sum_{i:\,(\mathcal{S}_i, \mathcal{T}_j) \in \mathcal{F}} x_{ij} + l_j \geq d_j - s_j, \quad \forall 1 \leq j \leq N, \\
& \sum_{j:\,(\mathcal{S}_i, \mathcal{T}_j) \in \mathcal{F}} x_{ij} \leq c_i, \quad \forall 1 \leq i \leq M, \\
& l_j,\, x_{ij} \geq 0.
\end{aligned}$$

The first constraint is the definition of shortage or lost sales for each product. The second constraint specifies that the sum of units produced at each plant cannot exceed its realized capacity. Without loss of generality, we assume that each product consumes one unit of capacity, otherwise we can re-scale the units for each product. Throughout the paper, $\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d})$ will be referred to as the shortage function. We first state an observation on the shortage function; this observation will be useful later in the analysis.

Lemma 4.1. $\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d})$ *is jointly convex in* $\mathbf{s}$, $\mathbf{c}$ *and* $\mathbf{d}$.

4.2.2 Robust Optimization Model for Inventory Decision

In the first stage, the firm faces the problem of minimizing total inventory while ensuring that the shortage is at most δ. In what follows, δ is referred to as the *shortage allowance*. We use $\mathcal{U} \subset \mathbb{R}_+^M \times \mathbb{R}_+^N$ to denote the uncertainty set that models the uncertainty of *both* plant capacity and product demand. Throughout the paper, we will consider polyhedral uncertainty sets; that is, $\mathcal{U}$ will always be defined by a system of linear inequalities.

Let $\mathbf{s} = (s_1, s_2, \ldots, s_N)$ be the vector of inventory for all products. The optimization problem can be formulated as the following worst-case (WM) model:

$$\begin{aligned} \textbf{(Problem-WM)} \quad \min \quad & \sum_{j=1}^{N} s_j \\ \text{s.t.} \quad & \Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d}) \leq \delta, \quad \forall (\mathbf{c}, \mathbf{d}) \in \mathcal{U}, \\ & s_j \geq 0, \quad \forall 1 \leq j \leq N. \end{aligned}$$

One reason to study the worst-case model is its computational and analytical tractability. In Section 4.3, we show that the worst-case model can be solved effectively using a delayed constraint generation algorithm. In Section 4.4, we show that if the uncertainty set satisfies a certain symmetric property, we can characterize the optimal solution and thus provide analytical expression for the optimal inventory levels.

The worst-case model also provides several other advantages compared with its stochastic counterpart. First, it is difficult and often impossible to accurately assess the probability of a plant disruption, because disruption can occur from so many different sources, whether from natural disaster, epidemics or factory fire Simchi-Levi et al. (2014). Because the inventory level can be very sensitive to the probability of disruption, considering the worst-case scenario may offer a more "robust" approach. Second, for small probability events such as plant disruptions, managers might find it useful to understand the maximum possible shortage of demand under wide range of scenarios. This understanding may lead them to better identify scenarios where the supply chain is most vulnerable. Finally, it has been suggested in Graves and Willems (2000) that it is easier for managers to communicate with customers by committing to a certain service level under a range of scenarios rather than providing a probabilistic guarantee, which is difficult to understand or verify.

4.3 Optimization Algorithm

In this section, we propose a constraint generation algorithm that allows us to solve large instances of Problem-WM. The section is divided into two parts. In the first part (Section 4.3.1), we derive several analytical results on the shortage function $\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d})$. In the second part (Section 4.3.2), we use the analytical results to propose an algorithm that is tailored to the Problem-WM model.

4.3.1 Analysis of the Shortage Function

Recall that by the convexity of the shortage function (Lemma 4.1), $\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d})$ is maximized at the extreme points of $\mathcal{U}$. In the formulation of Problem-WM, $\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d}) \leq \delta$ can be expanded into a set of linear constraints by introducing a set of auxiliary variables $\mathbf{x}^{\mathbf{c},\mathbf{d}}$, $\mathbf{l}^{\mathbf{c},\mathbf{d}}$ for each $\mathbf{c}$ and $\mathbf{d}$ (Section 4.2.1). Unfortunately, if we simply expand $\Pi(\cdot)$ using the formulation in Section 4.2.1 for all extreme points of $\mathcal{U}$, we would have an astronomical number of variables and constraints even for a system with just a handful of plants and products.

Similarly to Benders' decomposition, we can avoid adding a huge number of variables in the recourse functions by taking the dual of the recourse function. Next, we apply the strong duality theorem and develop the following lemma.

Lemma 4.2. *The shortage function,* $\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d})$ *can be rewritten as*

$$\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d}) = \max \quad \sum_{j=1}^{N}(d_j - s_j)q_j - \sum_{i=1}^{M} c_i p_i \tag{4.1}$$

$$s.t. \quad q_j \leq p_i, \quad \forall (\mathcal{S}_i, \mathcal{T}_j) \in \mathcal{F}, \tag{4.2}$$

$$q_j \leq 1, \quad \forall 1 \leq j \leq N, \tag{4.3}$$

$$p_i,\, q_j \geq 0, \quad \forall 1 \leq i \leq M, 1 \leq j \leq N. \tag{4.4}$$

where the constraints in the optimization problem are totally unimodular.

Because the original formulation of $\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d})$ has network flow type constraints, the totally unimodular property is not surprising. Let $V(\mathcal{F})$ be the set of all $(\mathbf{p}, \mathbf{q})$ that satisfies Equations (4.2-4.4), and $V^{\mathbb{N}}(\mathcal{F})$ be the set of all integer points in $V(\mathcal{F})$. By Lemma 4.2, all extreme points of $V(\mathcal{F})$ are in $V^{\mathbb{N}}(\mathcal{F})$, so we can replace continuous decision variables $(\mathbf{p}, \mathbf{q})$ with binary variables without loss of optimality.

Next, we state the proposition which derives reformulation for Problem-WM with N variables and 2^N linear constraints.

Proposition 4.3. *For each* $A \subset \{\mathcal{T}_1, \ldots, \mathcal{T}_N\}$, *let* $(\mathbf{c}^A, \mathbf{d}^A)$ *be the optimal solution of*

$$\max_{(\mathbf{c},\mathbf{d}) \in \mathcal{U}} \sum_{\mathcal{T}_j \in A} d_j - \sum_{\mathcal{S}_i \in P_{\mathcal{F}}(A)} c_i, \tag{4.5}$$

where $P_{\mathcal{F}}(A) = \{\mathcal{S}_i : (\mathcal{S}_i, \mathcal{T}_j) \in \mathcal{F}, \mathcal{T}_j \in A\}$ *is the set of plants that can produce at least one product in* A. *Then, Problem-WM can formulated as the following linear program (LP).*

$$\min \quad \sum_{j=1}^{N} s_j \tag{4.6}$$

$$s.t. \quad \sum_{\mathcal{T}_j \in A} (d_j^A - s_j) - \sum_{\mathcal{S}_i \in P_{\mathcal{F}}(A)} c_i^A \leq \delta, \quad \forall A \subset \{\mathcal{T}_1, \ldots, \mathcal{T}_N\}, \tag{4.7}$$

$$s_j \geq 0, \quad \forall 1 \leq j \leq N. \tag{4.8}$$

When N is not a large number, we can solve the exact linear programming formulation stated in Proposition 4.3. That is, for every possible $A \subset \{\mathcal{T}_1, \ldots, \mathcal{T}_N\}$, we first solve the optimization problem defined by Equation (4.5), and store the optimal solution $(\mathbf{c}^A, \mathbf{d}^A)$; once $(\mathbf{c}^A, \mathbf{d}^A)$ is found for every $A \subset \{\mathcal{T}_1, \ldots, \mathcal{T}_N\}$, we can solve the LP defined by Equations (4.6-4.8). In our computational experience, when the number of products $N \leq 20$, we can find $(\mathbf{c}^A, \mathbf{d}^A)$ for all $A \subset \{\mathcal{T}_1, \ldots, \mathcal{T}_N\}$ and solve the above formulation very quickly. However, for larger N, solving Problem-WM by generating all 2^N constraints is not efficient (see numerical experiments in Section 4.5).

In the proof of Proposition 4.3, we also derived a simple combinatorial expression for the shortage function. Next, we formally state the expression as a corollary. This expression would prove to be useful in our characterization of the optimal level of inventory in Section 4.4.1.

Corollary 4.4. *For any nonengative vector* $\mathbf{c}$,

$$\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d}) = \max_{A \subset \{\mathcal{T}_1, \ldots, \mathcal{T}_N\}} \sum_{\mathcal{T}_j \in A} (d_j - s_j) - \sum_{\mathcal{S}_i \in P_{\mathcal{F}}(A)} c_i. \tag{4.9}$$

4.3.2 Delayed Constraint Generation Algorithm

In this subsection, we present a delayed constraint generation algorithm that is significantly more efficient than enumerating all 2^N constraints defined by Equation (4.7). Roughly speaking, the algorithm starts with a LP formulation that is similar to the one defined by Equations (4.6-4.8), but only a subset of constraints. At each iteration, the algorithm adds a separating hyperplane (in the form of Equation (4.7)) if the currently inventory allocation solution $\mathbf{s}$ is infeasible. Algorithm 9 describes the outline of the constraint generation

algorithm.

Algorithm 9 Constraint Generation Algorithm

Let $\mathcal{A} = \varnothing$. Repeat the following steps:

1. Solve the relaxation of Problem-WM using constraints in $\mathcal{A}$:

$$\begin{aligned} \min \quad & \sum_{j=1}^{N} s_j \\ \text{s.t.} \quad & \sum_{\mathcal{T}_j \in A} (d_j^A - s_j) - \sum_{\mathcal{S}_i \in P_{\mathcal{F}}(A)} c_i^A \leq \delta, \quad \forall A \subset \mathcal{A}, \\ & s_j \geq 0, \quad \forall 1 \leq j \leq N. \end{aligned}$$

   Let $\mathbf{s}^*$ be the optimal solution.

2. Solve the optimization problem

$$\max_{\substack{A \subseteq \{\mathcal{T}_1, \ldots, \mathcal{T}_N\}, \\ (\mathbf{c},\mathbf{d}) \in \mathcal{U}}} \sum_{\mathcal{T}_j \in A} (d_j - s_j^*) - \sum_{\mathcal{S}_i \in P_{\mathcal{F}}(A)} c_i. \tag{4.10}$$

   Let its optimal value be $\delta(\mathcal{A})$, and its optimal solution be A^*, $(\mathbf{c}^{A^*}, \mathbf{d}^{A^*})$.

3. Check optimality: if $\delta(\mathcal{A}) \leq \delta$, stop; otherwise, let $\mathcal{A} \leftarrow \mathcal{A} \cup \{A^*\}$, and go back to Step 1. Note that $(\mathbf{c}^{A^*}, \mathbf{d}^{A^*})$ is the optimal solution for

$$\max_{(\mathbf{c},\mathbf{d}) \in \mathcal{U}} \sum_{\mathcal{T}_j \in A^*} (d_j - s_j^*) - \sum_{\mathcal{S}_i \in P_{\mathcal{F}}(A^*)} c_i.$$

It is important to note that Algorithm 9 applies the constraint generation on the reformulation presented in Proposition 4.3. Because the formulation in Proposition 4.3 has 2^N linear inequalities, it guarantees that our algorithm will terminate with at most that many iterations. Therefore, the run time of our algorithm depends mostly on the number of products, and independent of the number of extreme points in the uncertainty sets. This is in contrast with the standard cutting plane algorithms proposed for two-stage robust optimization problems Thiele et al. (2010), Zeng and Zhao (2013), where the number of constraints generated also depends on the number of extreme points in the uncertainty sets.

The main difficulty in the implementation of Algorithm 9 is to solve the optimization problem defined by Equation (4.10) in Step 2. While it is known that the optimization problem defined by Equation (4.10) is NP-hard Simchi-Levi and Wei (2014), we present a the hardness result in a stronger sense: we show that determining whether $\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d}) \leq \delta$

for all $(\mathbf{c}, \mathbf{d}) \in \mathcal{U}$ is NP-hard, *for any* process flexibility design $\mathcal{F}$.

Proposition 4.5. *For any fixed flexibility design $\mathcal{F}$, determining whether $\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d}) \leq \delta$ for all $(\mathbf{c}, \mathbf{d}) \in \mathcal{U}$ is NP-hard, for general shortage allowance δ and general nonnegative polyhedral uncertainty set $\mathcal{U}$.*

Proposition 4.5 provides us with the following important insight in designing algorithms for solving Problem-WM: even under a specific class of flexibility designs, one cannot derive polynomial-time algorithms unless $P = NP$, or have further assumptions on the uncertainty sets. Therefore, our approach for solving general Problem-WM is to convert it into a mixed integer linear program (MILP), and take the advantage of the powerful state-of-art MILP solvers. While our algorithm does not guarantee polynomial running time, in our computational experiments, we were able to quickly solve instances where the number of products in the system is less than one hundred.

To solve the optimization problem defined by Equation (4.10), we first convert it into the following bilinear program:

$$\begin{aligned} \max \quad & \sum_{j=1}^{N} (d_j - s_j^*) q_j - \sum_{i=1}^{M} c_i p_i \\ \text{s.t.} \quad & (\mathbf{c}, \mathbf{d}) \in \mathcal{U}, (\mathbf{p}, \mathbf{q}) \in V(\mathcal{F}). \end{aligned}$$

The bilinear program above is equivalent to the optimization problem defined by Equation (4.10). In particular, if $(\mathbf{c}^*, \mathbf{d}^*)$, $(\mathbf{p}^*, \mathbf{q}^*)$ is an extreme point optimal solution of the bilinear program, then let $A^* = \{\mathcal{T}_i | q_j^* = 1\}$ and we have that A^*, $(\mathbf{c}^*, \mathbf{d}^*)$ is an optimal solution for Equation (4.10).

Next, we apply the classical Mccormick envelopes McCormick (1976) and reformulate the bilinear program as the following mixed integer linear program (MILP) as follows.

$$\begin{aligned} \delta(\mathcal{A}) = \max \quad & \sum_{j=1}^{N} y_j - \sum_{j=1}^{N} s_j^* q_j - \sum_{i=1}^{M} z_i \\ \text{s.t.} \quad & p_i \geq q_j, \quad \forall (\mathcal{S}_i, \mathcal{T}_j) \in \mathcal{F}, \\ & y_j \leq d_j, y_j \leq d_j^{\max} q_j, \quad \forall j = 1, \ldots, N, \\ & z_i \geq 0, z_i \geq c_i - c_i^{\max} + c_i^{\max} p_i, \quad \forall i = 1, \ldots, M, \\ & p_i, q_j \in \{0, 1\}, \forall 1 \leq i \leq M, \quad \forall 1 \leq j \leq N, \end{aligned}$$

$$(\mathbf{c}, \mathbf{d}) \in \mathcal{U}.$$

Here, constants $d_j^{\max}$ and $c_i^{\max}$ are upper bounds of d_j and c_i for $(\mathbf{c}, \mathbf{d}) \in \mathcal{U}$.

We believe that there are significant advantages in solving the MILP reformulation above compared to the original bilinear programs. Over the last ten years, there has been vast improvement in solving mixing integer linear programs, and the MILP formulation allows us to take advantage of the progress made in state-of-the-art MILP solvers. Moreover, at each iteration, the MILP algorithm can use the MILP solutions from previous iterations as a warm start, therefore significantly improve the running time. In our computational experiments, we used Gurobi Mixed Integer Solver and were able to consistently solve the MILP formulations with up to one hundred products in an average of several seconds (see Section 4.5.1).

In our actual implementation, we do not require the solver to always find the optimal solution in Step 2 of Algorithm 9. Instead, we stop when the solver finds a feasible solution $(\mathbf{c}, \mathbf{d})$, $(\mathbf{p}, \mathbf{q})$ where its objective is strictly larger than δ. Stopping the MILP solver early saves time while still generating a valid cut in each iteration. After a feasible solution is found, we then apply the alternating direction method of multipliers (ADMM), i.e., fixing $(\mathbf{p}, \mathbf{q})$ or $(\mathbf{c}, \mathbf{d})$ while optimizing over the other, to strengthen our cut. Because optimizing $\sum_{j=1}^{N}(d_j - s_j^*)q_j - \sum_{i=1}^{M} c_i p_i$ over $(\mathbf{c}, \mathbf{d}) \in \mathcal{U}$ with fixed $(\mathbf{p}, \mathbf{q})$ (or over $(\mathbf{p}, \mathbf{q}) \in \mathcal{U}$ with fixed $(\mathbf{c}, \mathbf{d}) \in V(\mathcal{F})$) is simply solving a linear optimization, the ADMM can be performed very quickly. Note that ADMM is not guaranteed to converge to the global optimal solution. After the ADMM converges to $(\mathbf{p}, \mathbf{q})$ and $(\mathbf{c}, \mathbf{d})$, we let $A^* = \{\mathcal{T}_j | q_j = 1\}$, and $(\mathbf{c}^{A^*}, \mathbf{d}^{A^*}) = (\mathbf{c}, \mathbf{d})$.

4.4 Analysis for K-Chain Designs

To understand inventory allocation under different degrees of flexibility, in this section, we focus on the setting where there are N plants and N products ($M = N$). To study different degrees of process flexibility, we consider a canonical family of process flexibility designs known as the *K-chain*, where K is a strictly positive integer. A *K-chain* design is where plant 1 produces product 1 to product K, plant 2 produces product 2 to product $K + 1$, and in general plant i produces products $i, i + 1, \ldots, i + K - 1$. Because of its simple and

symmetric structure, this canonical family of flexibility designs have been analyzed in various studies Hopp et al. (2004), Wang and Zhang (2015).

Throughout the section, we study uncertainty sets where the demand and capacity uncertainties are separable. Namely, we have $\mathcal{U} = \mathcal{U}_c \times \mathcal{U}_d$, where $\mathcal{U}_c \subseteq \mathbb{R}^N$ is the plants capacity uncerainty set, and $\mathcal{U}_d \subseteq \mathbb{R}^N$ is the products demand uncerainty set. In Section 4.4.1, we will assume that the system is symmetric. That is, for any demand instance $\mathbf{d} \in \mathcal{U}_d$, every permutation of $\mathbf{d}$ is also in $\mathcal{U}_d$. Respectively, for any capacity instance $\mathbf{c} \in \mathcal{U}_c$, every permutation of $\mathbf{c}$ is in $\mathcal{U}_c$. For a fixed uncertainty set $\mathcal{U} = \mathcal{U}_c \times \mathcal{U}_d$, we define the quantitative measures $C^{\min}(t)$ and $D^{\max}(t)$ as follows.

Definition 4.6. *Define* $C^{\min}(t) := \min_{\mathbf{c}\in\mathcal{U}_c} \sum_{i=1}^{t} c_i$, *and* $D^{\max}(t) := \max_{\mathbf{d}\in\mathcal{U}_d} \sum_{j=1}^{t} d_j$.

Note that because both $\mathcal{U}_c$ and $\mathcal{U}_d$ only contain non-negative vectors, both $C^{\min}(t)$ and $D^{\max}(t)$ are non-decreasing with t.

4.4.1 Total Inventory Required by K-chain

We start with a lemma that helps us to derive the structure of the optimal inventory allocation in K-chain designs.

Lemma 4.7. *Suppose for any inventory allocation* $\mathbf{s} = (s_1, s_2, \ldots, s_N)$, *the rearranged allocation* $\phi(\mathbf{s}) = (s_2, s_3 \ldots, s_N, s_1)$ *(stock* s_2 *units of product 1, etc.) is feasible for the Problem-WM, then the optimal inventory allocation can be achieved by allocating inventory equally across all products.*

The symmetry of K-chain designs certainly satisfies the condition in Lemma 4.7, so we can assume the optimal inventory allocation has the property where $s_j = s$ for all product j. This property turns out to be very important for our analysis for two reasons. First, by restricting ourselves to study inventory allocations satisfying $s_j = s$ for all product j, we greatly simplify the original optimization problem to an optimization problem with just a single variable. Second, if $s_j = s$ for all product j, then both the inventories and the uncertainties are symmetric and this allows us to derive a simple condition for checking the feasibility of $\mathbf{s}$ in Problem-WM.

Next, we show that for integers $1 \leq K \leq N$, if $\mathcal{F}$ is a K-chain, then we can obtain a closed-form analytical expression for the optimal inventory level.

Proposition 4.8. *Let $\mathcal{F}$ be a K-chain, for some integer K between 1 and N. Suppose that $\mathcal{U}$ is symmetric and $\mathcal{U} = \mathcal{U}_c \times \mathcal{U}_d$. Then* $\mathbf{s}$ *is an optimal inventory allocation if $s_j = s^*$ for all $1 \leq j \leq N$ where*

$$s^* = \max\{\max_{1\leq t\leq N-K} \frac{D^{\max}(t) - C^{\min}(t+K-1) - \delta}{t}, \max_{N-k<t\leq N} \frac{D^{\max}(t) - C^{\min}(N) - \delta}{t}, 0\}. \tag{4.11}$$

Next, we will use Proposition 4.8 to study a set of specific uncertainty sets $\mathcal{U}_c$ and $\mathcal{U}_d$ to better understand the impact of K-chains on the optimal inventory level.

Example. In our example, we study a setting where the capacity of each plant is 1, and there can be at most one plant being completely disrupted. Therefore, we consider capacity uncertainty set

$$\mathcal{U}_c = \{\mathbf{c} | \sum_{i=1}^{N} c_i \geq N-1, 0 \leq c_i \leq 1, \forall 1 \leq i \leq N\}. \tag{4.12}$$

Also, we consider demand uncertainty set

$$\mathcal{U}_d = \{\mathbf{d} | \sum_{j=1}^{N} d_j \leq N + 2\sqrt{N}\sigma, \sum_{j=1}^{N} |d_j - 1| \leq \frac{\sqrt{3}\sigma N}{2} + \sigma\sqrt{N}, |d_j - 1| \leq \sqrt{3}\sigma, \forall j = 1 \ldots N\}. \tag{4.13}$$

where σ is some parameter between 0 and $\frac{1}{\sqrt{3}}$. $\mathcal{U}_d$ can be interpreted as the products having i.i.d. uniform demand distributions with mean 1 and standard deviation σ. In Appendix C, we provide a more detailed justification for this class of uncertainty sets. We note that under our choice of $\mathcal{U}_d$, for each integer t from 1 to N, $D^{\max}(t)$ has an exact analytical expression (see Appendix C). Moreover, for each integer t from 1 to N, it is easy to check that $C^{\min}(t) = t - 1$.

We perform a numerical study for $N = 12$. By varying values of σ, we can analyze how different demand variabilities affect the inventory level required for the K-chain flexibility designs. Using Appendix C and Proposition 4.8, we get that for $K = 1$, the optimal inventory level for 1-chain is equal to

$$12 \cdot \max\{1 + \sqrt{3}\sigma - \delta, \frac{6\sqrt{3}\sigma - (\delta - 1)}{6}, \frac{6\sqrt{3}\sigma - 1 - \delta}{10}, \frac{4\sqrt{3}\sigma + 1 - \delta}{12}, 0\}; \tag{4.14}$$

for $2 \leq K \leq 6$, the optimal inventory level for K-chain is equal to

$$12 \cdot \max\{\frac{6\sqrt{3}\sigma - (K-2+\delta)}{6}, \frac{6\sqrt{3}\sigma - 1 - \delta}{10}, \frac{4\sqrt{3}\sigma + 1 - \delta}{12}, 0\}; \tag{4.15}$$

and finally, for $7 \leq K \leq 12$, the optimal inventory level for K-chain is the same as that of full flexibility, which can be expressed as

$$12 \cdot \max\{\frac{6\sqrt{3}\sigma - 1 - \delta}{10}, \frac{4\sqrt{3}\sigma + 1 - \delta}{12}, 0\}. \tag{4.16}$$

Therefore, we can compute the optimal inventory level for any K-chain with different δ and σ. The following is a summary of the insights obtained.

First, full flexibility is never required to achieve the optimal inventory level. Indeed, a 7-chain (in a 12-products system) would always require the same amount of inventory as full flexibility, see Equation (4.16). Moreover, in our numerical examples, the 5-chain design achieves the same inventory level as full flexibility for all of the parameters we studied.

Second, while changing a dedicated design to a 2-chain design provides a large benefit, the benefit achieved by changing from 2-chain to 3-chain is also significant. This holds even when the demand variability, σ, is not high. For example, under medium level of variability ($\sigma = 0.3$) and zero shortage allowance ($\delta = 0$), changing from dedicated to the 2-chain design reduces total inventory by 66%, while changing from the 2-chain to the 3-chain further reduces the total inventory (compared to the inventory of dedicated design) by another 11%. This observation is somewhat different from the insight described in the process flexibility literature when no plant disruption is present. In that case, it has been consistently observed that almost all of the benefits are obtained by changing from dedicated to the 2-chain design. One intuitive way to explain the improvement achieved by a 3-chain design over a 2-chain is that while the 2-chain design is very effective in satisfying uncertain demand, the disruption would break the chain. Therefore, a disruption would greatly reduces 2-chain's ability to satisfy uncertain demand and more flexibility is required.

4.4.2 Inventory Allocation Strategy

In this section, we study a setting where the demand variabilities are different across products. As we will illustrate through an example, a decision maker needs to take the flexibility design into account when making inventory decisions. In particular, different degrees of flex-

ibility not only requires different levels of inventory, but also completely different inventory patterns.

To understand how process flexibility affects inventory decisions when products face different demand variability, we consider a stylized example with $2N$ products where products $1, 3, \ldots, 2N-1$ face the same high level of demand variability; and products $2, 4, \ldots, 2N$ face the same low level of demand variability. Therefore, we refer to products with odd labels as *high variability products*, while products with even labels as *low variability products.*

We construct the demand uncertainty set where the products demand mimicking $2N$ independent uniform distributions with the same mean; odd products (i.e., products with odd index) have standard deviation σ_H, while even products have standard deviation σ_L. This demand uncertainty set is chosen based on the discussion in Appendix C. In particular, consider the following demand uncertainty set with $\sigma_H > \sigma_L$:

$$\mathcal{U}_d = \Big\{\mathbf{d} \mid \sum_{j=1}^{2N} d_j \le 2N + 2\sqrt{(\sigma_H^2+\sigma_L^2)N}, \sum_{j=1}^{2N} |d_j - 1| \le \sqrt{3}(\sigma_H+\sigma_L)N + \frac{\sqrt{(\sigma_H^2+\sigma_L^2)N}}{2},$$
$$|d_{2j-1} - 1| \le \sqrt{3}\sigma_H, |d_{2j} - 1| \le \sqrt{3}\sigma_L, \forall 1 \le j \le N\Big\}.$$

The first two constraints are motivated by central limit theorem (CLT) to bound total variability among all products. The last two constraints bound the variability for each individual product. The odd products have higher demand variability than the even products. Similar to Section 4.4.1, we consider the capacity uncertainty set

$$\mathcal{U}_c = \{\mathbf{c} | \sum_{j=1}^{2N} c_j \ge 2N - 1, 0 \le c_j \le 1, \forall 1 \le j \le 2N\}.$$

Finally, in this subsection, we only consider the case where $\delta = 0$.

Note that $\mathcal{U}_d$ in our example is completely symmetric for odd and even products, respectively. Therefore, there exists an optimal inventory decision such that every odd product has the same inventory level s_H, and every even product has the same inventory level s_L. Next, using a pair of lemmas, we show that the inventory stategies for the high and low variability products under different degrees of flexibility can be completely different.

