Dynamic, Data-driven Decision-making in Revenue Management

by

Will (Wei) Ma

B.Math, University of Waterloo (2010)

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

September 2018

Signature redacted

Author ....
Sloan School of Management
July 15, 2018

Signature redacted

Certified by ....
David Simchi-Levi
Professor
Thesis Supervisor

Signature redacted

Accepted by ....
Patrick Jaillet, Professor
Co-director, Operations Research Center

Dynamic, Data-driven Decision-making in Revenue Management

by

Will (Wei) Ma

Submitted to the Sloan School of Management
on July 15, 2018, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

Motivated by applications in Revenue