Lemma 4.9. *When $\mathcal{F}$ is the dedicated or 2-chain design, there always exists an optimal inventory allocation with more inventory at the high variability products.*

Lemma 4.10. *Suppose* $N(2-\sqrt{3}\sigma_H-\sqrt{3}\sigma_L) \geq 2\sqrt{(\sigma_H^2+\sigma_L^2)N}+1$ *and* $\mathcal{F}$ *is the full flexibility design. Then, there always exists an optimal inventory allocation with more inventory at the low variability products.*

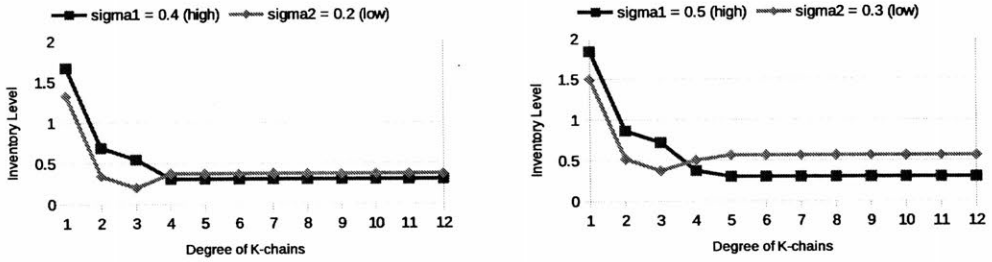
We note that the condition in Lemma 4.10, $N(2-\sqrt{3}\sigma_H-\sqrt{3}\sigma_L) \geq 2\sqrt{(\sigma_H^2+\sigma_L^2)N}+1$, is not restrictive. Because the left hand side of the condition scales linearly with N while the right hand side of the condition scales linearly with $\sqrt{N}$, for large N, the condition would hold for a wide range of (σ_H, σ_L). As an example, if $\sigma_H = 0.5$ and $\sigma_L = 0.25$, the condition in Lemma 4.10 would hold for any $N \geq 5$.

Lemma 4.9 and Lemma 4.10 illustrate an interesting effect: As one would expect, the firm holds more inventory for high variability demand when the degree of flexibility is small. This is similar to the classical newsvendor model, where more inventory is needed to achieve the same fill rate as the demand variability increases. But surprisingly, when the firm has high degree of flexibility (i.e. full flexibility), it is optimal to hold more inventory for products facing lower variability, and to hold less inventory for products with higher variability. We refer to such phenomenon as the *flipping effect*, since the inventory levels of two kinds of products flipped as degree of flexibility increases.

The flipping effect can be explained by relating our analysis to *Push and Pull* strategies. In a Pull strategy, the firm produces to order by applying capacity. In a Push strategy, the firm produces to stock, and then satisfies demand from inventory. When there is high degree of flexibility, high variability products are typically served using Pull (capacity) while low variability products are served using Push (inventory), so most inventory is allocated to the low variability products while less inventory is allocated to the high variability products. When there is low degree of flexibility, capacity is not enough to match supply with demand for high variability products, so more inventory is allocated to the high variability products.

To better understand how inventory allocations change with changes in the degree of flexibility, we compute the optimal inventory level for a specific system consisted of 12 plants, 6 high variability products and 6 low demand variability products. The computation is done using the computational algorithm proposed in Section 4.3. We test two pairs of parameters: $(\sigma_H = 0.4, \sigma_L = 0.2), (\sigma_H = 0.5, \sigma_L = 0.25)$. The finding of our results is summarized by Figure 4-2. In our numerical example, we assumed, without loss of optimality, that every odd product has the same inventory level s_1, and every even product has the same inventory level s_2. Indeed, Figure 4-2 illustrates the flipping effect as we increase the degree of flexibility.

Figure 4-2: Inventory under Asymmetric Demand in K-chains



The flipping effect also occurs in probabilistic settings where probability distributions of plant disruption and product demands are specified. We show similar results using numerical experiments in Section 4.5.2.

Our findings from Lemma 4.9, Lemma 4.10, and Figure 4-2 illustrate the importance of incorporating flexibility designs into inventory decisions. As a manufacturing firm changes its production system by adding flexibility, it is important to note that not only its optimal inventory levels would change, but the percentage mix of different inventories may change as well. Therefore, when a manufacturing system has process flexibility, it is not clear where inventories should be allocated. In these cases, the computational algorithm proposed in Section 4.3 provides a tool to effectively analyze optimal inventory positions under different flexibility designs.

4.5 Computational Experiments

4.5.1 Balanced System Example

In this part, we study a balanced flexibility system with equal number of plants and products (i.e. $M = N$). The computational result illustrates the efficiency of the constraint generation algorithm.

We assume that each plant has one unit of capacity, and at most 5 percent of the total capacity is lost due to disruption. The demand uncertainty set is given by Equation (4.13) with $\sigma = 0.3$, and the demand shortage allowance is set to $\delta = 0.02N$ (analogous to 2 percent of total mean demand). Additionally, we randomly generate the flexibility design so that each plant can produce at least one and up to five products.

We solve the example by the constraint generation algorithm (Algorithm 9). The compu-

tations were carried out using the Gurobi 6.0.3 solver on a 1.80GHz Intel CPU. In Table 4.1, we list the CPU time for different sizes of the system. For a system of 100 plants and 100 products, Algorithm 9 found the optimal inventory allocation within 400 seconds. The table also lists the number of constraints generated by Algorithm 9 before the optimal solution is found. The result shows that Algorithm 9 typically generates only a small fraction of the constraints, even though the full LP formulation (Equations 4.6–4.8) has over 10^{30} constraints. As a comparison, Table 4.1 shows the CPU time for solving the full LP formulation. This method is fast when the number of products is 10 because it avoids solving MILPs, but is significantly slower than the constraint generation algorithm for 20 products. For systems of size 50 and 100, there is not enough computer memory to generate the full LP.

Table 4.1: Constraint Generation Algorithm for the Balanced System

Number of Products (N)	10	20	50	100
CPU Time (sec)	0.47	2.25	62.68	380.9
# of Constraints Generated	21	70	327	1701
Total # of Constraints (2^N)	1024	$\sim 10^6$	$\sim 10^{15}$	$\sim 10^{30}$
CPU Time of full LP (sec)	0.06	78.77	—	—

4.5.2 Unbalanced System Example

We then consider an unbalanced system where the number of plants and products are unequal ($M \neq N$). This example is based on a real-world inspired data set of 16 vehicle models and 8 assembly plants at General Motors presented in Jordan and Graves (1995). We use the same plant capacities and mean product demands as in their paper, shown on the left chart of Figure 4-3. We assume that Product A to F (compact cars) have high variability demands, with a coefficient of variation equal to $\sigma_H = 0.4$; Product G to P (full-sized and luxury cars) have low variability demand with a coefficient of variation equal to $\sigma_L = 0.2$. The demand uncertainty set is then created based on the example in Section 4.4.2. The demand shortage allowance is set to 5 percent of the total mean demand. Furthermore, we assume that any one of the plants can be disrupted in the worst case.

Based on the "chaining" principle, Jordan and Graves (1995) proposed an ad hoc method to add process flexibility to the base design. They first cluster the plants and products into six groups, then add six links to connect six groups and create a "2-chain" design, see the right chart of Figure 4-3. Following their method, we also create "3-chain" and "4-chain"

Figure 4-3: Flexibility Design in the Asymmetric Example

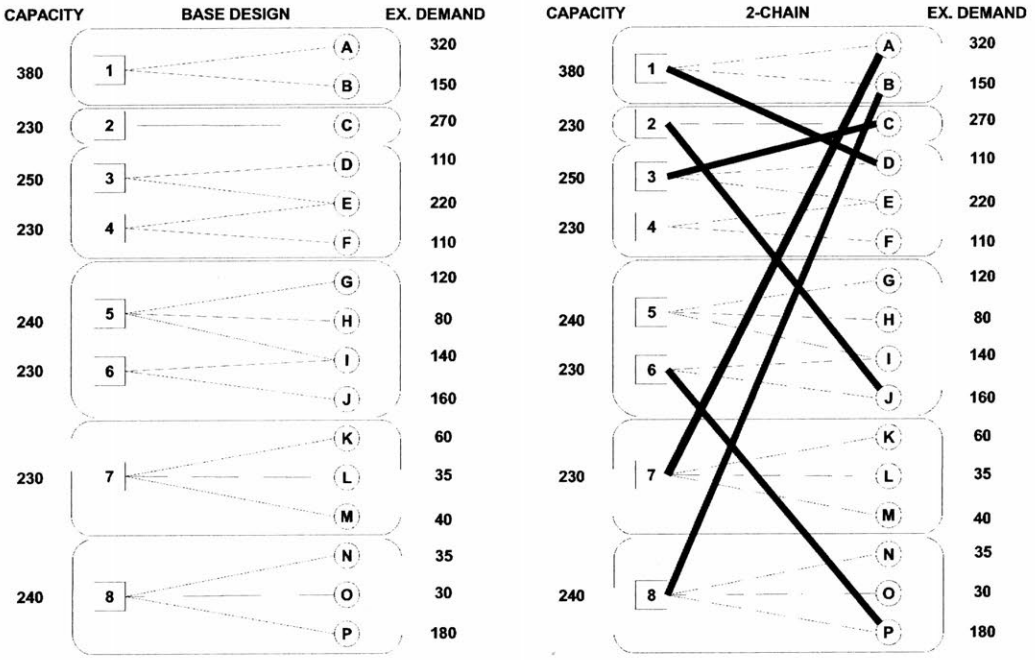


Figure 4-4: Flexibility Design in the Asymmetric Example (continued)

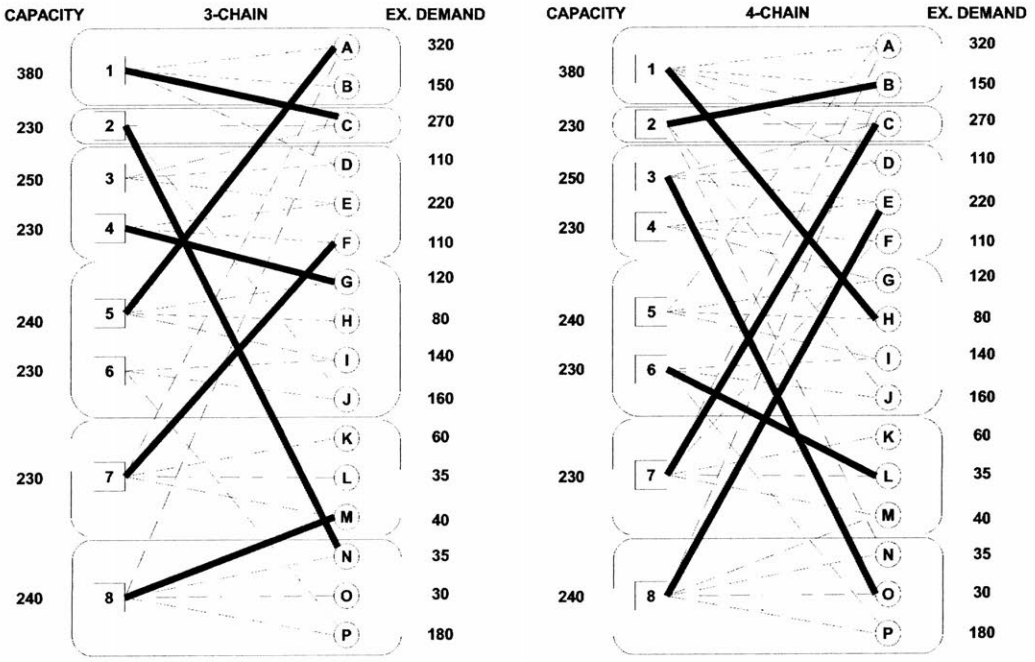


Table 4.2: Inventory Allocation for Unbalanced System

Flexibility Designs	Base Design	2-chain	3-chain	4-chain	Full Flexibility
Total Inventory	1790.4	1072.4	785.5	670.7	667.1
% for High Var Products	71.9%	69.6%	71.4%	51.5%	31.4%
# of Constrs Generated	24	34	40	30	18

designs by adding six links at a time, see Figure 4-4. Of course, there are many ways to add six links to the initial design. Our objective is not to find the "optimal" designs, but to create a family of designs with increasing degree of flexibility, and analyze how inventory decisions change within this family.

We apply Algorithm 9 to find the optimal inventory allocation for the four flexibility designs. We note that all the problems were solved within a second and only a few constraints were generated. Table 4.2 shows the total inventory required for different designs, as well as the percentage of inventory allocated to high variability products (Product A to F). The observation is consistent with the findings in Section 4.4 for the symmetric K-chain systems. While changing from a base design to a 2-chain design reduces total inventory, there is also significant inventory reduction achieved by changing from a 2-chain design to higher degrees of flexibility. Moreover, as the flipping effect predicts, when the degree of flexibility increases, less inventory is allocated to high variability products.

We note that the findings above are not unique to the worst case model, as similar results also hold for the stochastic model. We consider a case where product demands are independent and normally distributed truncated at two standard deviations. We assume that Product A to F (compact cars) have high variability demands, with a coefficient of variation equal to 0.5 prior to truncation; Product G to P (full-sized and luxury cars) have low variability demand, with a coefficient of variation equal to 0.3 prior to truncation. Each plant is assumed to be disrupted independently with probability 0.1.

The constraint under our stochastic model is to ensure that the expected fill rate (i.e., expected demand satisfied) is at least 95%. We note that while the constraint in our robust optimization model is closer to Type 1 service level (i.e., chance constraints), that Type 1 service level is non-convex and therefore poses computational difficulties (see Appendix C). For this reason, we focus instead on a stochastic model where the objective is to find inventory allocation such that the expected fill rate is at least 95%. The stochastic model is meant to verify the robustness of our insights in Section 4.4.

We note that the constraint of ensuring the expected fill rate to be at least 95% is significantly weaker than the worst-case constraint we studied previously. Therefore, the inventory level for the stochastic model is in general much lower than that of the robust optimization model. Table 4.3 shows the total inventory level and the percent of the inventory allocated to high variability products. As the degree of flexibility increases, the total inventory level decreases, while less inventory is allocated to high variability products.

Table 4.3: Inventory Allocation for Unbalanced System (Stochastic Model)

Flexibility Designs	Base Design	2-chain	3-chain	4-chain	Full Flexibility
Total Inventory	955.4	430.8	312.2	296.6	290.3
% High Var Products	72.54%	47.92%	33.44%	17.55%	4.62%

4.6 Extensions

4.6.1 Different Holding Costs and Lost Sales Costs

We consider an extension of the model in Section 4.2 where products have different holding cost and/or lost sales cost. Suppose product j has unit inventory holding cost h_j and unit lost sale cost b_j for all $j = 1, \ldots, N$. The inventory allocation problem can be formulated as

$$\begin{aligned} \min \quad & \sum_{j=1}^{N} h_j s_j \\ \text{s.t.} \quad & \Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d}) \leq \delta, \quad \forall (\mathbf{c}, \mathbf{d}) \in \mathcal{U}, \\ & s_j \geq 0, \forall 1 \leq j \leq N, \end{aligned} \tag{4.17}$$

where

$$\begin{aligned} \Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d}) = \min \quad & \sum_{j=1}^{N} b_j l_j \\ \text{s.t.} \quad & \sum_{i:(\mathcal{S}_i, \mathcal{T}_j) \in \mathcal{F}} x_{ij} + l_j \geq d_j - s_j, \quad \forall 1 \leq j \leq N, \\ & \sum_{j:(\mathcal{S}_i, \mathcal{T}_j) \in \mathcal{F}} x_{ij} \leq c_i, \quad \forall 1 \leq i \leq M, \\ & l_j, x_{ij} \geq 0. \end{aligned}$$

Problem (4.17) can be solved by a delayed constraint generation algorithm that is similar to Algorithm 9. For fixed inventory allocation $\mathbf{s}$, by taking the dual of the shortage function $\Pi(\cdot)$, a violating constraint of (4.17) is generated if the following bilinear program has an optimal value greater than shortage allowance δ:

$$\begin{aligned} \max_{(\mathbf{c},\mathbf{d})\in\mathcal{U}} \quad & \sum_{j=1}^{N}(d_j - s_j)q_j - \sum_{i=1}^{M} c_i p_i \\ \text{s.t.} \quad & q_j \le p_i, \quad \forall\,(\mathcal{S}_i, \mathcal{T}_j) \in \mathcal{F}, \\ & q_j \le b_j, \quad \forall\, 1 \le j \le N, \\ & p_i,\, q_j \ge 0, \quad \forall\, 1 \le i \le M, 1 \le j \le N. \end{aligned} \tag{4.18}$$

By Lemma 4.2, the constraints in (4.18) are totally unimodular, so the optimal solution of $\mathbf{p}$ and $\mathbf{q}$ is integral if b_j is an integer for all $j = 1, \ldots, N$.[1] Therefore, each bilinear term in (4.18) involves the product of a nonnegative integer variable and a nonnegative continuous value, which also admits a MILP reformulation based on McCormick envelops (see Gupte et al. (2013) for an example).

4.6.2 Time-to-Survive Model

In this subsection, we introduce a metric called Time-to-Survive (TTS), to measure the resilience of supply chain when the duration of the disruption is known. TTS of a given disruption scenario is defined as the time that a firm can maintain its customer service level under the disruption scenario. The TTS metric has been already adopted by the Ford Motor Company to assess risk exposure in its complex supply chain and evaluate Ford's risk mitigation strategies Simchi-Levi et al. (2014, 2015).

The definition of TTS is motivated by the concept of Time-to-Recover (TTR), which is the time for a facility to return to full capacity after a disruption. TTR is widely used to evaluate supply chain risk Hopp et al. (2012). If TTS is greater than TTR, then the disruption is not going to affect the firm's service level. On the other hand, when TTS for a specific facility is shorter than TTR, then the disruption will have an impact on service. Thus, an important challenge in supply chain risk management is to allocate inventory to different products to increase the supply chain's time to survive, and attempt to ensure that

[1] In general, if b_j's are rational numbers, they can be transformed into integer values by multiplying a common factor.

the TTS is always greater than TTR.

For this purpose, we define the *supply chain TTS* as the minimum (worst) TTS among all of the potential scenarios. The longer the supply chain TTS, the more robust the supply chain is. Below we show that the problem of allocating inventory under inventory budget to maximize supply chain TTS can be reduced to a special case of the model in Section 4.2.

Suppose that plant i has capacity c_i $(i = 1, \ldots, M)$, product j has a demand rate d_j per unit time $(j = 1, \ldots, N)$. Let r_j be the amount of inventory allocated to product j, and assume that the sum of inventory among all products cannot exceed a given budget R. Again, we use $\mathcal{U} \subset \mathbb{R}_+^M \times \mathbb{R}_+^N$ to denote the uncertainty set that models the uncertainty (and therefore all disruption scenarios) of plant capacity and product demand. We then formulate the problem of maximizing supply chain TTS as the following nonlinear program:

$$
\begin{aligned}
TTS = \max_{r_j, x_{ij}^{(\mathbf{c},\mathbf{d})}} \quad & t \\
\text{subject to} \quad & t \le \frac{r_j}{d_j - \sum_{i:(i,j)\in\mathcal{F}} x_{ij}^{(\mathbf{c},\mathbf{d})}}, \quad \forall 1 \le j \le N,\ (\mathbf{c},\mathbf{d}) \in \mathcal{U}, \\
& \sum_{j:(i,j)\in\mathcal{F}} x_{ij}^{(\mathbf{c},\mathbf{d})} \le c_i, \quad \forall 1 \le i \le M,\ (\mathbf{c},\mathbf{d}) \in \mathcal{U}, \\
& \sum_{j=1}^{N} r_j \le R, \\
& r_j \ge 0, x_{ij}^{(\mathbf{c},\mathbf{d})} \ge 0, \quad \forall 1 \le i \le M, 1 \le j \le N,\ (\mathbf{c},\mathbf{d}) \in \mathcal{U}.
\end{aligned}
$$

In this formulation, $x_{ij}^{(\mathbf{c},\mathbf{d})}$ denotes the production of product j by plant i if realized capacity and demand is $(\mathbf{c},\mathbf{d}) \in \mathcal{U}$. The first constraint follows the definition of TTS: In any scenario, the supply chain must survive the disruption by continuously supplying all products for at least t time units. The second constraint requires that production does not exceed plant capacity. The last constraint specifies that total inventory cannot exceed a given budget R.

Let $s_j = r_j/t$, we can reformulate the TTS model as

$$
\begin{aligned}
\min_{s_j, x_{ij}^{(\mathbf{c},\mathbf{d})} \ge 0} \quad & \sum_{j=1}^{M} s_j \\
\text{subject to} \quad & d_j - \sum_{i:(i,j)\in\mathcal{F}} x_{ij}^{(\mathbf{c},\mathbf{d})} \le s_j, \quad \forall 1 \le j \le N,\ (\mathbf{c},\mathbf{d}) \in \mathcal{U},
\end{aligned}
$$

$$\sum_{j:(i,j)\in\mathcal{F}} x_{ij}^{(\mathbf{c},\mathbf{d})} \leq c_i, \quad \forall 1 \leq i \leq M,\ (\mathbf{c},\mathbf{d}) \in \mathcal{U},$$

$$s_j \geq 0, x_{ij}^{(\mathbf{c},\mathbf{d})} \geq 0, \quad \forall 1 \leq i \leq M, 1 \leq j \leq N,\ (\mathbf{c},\mathbf{d}) \in \mathcal{U}.$$

This is a special case of the model in Section 4.2 where the demand shortage allowance is equal to $\delta = 0$. Therefore, the problem of maximizing the supply chain TTS can be solved using the algorithm we proposed in Section 4.3.

Chapter 5

Concluding Remarks

In the previous chapters, we proposed several methods that integrate online learning and decision making for operations management problems. In this chapter, we briefly summarize the results, and discuss some general challenges to apply dynamic learning and optimization.

5.1 Summary and Future Directions

In Chapter 2, we focus on the finite-horizon network revenue management problem in which an online retailer aims to maximize revenue from multiple products with limited inventory. As common in practice, the retailer does not know the expected demand at each price. The main contribution of our work is the development of a general algorithmic framework which learns mean demand and dynamically adjusts prices to maximize revenue. Our algorithm builds upon the Thompson sampling algorithm used for the multi-armed bandit problem by incorporating inventory constraints into the pricing strategy. Our algorithm proves to have both strong theoretical guarantees as well as promising numerical performance when compared to other algorithms developed for the same setting. We also show that the algorithm can be extended to the setting with contextual information.

Two key features of Thompson sampling are modelling flexibility and computation efficiency — the "sampling" step in the algorithm essentially decomposes learning and optimization in a computational way. This makes it possible to extend the idea of our algorithm to a broad range of problems. In fact, to the best of our knowledge, Chapter 2 is the first attempt to apply Thompson sampling for a *constrained* optimization problem, while previous works on Thompson sampling have generally focused on *unconstrained* optimization

problems, or special cases where constraints are defined in each individual period. Given that most real world systems are modelled using constrained optimization, we believe that Thompson sampling will find applications in other operations management problems.

In Chapter 3, we consider a dynamic pricing problem where the latent demand model is unknown but belongs to a finite set of demand functions. The seller faces a constraint that price can be changed at most m times. We propose a pricing policy that incurs a regret of $O(\log^{(m)} T)$, where T is the length of the sales horizon and $\log^{(m)} T$ is m iterations of logarithm. In addition, we show that this regret bound is the best possible up to a constant factor. We then implement the pricing algorithm at Groupon, a website that sells deals from local merchants. We design a process to generate linear demand function set from historical data, and use it as an input to our pricing algorithm. The algorithm incorporates Groupon's business rules, which allow at most one price decrease per deal. The field experiment shows that the algorithm has significantly improved both daily revenue and bookings per deal.

For the implementation at Groupon, we treat the initial price of each deal as input. This is because in the current practice, Groupon and local merchants need to negotiate these prices. A potential future direction is to optimize initial price given a description of deal features. This leads to an interesting problem of dynamic pricing for heterogeneous problems.

In Chapter 4, we consider a firm using process flexibility and inventory to mitigate risk from plant disruptions, with particular focus on the interplay between the two strategies. This interplay is modeled as a two-stage robust optimization problem. In the first stage, given the firm's process flexibility design, the firm optimizes its inventory levels for all products in order to guarantee that demand shortage never exceeds some constant. Uncertainties from plant disruptions and product demands are modeled using uncertainty sets. In the second stage, after disruption occurs, demands are realized, and the firm uses both inventory and process flexibility to minimize demand shortage. In particular, the firm may leverage its process flexibility, and reallocate production capacities to better match supply with demand. Our paper provides an effective delayed constraint generation algorithm to solve the robust optimization model. In our computational experience, the algorithm is capable of solving instances with up to 100 products. Moreover, we derive analytical results for a canonical family of flexibility designs known as K-chain designs, and find the following managerial insights. First, while a 2-chain design is not as effective as K-chain designs for $K > 2$,

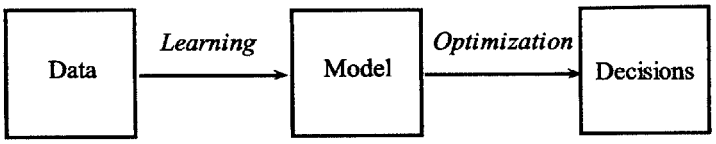


Figure 5-1: An Illustration of Open-Loop Decision Processes

full flexibility is often not needed. Second, when the firm has a high degree of flexibility, more inventory is allocated to products with low demand variability, and less inventory is allocated to products with high demand variability. We refer to this phenomenon as the "flipping effect."

There are multiple ways to extend the two-stage robust optimization model in Chapter 4 to other application domains. Since our algorithm exploits the network flow structure of the second stage problem, one potential application area is in transportation and logistics, where many problems possesses some network flow structures.

5.2 Overview of Dynamic Learning and Optimization

In the introduction of Chapter 1, we contrast two different approaches to deal with incomplete information in operations management. The classical approach, assumed by most operations management literature, applies the estimation process and decision making process sequentially. This can be thought as an "open-loop" process, illustrated in Figure 5-1. Although this approach is relatively simple, it ignores feedback from the real-world systems once decisions are made, so it does not have the ability to adjust the model based on feedback.

The methods proposed in Chapter 2–4 are of a different kind. We can think of them as "closed-loop" processes (Figure 5-2). In a closed-loop process, the decision maker continuously monitors feedback from previous decisions to refine her model.

The closed-loop decision processes are formally known as *Reinforcement Learning* and have long been studied by researchers in control theory, artificial intelligence and operations research (Sutton and Barto 1998). Reinforcement learning has found many successful applications in game playing, robotics and other engineering domains. Recent trend shows that reinforcement learning is also gaining momentum in operations management.

However, compared to other fields, there are several obstacles to adopt reinforcement

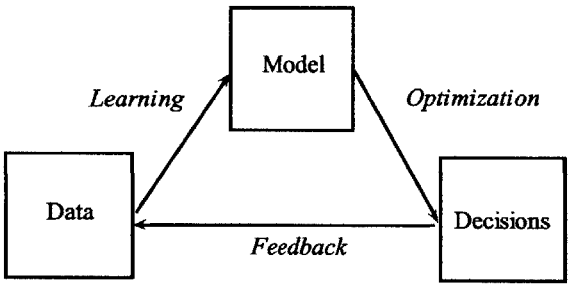


Figure 5-2: An Illustration of Closed-Loop Decision Processes

learning, or "closed-loop" decision processes, in operations management. First, most real world managerial problems are extremely complex. For example, the revenue management problem involves understanding consumer behaviour, defining seller's objectives and constraints, and perhaps coordinating with suppliers and various marketing channels. It is unlikely that there is a single model that correctly represents all these elements. Therefore, in addition to fine-tuning the model parameters, it may be necessary to adjust the model itself — a subject that we have not addressed in this thesis but may be of significant importance.

Second, in operations management, obtaining feedback from decisions can be costly. Suboptimal decisions may incur huge losses in firm's revenue and market share. In applications such as gaming and robotics, feedback can be collected relatively cheaply through simulation or lab experiments, whereas often the only way to get feedback in operations management is by interacting with the real world. Therefore, it is important to design learning and optimization methods that include the potential cost of experiments.

Appendix A

Technical Results for Chapter 2

A.1 Proofs of Theorem 2.1

Theorem. Suppose that the optimal solution(s) of OPT_{UB} are non-degenerate. If the demand distribution has bounded support, the regret of TS-fixed is bounded by

$$Regret(T) \leq O(\sqrt{T} \log T \log \log T).$$

A.1.1 Preliminaries

To prepare for the proof of the main result, we write OPT_{UB} in the standard form. For each arm $k = 1, \ldots, K$, we denote $r_k = \sum_{i=1}^{N} p_{ik} d_i(p_k)$ as the mean revenue and $b_{jk} = \sum_{i=1}^{N} a_{ij} d_i(p_k)$ as the expected consumption of resource j. We also add $M+1$ slack variables. Then we have

$$\begin{aligned} f^* = \max_{x,s} & \sum_{k=1}^{K} r_k x_k \\ \text{subject to} & \sum_{k=1}^{K} b_{jk} x_k + s_j = c_j \quad \text{for all } j = 1, \ldots, M \\ & \sum_{k=1}^{K} x_k + s_0 = 1 \\ & x_k \geq 0, \text{ for all } k = 1, \ldots, K \\ & s_j \geq 0, \text{ for all } j = 0, 1 \ldots, M. \end{aligned}$$

Assumption 1: Without loss of generality, we assume that $\sum_{i=1}^{N} p_{ik} \leq 1$ for all $k = 1, \ldots, K$ and $\max_{i,k}(p_{ik}/a_{ij}) \leq 1$ for all $j = 1, \ldots, M$ throughout this proof. Otherwise, we can rescale the unit of price. Similarly, we can assume $\sum_{i=1}^{N} a_{ij} \leq 1$ for all $i = 1, \ldots, N$ by rescale the units of resources. Also, we assume $c_j \leq \sum_{i=1}^{N} a_{ij}$ for all $j = 1, \ldots, M$ because $c_j > \sum_{i=1}^{N} a_{ij}$ implies that resource j is sufficient even for the maximum possible demand, so we can remove resource j from the constraints.

Assumption 2: To simplify the proof, we assume for now that OPT_{UB} has a unique optimal solution (x^*, s^*), and let $X^* = \{k \mid x_k^* > 0\}$ be the active price vectors (i.e. support) in the optimal solution and $S^* = \{j \mid s_j^* > 0\}$ be the active slack variables in the optimal solution. Let $|X^*|$ denote the cardinality of X^*. We also assume the optimal solution is non-degenerate, so $|X^*| + |S^*| = M + 1$. In the final part of the proof, we will extend the result to multiple optimal solutions.

We denote $r_k(\theta) = \sum_{i=1}^{N} p_{ik}\theta_{ik}(t)$ and $b_{jk}(\theta) = \sum_{i=1}^{N} a_{ij}\theta_{ik}(t)$ as the revenue and resource consumption under sampled demand, respectively. Then, $OPT(\theta)$ defined in TS-fixed (Algorithm 2) can be formulated equivalently as

$$
\begin{aligned}
f(\theta) = \max_{x,s} & \sum_{k=1}^{K} r_k(\theta) x_k \\
\text{subject to } & \sum_{k=1}^{K} b_{jk}(\theta) x_k + s_j = c_j \text{ for all } j = 1, \ldots, M \\
& \sum_{k=1}^{K} x_k + s_0 = 1 \\
& x_k \geq 0, \text{ for all } k = 1, \ldots, K \\
& s_j \geq 0, \text{ for all } j = 0, 1 \ldots, M.
\end{aligned}
$$

Let $X \subset \{k \mid x_1, \ldots, x_K\}$ and $S \subset \{j \mid s_0, s_1 \ldots, s_M\}$. Define constant $\Delta > 0$ as follows:

$$
\begin{aligned}
\Delta = \min\Big\{ & X^* \subsetneq X \text{ or } S^* \subsetneq S \mid \max_{y} f^* - \sum_{j=1}^{M} c_j y_j - y_0 \\
& \text{subject to } \sum_{j=1}^{M} b_{jk} y_j + y_0 \geq r_k \text{ for all } k \in X, y_j \geq 0 \text{ for all } j \in S \Big\}.
\end{aligned}
$$

And define

$$\epsilon = \min_{j=1,\ldots,M} \frac{c_j \Delta}{8 \sum_{i=1}^{N} a_{ij}}.$$

Because OPT_{UB} has a unique and non-degenerate optimal solution, any other basic solution would be strictly suboptimal. The constant Δ is the minimum difference between the optimal value, f^*, and the value of the objective function associated with other basic solutions.

We first present a few technical lemmas for the main proof.

Lemma A.1. *Consider problem $OPT(\theta)$ where the decision variables are restricted to a subset X and S such that $X^* \subsetneq X$ or $S^* \subsetneq S$. If $|\theta_{ik}(t) - d_{ik}| \leq \epsilon$ for all $i = 1, \ldots, N$ and $k \in X$, the optimal value of $OPT(\theta)$ satisfies $f(\theta) \leq f^* - \frac{3\Delta}{4}$.*

This lemma states that for any X such that $X^* \subsetneq X$ or S such that $S^* \subsetneq S$, as long as the difference between the sampled demand and the true demand is small enough, the optimal value of $OPT(\theta)$ is bounded away from f^*.

Proof. We first show that the defined constant, Δ, is indeed positive. According to the definition of (X^*, S^*), for any subset X such that $X^* \subsetneq X$ or any subset S such that $S^* \subsetneq S$, the following LP is either infeasible or has an optimal value strictly less than f^*.

$$\begin{aligned}
&\max_{x,s} \sum_{k \in X} r_k x_k \\
&\sum_{k \in X} b_{jk} x_k + s_j = c_j, \text{ for all } j = 1, \ldots, M \\
&\sum_{k \in X} x_k + s_0 = 1 \\
&x_k \geq 0 \text{ for all } k \in X \\
&s_j \geq 0 \text{ for all } j = 0, \ldots, M; s_j = 0 \text{ for all } j \notin S.
\end{aligned}$$

Let y_j $(j = 1, \ldots, M)$ be the dual variables associated with the first set of constraints, and y_0 be the dual variable associated with the second constraint. The dual of the LP is

$$\begin{aligned}
&\min_{y} \sum_{j=1}^{M} c_j y_j + y_0 \qquad (A.1) \\
&\text{subject to } \sum_{j=1}^{M} b_{jk} y_j + y_0 \geq r_k \quad \text{for all } k \in X \\
&y_j \geq 0 \text{ for all } j \in S.
\end{aligned}$$

Since the dual is feasible, its optimal value is either negative infinity or strictly less than f^* by strong duality, and thus Δ is positive.

Now consider problem $OPT(\theta)$ where the decision variables are restricted to a subset (X, S). Suppose $|\theta_{ik}(t) - d_{ik}| \leq \epsilon$ for all $i = 1, \ldots, N$ and $k \in X$. Let $\hat{x}_k$ $(k \in X)$ be the optimal solution to problem $OPT(\theta)$ with restricted (X, S), and let $\hat{y}_j$ $(j = 0, \ldots, M)$ be the optimal solution to the dual problem (A.1). We have

$$
\begin{aligned}
& f^* - f(\theta) \\
=& f^* - \sum_{k \in X} r_k(\theta)\hat{x}_k && \text{(A.2)} \\
\geq& f^* - \sum_{k \in X} r_k(\theta)\hat{x}_k + \sum_{j=1}^{M}(\sum_{k \in S} b_{jk}(\theta)\hat{x}_k - c_j)\hat{y}_j + (\sum_{k \in X} \hat{x}_k - 1)\hat{y}_0 && \text{(A.3)} \\
\geq& f^* - \sum_{k \in X} r_k\hat{x}_k + \sum_{j=1}^{M}(\sum_{k \in X} b_{jk}\hat{x}_k - c_j)\hat{y}_j + (\sum_{k \in X} \hat{x}_k - 1)\hat{y}_0 \\
& - (\sum_{k \in X} \hat{x}_k \sum_{i=1}^{N} p_{ik} + \sum_{k \in X} \hat{x}_k \sum_{j=1}^{M}(\hat{y}_j \sum_{i=1}^{N} a_{ij}))\epsilon && \text{(A.4)} \\
\geq& f^* - \sum_{k \in X} r_k\hat{x}_k + \sum_{j=1}^{M}(\sum_{k \in X} b_{jk}\hat{x}_k - c_j)\hat{y}_j + (\sum_{k \in X} \hat{x}_k - 1)\hat{y}_0 - (\sum_{k \in X} \hat{x}_k + \sum_{k \in X} \hat{x}_k \sum_{j=1}^{M} \hat{y}_j c_j)\frac{\Delta}{8} \\
& && \text{(A.5)} \\
\geq& \left(f^* - \sum_{j=1}^{M} c_j\hat{y}_j - \hat{y}_0 \right) + \sum_{k \in X} \hat{x}_k(\sum_{j=1}^{M} b_{jk}\hat{y}_j + \hat{y}_0 - r_k) - (1 + f^*)\frac{\Delta}{8} && \text{(A.6)} \\
\geq& \Delta + 0 - \frac{\Delta}{4} = \frac{3\Delta}{4}. && \text{(A.7)}
\end{aligned}
$$

Step (A.3) uses the fact that $\hat{y}_j \geq 0$ and $\sum_{k \in X} b_{jk}(\theta)\hat{x}_k \leq c_j$ for all $j \in S$, and also the fact that $\sum_{k \in X} b_{jk}(\theta)\hat{x}_k = c_j$ for all $j \notin S$. Step (A.4) uses the assumption that $|\theta_{ik}(t) - d_{ik}| \leq \epsilon$ for all $i = 1, \ldots, N$ and $k \in X$; therefore $|r_k(\theta) - r_k| \leq \sum_{i=1}^{N} p_{ik}\epsilon$ and $|b_{jk}(\theta) - b_{jk}| \leq \sum_{i=1}^{N} a_{ij}\epsilon$. In step (A.5), we use the definition of ϵ and the fact that $c_j \leq \sum_{i=1}^{N} a_{ij}$ and $\sum_{i=1}^{N} p_{ik} \leq 1$. Steps (A.6) and (A.7) use the fact that $\sum_{k \in X} \hat{x}_k \leq 1$ and $\sum_{j=1}^{M} c_j\hat{y}_j + \hat{y}_0 \leq f^* - \Delta \leq 1 - \Delta < 1$.

Lemma A.2. *We define $\mathcal{F}_{t-1}$ as the history prior to period t. We have*

$$
E\left[\frac{\Pr(\exists i, k \in X^* : \theta_{ik}(t) < d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})}{\Pr(\forall i, k \in X^* : \theta_{ik}(t) \geq d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})} \mid N_k(t-1)\ \forall k \in X^* \right] \leq O(\frac{1}{\Delta^{N|X^*|}}). \quad \text{(A.8)}
$$

Furthermore, if $N_k(t-1) \geq l \geq 32/\epsilon$ for all $k \in X^*$, we have

$$E\left[\frac{\Pr(\exists i, k \in X^* : \theta_{ik}(t) < d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})}{\Pr(\forall i, k \in X^* : \theta_{ik}(t) \geq d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})} \mid N_k(t-1)\ \forall k \in X^*\right] \leq O(\frac{1}{l\Delta^{N|X^*|+1}}). \tag{A.9}$$

Proof. Let $N_k(t-1)$ be the number of times that arm k has been played prior to period t. We have

$$\begin{aligned}
&E\left[\frac{\Pr(\exists i, k \in X^* : \theta_{ik}(t) < d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})}{\Pr(\forall i, k \in X^* : \theta_{ik}(t) \geq d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})}\right] \\
=&E\left[E[\frac{1}{\Pr(\forall i, k \in X^* : \theta_{ik}(t) \geq d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})} \mid N_k(t-1)\ \forall k \in X^*] - 1\right] \\
=&E\left[\prod_{k \in X^*}\prod_{i=1}^{N} E[\frac{1}{\Pr(\theta_{ik}(t) \geq d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})} \mid N_k(t-1)] - 1\right] \\
\leq&E\left[\prod_{k \in X^*}\prod_{i=1}^{N}(\frac{12}{\epsilon} + 1) - 1\right] = O\left(\frac{1}{\Delta^{N|X^*|}}\right).
\end{aligned}$$

The first equality is due to the fact that $\Pr(\exists i, k \in X^* : \theta_{ik}(t) < d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1}) = 1 - \Pr(\forall i, k \in X^* : \theta_{ik}(t) \geq d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})$ and uses the Tower rule. The second step uses the fact that conditioned on $N_k(t-1)$, the random variables $\Pr(\theta_{ik}(t) \geq d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})$ are independent. The last inequality uses Fact B.2.

Furthermore, if $N_k(t-1) \geq l \geq 32/\epsilon$ for all $k \in X^*$, we have

$$\begin{aligned}
&E\left[\frac{\Pr(\exists i, k \in X^* : \theta_{ik}(t) < d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})}{\Pr(\forall i, k \in X^* : \theta_{ik}(t) \geq d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})}\right] \\
=&E\left[\prod_{k \in X^*}\prod_{i=1}^{N} E[\frac{1}{\Pr(\theta_{ik}(t) \geq d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})} \mid N_k(t-1)] - 1\right] \\
\leq&E\left[\prod_{k \in X^*}\prod_{i=1}^{N}\left(O(\frac{1}{l\epsilon^2}) + 1\right) - 1\right] \\
=&\left(O(\frac{1}{l\epsilon^2}) + 1\right)^{N|X^*|} - 1 \leq O\left(\frac{1}{l\Delta^{N|X^*|+1}}\right).
\end{aligned}$$

The inequality again uses Fact B.2. Since $l \geq 32/\epsilon$, we have $O(1/(l\epsilon^2)) \leq O(1/(32\epsilon))$.

$$\left(O(\frac{1}{l\epsilon^2}) + 1\right)^{N|X^*|} - 1 \leq O\left(\frac{1}{l\epsilon^2}(N|X^*|)(1 + \frac{1}{32\epsilon})^{N|X^*|-1}\right) = O\left(\frac{1}{l\epsilon^{N|X^*|+1}}\right).$$

The inequality uses the fact that if $0 < x \leq c$, $(1+x)^y - 1 = \int_0^x y(1+u)^{y-1}du \leq xy(1+c)^{y-1}$.

Lemma A.3. *If $X_\epsilon(t) = X^*$ and $S(t) = S^*$, then for all $k = 1, \ldots, K$, there exists some constant $L > 0$ such that*

$$\left| \frac{x_k^*}{\sum_{k=1}^K x_k^*} - \frac{x_k(t)}{\sum_{k=1}^K x_k(t)} \right| \leq L\epsilon.$$

Proof. Define resource vector $c = (c_1, \ldots, c_M, 1)^T$. Let matrix B contain the optimal basis of OPT_{UB}, i.e., B is the coefficient matrix of OPT_{UB} whose columns are restricted to X^* and S^*. So we have $(x^*, s^*) = B^{-1}c$. Likewise, let $B(\theta)$ be the coefficient matrix with the same columns where the mean demands are replaced with sampled demands. Since $X_\epsilon(t) = X^*$, and $S(t) = S^*$, we have $(x(t), s(t))^T = B(\theta)^{-1}c$. In addition, we have $B(\theta) = B + \epsilon E$, where E is a matrix with elements in $[-1, 1]$, so

$$B(\theta)^{-1}c = (B + \epsilon E)^{-1}c = (I + \epsilon B^{-1}E)^{-1}B^{-1}c = (I - \epsilon B^{-1}E + O(\epsilon^2))B^{-1}c.$$

The last step is a result from the calculus of inverse matrix.

For $k \in X^*$, let e_k be the unit vector of length $|X^*| + |S^*| = M + 1$, whose element corresponding to x_k is one. We have

$$|x_k^* - x_k(t)| = |e_k^T(B^{-1} - B(\theta)^{-1})c| = |e_k^T(-\epsilon B^{-1}E + O(\epsilon^2))c| = O(\epsilon),$$

and by the Mean Value Theorem

$$\left| \frac{x_k^*}{\sum_{k=1}^K x_k^*} - \frac{x_k(t)}{\sum_{k=1}^K x_k(t)} \right| = O(\epsilon).$$

Lemma A.4. *Let τ be the minimum time t such that $N_k(t-1) \geq n$ for all $k \in X^*$. Then, for some $\gamma > 0$, we have*

$$E[\sum_{t=1}^{\tau} I(X_\epsilon(t) = X^*, S(t) = S^*)] \leq \frac{|X^*|}{\gamma} n.$$

Proof. Because we assume the optimal solution x^* is non-degenerate, we have $x_k^* > 0$ for all $k = 1, \ldots, K$. By Lemma A.3, if $X_\epsilon(t) = X^*$ and $S(t) = S^*$, then we have $|x_k^* - x_k(t)| = O(\epsilon)$ for all k. Suppose ϵ is small enough so that we have $x_k(t) \geq \gamma > 0$ for all k. (Note that we can reduce the value of ϵ defined in §A.1.1 to make this hold.) That means every arm is played with probability at least γ. So the expected time to play all arms in X^* for

at least n times is bounded by $|X^*|n/\gamma$.

A.1.2 Proof of Theorem 2.1

We now prove Theorem 2.1. For each time period $t = 1, \ldots, T$, we define the (possibly empty) set $X_\epsilon(t) = \{k : x_k(t) > 0, |\theta_{ik}(t) - d_{ik}| \leq \epsilon \forall i = 1, \ldots, N\}$. This set includes all arms in the optimal solution of $OPT(\theta)$ whose sampled demand is close to the true demand. Similarly, we define set $S(t) = \{j : s_j(t) > 0\}$. Recall that $N_k(T)$ is the number of times that the retailer offers price vector p_k during the selling season, and $Z_i(t)$ is the sales quantity of product i at time t (for all $i = 1, \ldots, N$, $t = 1, \ldots, T$).

From Section 2.4.1, we have

$$Regret(T) \leq f^*T - E[R(T)].$$

We need to provide a lower bound on the retailer's expected revenue, $E[R(T)]$, in order to complete the proof. To this end, we consider a new end period of the selling season: $T' = \lfloor (\sum_{k=1}^{K} x_k^*)T \rfloor$ and only count the retailer's revenue obtained from period 1 to period T', which is denoted as $E[R(T')]$. Note that if $\sum_{k=1}^{K} x_k^* < 1$, the optimal pricing strategy results in the retailer consuming all of a resource prior to the end of the selling horizon. Thus, we are essentially only considering revenue earned in periods prior to T', when we expect this resource to be fully consumed. (Clearly, $E[R(T')]$ is a lower bound of $E[R(T)]$.)

The retailer's total revenue $E[R(T)]$ is lower bounded by

$$E[R(T)] \geq E[R(T')] \geq \sum_{k \in X^*} r_k E[N_k(T')] - E[\sum_{j=1}^{M} \left(\max_{\substack{i=1,\ldots,N \\ k=1,\ldots,K}} \frac{p_{ik}}{a_{ij}} \right) (\sum_{i=1}^{N} \sum_{t=1}^{T'} a_{ij} Z_i(t) - I_j)^+]. \tag{A.10}$$

The first term on the right hand side, $\sum_{k \in X^*} r_k E[N_k(T')]$, is the expected revenue *without* inventory constraints when the retailer chooses arms in X^*; revenue obtained from arms not in X^* is ignored. This term can be decomposed as

$$\begin{aligned} & \sum_{k \in X^*} r_k E[N_k(T')] \\ &= \sum_{k \in X^*} r_k E\left[\sum_{t=1}^{T'} \frac{x_k(t)}{\sum_{k=1}^{K} x_k(t)} \right] \end{aligned}$$

$$
\begin{aligned}
&= \sum_{k\in X^*} r_k E\left[\sum_{t=1}^{T'} \frac{x_k^*(t)}{\sum_{k=1}^K x_k^*(t)}\right] + \sum_{k\in X^*} r_k E\left[\sum_{t=1}^{T'}\left(\frac{x_k(t)}{\sum_{k=1}^K x_k(t)} - \frac{x_k^*(t)}{\sum_{k=1}^K x_k^*(t)}\right)\right] \\
&\geq f^* \frac{T'}{\sum_{k=1}^K x_k^*(t)} + \sum_{k\in X^*} r_k E\left[\sum_{t=1}^{T'}\left(\frac{x_k(t)}{\sum_{k=1}^K x_k(t)} - \frac{x_k^*(t)}{\sum_{k=1}^K x_k^*(t)}\right)\right] \\
&\geq (f^*T - 1) - (\max_{k\in X^*} r_k) E[\sum_{t=1}^{T'} I(X_\epsilon(t) \neq X^* \text{ or } S(t) \neq S^*)] \\
&\quad + \sum_{k\in X^*} r_k E\left[\sum_{t=1}^{T'} I(X_\epsilon(t) = X^*, S(t) = S^*)\left(\frac{x_k(t)}{\sum_{k=1}^K x_k(t)} - \frac{x_k^*(t)}{\sum_{k=1}^K x_k^*(t)}\right)\right].
\end{aligned}
$$

Since no sales can be made when inventory runs out, the first term in Equation (A.10) overestimates the retailer's actual revenue. As a result, the second term in (A.10),

$$
E[\sum_{j=1}^{M} \max_{i,k}(p_{ik}/a_{ij}) \cdot (\sum_{i=1}^{N}\sum_{t=1}^{T'} a_{ij} Z_i(t) - I_j)^+],
$$

subtracts revenue obtained after resource consumption exceeds the inventory limit. For all $j = 1, \ldots, M$, $(\sum_{i=1}^{N}\sum_{t=1}^{T'} a_{ij} Z_i(t) - I_j)^+$ is the consumption of resource j that exceeds inventory budget I_j, and coefficient $\max_{i,k}(p_{ik}/a_{ij})$ is the maximum revenue that can be gained by adding one unit of resource j. Because we have assumed that $\max_{i,k}(p_{ik}/a_{ij}) \leq 1$ in Section A.1.1 for all $j = 1, \ldots, M$, we can simplify (A.10) as:

$$
\begin{aligned}
E[R(T')] &\geq \sum_{k\in X^*} r_k E[N_k(T')] - E[\sum_{j=1}^{M}(\sum_{i=1}^{N}\sum_{t=1}^{T'} a_{ij} Z_i(t) - I_j)^+] \\
&\geq (f^*T - 1) - E[\sum_{t=1}^{T'} I(X_\epsilon(t) \neq X^* \text{ or } S(t) \neq S^*)] \\
&\quad - \sum_{k\in X^*} r_k E\left[\sum_{t=1}^{T'} I(X_\epsilon(t) = X^*, S(t) = S^*)\left(\frac{x_k^*(t)}{\sum_{k=1}^K x_k^*(t)} - \frac{x_k(t)}{\sum_{k=1}^K x_k(t)}\right)\right] \\
&\quad - E[\sum_{j=1}^{M}(\sum_{i=1}^{N}\sum_{t=1}^{T'} a_{ij} Z_i(t) - I_j)^+]
\end{aligned}
$$

Because $r_k \leq 1$ for all $k = 1, \ldots, K$, the term $E[\sum_{t=1}^{T'} I(X_\epsilon(t) \neq X^* \text{ or } S(t) \neq S^*)]$ is an upper bound of the revenue loss when $X_\epsilon(t) \neq X^*$ or $S(t) \neq S^*$ happens, which we will bound in Part I of this proof. The term $\sum_{k\in X^*} r_k E[\sum_{t=1}^{T'} I(X_\epsilon(t) = X^*, S(t) = S^*)\left(\frac{x_k^*(t)}{\sum_{k=1}^K x_k^*(t)} - \frac{x_k(t)}{\sum_{k=1}^K x_k(t)}\right)]$ is the revenue loss in case $X_\epsilon(t) = X^*$ and $S(t) = S^*$, and we

consider it in Part II of the proof. The last term $E[\sum_{j=1}^{M}(\sum_{i=1}^{N}\sum_{t=1}^{T'} a_{ij}Z_i(t) - I_j)^+]$ will be bounded in Part III.

Part I: Bound the term:

$$E[\sum_{t=1}^{T'} I(X_\epsilon(t) \neq X^* \text{ or } S(t) \neq S^*)].$$

In this part, we bound the expected number of periods where $X_\epsilon(t) \neq X^*$ or $S(t) \neq S^*$ happens.

Case I(a): First, we consider the case where $X_\epsilon(t) \neq X^*$ and $f(\theta) \geq f^* - \frac{\Delta}{2}$ (or $S(t) \neq S^*$ and $f(\theta) \geq f^* - \frac{\Delta}{2}$). This is the case where $OPT(\theta)$ produces a suboptimal basis; note that the true mean revenue is less than $f^* - \Delta$ if a suboptimal basis is chosen, so intuitively this case should not happen very often.

Since $X_\epsilon(t) \neq X^*$, by Lemma A.1, we have

$$\sum_{k \in X_\epsilon(t)} r_k(\theta)x_k \leq f^* - 3\Delta/4,$$

so $\sum_{k \notin X_\epsilon(t)} r_k(\theta)x_k \geq \Delta/4$. Because $r_k(\theta) \leq 1$ for all $k = 1, \ldots, K$ and there can be at most $(M+1)$ arms with $x_k > 0$, there exists some arm $k \notin X_\epsilon(t)$ such that $x_k \geq \Delta/(4M+4)$, and thus arm k is selected with probability $x_k / \sum_{k=1}^{K} x_k \geq x_k \geq \Delta/(4M+4)$. Therefore, $\Delta/(4M+4)$ is the minimum probability that an arm not in $X_\epsilon(t)$ is played in Case 1(a).

By Fact B.1, if arm k has been pulled $N_k(t-1) \geq 2\log(NT)/\epsilon^2$ times (for any $\epsilon > 0$), the probability that $|\theta_{ik}(t) - d_{ik}| \geq \epsilon$ for some $i = 1, \ldots, N$ is bounded by $4/T$. For each period $t = 1, \ldots, T$, we define the event $A_t = \{\text{for all } k \text{ such that } N_k(t-1) \geq 2\log(NT)/\epsilon^2 : |\theta_{ik}(t) - d_{ik}| \leq \epsilon \, \forall i = 1, \ldots, N\}$, and we have $\Pr(A_t^c) \leq 4K/T$. So

$$\begin{aligned}
&\sum_{t=1}^{T} E\left[\Pr(X_\epsilon(t) \neq X^*, f(\theta) \geq f^* - \frac{\Delta}{2} \mid \mathcal{F}_{t-1})\right] \\
=&\sum_{t=1}^{T} E\left[\Pr(X_\epsilon(t) \neq X^*, f(\theta) \geq f^* - \frac{\Delta}{2} \mid \mathcal{F}_{t-1}) \mid A_t\right] \Pr(A_t) \\
&+ \sum_{t=1}^{T} E\left[\Pr(X_\epsilon(t) \neq X^*, f(\theta) \geq f^* - \frac{\Delta}{2} \mid \mathcal{F}_{t-1}) \mid A_t^c\right] \Pr(A_t^c) \\
\leq&\sum_{t=1}^{T} E\left[\Pr(X_\epsilon(t) \neq X^*, f(\theta) \geq f^* - \frac{\Delta}{2} \mid \mathcal{F}_{t-1}) \mid A_t\right]
\end{aligned}$$

$$+\sum_{t=1}^{T}\Pr(A_t^c) \leq \frac{4M+4}{\Delta}\cdot\frac{K\log NT}{\epsilon^2}+\sum_{t=1}^{T}\frac{4K}{T}=O(\frac{MK\log NT}{\Delta^3}).$$

The term $\frac{4M+4}{\Delta}\cdot\frac{K\log NT}{\epsilon^2}$ is the expected stopping time that all k arms have been played for $2\log(NT)/\epsilon^2$ times, in which case the event $\{X_\epsilon(t)\neq X^*, f(\theta)\geq f^*-\frac{\Delta}{2}\}$ cannot happen anymore under A_t.

The case $S(t)\neq S^*, f(\theta)\geq f^*-\frac{\Delta}{2}$ can be bounded in the same way so we omit the proof.

Case I(b): We then consider the case $f(\theta)<f^*-\frac{\Delta}{2}$ and therefore either $X_\epsilon(t)\neq X^*$ or $S(t)\neq S^*$. We will bound the total number of times that this case can happen:

$$\sum_{t=1}^{T}E\left[\Pr(f(\theta)<f^*-\frac{\Delta}{2}\mid\mathcal{F}_{t-1})\right].$$

Recall that X^* is the support of the optimal solution of the deterministic upper bound OPT_{UB}. If $\theta_{ik}(t)\geq d_{ik}-\epsilon/4$ for all $i=1,\ldots,N$ and $k\in X^*$, it is easily verifiable that $f(\theta)\geq f^*-\Delta/2$. So

$$\begin{aligned}
&\Pr(\forall i,k\in X^*:\theta_{ik}(t)\geq d_{ik}-\frac{\epsilon}{4}\mid\mathcal{F}_{t-1})\leq\Pr(f(\theta)\geq f^*-\frac{\Delta}{2}\mid\mathcal{F}_{t-1})\\
=&\Pr(X_\epsilon(t)=X^*,S(t)=S^*,f(\theta)\geq f^*-\frac{\Delta}{2}\mid\mathcal{F}_{t-1})\\
&+\Pr(X_\epsilon(t)\neq X^*\text{ or }S(t)\neq S^*,f(\theta)\geq f^*-\frac{\Delta}{2}\mid\mathcal{F}_{t-1})\\
=&\Pr(X_\epsilon(t)=X^*,S(t)=S^*\mid\mathcal{F}_{t-1})+\Pr(X_\epsilon(t)\neq X^*\text{ or }S(t)\neq S^*,f(\theta)\geq f^*-\frac{\Delta}{2}\mid\mathcal{F}_{t-1}).
\end{aligned}\tag{A.11}$$

In addition, we have

$$\Pr(f(\theta)<f^*-\frac{\Delta}{2}\mid\mathcal{F}_{t-1})\leq\Pr(\exists i,k\in X^*:\theta_{ik}(t)<d_{ik}-\frac{\epsilon}{4}\mid\mathcal{F}_{t-1}).\tag{A.12}$$

Combining (A.12) with (A.11), we have

$$\begin{aligned}
&\Pr(f(\theta)<f^*-\frac{\Delta}{2}\mid\mathcal{F}_{t-1})\\
\leq&\Pr(\exists i,k\in X^*:\theta_{ik}(t)<d_{ik}-\frac{\epsilon}{4}\mid\mathcal{F}_{t-1})\cdot\\
&\frac{\Pr(X_\epsilon(t)=X^*,S(t)=S^*\mid\mathcal{F}_{t-1})+\Pr(X_\epsilon(t)\neq X^*\text{ or }S(t)\neq S^*,f(\theta)\geq f^*-\frac{\Delta}{2}\mid\mathcal{F}_{t-1})}{\Pr(\forall i,k\in X^*:\theta_{ik}(t)\geq d_{ik}-\epsilon/4\mid\mathcal{F}_{t-1})}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{\Pr(\exists i, k \in X^* : \theta_{ik}(t) < d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})}{\Pr(\forall i, k \in X^* : \theta_{ik}(t) \geq d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})} \Pr(X_\epsilon(t) = X^*, S(t) = S^* \mid \mathcal{F}_{t-1}) \\
&\quad + \frac{\Pr(\exists i, k \in X^* : \theta_{ik}(t) < d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})}{\Pr(\forall i, k \in X^* : \theta_{ik}(t) \geq d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})} \cdot \\
&\quad s \Pr(X_\epsilon(t) \neq X^* \text{ or } S(t) \neq S^*, f(\theta) \geq f^* - \frac{\Delta}{2} \mid \mathcal{F}_{t-1}) \\
&\triangleq (\Phi_t) + (\Psi_t)
\end{aligned}
$$

Since we find it is more convenient to consider $\sqrt{(\Psi_t)}$ rather than (Ψ_t), we will bound

$$
\begin{aligned}
& E\left[\sum_{t=1}^{T} \Pr(f(\theta) < f^* - \frac{\Delta}{2} \mid \mathcal{F}_{t-1})\right] \\
\leq & E\left[\sum_{t=1}^{T} \sqrt{\Pr(f(\theta) < f^* - \frac{\Delta}{2} \mid \mathcal{F}_{t-1})}\right] \\
\leq & E\left[\sum_{t=1}^{T} \sqrt{(\Phi_t) + (\Psi_t)}\right] \\
\leq & E\left[\sum_{t=1}^{T} \left(\sqrt{(\Phi_t)} + \sqrt{(\Psi_t)}\right)\right].
\end{aligned}
$$

To bound (Φ_t), let τ_l denote the stopping time when all arms $k \in X^*$ have been selected for at least $l = 1, \ldots, T$ times.

$$
\begin{aligned}
& E[\sum_{t=1}^{T} (\Phi_t)] \\
= & E\left[\sum_{t=1}^{T} \frac{\Pr(\exists i, k \in X^* : \theta_{ik}(t) < d_{ik} - \epsilon \mid \mathcal{F}_{t-1})}{\Pr(\forall i, k \in X^* : \theta_{ik}(t) \geq d_{ik} - \epsilon \mid \mathcal{F}_{t-1})} \Pr(X_\epsilon(t) = X^*, S(t) = S^* \mid \mathcal{F}_{t-1})\right] \\
= & E\left[\sum_{t=1}^{T} \frac{\Pr(\exists i, k \in X^* : \theta_{ik}(t) < d_{ik} - \epsilon \mid \mathcal{F}_{t-1})}{\Pr(\forall i, k \in X^* : \theta_{ik}(t) \geq d_{ik} - \epsilon \mid \mathcal{F}_{t-1})} I(X_\epsilon(t) = X^*, S(t) = S^*)\right] \\
\leq & E\Big[\sum_{t=1}^{\tau_1} \frac{\Pr(\exists i, k \in X^* : \theta_{ik}(t) < d_{ik} - \epsilon \mid \mathcal{F}_{t-1})}{\Pr(\forall i, k \in X^* : \theta_{ik}(t) \geq d_{ik} - \epsilon \mid \mathcal{F}_{t-1})} I(X_\epsilon(t) = X^*, S(t) = S^*) \\
& + \sum_{l=1}^{T-1} \sum_{t=\tau_l+1}^{\tau_{l+1}} \frac{\Pr(\exists i, k \in X^* : \theta_{ik}(t) < d_{ik} - \epsilon \mid \mathcal{F}_{t-1})}{\Pr(\forall i, k \in X^* : \theta_{ik}(t) \geq d_{ik} - \epsilon \mid \mathcal{F}_{t-1})} I(X_\epsilon(t) = X^*, S(t) = S^*)\Big].
\end{aligned}
$$

The first step is by the Tower rule. The second step is an inequality because $\tau_T \geq T$. By

Lemma A.2, if $l < 32/\epsilon$, we have

$$
\begin{aligned}
=&E\left[\sum_{t=\tau_l+1}^{\tau_{l+1}} \frac{\Pr(\exists i,k\in X^*:\theta_{ik}(t)<d_{ik}-\epsilon/4\mid \mathcal{F}_{t-1})}{\Pr(\forall i,k\in X^*:\theta_{ik}(t)\geq d_{ik}-\epsilon/4\mid \mathcal{F}_{t-1})} I(X_\epsilon(t)=X^*,S(t)=S^*)\right]\\
=&E\left[\sum_{t=\tau_l+1,X_\epsilon(t)=X^*,S(t)=S^*}^{\tau_{l+1}} \frac{\Pr(\exists i,k\in X^*:\theta_{ik}(t)<d_{ik}-\epsilon/4\mid \mathcal{F}_{t-1})}{\Pr(\forall i,k\in X^*:\theta_{ik}(t)\geq d_{ik}-\epsilon/4\mid \mathcal{F}_{t-1})}\right]\\
\leq&E\left[\sum_{t=\tau_l+1,X_\epsilon(t)=X^*,S(t)=S^*}^{\tau_{l+1}} O\left(\frac{1}{\Delta^{N(M+1)}}\right)\right]=\frac{|X^*|}{\gamma}\cdot O\left(\frac{1}{\Delta^{N|X^*|}}\right)
\end{aligned}
$$

The last step is due to Lemma A.4. Similarly, if $l \geq 32/\epsilon$, we have

$$
\begin{aligned}
&E\left[\sum_{t=\tau_l+1}^{\tau_{l+1}} \frac{\Pr(\exists i,k\in X^*:\theta_{ik}(t)<d_{ik}-\epsilon/4\mid \mathcal{F}_{t-1})}{\Pr(\forall i,k\in X^*:\theta_{ik}(t)\geq d_{ik}-\epsilon/4\mid \mathcal{F}_{t-1})} I(X_\epsilon(t)=X^*,S(t)=S^*)\right]\\
=&E\left[\sum_{t=\tau_l+1,X_\epsilon(t)=X^*,S(t)=S^*}^{\tau_{l+1}} \frac{\Pr(\exists i,k\in X^*:\theta_{ik}(t)<d_{ik}-\epsilon/4\mid \mathcal{F}_{t-1})}{\Pr(\forall i,k\in X^*:\theta_{ik}(t)\geq d_{ik}-\epsilon/4\mid \mathcal{F}_{t-1})}\right]\\
\leq&E\left[\sum_{t=\tau_l+1,X_\epsilon(t)=X^*,S(t)=S^*}^{\tau_{l+1}} O\left(\frac{1}{l\Delta^{NM+N+1}}\right)\right]\\
=&\frac{|X^*|}{\gamma}O\left(\frac{1}{l\Delta^{N|X^*|+1}}\right)
\end{aligned}
$$

Summing over these terms, we have

$$
\begin{aligned}
&E[\sum_{t=1}^{T}(\Phi_t)]\\
=&E\left[\sum_{t=1}^{T} \frac{\Pr(\exists i,k\in X^*:\theta_{ik}(t)<d_{ik}-\epsilon/4\mid \mathcal{F}_{t-1})}{\Pr(\forall i,k\in X^*:\theta_{ik}(t)\geq d_{ik}-\epsilon/4\mid \mathcal{F}_{t-1})} \Pr(X_\epsilon(t)=X^*,S(t)=S^*\mid \mathcal{F}_{t-1})\right]\\
\leq&E\left[\sum_{l=1}^{32/\epsilon} O\left(\frac{|X^*|}{\gamma\Delta^{N|X^*|}}\right)+\sum_{l=32/\epsilon+1}^{T-1} O\left(\frac{|X^*|}{\gamma l\Delta^{N|X^*|+1}}\right)\right]\\
=&O\left(\frac{|X^*|\log T}{\gamma\Delta^{N|X^*|+1}}\right).
\end{aligned}
$$

Applying the Cauchy-Schwartz inequality twice, it holds that

$$
\sum_{t=1}^{T} E[\sqrt{(\Phi_t)}]\leq\sum_{t=1}^{T}\sqrt{E\left[(\Phi_t)\right]}
$$

$$
\begin{aligned}
&= \sum_{t=1}^{T} \sqrt{E\left[(\Phi_t)\right]} \cdot \sqrt{1} \\
&\leq \sqrt{\sum_{t=1}^{T} E\left[(\Phi_t)\right]} \sqrt{T} \\
&= O(\frac{\sqrt{|X^*|T \log T}}{\gamma \Delta^{(N|X^*|+1)/2}})
\end{aligned}
$$

The inequalities use Cauchy-Schwartz inequality (cf. Fact B.3).

To bound (Ψ_t), we have

$$
\begin{aligned}
&\sum_{t=1}^{T} E[\sqrt{(\Psi_t)}] \\
&= \sum_{t=1}^{T} E[\sqrt{\frac{\Pr(\exists i, k \in X^* : \theta_{ik}(t) < d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})}{\Pr(\forall i, k \in X^* : \theta_{ik}(t) \geq d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})}} \cdot \\
&\sqrt{\Pr(X_\epsilon(t) \neq X^* \text{ or } S(t) \neq S^*, f(\theta) \geq f^* - \Delta 2 \mid \mathcal{F}_{t-1})}] \\
&\leq \sum_{t=1}^{T} \sqrt{E\left[\frac{\Pr(\exists i, k \in X^* : \theta_{ik}(t) < d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})}{\Pr(\forall i, k \in X^* : \theta_{ik}(t) \geq d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})}\right]} \\
&\quad \cdot \sqrt{E\left[\Pr(X_\epsilon(t) \neq X^* \text{ or } S(t) \neq S^*, f(\theta) \geq f^* - \frac{\Delta}{2} \mid \mathcal{F}_{t-1})\right]} \\
&\leq O(\frac{1}{\Delta^{(N|X^*|)/2}}) \sum_{t=1}^{T} \sqrt{E\left[\Pr(X_\epsilon(t) \neq X^* \text{ or } S(t) \neq S^*, f(\theta) \geq f^* - \frac{\Delta}{2} \mid \mathcal{F}_{t-1})\right]} \\
&\leq O(\frac{1}{\Delta^{(N|X^*|)/2}}) \sqrt{\sum_{t=1}^{T} E\left[\Pr(X_\epsilon(t) \neq X^* \text{ or } S(t) \neq S^*, f(\theta) \geq f^* - \frac{\Delta}{2} \mid \mathcal{F}_{t-1})\right]} \sqrt{T} \\
&\leq O(\frac{1}{\Delta^{(N|X^*|)/2}} \cdot \frac{\sqrt{MK \log NT}}{\Delta^{3/2}} \sqrt{T}) \\
&= O(\frac{1}{\Delta^{(N|X^*|+3)/2}} \sqrt{MKT \log NT})
\end{aligned}
$$

The first inequality uses Fact B.3, the second inequality uses Lemma A.2, the third inequality uses Fact B.3, and the last inequality uses the result in Case 1(a).

Finally, we have

$$
\begin{aligned}
&E\left[\sum_{t=1}^{T} \Pr(f(\theta) < f^* - \frac{\Delta}{2} \mid \mathcal{F}_{t-1})\right] \\
&\leq E\left[\sum_{t=1}^{T} \left(\sqrt{(\Phi_t)} + \sqrt{(\Psi_t)}\right)\right]
\end{aligned}
$$

$$\leq O(\frac{\sqrt{|X^*|T\log T}}{\Delta^{(N|X^*|+1)/2}}) + O(\frac{1}{\Delta^{(N|X^*|+3)/2}}\sqrt{MKT\log NT})$$
$$\leq O(\frac{1}{\Delta^{(N|X^*|+3)/2}}\sqrt{MKT\log NT}).$$

To summarize for the entire Part I, we show that

$$\begin{aligned}
&E\left[\sum_{t=1}^{T'} I(X_\epsilon(t) \neq X^* \text{ or } S(t) \neq S^*)\right] \\
\leq &E[\sum_{t=1}^{T} I(X_\epsilon(t) \neq X^* \text{ or } S(t) \neq S^*)] \\
= &E\left[\sum_{t=1}^{T} I(X_\epsilon(t) \neq X^* \text{ or } S(t) \neq S^*, f(\theta) \geq f^* - \frac{\Delta}{2})\right] \\
&+ E\left[\sum_{t=1}^{T} I(X_\epsilon(t) \neq X^* \text{ or } S(t) \neq S^*, f(\theta) < f^* - \frac{\Delta}{2})\right] \\
\leq &O(\frac{MK\log NT}{\Delta^3}) + O(\frac{1}{\Delta^{(N|X^*|+3)/2}}\sqrt{MKT\log NT}) \\
= &O(\sqrt{T\log T}).
\end{aligned}$$

Part II: Bound the term

$$\sum_{k\in X^*} r_k E[\sum_{t=1}^{T'} I(X_\epsilon(t) = X^*, S(t) = S^*)\left(\frac{x_k^*(t)}{\sum_{k=1}^K x_k^*(t)} - \frac{x_k(t)}{\sum_{k=1}^K x_k(t)}\right)].$$

In this part, we bound the revenue loss when $X_\epsilon(t) = X^*$ and $S(t) = S^*$ happens. We first define a sequence of constants as follows:

$$\epsilon_1 = \min\{\epsilon, T'^{-\frac{1}{4}}\}, \epsilon_2 = \min\{\epsilon, T'^{-\frac{3}{8}}\}, \ldots, \epsilon_n = \min\{\epsilon, T'^{-\frac{2^n-1}{2^{n+1}}}\}, \ldots$$

and let τ_ν be the minimum t such that $N_k(t-1) \geq 2\log T'/\epsilon_\nu^2$ for all $k \in X^*$. By Lemma A.3 and Fact B.1, there exists a constant L such that if $X_\epsilon(t) = X^*$, $S(t) = S^*$ and $t \geq \tau_\nu$, we have

$$\Pr(\left|\frac{x_k(t)}{\sum_{k=1}^K x_k(t)} - \frac{x_k^*}{\sum_{k=1}^K x_k^*}\right| \geq L\epsilon_\nu) \leq \frac{4N}{T'}.$$

Let n be the smallest number such that $2\log T'/\epsilon_n^2 \geq T'$. We have

$$
\begin{aligned}
&E[\sum_{t=1}^{T'} I(X_\epsilon(t)=X^*, S(t)=S^*)(\frac{x_k^*}{\sum_{k=1}^K x_k^*} - \frac{x_k(t)}{\sum_{k=1}^K x_k(t)})] \\
=&E[\sum_{t=1}^{T'} I(X_\epsilon(t)=X^*, S(t)=S^*, t<\tau_1)(\frac{x_k^*}{\sum_{k=1}^K x_k^*} - \frac{x_k(t)}{\sum_{k=1}^K x_k(t)})] \\
&+E[\sum_{\nu=1}^{n-1}\sum_{t=1}^{T'} I(X_\epsilon(t)=X^*, S(t)=S^*, \tau_\nu \leq t < \tau_{\nu+1})(\frac{x_k^*}{\sum_{k=1}^K x_k^*} - \frac{x_k(t)}{\sum_{k=1}^K x_k(t)})] \\
\leq&E[\sum_{t=1}^{T'} I(X_\epsilon(t)=X^*, S(t)=S^*, t<\tau_1)] \\
&+E[\sum_{\nu=1}^{n-1}\sum_{t=1}^{T'} I(X_\epsilon(t)=X^*, S(t)=S^*, \tau_\nu \leq t < \tau_{\nu+1})\cdot \\
&I(\frac{x_k^*}{\sum_{k=1}^K x_k^*} - \frac{x_k(t)}{\sum_{k=1}^K x_k(t)} \leq L\epsilon_\nu)\cdot L\epsilon_\nu] \\
&+E[\sum_{\nu=1}^{n-1}\sum_{t=1}^{T'} I(X_\epsilon(t)=X^*, S(t)=S^*, \tau_\nu \leq t < \tau_{\nu+1}) I(\frac{x_k^*}{\sum_{k=1}^K x_k^*} - x_k(t) > L\epsilon_\nu)] \\
\leq&E[\sum_{t=1}^{T'} I(X_\epsilon(t)=X^*, S(t)=S^*, t<\tau_1)] \\
&+E[\sum_{\nu=1}^{n-1}\sum_{t=1}^{T'} I(X_\epsilon(t)=X^*, S(t)=S^*, \tau_\nu \leq t < \tau_{\nu+1})(L\epsilon_\nu + \frac{4N}{T'})]
\end{aligned}
$$

By Lemma A.4, we have

$$
E[\sum_{t=1}^{T'} I(X_\epsilon(t)=X^*, S(t)=S^*, t<\tau_\nu)] \leq \frac{|X^*|}{\gamma} T'^{\frac{2^\nu-1}{2^\nu}} \log T', \text{ for all } \nu.
$$

By definition of n, we have $2\log T'/\epsilon_{n-1}^2 < T'$, or equivalently $\epsilon_{n-1}^2 = T'^{-\frac{2^{n-1}-1}{2^n}} > 2\log T'/T'$. Simple algebra shows that $n < \log(2\log T'/\log(2\log T')) = O(\log\log T)$. Therefore,

$$
\begin{aligned}
&E[\sum_{t=1}^{T'} I(X_\epsilon(t)=X^*, S(t)=S^*)(\frac{x_k^*}{\sum_{k=1}^K x_k^*} - \frac{x_k(t)}{\sum_{k=1}^K x_k(t)})] \\
\leq&E[\sum_{t=1}^{T'} I(X_\epsilon(t)=X^*, S(t)=S^*, t<\tau_1)]
\end{aligned}
$$

$$
\begin{aligned}
&+\sum_{\nu=1}^{n-1} E[\sum_{t=1}^{T'} I(X_\epsilon(t)=X^*, S(t)=S^*, \tau_\nu \le t < \tau_{\nu+1})(L\epsilon_\nu + \frac{4N}{T'})] \\
\le & E[\sum_{t=1}^{T'} I(X_\epsilon(t)=X^*, S(t)=S^*, t<\tau_1)] \\
&+\sum_{\nu=1}^{n-1} E[\sum_{t=1}^{T'} I(X_\epsilon(t)=X^*, S(t)=S^*, t<\tau_{\nu+1})]\cdot L\epsilon_\nu \\
&+\sum_{\nu=1}^{n-1} E[\sum_{t=1}^{T'} I(X_\epsilon(t)=X^*, S(t)=S^*, \tau_\nu \le t<\tau_{\nu+1})]\cdot \frac{4N}{T'} \\
\le & \frac{|X^*|}{\gamma}\sqrt{T'}\log T' + \sum_{\nu=1}^{n-1}\left(\frac{|X^*|}{\gamma} T'^{\frac{2^{\nu+1}-1}{2^{\nu+1}}}\log(T')\cdot L\epsilon_\nu\right) + T'\cdot\frac{4N}{T'} \\
\le & \frac{|X^*|}{\gamma}\sqrt{T'}\log T' + (n-1)\frac{|X^*|}{\gamma}L\sqrt{T'}\log T' + 4N \\
= & \frac{|X^*|}{\gamma}\sqrt{T'}\log T' + O(\log\log T)\frac{|X^*|}{\gamma}L\sqrt{T'}\log T' + 4N.
\end{aligned}
$$

The second to last step uses the fact that $T'^{\frac{2^{\nu+1}-1}{2^{\nu+1}}}\epsilon_\nu = \sqrt{T'}$ for all ν. To recap, because $r_k \le 1$ for all k, we have the following bound

$$
\begin{aligned}
&\sum_{k\in X^*} r_k E[\sum_{t=1}^{T'} I(X_\epsilon(t)=X^*, S(t)=S^*)(\frac{x_k^*}{\sum_{k=1}^K x_k^*} - \frac{x_k(t)}{\sum_{k=1}^K x_k(t)})] \\
\le & \frac{|X^*|^2}{\gamma}\sqrt{T'}\log T' + O(\log\log T)\frac{|X^*|^2}{\gamma}L\sqrt{T'}\log T' + 4N \\
= & O(\sqrt{T}\log T\log\log T).
\end{aligned}
$$

Part III: Bound term

$$
E[\sum_{j=1}^{M}(\sum_{i=1}^{N}\sum_{t=1}^{T'} a_{ij}Z_i(t) - I_j)^+].
$$

For any resource $j = 1, \ldots, M$, we let $c_j' = c_j/(\sum_{k=1}^K x_k^*)$. This is interpreted as the expected rate of inventory consumption of resource j in the first T' time periods. Since $I_j = c_j T \ge c_j' T'$, we have

$$
E[\left(\sum_{i=1}^{N}\sum_{t=1}^{T'} a_{ij}Z_i(t) - I_j\right)^+]
$$

$$\leq E[\sum_{i=1}^{N}\sum_{t=1}^{T'} a_{ij} I(X_\epsilon(t) \neq X^* \text{ or } S(t) \neq S^*)]$$
$$+ E[\left(\sum_{i=1}^{N}\sum_{t=1}^{T'} a_{ij} Z_i(t) I(X_\epsilon(t) = X^*, S(t) = S^*) - c_j' T'\right)^+].$$

We show in Part I of the proof that

$$E[\sum_{i=1}^{N}\sum_{t=1}^{T'} a_{ij} I(X_\epsilon(t) \neq X^* \text{ or } S(t) \neq S^*)]$$
$$\leq O(\frac{MK \log NT}{\Delta^3}) + O(\frac{1}{\Delta^{(N|X^*|+3)/2}} \sqrt{MKT \log NT}).$$

Furthermore, we have

$$E[\left(\sum_{i=1}^{N}\sum_{t=1}^{T'} a_{ij} Z_i(t) I(X_\epsilon(t) = X^*, S(t) = S^*) - c_j' T'\right)^+]$$
$$\leq E[\Big(\sum_{i=1}^{N}\sum_{t=1}^{T'} (a_{ij} Z_i(t) I(X_\epsilon(t) = X^*, S(t) = S^*, t < \tau_1) - c_j')$$
$$+ \sum_{\nu=1}^{n-1}\sum_{t=1}^{T'} (\sum_{i=1}^{N} a_{ij} Z_i(t) I(X_\epsilon(t) = X^*, S(t) = S^*, \tau_\nu \leq t < \tau_{\nu+1}) - c_j')\Big)^+]$$
$$\leq E[\Big(\sum_{i=1}^{N}\sum_{t=1}^{T'} (a_{ij} Z_i(t) I(X_\epsilon(t) = X^*, S(t) = S^*, t < \tau_1) - c_j')\Big)^+$$
$$+ \sum_{\nu=1}^{n-1}\Big(\sum_{t=1}^{T'} (\sum_{i=1}^{N} a_{ij} Z_i(t) I(X_\epsilon(t) = X^*, S(t) = S^*, \tau_\nu \leq t < \tau_{\nu+1}) - c_j')\Big)^+]$$
$$\leq E[\sum_{i=1}^{N}\sum_{t=1}^{\tau_1} a_{ij} + \sum_{\nu=1}^{n-1}\Big(\sum_{t=1}^{T'} (\sum_{i=1}^{N} a_{ij} \dot{Z}_i(t) I(X_\epsilon(t) = X^*, S(t) = S^*, \tau_\nu \leq t < \tau_{\nu+1}) - c_j')\Big)^+].$$

For each $\nu = 1 \ldots, n$, let $U_i^\nu(t)$ for each $i = 1, \ldots, N$ and $t = 1, \ldots, T'$ be a Bernoulli random variable with success probability $\sum_{k=1}^{K} d_{ik}(x_k^*/(\sum_{k=1}^{K} x^*) + L\epsilon_\nu)$. Recall that

$$(\sum_{k=1}^{K} d_{ik} x_k^*)/(\sum_{k=1}^{K} x^*)$$

is the expected sales of product i by choosing arms according to the optimal solution x^*. Under the case $\{X_\epsilon(t) = X^*, S(t) = S^*, \tau_\nu \leq t < \tau_{\nu+1}\}$, the event $x_k(t)/(\sum_{k=1}^{K} x_k(t)) \leq x_k^*/(\sum_{k=1}^{K} x^*) + L\epsilon_\nu$ happens with probability at least $1 - 4N/T$ (by Lemma A.3 and B.1),

so we can upper bound $Z_i(t)$ by $U_i^\nu(t)$ with high probability.

Conditioning on τ_ν and τ_{nu+1},

$$
\begin{aligned}
&E[\Big(\sum_{t=1}^{T'}(\sum_{i=1}^{N} a_{ij}Z_i(t)I(X_\epsilon(t)=X^*,S(t)=S^*,\tau_\nu\le t<\tau_{\nu+1})-c'_j)\Big)^+ \mid \tau_\nu,\tau_{\nu+1}] \\
\le&E[\Big(\sum_{t=\tau_\nu}^{\tau_{\nu+1}-1}(\sum_{i=1}^{N} a_{ij}U_i^\nu(t)-c'_j)\Big)^+ \mid \tau_\nu,\tau_{\nu+1}]+E[\sum_{t=1}^{T'} I(\tau_\nu\le t<\tau_{\nu+1})\frac{4N}{T'}] \\
\le&\frac{\sqrt{\sigma_\nu^2(\tau_{\nu+1}-\tau_\nu)+(KL\epsilon_\nu)^2(\tau_{\nu+1}-\tau_\nu)^2}+KL\epsilon_\nu(\tau_{\nu+1}-\tau_\nu)}{2} \\
&+E[\sum_{t=1}^{T'} I(\tau_\nu\le t<\tau_{\nu+1})\frac{4N}{T'}].
\end{aligned}
$$

Since $U_i^\nu(t)$ are i.i.d., we use Fact B.4 in the last inequality. The constant σ_ν^2 is the variance of $\sum_{i=1}^N a_{ij}U_i^\nu(t)$; note that $\sigma_\nu^2 \le (\sum_{i=1}^N a_{ij})/4 \le 1/4$ because the maximum variance of a Bernoulli random variable is 1/4. We also use the fact that $E[\sum_{i=1}^N a_{ij}U_i^\nu(t)-c'_j]=KL\epsilon_\nu$.

Summing over ν, we have

$$
\begin{aligned}
&E[\sum_{i=1}^{N}\sum_{t=1}^{\tau_1} a_{ij}+\sum_{\nu=1}^{n-1}\Big(\sum_{t=1}^{T'}(\sum_{i=1}^{N} a_{ij}Z_i(t)I(X_\epsilon(t)=X^*,S(t)=S^*,\tau_\nu\le t<\tau_{\nu+1})-c'_j)\Big)^+] \\
\le&\frac{|X^*|}{\gamma}\sqrt{T'}\log T'+\sum_{\nu=1}^{n-1}E[\frac{\sqrt{\sigma_\nu^2(\tau_{\nu+1}-\tau_\nu)+(KL\epsilon_\nu)^2(\tau_{\nu+1}-\tau_\nu)^2}+KL\epsilon_\nu(\tau_{\nu+1}-\tau_\nu)}{2}] \\
&+\sum_{\nu=1}^{n-1}E[\sum_{t=1}^{T'} I(\tau_\nu\le t\le\tau_{\nu+1})\frac{4N}{T'}] \\
\le&\frac{|X^*|}{\gamma}\sqrt{T'}\log T'+\sum_{\nu=1}^{n-1}E[\frac{\sqrt{1/4*(\tau_{\nu+1}-\tau_\nu)}+KL\epsilon_\nu(\tau_{\nu+1}-\tau_\nu)+KL\epsilon_\nu(\tau_{\nu+1}-\tau_\nu)}{2}] \\
&+T'\cdot\frac{4N}{T'} \\
\le&\frac{|X^*|}{\gamma}\sqrt{T'}\log T'+\sum_{\nu=1}^{n-1}E[\frac{\sqrt{(\tau_{\nu+1}-\tau_\nu)}}{4}+LK\epsilon_\nu(\tau_{\nu+1}-\tau_\nu)]+4N \\
\le&\frac{|X^*|}{\gamma}\sqrt{T'}\log T'+\frac{1}{4}E[\sum_{\nu=1}^{n-1}\sqrt{(\tau_{\nu+1}-\tau_\nu)}]+O(K\frac{|X^*|}{\gamma}\sqrt{T}\log T\log\log T)+4N \\
\le&\frac{|X^*|}{\gamma}\sqrt{T'}\log T'+\frac{1}{4}E[\sqrt{\sum_{\nu=1}^{n-1}(\tau_{\nu+1}-\tau_\nu)}\sqrt{\sum_{\nu=1}^{n-1}1}]+O(K\frac{|X^*|}{\gamma}\sqrt{T}\log T\log\log T)+4N \\
\le&\frac{|X^*|}{\gamma}\sqrt{T'}\log T'+\frac{1}{4}\sqrt{E[\sum_{\nu=1}^{n-1}(\tau_{\nu+1}-\tau_\nu)]\sqrt{n-1}}+O(K\frac{|X^*|}{\gamma}\sqrt{T}\log T\log\log T)+4N
\end{aligned}
$$

$$\leq \frac{|X^*|}{\gamma}\sqrt{T'}\log T' + O(\sqrt{T})O(\sqrt{\log\log T}) + O(K\frac{|X^*|}{\gamma}\sqrt{T}\log T\log\log T) + 4N.$$

The first inequality uses the fact that $E[\tau_1] \leq \frac{|X^*|}{\gamma}\sqrt{T'}\log T'$. The fourth inequality uses the result from Part II:

$$\sum_{\nu=1}^{n-1} E[\epsilon_\nu(\tau_{\nu+1} - \tau_\nu)] \leq O(\frac{|X^*|}{\gamma}\sqrt{T}\log T\log\log T).$$

The fifth and sixth inequalities are due to the Cauchy-Schwartz inequality (Fact B.3). The last inequality is due to $E[\tau_n] = O(T)$ and $n = O(\log\log T)$, which are results from Part II.

To summarize, we have

$$\begin{aligned}
&E[\sum_{j=1}^{M}(\sum_{i=1}^{N}\sum_{t=1}^{T'} a_{ij}Z_i(t) - I_j)^+] \\
\leq& M\Big(O(\frac{MK\log NT}{\Delta^3}) + O(\frac{1}{\Delta^{(N|X^*|+3)/2}}\sqrt{MKT\log NT}) \\
&+ O(\sqrt{T}\log T) + O(\sqrt{T})O(\sqrt{\log\log T}) + O(\sqrt{T}\log T\log\log T)\Big) \\
=& O(\sqrt{T}\log T\log\log T).
\end{aligned}$$

Combining Parts I, II and III completes the proof.

The regret is bounded by

$$\begin{aligned}
Regret(T) \leq& f^*T - E[R(T')] \\
\leq& f^*T - f^*T + 1 + (\max_{k\in X^*} r_k)E[\sum_{t=1}^{T'} I(X_\epsilon(t) \neq X^* \text{ or } S(t) \neq S^*)] \\
&+ \sum_{k\in X^*} r_k E[\sum_{t=1}^{T'} I(X_\epsilon(t) = X^*, S(t) = S^*)\left(\frac{x_k^*(t)}{\sum_{k=1}^K x_k^*(t)} - \frac{x_k(t)}{\sum_{k=1}^K x_k(t)}\right)] \\
&+ E[\sum_{j=1}^{M}(\sum_{i=1}^{N}\sum_{t=1}^{T'} a_{ij}Z_i(t) - I_j)^+] \\
\leq& O(\sqrt{T\log T}) + O(\sqrt{T}\log T\log\log T) + O(\sqrt{T}\log T\log\log T) \\
=& O(\sqrt{T}\log T\log\log T).
\end{aligned}$$

Extending the proof for multiple optimal solutions.

We consider the case where the optimal solution of OPT_{UB} is any convex combination

of two extreme points (x^*, s^*) and $(\bar{x}, \bar{s})$. The case of more than two optimal extreme points can be resolved using the same method. Define $X^* = \{k \mid x_k^* > 0\}, S^* = \{j \mid s_j^* > 0\}, \bar{X} = \{k \mid \bar{x}_k > 0\}, \bar{S} = \{j \mid \bar{s_j} > 0\}$.

In order to modify Lemma A.1, we define constant Δ as follows:

$$\begin{aligned}\Delta = \min\big\{X, S :&(X^* \subsetneq X \text{ or } S^* \subsetneq S) \text{ and } (\bar{X} \subsetneq X \text{ or } \bar{S} \subsetneq S) : \\ &\max_y f^* - \sum_{j=1}^{M} c_j y_j - y_0 \\ &\text{subject to } \sum_{j=1}^{M} b_{jk} y_j + y_0 \geq r_k \text{ for all } k \in X, y_j \geq 0 \text{ for all } j \in S\big\}.\end{aligned}$$

Then we define constant ϵ the same as in Section A.1.1. The equivalence of Lemma A.1 is the following, and we omit the proof since it is almost identical to the proof of Lemma A.1.

Lemma A.5. *Consider problem $OPT(\theta)$ where the decision variables are restricted to a subset X and S that do not satisfy either of the conditions: 1) $X^* \subset X$ and $S^* \subset S$; 2) $\bar{X} \subset X$ and $\bar{S} \subset S$. If $|\theta_{ik}(t) - d_{ik}| \leq \epsilon$ for all $i = 1, \ldots, N$ and $k \in X$, the optimal value of $OPT(\theta)$ satisfies $f(\theta) \leq f^* - \frac{3\Delta}{4}$.*

For any period $t = 1, \ldots, T$, the definitions of the sets $X_\epsilon(t)$ and $S(t)$ remain the same. We define indicator functions $I_t^* = I(X_\epsilon(t) = X^*, S(t) = S^*)$ and $\bar{I}_t = I(X_\epsilon(t) = \bar{X}, S(t) = \bar{S})$. Let

$$T' = \min\{\tau : \sum_{t=1}^{\tau} \left(\frac{I_t^*}{\sum_{k=1}^{K} x_k^*} + \frac{\bar{I}_t}{\sum_{k=1}^{K} \bar{x}_k} + (1 - I_t^* - \bar{I}_t) \right) \geq T\}.$$

The new definition of T' represents a random variable interpreted as the time when inventory runs out. We again consider the retailer's revenue before period T' (note that $T' \leq T$ almost surely):

$$\begin{aligned}E[R(T)] \geq& E[R(T')] \\ \geq& \sum_{k=1}^{K} r_k E[N_k(T')] - E[\sum_{j=1}^{M} (\sum_{i=1}^{N} \sum_{t=1}^{T'} a_{ij} Z_i(t) - I_j)^+] \\ =& \sum_{k=1}^{K} r_k E\left[\sum_{t=1}^{T'} \frac{x_k(t)}{\sum_{k=1}^{K} x_k(t)}\right] - E[\sum_{j=1}^{M} (\sum_{i=1}^{N} \sum_{t=1}^{T'} a_{ij} Z_i(t) - I_j)^+] \\ =& \sum_{k=1}^{K} r_k E\left[\sum_{t=1}^{T'} \frac{x_k(t)}{\sum_{k=1}^{K} x_k(t)} (I_t^* + \bar{I}_t)\right] + \sum_{k=1}^{K} r_k E\left[\sum_{t=1}^{T'} \frac{x_k(t)}{\sum_{k=1}^{K} x_k(t)} (1 - \bar{I}_t - I_t^*)\right]\end{aligned}$$

$$
\begin{aligned}
&- E[\sum_{j=1}^{M}(\sum_{i=1}^{N}\sum_{t=1}^{T'} a_{ij}Z_i(t) - I_j)^+] \\
=&\sum_{k=1}^{K} r_k E\left[\sum_{t=1}^{T'}(\frac{x_k^*}{\sum_{k=1}^{K} x_k^*} I_t^* + \frac{\bar{x}_k}{\sum_{k=1}^{K}\bar{x}_k}\bar{I}_t + (1-\bar{I}_t - I_t^*))\right] \\
&- \sum_{k=1}^{K} r_k E\left[\sum_{t=1}^{T'}\left(\frac{x_k^*(t)}{\sum_{k=1}^{K} x_k^*(t)} - \frac{x_k(t)}{\sum_{k=1}^{K} x_k(t)}\right) I_t^*\right] \\
&- \sum_{k=1}^{K} r_k E\left[\sum_{t=1}^{T'}\left(\frac{\bar{x}_k}{\sum_{k=1}^{K}\bar{x}_k} - \frac{x_k(t)}{\sum_{k=1}^{K} x_k(t)}\right)\bar{I}_t)\right] \\
&+ \sum_{k=1}^{K} r_k E\left[\sum_{t=1}^{T'}(\frac{x_k(t)}{\sum_{k=1}^{K} x_k(t)} - 1)(1-\bar{I}_t - I_t^*)\right] - E[\sum_{j=1}^{M}(\sum_{i=1}^{N}\sum_{t=1}^{T'} a_{ij}Z_i(t) - I_j)^+] \\
\geq& f^*T - \sum_{k=1}^{K} r_k E\left[\sum_{t=1}^{T'}\left(\frac{x_k^*(t)}{\sum_{k=1}^{K} x_k^*(t)} - \frac{x_k(t)}{\sum_{k=1}^{K} x_k(t)}\right) I_t^*\right] \\
&- \sum_{k=1}^{K} r_k E\left[\sum_{t=1}^{T'}\left(\frac{\bar{x}_k}{\sum_{k=1}^{K}\bar{x}_k} - \frac{x_k(t)}{\sum_{k=1}^{K} x_k(t)}\right)\bar{I}_t)\right] \\
&- E[\sum_{t=1}^{T'}(1-\bar{I}_t - I_t^*)] - E[\sum_{j=1}^{M}(\sum_{i=1}^{N}\sum_{t=1}^{T'} a_{ij}Z_i(t) - I_j)^+].
\end{aligned}
$$

The last inequality uses the definition of T'. Similar to the case of having a unique optimal solution, we bound $E[\sum_{t=1}^{T'}(1-\bar{I}_t - I_t^*)]$ in Part I of this proof, the term

$$
\sum_{k=1}^{K} r_k E\left[\sum_{t=1}^{T'}\left(\frac{x_k^*(t)}{\sum_{k=1}^{K} x_k^*(t)} - \frac{x_k(t)}{\sum_{k=1}^{K} x_k(t)}\right) I_t^*\right] + \sum_{k=1}^{K} r_k E\left[\sum_{t=1}^{T'}\left(\frac{\bar{x}_k}{\sum_{k=1}^{K}\bar{x}_k} - \frac{x_k(t)}{\sum_{k=1}^{K} x_k(t)}\right)\bar{I}_t)\right]
$$

in Part II of the proof, and the last term $E[\sum_{j=1}^{M}(\sum_{i=1}^{N}\sum_{t=1}^{T'} a_{ij}Z_i(t) - I_j)^+]$ in Part III.

To prove Part I, we consider two cases: (a) $I_t^* = \bar{I}_t = 0$, and $f(\theta) \geq f^* - \Delta/2$; (b) $f(\theta) \leq f^* - \Delta/2$. The proof of Case (a) follows almost step by step, except that we now use Lemma A.5, which replaces Lemma A.1 used in the unique optimal solution case. In Case (b), we have

$$
\begin{aligned}
&\Pr(f(\theta) < f^* - \frac{\Delta}{2} \mid \mathcal{F}_{t-1}) \\
\leq&\Pr(\left\{\exists i,k \in X^* : \theta_{ik}(t) < d_{ik} - \frac{\epsilon}{4}\right\} \cap \left\{\exists i,k \in \bar{X} : \theta_{ik}(t) < d_{ik} - \frac{\epsilon}{4}\right\} \mid \mathcal{F}_{t-1})
\end{aligned}
$$

$$
\begin{aligned}
&\cdot \frac{\Pr(I_t^* = 1 \mid \mathcal{F}_{t-1}) + \Pr(\bar{I}_t = 1 \mid \mathcal{F}_{t-1}) + \Pr(I_t^* = \bar{I}_t = 0, f(\theta) \geq f^* - \frac{\Delta}{2} \mid \mathcal{F}_{t-1})}{\Pr(\{\forall i, k \in X^* : \theta_{ik}(t) \geq d_{ik} - \epsilon/4\} \cup \{\forall i, k \in \bar{X} : \theta_{ik}(t) \geq d_{ik} - \epsilon/4\} \mid \mathcal{F}_{t-1})} \\
=&\frac{\Pr(\{\exists i, k \in X^* : \theta_{ik}(t) < d_{ik} - \frac{\epsilon}{4}\} \cap \{\exists i, k \in \bar{X} : \theta_{ik}(t) < d_{ik} - \frac{\epsilon}{4}\} \mid \mathcal{F}_{t-1})}{\Pr(\{\forall i, k \in X^* : \theta_{ik}(t) \geq d_{ik} - \epsilon/4\} \cup \{\forall i, k \in \bar{X} : \theta_{ik}(t) \geq d_{ik} - \epsilon/4\} \mid \mathcal{F}_{t-1})} \\
&\cdot \Pr(I_t^* = 1 \mid \mathcal{F}_{t-1}) \\
&+ \frac{\Pr(\{\exists i, k \in X^* : \theta_{ik}(t) < d_{ik} - \frac{\epsilon}{4}\} \cap \{\exists i, k \in \bar{X} : \theta_{ik}(t) < d_{ik} - \frac{\epsilon}{4}\} \mid \mathcal{F}_{t-1})}{\Pr(\{\forall i, k \in X^* : \theta_{ik}(t) \geq d_{ik} - \epsilon/4\} \cup \{\forall i, k \in \bar{X} : \theta_{ik}(t) \geq d_{ik} - \epsilon/4\} \mid \mathcal{F}_{t-1})} \\
&\cdot \Pr(\bar{I}_t = 1 \mid \mathcal{F}_{t-1}) \\
&+ \frac{\Pr(\{\exists i, k \in X^* : \theta_{ik}(t) < d_{ik} - \frac{\epsilon}{4}\} \cap \{\exists i, k \in \bar{X} : \theta_{ik}(t) < d_{ik} - \frac{\epsilon}{4}\} \mid \mathcal{F}_{t-1})}{\Pr(\{\forall i, k \in X^* : \theta_{ik}(t) \geq d_{ik} - \epsilon/4\} \cup \{\forall i, k \in \bar{X} : \theta_{ik}(t) \geq d_{ik} - \epsilon/4\} \mid \mathcal{F}_{t-1})} \\
&\cdot \Pr(I_t^* = \bar{I}_t = 0, f(\theta) \geq f^* - \frac{\Delta}{2} \mid \mathcal{F}_{t-1}) \\
\leq&\frac{\Pr(\exists i, k \in X^* : \theta_{ik}(t) < d_{ik} - \frac{\epsilon}{4} \mid \mathcal{F}_{t-1})}{\Pr(i, k \in X^* : \theta_{ik}(t) \geq d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})} \Pr(I_t^* = 1 \mid \mathcal{F}_{t-1}) \\
&+ \frac{\Pr(\exists i, k \in \bar{X} : \theta_{ik}(t) < d_{ik} - \frac{\epsilon}{4} \mid \mathcal{F}_{t-1})}{\Pr(\forall i, k \in \bar{X} : \theta_{ik}(t) \geq d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})} \Pr(\bar{I}_t = 1 \mid \mathcal{F}_{t-1}) \\
&+ \frac{\Pr(\exists i, k \in X^* : \theta_{ik}(t) < d_{ik} - \frac{\epsilon}{4} \mid \mathcal{F}_{t-1})}{\Pr(\forall i, k \in X^* : \theta_{ik}(t) \geq d_{ik} - \epsilon/4 \mid \mathcal{F}_{t-1})} \Pr(I_t^* = \bar{I}_t = 0, f(\theta) \geq f^* - \frac{\Delta}{2} \mid \mathcal{F}_{t-1}) \\
\triangleq&(\Phi_t) + (\Gamma_t) + (\Psi_t)
\end{aligned}
$$

Therefore, we get

$$
\begin{aligned}
&E\left[\sum_{t=1}^{T} \Pr(f(\theta) < f^* - \frac{\Delta}{2} \mid \mathcal{F}_{t-1})\right] \\
\leq&E\left[\sum_{t=1}^{T} \sqrt{\Pr(f(\theta) < f^* - \frac{\Delta}{2} \mid \mathcal{F}_{t-1})}\right] \\
\leq&E\left[\sum_{t=1}^{T} \sqrt{(\Phi_t) + (\Gamma_t) + (\Psi_t)}\right] \\
\leq&E\left[\sum_{t=1}^{T} \left(\sqrt{(\Phi_t)} + \sqrt{(\Gamma_t)} + \sqrt{(\Psi_t)}\right)\right].
\end{aligned}
$$

We can then bound each of these terms as before:

$$
E[\sum_{t=1}^{T} \sqrt{(\Phi_t)}], E[\sum_{t=1}^{T} \sqrt{(\Gamma_t)}], E[\sqrt{(\Psi_t)}].
$$

This finishes the proof of Part I.

Part II also requires little change: we can establish the bound for

$$E\left[\sum_{t=1}^{T'}\left(\frac{x_k^*(t)}{\sum_{k=1}^K x_k^*(t)}-\frac{x_k(t)}{\sum_{k=1}^K x_k(t)}\right)I_t^*\right]$$

and

$$E\left[\sum_{t=1}^{T'}\left(\frac{\bar{x}_k}{\sum_{k=1}^K \bar{x}_k}-\frac{x_k(t)}{\sum_{k=1}^K x_k(t)}\right)\bar{I}_t)\right]$$

respectively, using the same proof as before.

As for Part III, for any resource $j=1,\ldots,M$, we have the following decomposition

$$\begin{aligned}
&E[\left(\sum_{i=1}^N\sum_{t=1}^{T'}a_{ij}Z_i(t)-c_jT\right)^+]\\
\leq&E[\sum_{i=1}^N\sum_{t=1}^{T'}a_{ij}(1-\bar{I}_t-I_t^*)]+E[\left(\sum_{i=1}^N\sum_{t=1}^{T'}(a_{ij}Z_i(t)-\frac{c_j}{\sum_{k=1}^K x_k^*(t)})I_t^*\right)^+]\\
&+E[\left(\sum_{i=1}^N\sum_{t=1}^{T'}(a_{ij}Z_i(t)-\frac{c_j}{\sum_{k=1}^K x_k(t))})\bar{I}_t\right)^+].
\end{aligned}$$

Then, the original proof of Part III can be applied separately to each of the three terms.

A.2 Useful Facts

Fact B.1

Let $\theta_{ik}(t)$ be the sampled demand of product i under price p_k at time t, and d_{ik} be the mean demand of product i under price p_k. Let $N_k(t-1)$ be the number of times that price p_k is offered before time t. For any $\epsilon>0$, Agrawal and Goyal (2013) (Lemma 2 and Lemma 3) show that

$$\Pr(|\theta_{ik}(t)-d_{ik}|\geq\epsilon\mid\mathcal{F}_{t-1})\leq 4e^{-\epsilon^2N_k(t-1)/2}.$$

In particular, if $N_k(t-1)\geq 2\log T/\epsilon^2$, we have

$$\Pr(|\theta_{ik}(t)-d_{ik}|\geq\epsilon\mid\mathcal{F}_{t-1})\leq\frac{4}{T}.$$

Proof. This proof is slightly modified from Agrawal and Goyal (2013). We can derive the above inequalities using the fact that the $(i+1)$th order statistic of $n+1$ uniformly

distributed variables is distributed as $Beta(i+1, n-i+1)$. Therefore, we have

$$\begin{aligned}
&\Pr(|\theta_{ik}(t) - d_{ik}| \geq \epsilon \mid \mathcal{F}_{t-1}) \\
=&\Pr(\theta_{ik}(t) \geq d_{ik} + \epsilon \mid \mathcal{F}_{t-1}) + \Pr(\theta_{ik}(t) \leq d_{ik} - \epsilon \mid \mathcal{F}_{t-1}) \\
=&\Pr(\sum_{i=1}^{N_k(t-1)+1} X_i \leq \hat{d}_{ik}(t) N_k(t-1) \mid \mathcal{F}_{t-1}) + \Pr(\sum_{i=1}^{N_k(t-1)+1} Y_i \geq \hat{d}_{ik}(t) N_k(t-1) + 1 \mid \mathcal{F}_{t-1}),
\end{aligned}$$

where X_i's are i.i.d. Bernoulli random variables with mean $d_{ik} + \epsilon$, Y_i's are i.i.d. Bernoulli random variables with mean $d_{ik} - \epsilon$, and $\hat{d}_{ik}(t)$ is the empirical mean demand of product i under price p_k for the first $t-1$ periods. By Hoeffding's inequality, we have

$$\Pr(|\hat{d}_{ik}(t) - d_{ik}| \geq \frac{\epsilon}{2} \mid \mathcal{F}_{t-1}) \leq 2e^{-\epsilon^2 N_k(t-1)/2}.$$

So

$$\begin{aligned}
&\Pr(\sum_{i=1}^{N_k(t-1)+1} X_i \leq \hat{d}_{ik}(t) N_k(t-1) \mid \mathcal{F}_{t-1}) + \Pr(\sum_{i=1}^{N_k(t-1)+1} Y_i \geq \hat{d}_{ik}(t) N_k(t-1) + 1 \mid \mathcal{F}_{t-1}) \\
\leq&\Pr(\sum_{i=1}^{N_k(t-1)+1} X_i \leq (d_{ik} + \frac{\epsilon}{2}) N_k(t-1) \mid \mathcal{F}_{t-1}) \\
&+ \Pr(\sum_{i=1}^{N_k(t-1)+1} Y_i \geq (d_{ik} - \frac{\epsilon}{2}) N_k(t-1) + 1 \mid \mathcal{F}_{t-1}) + 2e^{-\epsilon^2 N_k(t-1)/2} \\
\leq&\Pr(\sum_{i=1}^{N_k(t-1)} X_i \leq (d_{ik} + \frac{\epsilon}{2}) N_k(t-1) \mid \mathcal{F}_{t-1}) + \Pr(\sum_{i=1}^{N_k(t-1)} Y_i \geq (d_{ik} - \frac{\epsilon}{2}) N_k(t-1) \mid \mathcal{F}_{t-1}) \\
&+ 2e^{-\epsilon^2 N_k(t-1)/2} \\
\leq&e^{-\epsilon^2 N_k(t-1)/2} + e^{-\epsilon^2 N_k(t-1)/2} + 2e^{-\epsilon^2 N_k(t-1)/2} = 4e^{-\epsilon^2 N_k(t-1)/2}.
\end{aligned}$$

The last inequality again uses the Hoeffding bound.

Fact B.2

Let $\theta_{ik}(t)$ be the sampled demand of product i under price p_k at time t, and let d_{ik} be the mean demand of product i under price p_k. Let $N_k(t-1)$ be the number of times that price

p_k is offered before time t. For any $\epsilon > 0$, Agrawal and Goyal (2013) (Lemma 4) show that

$$E[\frac{1}{\Pr(\theta_{ik}(t) \geq d_{ik} - \epsilon \mid \mathcal{F}_{t-1})} \mid N_k(t-1)] \leq 1 + \frac{3}{\epsilon}.$$

Furthermore, if $N_k(t-1) \geq 8/\epsilon$, it holds that

$$E[\frac{1}{\Pr(\theta_{ik}(t) \geq d_{ik} - \epsilon \mid \mathcal{F}_{t-1})} \mid N_k(t-1)] \leq 1 + O(\frac{1}{\epsilon^2 N_k(t-1)}).$$

(In fact, the bound proved by Agrawal and Goyal (2013) is tighter than $O(\frac{1}{\epsilon^2 N_k(t-1)})$, but we only need the looser bound above in the proof.)

Fact B.3

We use two forms of the Cauchy-Schwartz inequality in the proof. Suppose that $a_i \geq 0$ and $b_i \geq 0$ for all $i = 1, \ldots, n$, we have

$$\sqrt{\sum_{i=1}^{n} a_i}\sqrt{\sum_{i=1}^{n} b_i} \geq \sum_{i=1}^{n} \sqrt{a_i b_i}.$$

Suppose that $A \geq 0$ and $B \geq 0$ almost surely, we have

$$\sqrt{E[A]}\sqrt{E[B]} \geq E[\sqrt{AB}].$$

Fact B.4

Given constants $c > 0$, $\epsilon \geq 0$ and $\sigma \geq 0$, suppose C_i are i.i.d random variables with mean $c + \epsilon$ and variance σ^2 for all $i = 1, \ldots, t$. Gallego and Van Ryzin (1994) uses the following inequality (in the proof of Theorem 3):

$$E[(\sum_{i=1}^{t} U_i - ct)^+] \leq \frac{\sqrt{\sigma^2 t + (\epsilon t)^2} + \epsilon t}{2}.$$

In fact, the following looser bound is enough for our proof:

$$E[(\sum_{i=1}^{t} U_i - ct)^+] \leq E[|\sum_{i=1}^{t} U_i - ct|] \leq \sqrt{E[(\sum_{i=1}^{t} U_i - ct)^2]} = \sqrt{\sigma^2 t + (\epsilon t)^2}.$$

Appendix B

Technical Results for Chapter 3

B.1 Proofs of the Results in Section 3.3

B.1.1 Proof of Theorem 3.3

Proof. Proof of Theorem 3.3. Suppose d_1 is the underlying demand function. The regret under demand d_1 can be decomposed as

$$\text{Regret}_1^{\mathsf{mPC}}(T) = \mathbb{E}_1\left[\sum_{t=1}^{T}(r_1^* - r_1(P_t))\right] = \sum_{\ell=0}^{m}\mathbb{E}_1\left[\sum_{t=\tau_\ell+1}^{\tau_{\ell+1}}(r_1^* - r_1(P_t))\right].$$

We first consider the case where $\log^{(m)} T > 0$. By definition, $\tau_1 = \left\lceil M_\Phi(P_0^*)\log^{(m)} T\right\rceil$, so the regret during Phase 0 is equal to

$$\mathbb{E}_1\left[\sum_{t=1}^{\tau_1}(r_1^* - r_1(P_t))\right] = \left\lceil M_\Phi(P_0^*)\log^{(m)} T\right\rceil(r_1^* - r_1(P_0^*)). \tag{B.1}$$

Next, we show that for each $1 \le \ell \le m$, the regret during Phase ℓ is bounded by

$$\mathbb{E}_1\left[\sum_{t=\tau_\ell+1}^{\tau_{\ell+1}}(r_1^* - r_1(P_t))\right] \le \frac{2M_\Phi^* r_1^*}{\log^{(m-\ell)} T} + \frac{2r_1^*}{(\log^{(m-\ell)} T)^2}, \tag{B.2}$$

where $M_\Phi^* = \max_{i\in\{1,\cdots,K\}} M_\Phi(p_i^*)$.

For $1 \le \ell \le m$, the regret during Phase ℓ satisfies the following bound:

$$\begin{aligned}
&\mathbb{E}_1\left[\sum_{t=\tau_\ell+1}^{\tau_{\ell+1}} (r_1^* - r_1(P_t))\right] \\
=&\mathbb{E}_1\left[(\tau_{\ell+1} - \tau_\ell) \times (r_1^* - r_1(P_\ell^*))\right] \\
\le&\mathbb{E}_1\left[\left(M_\Phi(P_\ell^*)\log^{(m-\ell)} T + 1\right) \times (r_1^* - r_1(P_\ell^*))\right] \\
\le&\left(M_\Phi^*\log^{(m-\ell)} T + 1\right)\sum_{i=1}^{K}(r_1^* - r_1(p_i^*)) \times \mathbb{P}_1(P_\ell^* = p_i^*) \qquad \text{(B.3)} \\
\le&\left(M_\Phi^*\log^{(m-\ell)} T + 1\right) r_1^* \times \sum_{i=2}^{K}\mathbb{P}_1(P_\ell^* = p_i^*).
\end{aligned}$$

In the expression above, the expectation is taken on the price offered in Phase ℓ, P_ℓ^*, which is a random variable depending on the realized demand in phases $0, \cdots, \ell-1$. In equation (B.3), we use the fact that for all $\ell = 1, \cdots, m$, the offered price $P_\ell^* \in \{p_1^*, \cdots, p_K^*\}$ (see line 10 of mPC).

To complete the proof of inequality (B.2), we prove the following inequality:

$$\sum_{i=2}^{K}\mathbb{P}_1(P_\ell^* = p_i^*) \le \frac{2}{(\log^{(m-\ell)} T)^2}. \qquad \text{(B.4)}$$

By the definition of mPC, the choice of price P_ℓ^* is determined by the sample mean $\bar{X}_{\ell-1}$ in Phase $\ell-1$, so we have

$$\sum_{i=2}^{K}\mathbb{P}_1(P_\ell^* = p_i^*) = \mathbb{P}_1\left(|\bar{X}^{\ell-1} - d_1(P_{\ell-1}^*)| \ge |\bar{X}^{\ell-1} - d_i(P_{\ell-1}^*)| \text{ for some } i \ne 1\right).$$

Now, if $|\bar{X}^{\ell-1} - d_1(P_{\ell-1}^*)| \ge |\bar{X}^{\ell-1} - d_i(P_{\ell-1}^*)|$ for some $i \ne 1$, we have

$$|\bar{X}^{\ell-1} - d_1(P_{\ell-1}^*)| \ge \frac{1}{2}\left(|\bar{X}^{\ell-1} - d_i(P_{\ell-1}^*)| + |\bar{X}^{\ell-1} - d_1(P_{\ell-1}^*)|\right) \ge \frac{1}{2}|d_i(P_{\ell-1}^*) - d_1(P_{\ell-1}^*)|,$$

where the last step uses the triangle inequality. This leads to the following bound:

$$\sum_{i=2}^{K}\mathbb{P}_1(P_\ell^* = p_i^*) \le \mathbb{P}_1\left(\left|\bar{X}^{\ell-1} - d_1(P_{\ell-1}^*)\right| \ge \frac{1}{2}\min_{i\ne 1}\left|d_1(P_{\ell-1}^*) - d_i(P_{\ell-1}^*)\right|\right). \qquad \text{(B.5)}$$

Given price $P_{\ell-1}^*$, sample mean $\bar{X}_{\ell-1}$ is the average of i.i.d. random variables with mean

$d_1(P_{\ell-1})$. Because demand in each period is lighted-tailed with parameters (σ, b), we can apply the Chernoff inequality: conditioning on $P^*_{\ell-1}$, for any $\epsilon > 0$, it holds that

$$\mathbb{P}_1\left(|\bar{X}^{\ell-1} - d_1(P^*_{\ell-1})| \geq \epsilon \middle| P^*_{\ell-1}\right) \leq 2\exp\left(-(\tau_\ell - \tau_{\ell-1})(\frac{\epsilon^2}{2\sigma^2} \wedge \frac{\epsilon}{2b})\right).$$

Let $\epsilon = \frac{1}{2}\min_{i\neq 1}\left|d_1(P^*_{\ell-1}) - d_i(P^*_{\ell-1})\right|$. Because $\tau_\ell - \tau_{\ell-1} = \left\lceil M_\Phi(P^*_{\ell-1})\log^{(m-\ell+1)} T\right\rceil$, we have

$$\begin{aligned}
&\mathbb{P}_1\left(\left|\bar{X}^{\ell-1} - d_1(P^*_{\ell-1})\right| \geq \frac{1}{2}\min_{i\neq 1}\left|d_1(P^*_{\ell-1}) - d_i(P^*_{\ell-1})\right| \middle| P^*_{\ell-1}\right) \\
\leq& 2\mathbb{E}_1\left[\exp\left(-\left\lceil M_\Phi(P^*_{\ell-1})\log^{(m-\ell+1)} T\right\rceil(\frac{\epsilon^2}{2\sigma^2} \wedge \frac{\epsilon}{2b})\right) \middle| P^*_{\ell-1}\right] \\
=& 2\exp\left(-\left\lceil M_\Phi(P^*_{\ell-1})\log^{(m-\ell+1)} T\right\rceil(\frac{\epsilon^2}{2\sigma^2} \wedge \frac{\epsilon}{2b})\right) \\
\leq& 2\exp\left(-M_\Phi(P^*_{\ell-1})\log^{(m-\ell+1)} T(\frac{\epsilon^2}{2\sigma^2} \wedge \frac{\epsilon}{2b})\right) \\
\leq& 2\exp\left(-2\log^{(m-\ell+1)} T\right) \qquad \text{(B.6)} \\
=& \frac{2}{\left(\log^{(m-\ell)} T\right)^2},
\end{aligned}$$

where step (B.6) uses the definition

$$M_\Phi(P^*_{\ell-1}) = 2 \times \left(\frac{2\sigma^2}{\frac{1}{4}\min_{i\neq j}\left(d_i(P^*_{\ell-1}) - d_j(P^*_{\ell-1})\right)^2} \vee \frac{2b}{\frac{1}{2}\min_{i\neq j}\left|d_i(P^*_{\ell-1}) - d_j(P^*_{\ell-1})\right|}\right).$$

By integrating over the realizations of $P^*_{\ell-1}$ in the above bound, we have established inequality (B.4), which in turn proves (B.2).

Combining equations (B.1) and (B.2), we can prove the regret bound on mPC under demand d_1 as follows:

$$\begin{aligned}
&\text{Regret}_1^{\text{mPC}}(T) = \mathbb{E}_1\left[\sum_{t=1}^{\tau_1}(r_1^* - r_1(P_t))\right] + \sum_{\ell=1}^{m}\mathbb{E}_1\left[\sum_{t=\tau_\ell+1}^{\tau_{\ell+1}}(r_1^* - r_1(P_t))\right] \\
\leq& \left(M_\Phi(P_0^*)\log^{(m)} T + 1\right)(r_1^* - r_1(P_0^*)) + \sum_{\ell=1}^{m}\left(\frac{2M_\Phi^* r_1^*}{\log^{(m-\ell)} T} + \frac{2r_1^*}{(\log^{(m-\ell)} T)^2}\right).
\end{aligned}$$

Since $\log^{(m)} T > 0$, it is easily verified that $\log^{(m-\ell)} T \geq e^{\ell-1}$ for all $\ell \geq 1$, so

$$\sum_{\ell=1}^{m} \frac{1}{\log^{(m-\ell)} T} \leq \sum_{\ell=1}^{\infty} \frac{1}{e^{\ell-1}} \leq 2, \quad \sum_{\ell=1}^{m} \frac{1}{(\log^{(m-\ell)} T)^2} \leq \sum_{\ell=1}^{\infty} \frac{1}{e^{2\ell-2}} \leq \frac{3}{2}.$$

Therefore,

$$\begin{aligned}\mathrm{Regret}_1^{\mathsf{mPC}}(T) \leq & \left(M_\Phi(P_0^*) \log^{(m)} T + 1\right)(r_1^* - r_1(P_0^*)) + 4M_\Phi^* r_1^* + 3r_1^* \\ \leq & M_\Phi(P_0^*)(r_1^* - r_1(P_0^*)) \log^{(m)} T + 4M_\Phi^* r_1^* + 4r_1^*.\end{aligned}$$

The minimax regret of demand set Φ is bounded by

$$\mathrm{Regret}_\Phi^{\mathsf{mPC}}(T) = \max_{i=1,\ldots,K} \mathrm{Regret}_i^{\mathsf{mPC}}(T) \leq C_\Phi(P_0^*) \log^{(m)} T + 4M_\Phi^* r^* + 4r^*,$$

where $C_\Phi(P_0^*) = \max_{i\in\{1,\cdots,K\}}\{M_\Phi(P_0^*)(r_i^* - r_i(P_0^*))\}$ and $r^* = \max_{i\in\{1,\cdots,K\}} r_i^*$.

If $\log^{(m)} T = 0$, let $m' \leq m$ be the largest integer such that $\log^{(m')} T > 0$. Clearly, $\log^{(m')} T \leq 1$. In this case, policy mPC applied to T periods uses only m' price changes, so

$$\mathrm{Regret}_\Phi^{\mathsf{mPC}}(T) \leq C_\Phi(P_0^*) \log^{(m')} T + 4M_\Phi^* r^* + 4r^* \leq C_\Phi(P_0^*) + 4M_\Phi^* r^* + 4r^*.$$

Combining both cases for $\log^{(m)} T > 0$ and $\log^{(m)} T = 0$, we have

$$\mathrm{Regret}_\Phi^{\mathsf{mPC}}(T) \leq C_\Phi(P_0^*) \max\{\log^{(m)} T, 1\} + 4M_\Phi^* r^* + 4r^*.$$

□

B.1.2 Proof of Lemma 3.7

Proof. Proof of Lemma 3.7.

Let $h_t = (p_1, x_1, \ldots, p_t, x_t)$ be a realization of $H_t = (P_1, X_1, \cdots, P_t, X_t)$. We first assume $\mathbb{P}_i^\pi(H_t = h_t) > 0$, so we have

$$\mathbb{P}_i^\pi(H_t = h_t) = \prod_{s=1}^{t} \mathbb{P}_i^\pi(D(p_s) = x_s) \prod_{s=1}^{t-1} \mathbb{P}_i^\pi(P_{s+1} = p_{s+1} \mid H_s = h_s)$$

$$= \prod_{s=1}^{t} \left(\mathbb{P}_{i'}^{\pi}(D(p_s) = x_s) \cdot \frac{\mathbb{P}_{i}^{\pi}(D(p_s) = x_s)}{\mathbb{P}_{i'}^{\pi}(D(p_s) = x_s)} \right) \prod_{s=1}^{t-1} \mathbb{P}_{i}^{\pi}(P_{s+1} = p_{s+1} \mid H_s = h_s) \tag{B.7}$$

$$\geq \prod_{s=1}^{t} (\mathbb{P}_{i'}^{\pi}(D(p_s) = x_s) \cdot \kappa_\Gamma) \prod_{s=1}^{t-1} \mathbb{P}_{i}^{\pi}(P_{s+1} = p_{s+1} \mid H_s = h_s) \tag{B.8}$$

$$= \kappa_\Gamma^t \prod_{s=1}^{t} \mathbb{P}_{i'}^{\pi}(D(p_s) = x_s) \prod_{s=1}^{t-1} \mathbb{P}_{i}^{\pi}(P_{s+1} = p_{s+1} \mid H_s = h_s)$$

$$= \kappa_\Gamma^t \prod_{s=1}^{t} \mathbb{P}_{i'}^{\pi}(D(p_s) = x_s) \prod_{s=1}^{t-1} \mathbb{P}_{i'}^{\pi}(P_{s+1} = p_{s+1} \mid H_s = h_s) \tag{B.9}$$

$$= \kappa_\Gamma^t \mathbb{P}_{i'}^{\pi}(H_t = h_t).$$

Step (B.7) uses the third condition of (Γ), which states that all demand functions have the same support under a given price, so $\mathbb{P}_{i'}^{\pi}(D(p_s) = x_s) \neq 0$. Step (B.8) uses the fourth condition of (Γ). Step (B.9) holds because price P_{s+1} is determined by policy π and realized history h_s, and is independent of the underlying demand model. Note that if π is a deterministic policy, we always have $\mathbb{P}_{i}^{\pi}(P_{s+1} = p_{s+1} \mid H_s = h_s) = 1$ for all i.

Finally, if $\mathbb{P}_{i}^{\pi}(H_t = h_t) = 0$, we have $\mathbb{P}_{i'}^{\pi}(H_t = h_t) = 0$, too. This is again due to the third condition of (Γ), which states that all demand functions have the same support under a given price. □

B.1.3 Proof of Proposition 3.9

Proof. Proof of Proposition 3.9. Let m be the integer such that $\tau_m < T \leq \tau_{m+1}$. Suppose d_1 is the underlying demand function. The regret under demand d_1 can be composed as

$$\text{Regret}_1^{\text{uPC}}(T) = \mathbb{E}_1 \left[\sum_{t=1}^{T} (r_1^* - r_1(P_t)) \right] \leq \sum_{\ell=0}^{m} \mathbb{E}_1 \left[\sum_{t=\tau_\ell+1}^{\tau_{\ell+1}} (r_1^* - r_1(P_t)) \right].$$

The regret during Phase 0 is equal to

$$\mathbb{E}_1 \left[\sum_{t=1}^{\tau_1} (r_1^* - r_1(P_t)) \right] = \lceil M_\Phi(P_0^*) \rceil (r_1^* - r_1(P_0^*)).$$

For $1 \leq \ell \leq m$, the offered price $P_\ell^* \in \{p_1^*, \cdots, p_K^*\}$, so the regret during Phase ℓ is

bounded by:

$$\begin{aligned}
&\mathbb{E}_1\left[\sum_{t=\tau_\ell+1}^{\tau_{\ell+1}}(r_1^*-r_1(P_t))\right]\\
=&\mathbb{E}_1\left[(\tau_{\ell+1}-\tau_\ell)\times(r_1^*-r_1(P_\ell^*))\right]\\
\leq&\mathbb{E}_1\left[\left(M_\Phi(P_\ell^*)e^{(\ell)}+1\right)\times(r_1^*-r_1(P_\ell^*))\right]\\
\leq&\left(M_\Phi^* e^{(\ell)}+1\right)\sum_{i=1}^{K}(r_1^*-r_1(p_i^*))\times\mathbb{P}_1(P_\ell^*=p_i^*)\\
\leq&\left(M_\Phi^* e^{(\ell)}+1\right)r_1^*\times\sum_{i=2}^{K}\mathbb{P}_1(P_\ell^*=p_i^*).
\end{aligned}$$

By the definition of the uPC policy, the choice of price P_ℓ^* is determined by the sample mean $\bar{X}_{\ell-1}$ in Phase $\ell-1$. Similar to the proof of Theorem 3.3, letting

$$\epsilon=\frac{1}{2}\min_{i\neq 1}\left|d_1(P_{\ell-1}^*)-d_i(P_{\ell-1}^*)\right|,$$

we have

$$\begin{aligned}
&\sum_{i=2}^{K}\mathbb{P}_1(P_\ell^*=p_i^*)\\
\leq&\mathbb{P}_1\left(\left|\bar{X}^{\ell-1}-d_1(P_{\ell-1}^*)\right|\geq\epsilon\right)\\
=&\mathbb{E}_1\left[\mathbb{P}_1\left(\left|\bar{X}^{\ell-1}-d_1(P_{\ell-1}^*)\right|\geq\epsilon\middle|P_{\ell-1}^*\right)\right]\\
\leq&\mathbb{E}_1\left[2\mathbb{E}_1\left[\exp\left(-(\tau_\ell-\tau_{\ell-1})(\frac{\epsilon^2}{2\sigma^2}\wedge\frac{\epsilon}{2b})\right)\middle|P_{\ell-1}^*\right]\right]\\
\leq&\mathbb{E}_1\left[2\mathbb{E}_1\left[\exp\left(-M_\Phi(P_{\ell-1}^*)e^{(\ell-1)}(\frac{\epsilon^2}{2\sigma^2}\wedge\frac{\epsilon}{2b})\right)\middle|P_{\ell-1}^*\right]\right]\\
\leq&\mathbb{E}_1\left[2\mathbb{E}_1\left[\exp\left(-2e^{(\ell-1)}\right)\middle|P_{\ell-1}^*\right]\right]\\
=&2/(e^{(\ell)})^2.
\end{aligned}$$

So

$$\mathbb{E}_1\left[\sum_{t=\tau_\ell+1}^{\tau_{\ell+1}}(r_1^*-r_1(P_t))\right]\leq\left(M_\Phi^* e^{(\ell)}+1\right)r_1^*\cdot\frac{2}{(e^{(\ell)})^2}=\frac{2M_\Phi^* r_1^*}{e^{(\ell)}}+\frac{2r_1^*}{(e^{(\ell)})^2}.$$

In sum, the regret of uPC under demand d_1 is bounded by

$$\begin{aligned}
&\text{Regret}_1^{\text{uPC}}(T) = \mathbb{E}_1\left[\sum_{t=1}^{\tau_1}(r_1^* - r_1(P_t))\right] + \sum_{\ell=1}^{m}\mathbb{E}_1\left[\sum_{t=\tau_\ell+1}^{\tau_{\ell+1}}(r_1^* - r_1(P_t))\right] \\
\leq &(M_\Phi(P_0^*)+1)(r_1^* - r_1(P_0^*)) + \sum_{\ell=1}^{m}\left(\frac{2M_\Phi^* r_1^*}{e^{(\ell)}} + \frac{2r_1^*}{(e^{(\ell)})^2}\right) \\
\leq &M_\Phi(P_0^*)(r_1^* - r_1(P_0^*)) + r_1^* + (2M_\Phi^* r_1^* + r_1^*) \\
= &M_\Phi(P_0^*)(r_1^* - r_1(P_0^*)) + 2M_\Phi^* r_1^* + 2r_1^*.
\end{aligned}$$

The minimax regret of demand set Φ is given by

$$\text{Regret}_\Phi^{\text{uPC}}(T) = \max_{i=1,\ldots,K}\text{Regret}_i^{\text{uPC}}(T) \leq C_\Phi(P_0^*) + 2M_\Phi^* r^* + 2r^*,$$

where $C_\Phi(P_0^*) = \max_{i\in\{1,\cdots,K\}}\{M_\Phi(P_0^*)(r_i^* - r_i(P_0^*))\}$ and $r^* = \max_{i\in\{1,\cdots,K\}} r_i^*$.

□

B.1.4 Proof of Proposition 3.10

Proof. Proof of Proposition 3.10.

Consider a price set $\mathcal{P} = \{1, 2\}$ and two demand functions

$$d_1(1) = 0.6, d_1(2) = 0.25; d_2(1) = 0.4, d_2(2) = 0.25.$$

Demand per period has a Bernoulli distribution. It is clear that the optimal prices are $p_1^* = 1, p_2^* = 2$. This demand model violates Assumption 3.1, because $p_2^* = 2$ is not a discriminative price. We show that for this model, any non-anticipating policy must have a regret of $\Omega(\log T)$.

The one period regret for not using the optimal price is $a = 0.1$ under either demand function. For any policy, we let T_1 be the number of the times that $p = 1$ is used.

We prove the result by contradiction. Suppose $\text{Regret}_2(T) = a \cdot \mathbb{E}_2[T_1] = o(1)\cdot\log T$ and $\text{Regret}_1(T) = a(\mathbb{E}_1[T - T_1]) = o(1)\cdot\log T$. The change-of-measure inequality (see proof of Lemma 3.7) implies that for any event A,

$$\mathbb{P}_2(A) \leq \mathbb{E}_1[1_A \exp(bT_1)].$$

where $b = \log(0.6/0.4)$.

Consider the event: $A = \{T_1 \leq \log T/(2b)\}$, then we have

$$\mathbb{P}_2(A) \leq \mathbb{P}_1(A)\exp(b \cdot \log T/(2b)) = \mathbb{P}_1(A)\sqrt{T}.$$

By Markov's inequality,

$$\mathbb{P}_1(A) = \mathbb{P}_1(T - T_1 \geq T - \log T/(2b)) \leq \frac{\mathbb{E}_1[T - T_1]}{T - \log T/(2b)} = \frac{o(1)\log T}{T - \log T/(2b)}.$$

Thus, we have

$$\mathbb{P}_2(A) \leq \frac{o(1)\sqrt{T}\log T}{T - \log T/(2b)} = o(1).$$

Using Markov's inequality again, we get

$$\mathbb{E}_2[T_1] \geq \frac{\log T}{2b}\mathbb{P}_2(T_1 \geq \frac{\log T}{2b}) = \frac{\log T}{2b}(1 - \mathbb{P}_2(A)) = \frac{\log T}{2b}(1 - o(1)).$$

This contradicts the assumption that $\mathbb{E}_2[T_1] = o(1) \cdot \log T$. □

B.1.5 Proof of Proposition 3.12

Proof. Proof of Proposition 3.12. Suppose d_1 is the underlying demand function. Let $k \leq K - 1$ be the number of iterations in the while loop.

The regret under demand d_1 can be composed as

$$\text{Regret}_1^{\mathsf{kPC}}(T) = \mathbb{E}_1\left[\sum_{t=1}^{T}(r_1^* - r_1(P_t))\right] \leq \mathbb{E}_1\left[\sum_{\ell=0}^{k}\sum_{t=\tau_\ell+1}^{\tau_{\ell+1}}(r_1^* - r_1(P_t))\right].$$

Let $\epsilon = \frac{1}{2}\min_{i:d_1(P^*_{\ell-1})\neq d_i(P^*_{\ell-1})}\left|d_1(P^*_{\ell-1}) - d_i(P^*_{\ell-1})\right|$. The probability that demand d_1 is eliminated in phase $\ell < k$ is bounded by

$$\begin{aligned}
&\mathbb{P}_1\left(|\bar{X}^{\ell-1} - d_1(P^*_{\ell-1})| \geq |\bar{X}^{\ell-1} - d_i(P^*_{\ell-1})| \text{ for some } i \neq 1\right)\\
\leq&\mathbb{P}_1\left(\left|\bar{X}^{\ell-1} - d_1(P^*_{\ell-1})\right| \geq \epsilon\right) &\text{(B.10)}\\
=&\mathbb{E}_1\left[\mathbb{P}_1\left(\left|\bar{X}^{\ell-1} - d_1(P^*_{\ell-1})\right| \geq \epsilon \middle| P^*_{\ell-1}\right)\right]\\
\leq&\mathbb{E}_1\left[2\mathbb{E}_1\left[\exp\left(-(\tau_\ell - \tau_{\ell-1})(\frac{\epsilon^2}{2\sigma^2} \wedge \frac{\epsilon}{2b})\right)\middle| P^*_{\ell-1}\right]\right] &\text{(B.11)}
\end{aligned}$$

$$
\begin{aligned}
&\leq \mathbb{E}_1\left[2\mathbb{E}_1\left[\exp\left(-\tilde{M}_\Phi(P^*_{\ell-1})\log T\left(\frac{\epsilon^2}{2\sigma^2}\wedge\frac{\epsilon}{2b}\right)\right)\middle| P^*_{\ell-1}\right]\right] \\
&\leq \mathbb{E}_1\left[2\mathbb{E}_1\left[\exp\left(-\log T\right)\middle| P^*_{\ell-1}\right]\right] \\
&= 2/T.
\end{aligned}
$$

Inequality (B.10) is proved in Theorem 3.3, and (B.11) uses the Chernoff bound. Since $k \leq K-1$, we have

$$
\mathbb{P}_1\left(|\bar{X}^{\ell-1} - d_1(P^*_{\ell-1})| \geq |\bar{X}^{\ell-1} - d_i(P^*_{\ell-1})| \text{ for some } i \neq 1, 0 \leq \ell < k\right) \leq \frac{2(K-1)}{T}.
$$

For each of the learning phase ($0 \leq \ell \leq k-1$), the regret is bounded by

$$
\mathbb{E}_1\left[\sum_{t=\tau_\ell+1}^{\tau_{\ell+1}} (r_1^* - r_1(P_t))\right] = \mathbb{E}_1\left[\left[\tilde{M}_A(P^*_\ell)\log T\right](r_1^* - r_1(P^*_\ell))\right] \leq \tilde{M}_\Phi r_1^* \log T + r_1^*.
$$

The regret in the earning phase ($\ell = k$) is bounded by

$$
\mathbb{E}_1\left[\sum_{t=\tau_k+1}^{T} (r_1^* - r_1(P_t))\right] \leq T r_1^* \mathbb{P}_1(P_k \neq p_i^*).
$$

So the regret of kPC under demand d_1 is bounded by

$$
\begin{aligned}
&\text{Regret}_1^{\text{kPC}}(T) = \mathbb{E}_1\left[\sum_{\ell=0}^{k-1}\sum_{t=\tau_\ell+1}^{\tau_{\ell+1}} (r_1^* - r_1(P_t))\right] + \mathbb{E}_1\left[\sum_{t=\tau_k+1}^{T} (r_1^* - r_1(P_t))\right] \\
&\leq (K-1)(\tilde{M}_\Phi r_1^* \log T + r_1^*) + T r_1^* \frac{2(K-1)}{T} \\
&= (K-1)\tilde{M}_\Phi r_1^* \log T + 3(K-1) r_1^*.
\end{aligned}
$$

The minimax regret of demand set Φ is given by

$$
\text{Regret}_\Phi^{\text{kPC}}(T) = \max_{i=1,\ldots,K} \text{Regret}_i^{\text{kPC}}(T) \leq (K-1)\tilde{M}_\Phi r^* \log T + 3(K-1) r^*.
$$

□

Appendix C

Technical Results for Chapter 4

C.1 Proofs

C.1.1 Proof of Lemma 4.1

Proof. Proof of Lemma 4.1. For any $\mathbf{s}^1$, $\mathbf{c}^1$, $\mathbf{d}^1$ and $\mathbf{s}^2$, $\mathbf{c}^2$, $\mathbf{d}^2$. Let $\mathbf{x}^1$, $\mathbf{l}^1$ and $\mathbf{x}^2$, $\mathbf{l}^2$ be the optimal solutions for the optimization problems defining $\Pi(\mathcal{F}, \mathbf{s}^1, \mathbf{c}^1, \mathbf{d}^1)$ and $\Pi(\mathcal{F}, \mathbf{s}^2, \mathbf{c}^2, \mathbf{d}^2)$. For any $0 \leq \lambda \leq 1$, clearly $\lambda\mathbf{x}^1 + (1-\lambda)\mathbf{x}^2$, $\lambda\mathbf{l}^1 + (1-\lambda)\mathbf{l}^2$ is feasible for the optimization problem defining $\Pi(\mathcal{F}, \lambda\mathbf{s}^1 + (1-\lambda)\mathbf{s}^2, \lambda\mathbf{c}^1 + (1-\lambda)\mathbf{c}^2, \lambda\mathbf{d}^1 + (1-\lambda)\mathbf{d}^2)$ and therefore, we have

$$\begin{aligned} &\Pi(\mathcal{F}, \lambda\mathbf{s}^1 + (1-\lambda)\mathbf{s}^2, \lambda\mathbf{c}^1 + (1-\lambda)\mathbf{c}^2, \lambda\mathbf{d}^1 + (1-\lambda)\mathbf{d}^2) \\ &\leq \lambda\Pi(\mathcal{F}, \mathbf{s}^1, \mathbf{d}^1, \mathbf{d}^1) + (1-\lambda)\Pi(\mathcal{F}, \mathbf{s}^2, \mathbf{d}^2, \mathbf{d}^2). \end{aligned}$$

□

C.1.2 Proof of Lemma 4.2

Proof. Proof of Lemma 4.2. First, note that we can view the LP formulation of $\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d})$ as a network flow problem: Consider a network consisting of all the plant and product nodes, a source node, and a sink node. Create arcs

- from the source node to each plant node i with upper bound c_i
- from the source node to each product node j with lower bound 0

- from each plant node i to each product node j with lower bound 0, if (i,j) is in the flexibility design $\mathcal{F}$

- from each product node j to the sink node, with lower bound $(d_j - s_j)^+$.

Suppose the arc from the source node to product node j has one unit of cost, and other arcs have zero cost. Let x_{ij} be the flow on arc from plant node i to product node j, and l_j be the flow on arc from the source node to product node j. Then this minimum cost flow problem is exactly the primal formulation of $\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d})$.

Using the strong duality theorem, we can express $\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d})$ by the following dual formulation.

$$\begin{aligned} \Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d}) = \max & \sum_{j=1}^{N} (d_j - s_j) q_j - \sum_{i=1}^{M} c_i p_i \qquad \text{(C.1)} \\ q_j - p_i \leq 0, & \quad \forall (\mathcal{S}_i, \mathcal{T}_j) \in \mathcal{F} \\ q_j \leq 1, & \quad \forall 1 \leq j \leq N \\ p_i,\ q_j \geq 0, & \forall 1 \leq i \leq M, \forall 1 \leq j \leq N. \end{aligned}$$

The constraints in the dual formulation above are totally unimodular. This is an immediate result as the primal formulation of $\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d})$ is a network flow problem.

□

C.1.3 Proof of Proposition 4.3

Proof. Proof of Proposition 4.3. First, observe that for any nonnegative vector $\mathbf{c}$, if $\mathbf{p}$, $\mathbf{q}$ is an optimal solution, we can without loss of generality assume that $p_i = \max\{q_j \mid (\mathcal{S}_i, \mathcal{T}_j) \in \mathcal{F}\}$. Combining this observation with Lemma 4.2, we have

$$\begin{aligned} \Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d}) = \max & \sum_{j=1}^{N} (d_j - s_j) q_j - \sum_{i=1}^{M} c_i p_i \\ p_i = \max & \{q_j \mid (\mathcal{S}_i, \mathcal{T}_j) \in \mathcal{F}\}, \\ \mathbf{p} \in \{0,1\}^M, & \ \mathbf{q} \in \{0,1\}^N, \forall 1 \leq i \leq M, \forall 1 \leq j \leq N. \end{aligned}$$

For each $\mathbf{q} \in \{0,1\}^N$, we can replace $\mathbf{q}$ by defining set $A = \{\mathcal{T}_j | q_j = 1\}$. Then $\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d})$

can be rewritten as follows.

$$\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d}) = \max_{A \subset \{\mathcal{T}_1, \ldots, \mathcal{T}_N\}} \sum_{\mathcal{T}_j \in A} (d_j - s_j) - \sum_{\mathcal{S}_i \in P_{\mathcal{F}}(A)} c_i.$$

Because $\mathbf{c}$ is nonnegative for any $(\mathbf{c}, \mathbf{d}) \in \mathcal{U}$, Problem-WM can be rewritten as

$$\min \sum_{j=1}^{N} s_j \tag{C.2}$$

$$\text{subject to} \quad \max_{A \subset \{\mathcal{T}_1, \ldots, \mathcal{T}_N\}} \Big\{ \sum_{\mathcal{T}_j \in A} (d_j - s_j) - \sum_{\mathcal{S}_i \in P_{\mathcal{F}}(A)} c_i \Big\} \leq \delta, \forall (\mathbf{c}, \mathbf{d}) \in \mathcal{U}, \tag{C.3}$$

$$s_j \geq 0, \forall 1 \leq j \leq N. \tag{C.4}$$

Rearranging the first set constraints described by Equation (C.3), we get

$$\max_{(\mathbf{c},\mathbf{d}) \in \mathcal{U}} \Big\{ \sum_{\mathcal{T}_j \in A} (d_j - s_j) - \sum_{\mathcal{S}_i \in P_{\mathcal{F}}(A)} c_i \Big\} \leq \delta, \forall A \subset \{\mathcal{T}_1, \ldots, \mathcal{T}_N\}. \tag{C.5}$$

Now, observe that the optimal solution for Equation (C.5) for each fixed

$$A \subset \{\mathcal{T}_1, \ldots, \mathcal{T}_N\}$$

is independent of $\mathbf{s}$. Therefore, the vector $(\mathbf{c}^A, \mathbf{d}^A)$ defined in Proposition 4.3 is always the optimal solution for Equation (C.5). As a result, we have that

$$\begin{aligned} & \max_{A \subset \{\mathcal{T}_1, \ldots, \mathcal{T}_N\}} \Big\{ \sum_{\mathcal{T}_j \in A} (d_j - s_j) - \sum_{\mathcal{S}_i \in P_{\mathcal{F}}(A)} c_i \Big\} \leq \delta, \forall (\mathbf{c}, \mathbf{d}) \in \mathcal{U} \\ \iff \quad & \sum_{\mathcal{T}_j \in A} (d_j^A - s_j) - \sum_{\mathcal{S}_i \in P_{\mathcal{F}}(A)} c_i^A \leq \delta, \forall A \subset \{\mathcal{T}_1, \ldots, \mathcal{T}_N\}, \end{aligned}$$

and the result of Proposition 4.3 follows. □

C.1.4 Proof of Lemma 4.7

Proof. Proof of Lemma 4.7. If two inventory allocations $\mathbf{s} = (s_1, s_2, \ldots, s_N)$ and $\mathbf{s}' = (s'_1, s'_2, \ldots, s'_N)$ are feasible for Problem-WM, then by Lemma 4.1, the convex combination of $\mathbf{s}$ and $\mathbf{s}'$ is also feasible for Problem-WM. By assumption of symmetry, if the inventory allocation $\mathbf{s}$ is optimal for Problem-WM, then $\sigma(\mathbf{s}), \sigma^2(\mathbf{s}), \ldots, \sigma^{N-1}(\mathbf{s})$ are also optimal.

Therefore, their convex combination $\bar{\mathbf{s}} = (\bar{s}, \bar{s}, \ldots, \bar{s})$, where $\bar{s} = \sum_{i=1}^{N} s_i/N$, is also optimal for Problem-WM. □

C.1.5 Proof of Proposition 4.8

To prove Proposition 4.8, we first develop a technical lemma that characterizes the worst-case lost sales of $\mathcal{F}$ when $s_j = s$ for all $1 \leq j \leq N$. The key idea to this lemma is to take advantage of the symmetries in $\mathcal{U}_c$ and $\mathcal{U}_d$.

Lemma C.1. *Fix an arbitrary flexibility structure $\mathcal{F}$ and suppose $s_j = s$ for all $1 \leq j \leq N$. Then*

$$\max_{(\mathbf{c},\mathbf{d})\in\mathcal{U}_c\times\mathcal{U}_d} \Pi(\mathcal{F},\mathbf{s},\mathbf{c},\mathbf{d}) = \max_{1\leq t\leq N} \Big(D^{\max}(t) - t\cdot s - C^{\min}(\delta^t(\mathcal{F})) \Big), \tag{C.6}$$

where $\delta^t(\mathcal{F})$ is the minimal value of $|P_{\mathcal{F}}(A)|$ for any $A \subset \{\mathcal{T}_1, \ldots \mathcal{T}_N\}$ such that $|A| = t$.

Proof. Proof of Lemma C.1. By Equation (4.9) in Corollary 4.4,

$$\begin{aligned}
\max_{(\mathbf{c},\mathbf{d})\in\mathcal{U}_c\times\mathcal{U}_d} \Pi(\mathcal{F},\mathbf{s},\mathbf{c},\mathbf{d}) &= \max_{A\subset\{\mathcal{T}_1,\ldots,\mathcal{T}_N\}} \max_{(\mathbf{c},\mathbf{d})\in\mathcal{U}_c\times\mathcal{U}_d} \Big(\sum_{\mathcal{T}_j\in A} (d_j - s) - \sum_{\mathcal{S}_i\in P_{\mathcal{F}}(A)} c_i \Big) \\
&= \max_{A\subset\{\mathcal{T}_1,\ldots,\mathcal{T}_N\}} \Big(D^{\max}(|A|) - |A|\cdot s - C^{\min}(|P_{\mathcal{F}}(A)|) \Big) \\
&\geq \max_{1\leq t\leq N} \Big(D^{\max}(t) - t\cdot s - C^{\min}(\delta^t(\mathcal{F})) \Big).
\end{aligned}$$

Define set A^* as set the maximizes

$$\max_{A\subset\{\mathcal{T}_1,\ldots,\mathcal{T}_N\}} D^{\max}(|A|) - |A|\cdot s - C^{\min}(|P_{\mathcal{F}}(A)|),$$

and define $t^* = |A^*|$. Because $C^{\min}(t)$ is nondecreasing with t, we have $|P_{\mathcal{F}}(A^*)| = \delta^{t^*}(\mathcal{F})$. Therefore, we have

$$\begin{aligned}
\max_{(\mathbf{c},\mathbf{d})\in\mathcal{U}_c\times\mathcal{U}_d} \Pi(\mathcal{F},\mathbf{s},\mathbf{c},\mathbf{d}) &= D^{\max}(t^*) - t^*\cdot s - C^{\min}(\delta^{t^*}(\mathcal{F})) \\
&\leq \max_{1\leq t\leq N} \Big(D^{\max}(t) - t\cdot s - C^{\min}(\delta^t(\mathcal{F})) \Big).
\end{aligned}$$

Combining both inequalities we obtain the desired result. □

Now, we are ready to prove Proposition 4.8.

Proof. Proof of Proposition 4.8. Because $\mathcal{F}$ is a K-chain, for any integer $1 \leq t \leq N$, if we take $A = \{\mathcal{T}_1, \ldots\ldots, \mathcal{T}_t\}$, then $|P_{\mathcal{F}}(A)| = \delta^t(\mathcal{F})$. Moreover, if $1 \leq t \leq N - K + 1$, then $|P_{\mathcal{F}}(A)| = t + K - 1$ and if $N - K + 1 < t \leq N$, then $|P_{\mathcal{F}}(A)| = N$. Thus, we have

$$\delta^t(\mathcal{F}) = \begin{cases} t + K - 1 & \text{if } 1 \leq t \leq N - k + 1, \\ N & \text{if } N - k + 1 < t \leq N. \end{cases} \tag{C.7}$$

Combining this with Lemma C.1, we get that $\max_{(\mathbf{c},\mathbf{d}) \in \mathcal{U}_c \times \mathcal{U}_d} \Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d}) \leq \delta$ if and only if

$$D^{\max}(t) - ts - C^{\min}(t + K - 1) \leq \delta, \forall 1 \leq t \leq N - K$$
$$\text{and} \qquad D^{\max}(t) - ts - C^{\min}(N) \leq \delta, \forall N - k \leq t \leq N.$$

Therefore if $s_j = s^*$ for $1 \leq j \leq N$ is an optimal inventory, then s^* must be the smallest nonnegative quantity so that s satisfies the two equations above. This implies that

$$s^* = \max\{\max_{1 \leq t \leq N-K} \frac{D^{\max}(t) - C^{\min}(t + K - 1) - \delta}{t}, \max_{N-k<t\leq N} \frac{D^{\max}(t) - C^{\min}(N) - \delta}{t}, 0\}.$$

□

We note that an alternative proof for Proposition 4.3 can be also derived by converting Problem-WM into a suitable robust network flow problem and apply the result of Atamtürk and Zhang (2007). We choose use proof above because the application of the structure of Problem-WM makes the proof much more intuitive. Moreover, by avoiding invoking the result of Atamtürk and Zhang (2007), it makes our paper self contained.

C.1.6 Proofs of Lemma 4.9 and 4.10

Recall that in Section 4.4.2, that we assume without loss of generality, that all odd (high variability) products have the same inventory level s_H and all even (low variability) products have the same inventory level s_L. For simplicity, we slightly abuse the notation by using s to denote both $[s_H, s_L]$ and $[s_1, ..., s_{2N}]$, where $s_{2i-1} = s_H$, and $s_{2i} = s_L$ for $1 \leq i \leq N$.

Note that for any $B \subset \{\mathcal{S}_1, \ldots, \mathcal{S}_{2N}\}$, and any $\mathbf{c} \in \mathcal{U}_C$, by construction of $\mathcal{U}_C$, we have

$$\sum_{\mathcal{S}_i \in B} c_i \geq |B| - 1. \tag{C.8}$$

Proof. Proof of Lemma 4.9. For each integer i from 1 to N, define $\mathbf{d}^{2i-1}$ to be the vector that $d^{2i-1}_{2i-1} = 1 + \sqrt{3}\sigma_H$, $d^{2i-1}_j = 1$ for all j not equal to $2i-1$, and $\mathbf{d}^{2i}$ be the vector that $d^{2i}_{2i} = 1 + \sqrt{3}\sigma_L$, $d^{2i}_j = 1$ for all j not equal to $2i$. It is easy to check that for all $1 \leq j \leq 2N$, $\mathbf{d}^j$ is in $\mathcal{U}_d$. Moreover, recall that $\mathbf{c}^i$ is the vector such that $c^i_i = 0$ and $c^i_j = 1$ for all $1 \leq j \neq i \leq 2N$, for each i from 1 to $2N$.

By Equation (4.9), and our assumption that $\delta = 0$, $\mathbf{s}$ is a feasible solution for Problem-WM if and only if

$$\sum_{\mathcal{T}_j \in A} (d_j - s_j) - \sum_{\mathcal{S}_i \in P_{\mathcal{F}}(A)} c_i \leq 0, \quad \forall A \subset \{\mathcal{T}_1, \ldots, \mathcal{T}_{2N}\}, (\mathbf{c}, \mathbf{d}) \in \mathcal{U}_c \times \mathcal{U}_d.$$

Consider the case where $\mathcal{F}$ is the dedicated design. We next show that $s_H = 1 + \sqrt{3}\sigma_H$, and $s_L = 1 + \sqrt{3}\sigma_L$ is an optimal solution for Problem-WM. First, for each $1 \leq i \leq N$, $d_{2i-1} \leq 1 + \sqrt{3}\sigma_H = s_{2i-1}$ and $d_{2i} \leq 1 + \sqrt{3}\sigma_L = s_{2i}$ which therefore implies that for all $A \subset \{\mathcal{T}_1, \ldots, \mathcal{T}_{2N}\}$ and $(\mathbf{c}, \mathbf{d}) \in \mathcal{U}_c \times \mathcal{U}_d$,

$$\sum_{\mathcal{T}_j \in A} (d_j - s_j) - \sum_{\mathcal{S}_i \in P_{\mathcal{F}}(A)} c_i \leq \sum_{\mathcal{T}_j \in A} (d_j - s_j) \leq 0.$$

Therefore, $\mathbf{s}$ is feasible. Also, because $\mathbf{d}^1, \mathbf{d}^2 \in \mathcal{U}_d$ and $\mathbf{c}^1, \mathbf{c}^2 \in \mathcal{U}_c$, for any feasible solution $\mathbf{s}'$ of Problem-WM, we must have

$$(d^1_1 - s'_H) - \sum_{\mathcal{S}_i \in P_{\mathcal{F}}(\{1\})} c^1_i = 1 + \sqrt{3}\sigma_H - s'_H \leq 0,$$
$$\text{and } (d^2_2 - s'_L) - \sum_{\mathcal{S}_i \in P_{\mathcal{F}}(\{1\})} c^2_i = 1 + \sqrt{3}\sigma_L - s'_L \leq 0.$$

This implies that any feasible solution $\mathbf{s}' \geq \mathbf{s}$, and therefore, $\mathbf{s}$ is the unique optimal solution. Note that $s_H > s_L$, and we are done with the case where $\mathcal{F}$ is the dedicated design.

Next, consider the case where $\mathcal{F}$ is the 2-chain design. Let $s_H = t + \sqrt{3}\sigma_H$, and $s_L =$

$t + \sqrt{3}\sigma_L$, where

$$t = \max\{0, \frac{D^{\max} - 2N + 1 - \sqrt{3}N(\sigma_H + \sigma_L)}{2N}\}, \text{where } D^{\max} = \max_{\mathbf{d} \in \mathcal{U}_d} \sum_{j=1}^{2N} d_j.$$

We next show that s is an optimal solution for Problem-WM.

First, we check the feasibility of s. For any $A \subsetneq \{\mathcal{T}_1, \ldots, \mathcal{T}_{2N}\}$, let

$$A_H = A \cup \{\mathcal{T}_1, \ldots, \mathcal{T}_{2N-1}\}$$

and

$$A_L = A \cup \{\mathcal{T}_2, \ldots, \mathcal{T}_{2N}\}.$$

Note that because $\mathcal{F}$ is a 2-chain, $|P_{\mathcal{F}}(A)| \geq |A| + 1$, which implies that

$$\sum_{\mathcal{S}_i \in P_{\mathcal{F}}(A)} c_i \geq |A|, \forall \mathbf{c} \in \mathcal{U}_c$$

Therefore, for any $(\mathbf{c}, \mathbf{d}) \in \mathcal{U}_c \times \mathcal{U}_d$, we have

$$\begin{aligned}
\sum_{\mathcal{T}_j \in A} (d_j - s_j) - \sum_{\mathcal{S}_i \in P_{\mathcal{F}}(A)} c_i &\leq \sum_{\mathcal{T}_j \in A} (d_j - s_j) - |A| \\
&\leq \sum_{\mathcal{T}_j \in A_H} (d_j - s_H - 1) + \sum_{\mathcal{T}_j \in A_L} (d_j - s_L - 1) \\
&\leq \sum_{\mathcal{T}_j \in A_H} (d_j - \sqrt{3}\sigma_H - 1) + \sum_{\mathcal{T}_j \in A_L} (d_j - \sqrt{3}\sigma_L - 1) \\
&\leq 0.
\end{aligned}$$

Finally, when $A = \{\mathcal{T}_1, \ldots, \mathcal{T}_{2N}\}$,

$$\begin{aligned}
\sum_{\mathcal{T}_j \in A} (d_j - s_j) - \sum_{\mathcal{S}_i \in P_{\mathcal{F}}(A)} c_i &\leq \sum_{j=1}^{2N} d_j - N(s_H + s_L + 2t) - 2N + 1 \\
&\leq D^{\max} - 2N + 1 - N(\sqrt{3}\sigma_H + \sqrt{3}\sigma_L + 2t) \\
&= D^{\max} - 2N + 1 - \sqrt{3}N(\sigma_H + \sigma_L) - 2Nt \\
&\leq 0,
\end{aligned}$$

where the last inequality holds because of our choice of t. Therefore, $\mathbf{s}$ is a feasible solution.

Because $\mathbf{d}^1, \mathbf{d}^2 \in \mathcal{U}_d$, $\mathbf{c}^1, \mathbf{c}^2 \in \mathcal{U}_c$, for any feasible solution $\mathbf{s}'$ of Problem-WM, we must have

$$(d_1^1 - s'_H) - \sum_{\mathcal{S}_i \in P_{\mathcal{F}}(\{1\})} c_i^1 = \sqrt{3}\sigma_H - s'_H \leq 0,$$
$$\text{and } (d_2^2 - s'_L) - \sum_{\mathcal{S}_i \in P_{\mathcal{F}}(\{1\})} c_i^2 = \sqrt{3}\sigma_L - s'_L \leq 0.$$

Therefore, if $D^{\max} - 2N + 1 - \sqrt{3}N(\sigma_H + \sigma_L) \leq 0$, we have that any feasible solution $\mathbf{s}' \geq \mathbf{s}$, implying that $\mathbf{s}$ is the unique optimal solution.

If $D^{\max} - 2N + 1 - \sqrt{3}N(\sigma_H + \sigma_L) > 0$, let $\mathbf{d}^*$ be the vector such that $\sum_{i=1}^{2N} d_j^* = D^{\max}$. Then for any feasible solution $\mathbf{s}'$, we must have that

$$\sum_{i=1}^{2N}(d_j - s'_j) - \sum_{i=1}^{2N} c_i^1 = D^{\max} - N(s'_H + s'_L) - 2N + 1 \leq 0$$
$$\implies N(s'_H + s'_L) \geq D^{\max} - 2N + 1.$$

Now, observe that the total inventory for $\mathbf{s}$ is equal to

$$\begin{aligned} N(s_H + s_L) &= N(\sqrt{3}\sigma_H + \sqrt{3}\sigma_L + 2t) \\ &= \sqrt{3}N(\sigma_H + \sigma_L) + D^{\max} - 2N + 1 - \sqrt{3}N(\sigma_H + \sigma_L) \\ &= D^{\max} - 2N + 1. \end{aligned}$$

Therefore, the total inventory level of $\mathbf{s}$ is less or equal to the total inventory level of any feasible solution $\mathbf{s}'$. Hence, $\mathbf{s}$ is the optimal inventory. By our definition of $\mathbf{s}$, we have $s_H > s_L$ and we are done. □

Proof. Proof of Lemma 4.10. Let $D^{\max} = \max_{\mathbf{d} \in \mathcal{U}_d} \sum_{j=1}^{2N} d_j$. By our choice of uncertainty set, note that $D^{\max} \leq 2N + 2\sqrt{(\sigma_H^2 + \sigma_L^2)N} + 1$. Now, consider inventory allocation $\mathbf{s}$ such that $s_H = (1 - \sqrt{3}\sigma_H)t$ and $s_L = (1 - \sqrt{3}\sigma_L)t$, where

$$t = \frac{D^{\max} - 2N + 1}{N(2 - \sqrt{3}\sigma_H - \sqrt{3}\sigma_L)} \leq \frac{2\sqrt{(\sigma_H^2 + \sigma_L^2)N} + 1}{N(2 - \sqrt{3}\sigma_H - \sqrt{3}\sigma_L)}.$$

By the assumption in Lemma 4.10, we must have $t \leq 1$.

Therefore, we have $s_H \leq 1 - \sqrt{3}\sigma_H$ and $s_L \leq 1 - \sqrt{3}\sigma_L$. Because for any $\mathbf{d} \in \mathcal{U}_d$, $d_{2i-1} \geq 1 - \sqrt{3}\sigma_H$ and $d_{2i} \geq 1 - \sqrt{3}\sigma_L$ for each $1 \leq i \leq N$, we have that $\mathbf{d} - \mathbf{s}$ is always non-negative. Therefore, for any $A \subset \{\mathcal{T}_1, \ldots, \mathcal{T}_{2N}\}$, $\mathbf{c} \in \mathcal{U}_c$, $\mathbf{d} \in \mathcal{U}_d$,

$$\begin{aligned}
\sum_{\mathcal{T}_j \in A}(d_j - s_j) - \sum_{\mathcal{S}_i \in P_{\mathcal{F}}(A)} c_i &= \sum_{\mathcal{T}_j \in A}(d_j - s_j) - 2N + 1 \\
&\leq \sum_{j=1}^{2N}(d_j - s_j) - 2N + 1 \\
&= \sum_{j=1}^{2N} d_j - N(s_H + s_L) - 2N + 1 \\
&= \sum_{j=1}^{2N} d_j - (D^{\max} - 2N + 1) - 2N + 1 \\
&= \sum_{j=1}^{2N} d_j - D^{\max} \leq 0.
\end{aligned}$$

This implies that $\mathbf{s}$ is feasible.

Next, for any feasible solution $\mathbf{s}'$, we must have that

$$\begin{aligned}
&\sum_{i=1}^{2N}(d_j - s'_j) - \sum_{i=1}^{2N} c_i^1 = D^{\max} - N(s'_H + s'_L) - 2N + 1 \leq 0 \\
&\quad \Longrightarrow N(s'_H + s'_L) \geq D^{\max} - 2N + 1 = N(s_H + s_L).
\end{aligned}$$

And therefore, $\mathbf{s}$ must be the optimal inventory position. Because $\sigma_H > \sigma_L$, we have $s_H = (1 - \sqrt{3}\sigma_H)t < (1 - \sqrt{3}\sigma_L)t = s_L$ and we are done. □

C.2 Computing $D^{\max}(t)$

Here, we describe a general formula for computing $D^{\max}(t)$, when $\mathcal{U}_d$ is defined by

$$\mathcal{U}_d = \{\mathbf{d} \,|\, \sum_{j=1}^{N} d_j \leq N + \gamma, \sum_{j=1}^{N} |d_j - 1| \leq \beta, |d_j - 1| \leq \alpha, \forall 1 \leq j \leq N\}, \tag{C.9}$$

where α, β and γ are real parameters.

Lemma C.2. *Suppose $\mathcal{U}_d$ is defined by Equation* (C.10) *and $\beta \geq \gamma$, then*

$$D^{\max}(t) = \begin{cases} t(1+\alpha) & \text{if } 0 \leq t \leq \lfloor \frac{\beta+\gamma}{2\alpha} \rfloor, \\ t + \frac{\beta+\gamma}{2} & \text{if } \lfloor \frac{\beta+\gamma}{2\alpha} \rfloor < t < N - \lceil \frac{\beta-\gamma}{2\alpha} \rceil, \\ t + \gamma + (N-t)\alpha & \text{if } N - \lceil \frac{\beta-\gamma}{2\alpha} \rceil \leq t \leq N. \end{cases} \tag{C.10}$$

Proof. Proof of Lemma C.2. Let $\mathbf{d}^*$ be the vector such that

$$d_j^* = \begin{cases} (1+\alpha) & \text{if } 0 \leq j \leq \lfloor \frac{\beta+\gamma}{2\alpha} \rfloor, \\ 1 + (\frac{\beta+\gamma}{2} - \lfloor \frac{\beta+\gamma}{2\alpha} \rfloor \alpha) & \text{if } \lfloor \frac{\beta+\gamma}{2\alpha} \rfloor < t \leq \lceil \frac{\beta+\gamma}{2\alpha} \rceil, \\ 1 & \text{if } \lceil \frac{\beta+\gamma}{2\alpha} \rceil < t < N - \lceil \frac{\beta-\gamma}{2\alpha} \rceil, \\ 1 - (\frac{\beta-\gamma}{2} - \lfloor \frac{\beta-\gamma}{2\alpha} \rfloor \alpha) & \text{if } N - \lceil \frac{\beta-\gamma}{2\alpha} \rceil < t \leq N - \lfloor \frac{\beta-\gamma}{2\alpha} \rfloor, \\ 1 - \alpha & \text{if } N - \lceil \frac{\beta-\gamma}{2\alpha} \rceil < t \leq N. \end{cases}$$

It is easy to check that $\mathbf{d}^* \in \mathcal{U}_d$.

For any integer $0 \leq t \leq \lfloor \frac{\beta+\gamma}{2\alpha} \rfloor$, for any $\mathbf{d} \in \mathcal{U}_d$ because $|d_j - 1| \leq \alpha, \forall 1 \leq j \leq N$, clearly $\sum_{j=1}^t d_j \leq t(1+\alpha)$. But $\sum_{j=1}^t d_j^* = t(1+\alpha)$, and this implies $D^{\max}(t) = t(1+\alpha)$.

For any integer $\lfloor \frac{\beta+\gamma}{2\alpha} \rfloor < t < N - \lceil \frac{\beta-\gamma}{2\alpha} \rceil$, for any $\mathbf{d} \in \mathcal{U}_d$ because $\sum_{j=1}^N d_j \leq N + \gamma, \sum_{j=1}^N |d_j - 1| \leq \beta$, we must have $\sum_{j=1}^t d_j \leq t + \frac{\beta+\gamma}{2}$. But $\sum_{j=1}^t d_j^* = t + \frac{\beta+\gamma}{2}$, we get $D^{\max}(t) = t + \frac{\beta+\gamma}{2}$.

Finally, for any integer $N - \lceil \frac{\beta-\gamma}{2\alpha} \rceil \leq t \leq N$, for any $\mathbf{d} \in \mathcal{U}_d$ because $\sum_{j=1}^N d_j \leq N + \gamma, |d_j - 1| \leq \alpha, \forall 1 \leq j \leq N$, we must have $\sum_{j=1}^t d_j \leq t + \gamma + (N-t)\alpha$. But $\sum_{j=1}^t d_j^* = t + \gamma + (N-t)\alpha$, we get $D^{\max}(t) = t + \gamma + (N-t)\alpha$.

□

In Section 4.4, we choose $N = 12$ and $\mu = 1$, and set $\gamma = 4\sqrt{3}\sigma$, $\beta = 8\sqrt{3}\sigma$ and $\alpha = \sqrt{3}\sigma$. Substitute this into Equation (C.10), we get

$$D^{\max}(t) = \begin{cases} 6(1+\sqrt{3}\sigma) & \text{if } 1 \leq t \leq 6, \\ t + 6\sqrt{3}\sigma & \text{if } 7 \leq t \leq 10, \\ t + (16-t)\sqrt{3}\sigma & \text{if } 11 \leq t \leq 12. \end{cases} \tag{C.11}$$

Applying Equation (C.10), we get that for $2 \leq K \leq 12$, the optimal inventory level for

K-chain is equal to

$$12 \cdot \max\{\frac{6\sqrt{3}\sigma - (K-2+\delta)}{6}, \frac{6\sqrt{3}\sigma - 1 - \delta}{10}, \frac{5\sqrt{3}\sigma - \delta}{11}, \frac{4\sqrt{3}\sigma + 1 - \delta}{12}, 0\}. \quad \text{(C.12)}$$

C.3 Type 1 Service Level

The Type 1 service level is an event-oriented performance guarantee. In the context of demand shortage, the Type 1 service level ensures that the probability of total shortage being greater than δ is less than or equal to ϵ, for some constants δ and ϵ. Therefore, given that $\mathbf{C}$, $\mathbf{D}$ are probabilistic distributions, the optimization problem with Type 1 service level constraint is defined as follows.

$$\min \sum_{j=1}^{N} s_j \quad \text{(C.13)}$$

$$\text{s.t. } \mathbb{P}_{\mathbf{C},\mathbf{D}}[\Pi(\mathcal{F}, \mathbf{s}, \mathbf{C}, \mathbf{D}) \leq \delta] \geq 1 - \epsilon \quad \text{(C.14)}$$

$$s_j \geq 0, \forall 1 \leq j \leq N. \quad \text{(C.15)}$$

Unfortunately, the service level constraint is non-convex. For example, consider a setting with two products and two plants, where demand for each product is always equal to 1 and the plant capacities are i.i.d. random with each plant having capacity equal to 1 with probability 0.9 and 0 with probability 0.1. In this case, if the firm wants to guarantee that the probability of total shortage being less than 0 is less than or equal to 0.1 ($\delta = 0$ and $\epsilon = 0.1$), then either the inventory allocation $\mathbf{s}^1 = [1, 0]$ or $\mathbf{s}^2 = [0, 1]$ is a feasible solution for Equation (C.14). However, note that $0.5\mathbf{s}^1 + 0.5\mathbf{s}^2 = [0.5, 0.5]$, a convex combination of $\mathbf{s}^1$ and $\mathbf{s}^2$, is not feasible for Equation (C.14). As a result, because of the non-convex nature of Type 1 service level, solving the optimization problem defined by Equation (C.13)-(C.15) is generally very difficult.

C.4 Hardness Result

We prove that it is NP-hard to determine whether $\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d}) \leq \delta$ for all $(\mathbf{c}, \mathbf{d}) \in \mathcal{U}$, under any *fixed* flexibility designs $\mathcal{F}$. The result demonstrates that the computational complexity can arise solely from the linear inequalities in the uncertainty sets. The proof presented in

this section is different from the result presented Simchi-Levi and Wei (2014), which proved that determining whether $\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d}) \leq \delta$ for *general* flexibility designs $\mathcal{F}$ is NP-hard.

First, we present the result of Rohn (2000), which states that the following class of optimization problems is NP-hard.

$$\max \sum_{j=1}^{n} |y_i| \tag{C.16}$$

$$\text{s.t. } \mathbf{Ax} = \mathbf{y}, \tag{C.17}$$

$$-1 \leq x_i \leq 1, \forall 1 \leq i \leq m, \tag{C.18}$$

where $\mathbf{A}$ is an arbitrary matrix with dimension $m \times n$.

Next, we prove the hardness result (Proposition 4.5) for determining $\Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d}) \leq \delta$, by showing that any problem defined by Equations (C.16-C.18) can be reduced to optimizing $\max_{(\mathbf{c},\mathbf{d})\in\mathcal{U}} \Pi(\mathcal{F}, \mathbf{s}, \mathbf{c}, \mathbf{d})$.

Proof. Proof of Proposition 4.5. Consider the optimization problem defined by Equations (C.16-C.18). Because the feasible region of $\mathbf{x}$ is bounded, $\mathbf{y}$ must also be bounded. Let L_1 (and L_2) be the smallest non-negative number such that for any $\mathbf{y}$ in the set $\{\mathbf{y}|\mathbf{Ax} = \mathbf{y}, -1 \leq x_i \leq 1, \forall 1 \leq i \leq m\}$,

$$\min_{1\leq j\leq n} y_j + L_1 \geq 0 \text{ and } \min_{1\leq j\leq n} -y_j + L_2 \geq 0.$$

Then, the optimization problem defined by Equations (C.16-C.18) can be reformulated as

$$\max \sum_{j=1}^{n} (d_j - L_1)^+ + \sum_{j=1}^{n} (d_{j+n} - L_2)^+ \tag{C.19}$$

$$\text{s.t. } \mathbf{Ax} = \mathbf{y}, \tag{C.20}$$

$$d_j = y_j + L_1, \forall 1 \leq j \leq n, \tag{C.21}$$

$$d_{j+n} = -y_j + L_2, \forall 1 \leq j \leq n, \tag{C.22}$$

$$-1 \leq x_i \leq 1, \forall 1 \leq i \leq m. \tag{C.23}$$

Let $\mathcal{U}_d$ be the set of $\mathbf{d}$ where there exists $\mathbf{x} \in [-1, 1]^m$, $\mathbf{y} \in \mathbb{R}^n$ such that Equations (C.20-C.22) are satisfied. Note that $\mathcal{U}_d$ is a nonnegative polyhedral set.

Let $\mathbf{s}$ be the vector where $s_i = L_1$ and $s_{i+n} = L_2$ for $1 \leq i \leq n$. Consider the nonnegative

polyhedral uncertainty set $\mathcal{U}$ where for all $(\mathbf{c},\mathbf{d}) \in \mathcal{U}$, we must have $\mathbf{c} = \mathbf{0}$, $\mathbf{d} \in \mathcal{U}_d$. Then, for any flexibility design $\mathcal{F}$, by Equation 4.9,

$$\max_{(\mathbf{c},\mathbf{d})\in\mathcal{U}} \Pi(\mathcal{F},\mathbf{s},\mathbf{c},\mathbf{d}) = \max_{\mathbf{d}\in\mathcal{U}_d}\Big\{ \max_{A\subset\{\mathcal{T}_1,\ldots,\mathcal{T}_N\}} \sum_{\mathcal{T}_j\in A}(d_j - s_j)\Big\} = \max_{\mathbf{d}\in\mathcal{U}_d}\sum_{j=1}^{2n}(d_j - s_j)^+.$$

But $\max_{\mathbf{d}\in\mathcal{U}_d}\sum_{j=1}^{2n}(d_j - s_j)^+$ is equivalent to the optimization problem defined by Equations (C.16-C.18), which is a NP-hard problem. Therefore, we have that computing

$$\max_{(\mathbf{c},\mathbf{d})\in\mathcal{U}} \Pi(\mathcal{F},\mathbf{s},\mathbf{c},\mathbf{d}) \tag{C.24}$$

for any fixed $\mathcal{F}$ is NP-hard.

Finally, by the classical Ellipsoid method, the problem of determining whether

$$\Pi(\mathcal{F},\mathbf{s},\mathbf{c},\mathbf{d}) \leq \delta$$

for general δ is in the same complexity class solving the optimization problem, and we are done. □

Proposition 4.5 has the following important implication for Problem-WM. Even with the simplest possible flexibility design $\mathcal{F}$ (i.e. dedicated or full flexibility design), determining whether $\Pi(\mathcal{F},\mathbf{s},\mathbf{c},\mathbf{d}) \leq \delta$ is NP-hard, unless P=NP, there does not exist a formulation for Problem-WM with a polynomial number of linear inequalities.

C.5 Choosing Uncertainty Sets

Like many robust optimization model, the choice of uncertainty sets (confidence set) plays a crucial role to the quality of the solutions. In a robust optimization models, we rarely wish to protect our selves against all possible uncertainty realizations but only against scenarios that "likely" to occur. In this section, we propose a method to restrict the demand uncertainty in our robust optimization model. Our method mimic the approach of traditional robust optimization (see for example, Ben-Tal et al. 2009, Bandi and Bertsimas 2012), that uses the properties of stochastic demand distributions.

C.5.1 Structure of the Proposed Uncertainty Sets

Suppose that the demand for each product j is denoted by some random variable D_j with mean $\bar{d}_j$. For the worst-case model, we apply the following set of linear inequalities to restrict the variations of demand.

$$\{(\mathbf{c},\mathbf{d})|\sum_{j=1}^{N} d_j \leq \sum_{j=1}^{N} \bar{d}_j + \gamma, \sum_{j=1}^{N} |d_j - \bar{d}_j| \leq \beta, |d_j - \mu_j| \leq \alpha_j, \forall 1 \leq j \leq N\}, \tag{C.25}$$

with parameters γ, β and α_j for $1 \leq j \leq N$. The parameter γ allows us to control the deviation of total realized demand from the total expected demand; the parameter β allows us to control the total absolute (L_1) deviation between the realized demand and the mean demand; and finally, the parameters α_j for $1 \leq j \leq N$ allows us to control the deviation of each product's realized demand from its mean. To choose the values of α_j, β and γ, one can fit the numbers with empirical data. Alternatively, we can choose the values of α_j, β and γ by properties of the stochastic demand distributions. In Section C.5.2, we provide a specific example in which we choose the values of α_j, β and γ based on the properties of the stochastic demand distributions.

In our context, the variabilities of plant capacities are typically low, and the most of the uncertainty come from plant disruptions due to low probability, unforeseeable events. Thus, the capacity uncertainty sets are chosen to reflect possible plant disruptions. We propose two approaches to model capacity uncertainty.

The first approaches is what we called a *scenario-based* approach. Typically, we use this approach when the scenarios are low probability and independent events, and the probabilities are difficult to estimate. In this case, we consider a finite set Ω as the set of disruption scenarios, and for each $\omega \in \Omega$, we let $\mathbf{c}^\omega$ be the plant capacities corresponding to disruption scenario ω. Thus, to model the variations of capacity, we simply consider the uncertainty set which is the convex combination of all of those disruption scenarios.

$$\{(\mathbf{c},\mathbf{d})|\sum_{\omega\in\Omega} \lambda_\omega \mathbf{c}^\omega, \sum_{\omega\in\Omega} \lambda_\omega = 1, \lambda_\omega \geq 0, \forall \omega \in \Omega\}. \tag{C.26}$$

In a specific example, if $\bar{\mathbf{c}}$ is the capacity vector with no disruption, and the likelihood of plant disruption is low and disruptions at plants independent, then we take the convex combination of the scenarios where exactly one plant is disrupted. In that case, we have

$\Omega = \{1, \ldots, M\}$, where for each $1 \leq i \leq M$, we define $\mathbf{c}^i$ to be the vector such that $c^i_i = 0$ and $c^i_{i'} = \bar{c}_{i'}$ for $1 \leq i' \neq i \leq M$.

The second approach is similar to our approach in modelling the demand uncertainty. That is, we restrict the capacity uncertainty using the following linear inequalities to limit the deviation of total realized capacity from the capacity with no disruption:

$$\{(\mathbf{c}, \mathbf{d}) | \sum_{i=1}^{M} c_i \geq \sum_{i=1}^{M} \bar{c}_i - \gamma, 0 \leq c_i \leq \bar{c}_i, \forall 1 \leq i \leq M.\}. \tag{C.27}$$

Vector $\bar{\mathbf{c}}$ is the plant capacities with no disruption, and while parameter γ bounds the total loss of capacity for all of the plants. When the disruption probabilities are known, we can select some γ such that the random plant capacities will satisfy the inequality $\sum_{i=1}^{M} \bar{c}_i - \gamma$ with high probability. Also, because we are assuming that the capacity of plants can only decrease due to disruption, in the uncertainty set, we also have the constraints $c_i \leq \bar{c}_i$, for each i from 1 to M.

Finally, there also exists potentially pooling between capacity uncertainty and demand uncertainty; i.e. the worst possible disruption probably does not occur simultaneously with the worst possible demand scenario. The pooling between capacity uncertainty and demand uncertainty can be modelled by adding linear inequalities that involves both $\mathbf{c}$ and $\mathbf{d}$, i.e., inequalities of type

$$\{(\mathbf{c}, \mathbf{d}) | \sum_{j=1}^{N} d_j - \sum_{i=1}^{M} c_i \geq \gamma\}. \tag{C.28}$$

C.5.2 Selecting Parameters to Model Demand Uncertainty

Here, we provide a concrete example of selecting parameters α_j, β and γ for Equation (C.25) to model the uncertainties of product demand. While our example assumes that the demands are uniform distributed, our method can be easily extended to other stochastic demand distributions with bounded support.

Suppose product demands are independent and identically distributed (i.i.d.) uniform random variables with mean μ and standard deviation σ. Let D_j be the stochastic demand for product j. Note that because demand are uniformly distributed, the support of the distribution is $[\mu - \sqrt{3}\sigma, \mu + \sqrt{3}\sigma]$. As a result, we set $\alpha_j = \sqrt{3}\sigma$ for $1 \leq j \leq N$.

Motivated by the classical central limit theorem (CLT), we set

$$\beta = \mathbb{E}[\sum_{j=1}^{N} |D_j - \mu|] + \theta \sqrt{\mathbb{V}[\sum_{j=1}^{N} |D_j - \mu|]},$$

where $\mathbb{E}[\cdot]$ and $\mathbb{V}[\cdot]$ are expectation and variance functions, and θ being the parameter to control the conservatism of the inequality. Note that $|D_1 - \mu|$ is uniformly distributed over $[0, \sqrt{3}\sigma]$, and which implies that the expected value and standard deviation of $|D_1 - \mu|$ are equal to $\frac{\sqrt{3}\sigma}{2}$ and $\frac{\sigma}{2}$ respectively. Therefore, in this example, we have

$$\beta = \frac{\sqrt{3}\sigma N}{2} + \frac{\theta\sigma\sqrt{N}}{2}.$$

Similarly, we use CLT to set

$$\gamma = \theta \sqrt{\mathbb{V}[\sum_{j=1}^{N} D_j]} = \theta\sqrt{N}\sigma.$$

In sum, we have

$$\{(\mathbf{c}, \mathbf{d}) | \sum_{j=1}^{N} d_j \leq \mu N + \theta\sqrt{N}\sigma, \sum_{j=1}^{N} |d_j - \mu| \leq \frac{\sqrt{3}\sigma N}{2} + \frac{\theta\sigma\sqrt{N}}{2},$$
$$|d_j - \mu| \leq \sqrt{3}\sigma, \forall 1 \leq j \leq N\}. \tag{C.29}$$

The parameter θ can be selected through either simple rule of thumb (i.e., $\theta = 2$ or $\theta = 3$), or more carefully so that the uncertainty set has certain probabilistic guarantees. The probabilistic guarantees can be checked using a monte carlo simulation; compute the empirical probability of $\mathbf{D}$ being in the set defined by Equation (C.29) using sampled demand. For example, for $N = 10$ and D_j being uniformly distributed over interval $[0, 2]$, if we pick $\theta = 2$, the monte carlo simulation shows that $[D_1, \ldots, D_N]$ lies in the demand uncertainty set about 96% of the time.

Also, we can use the concentration inequalities to bound the probabilities that the demand would falls into the set defined by Equation (C.29). In that case, we can apply the

Hoeffding's inequality to obtain that

$$\mathbb{P}[\sum_{j=1}^{N} D_j > \mu N + \theta\sqrt{N}\sigma] \leq e^{-\theta^2/6}$$

$$\text{and } \mathbb{P}[\sum_{j=1}^{N} |D_j - \mu| > \frac{\sqrt{3}\sigma N}{2} + \frac{\theta\sigma\sqrt{N}}{2}] \leq e^{-\theta^2/6}.$$

Therefore, by union bound, we get

$$\mathbb{P}[\sum_{j=1}^{N} D_j \leq \mu N + \theta\sqrt{N}\sigma, \sum_{j=1}^{N} |D_j - \mu| \leq \frac{\sqrt{3}\sigma N}{2} + \frac{\theta\sigma\sqrt{N}}{2}] \leq 1 - 2e^{-\theta^2/6}. \tag{C.30}$$

Note that if we use the bound from Equation (C.30) to ensure that the probability of $\mathbf{D}$ being in the set defined by Equation (C.29) is at least 96%, we would need to set θ to be approximately 4.4. This is significantly more conservative than what we get in the monte carlo simulation. Nevertheless, the bound from Equation (C.30) holds for all N, and any distribution that is supported over $[1-\sqrt{\sigma}, 1+\sqrt{\sigma}]$. Moreover, when N grows large, $4.4\sqrt{N}$ is much smaller comparing to N, thus providing us with a good bound for controlling the maximum total product demand.

Bibliography

Agrawal, S. and Goyal, N. (2011). Analysis of thompson sampling for the multi-armed bandit problem. *arXiv preprint arXiv:1111.1797*.

Agrawal, S. and Goyal, N. (2013). Further optimal regret bounds for thompson sampling. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, pages 99–107.

Araman, V. F. and Caldentey, R. (2009). Dynamic pricing for nonperishable products with demand learning. *Operations Research*, 57(5):1169–1188.

Ardestani-Jaafari, A. and Delage, E. (2014). The value of flexibility in robust location-transportation problem. *Working Paper*.

Arreola-Risa, A. and DeCroix, G. (1998). Inventory management under random supply disruptions and partial backorders. *Naval Research Logistics*, 45(7):687–703.

Atamtürk, A. and Zhang, M. (2007). Two-stage robust network flow and design under demand uncertainty. *Operations Research*, 55(4):662–673.

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002a). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256.

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

Aviv, Y. and Pazgal, A. (2005a). Dynamic pricing of short life-cycle products through active learning. working paper, Olin School Business, Washington University, St. Louis, MO.

Aviv, Y. and Pazgal, A. (2005b). A partially observed markov decision process for dynamic pricing. *Management Science*, 51(9):1400–1416.

Aviv, Y. and Vulcano, G. (2012). Dynamic list pricing. In Özer, Ö. and Phillips, R., editors, *The Oxford Handbook of Pricing Management*, pages 522–584. Oxford University Press, Oxford.

Badanidiyuru, A., Kleinberg, R., and Slivkins, A. (2013). Bandits with knapsacks. In *IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 207–216. IEEE.

Bandi, C. and Bertsimas, D. (2012). Tractable stochastic analysis in high dimensions via robust optimization. *Mathematical programming*, 134(1):23–70.

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2009). *Robust optimization*. Princeton University Press.

Bertsimas, D., Brown, D., and Caramanis, C. (2011). Theory and applications of robust optimization. *SIAM review*, 53(3):464–501.

Bertsimas, D. and Perakis, G. (2006). Dynamic pricing: A learning approach. In Lawphongpanich, S., Hearn, D. W., and Smith, M. J., editors, *Mathematical and Computational Models for Congestion Charging*, volume 101 of *Applied Optimization*, pages 45–79. Springer US.

Besbes, O. and Sauré, D. (2014). Dynamic pricing strategies in the presence of demand shifts. *Manufacturing & Service Operations Management*, 16(4):513–528.

Besbes, O. and Zeevi, A. (2009). Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research*, 57(6):1407–1420.

Besbes, O. and Zeevi, A. (2011). On the minimax complexity of pricing in a changing environment. *Operations Research*, 59(1):66–79.

Besbes, O. and Zeevi, A. (2012). Blind network revenue management. *Operations Research*, 60(6):1537–1550.

Besbes, O. and Zeevi, A. (2015). On the (surprising) sufficiency of linear models for dynamic pricing with demand learning. *Management Science*, 61(4):723–739.

Bitran, G. R. and Mondschein, S. V. (1997). Periodic pricing of seasonal products in retailing. *Management Science*, 43(1):64–79.

Bollapragada, R., Rao, U. S., and Zhang, J. (2004). Managing inventory and supply performance in assembly systems with random supply capacity and demand. *Management Science*, 50(12):1729–1743.

Boyacı, T. and Özer, Ö. (2010). Information acquisition for capacity planning via pricing and advance selling: When to stop and act? *Operations Research*, 58(5):1328–1349.

Broder, J. (2011). *Online Algorithms For Revenue Management*. PhD thesis, Cornell University.

Broder, J. and Rusmevichientong, P. (2012). Dynamic pricing under a general parametric choice model. *Operations Research*, 60(4):965–980.

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

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

Chapelle, O. and Li, L. (2011). An empirical evaluation of thompson sampling. In *Advances in Neural Information Processing Systems*, pages 2249–2257.

Chen, Q., Jasin, S., and Duenyas, I. (2014). Adaptive parametric and nonparametric multi-product pricing via self-adjusting controls. Working paper, Ross School of Business, University of Michigan.

Chen, Q., Jasin, S., and Duenyas, I. (2015a). Real-time dynamic pricing with minimal and flexible price adjustments. *Management Science*. To appear.

Chen, X., Owen, Z., Pixton, C., and Simchi-Levi, D. (2015b). A statistical learning approach to personalization in revenue management. *Available at SSRN: http://ssrn.com/abstract=2579462*.

Cheung, W. C., Simchi-Levi, D., and Wang, H. (2015). Dynamic pricing and demand learning with limited price experimentation. Working paper, Massachusetts Institute of Technology; Available at SSRN 2457296.

Chou, M., Teo, C.-P., and Zheng, H. (2011). Process flexibility revisited: The graph expander and its applications. *Operations Research*, 59(5):1090–1105.

Chou, M. C., Chua, G. A., Teo, C.-P., and Zheng, H. (2010). Design for process flexibility: Efficiency of the long chain and sparse structure. *Operations Research*, 58(1):43–58.

Culp, S. (2013). Supply chain disruption a major threat to business. *Forbes*.

DeCroix, G. A. (2013). Inventory management for an assembly system subject to supply disruptions. *Management Science*, 59(9):2079–2092.

den Boer, A. and Zwart, B. (2014). Simultaneously learning and optimizing using controlled variance pricing. *Management Science*, 60(3):770–783.

den Boer, A. V. (2014). Dynamic pricing with multiple products and partially specified demand distribution. *Mathematics of operations research*, 39(3):863–888.

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

Farias, V. and Van Roy, B. (2010). Dynamic pricing with a prior on market response. *Operations Research*, 58(1):16–29.

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

Fine, C. H. and Freund, R. M. (1990). Optimal investment in product-flexible manufacturing capacity. *Management Science*, 36(4):449–466.

Gabrel, V., Murat, C., and Thiele, A. (2014). Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3):471–483.

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

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

Garivier, A. and Cappé, O. (2011). The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Conference on Learning Theory (COLT)*, pages 359–376.

Gittins, J., Glazebrook, K., and Weber, R. (2011). *Multi-armed bandit allocation indices*. John Wiley & Sons.

Graves, S. C. and Willems, S. P. (2000). Optimizing strategic safety stock placement in supply chains. *Manufacturing & Service Operations Management*, 2(1):68–83.

Gupte, A., Ahmed, S., Cheon, M. S., and Dey, S. (2013). Solving mixed integer bilinear problems using milp formulations. *SIAM Journal on Optimization*, 23(2):721–744.

Gürler, Ü. and Parlar, M. (1997). An inventory problem with two randomly available suppliers. *Operations Research*, 45(6):904–918.

Harrison, J., Keskin, N., and Zeevi, A. (2012). Bayesian dynamic pricing policies: Learning and earning under a binary prior distribution. *Management Science*, 58(3):570–586.

Hedde, C. (2014). 2013 Natural Catastrophe Year in Review - US/Global Natural Catastrophe Update. Webinar.

Hopp, W., Iravani, S., and Liu, Z. (2012). Mitigating the impact of disruptions in supply chains. In Gurnani, H., Mehrotra, A., and Ray, S., editors, *Supply Chain Disruptions*, pages 21–49. Springer London.

Hopp, W. J., Tekin, E., and Van Oyen, M. P. (2004). Benefits of skill chaining in serial production lines with cross-trained workers. *Management Science*, 50(1):83–98.

Jasin, S. and Kumar, S. (2012). A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. *Mathematics of Operations Research*, 37(2):313–345.

Jordan, W. C. and Graves, S. C. (1995). Principles on the benefits of manufacturing process flexibility. *Management Science*, 41(4):577–594.

Kaufmann, E., Korda, N., and Munos, R. (2012). Thompson sampling: An asymptotically optimal finite-time analysis. In *Algorithmic Learning Theory*, pages 199–213. Springer.

Keskin, N. B. and Zeevi, A. (2014). Dynamic pricing with an unknown demand model: Asymptotically optimal semi-myopic policies. *Operations Research*, 62(5):1142–1167.

Kleinberg, R. and Leighton, T. (2003). The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pages 594–605. IEEE.

Lai, T. L. and Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22.

Lei, M. Y., Jasin, S., and Sinha, A. (2014). Near-optimal bisection search for nonparametric dynamic pricing with inventory constraint. Working paper, Ross School of Business, University of Michigan.

McCormick, G. P. (1976). Computability of global solutions to factorable nonconvex programs: Part I – convex underestimating problems. *Mathematical programming*, 10(1):147–175.

Mersereau, A. J., Rusmevichientong, P., and Tsitsiklis, J. N. (2009). A structured multiarmed bandit problem and the greedy policy. *IEEE Transactions on Automatic Control*, 54(12):2787–2802.

Meyer, R. R., Rothkopf, M. H., and Smith, S. A. (1979). Reliability and inventory in a production-storage system. *Management Science*, 25(8):799–807.

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

Robbins, H. (1952). Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535.

Rohn, J. (2000). Computing the norm $\|a\|_{\infty,1}$ is np-hard. *Linear and Multilinear Algebra*, 47(3):195–204.

Rothschild, M. (1974). A two-armed bandit theory of market pricing. *Journal of Economic Theory*, 9(2):185–202.

Rusmevichientong, P. and Tsitsiklis, J. N. (2010). Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411.

Russo, D. and Van Roy, B. (2014). Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243.

Secomandi, N. (2008). An analysis of the control-algorithm re-solving issue in inventory and revenue management. *Manufacturing & Service Operations Management*, 10(3):468–483.

Simchi-Levi, D. (2010). *Operations Rules: Delivering Customer Value through Flexible Operations.* The MIT Press.

Simchi-Levi, D., Schmidt, W., and Wei, Y. (2014). From superstorms to factory fires: Managing unpredictable supply chain disruptions. *Havard Business Review*, 92(1–2):96–101.

Simchi-Levi, D., Schmidt, W., Wei, Y., Zhang, P. Y., Combs, K., Ge, Y., Gusikhin, O., Sander, M., and Zhang, D. (2015). Identifying risks and mitigating disruptions in the automotive supply chain. *Forthcoming in Interfaces.*

Simchi-Levi, D. and Wei, Y. (2012). Understanding the performance of the long chain and sparse designs in process flexibility. *Operations Research*, 60(5):1125–1141.

Simchi-Levi, D. and Wei, Y. (2014). Worst-case analysis of process flexibility designs. *Forthcoming in Operations Research.*

Sodhi, M. S. and Tang, C. S. (2012). Strategic approaches for mitigating supply chain risks. In Hillier, F. S., editor, *Managing Supply Chain Risk*, volume 172 of *International Series in Operations Research and Management Science*, pages 95–108. Springer, New York.

Song, J.-S. and Zipkin, P. H. (1996). Inventory control with information about supply conditions. *Management Science*, 42(10):1409–1419.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: An introduction.* MIT press.

Talluri, K. T. and van Ryzin, G. J. (2005). *Theory and Practice of Revenue Management.* Springer-Verlag.

Tang, C. and Tomlin, B. (2008). The power of flexibility for mitigating supply chain risks. *International Journal of Production Economics*, 116(1):12–27.

Thiele, A., Terry, T., and Epelman, M. (2010). Robust linear optimization with recourse. University of Michigan, IOE Technical Report TR09-01.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, pages 285–294.

Tomlin, B. (2006). On the value of mitigation and contingency strategies for managing supply chain disruption risks. *Management Science*, 52(5):639–657.

Tomlin, B. and Wang, Y. (2005). On the value of mix flexibility and dual sourcing in unreliable newsvendor networks. *Manufacturing & Service Operations Management*, 7(1):37–57.

Wang, X. and Zhang, J. (2015). Process flexibility: A distribution-free bound on the performance of k-chain. *Operations Research.*

Wang, Z., Deng, S., and Ye, Y. (2014). Close the gaps: A learning-while-doing algorithm for single-product revenue management problems. *Operations Research*, 62(2):318–331.

Zbaracki, M. J., Ritson, M., Levy, D., Dutta, S., and Bergen, M. (2004). Managerial and customer costs of price adjustment: direct evidence from industrial markets. *Review of Economics and Statistics*, 86(2):514–533.

Zeng, B. and Zhao, L. (2013). Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters*, 41(5):457–461.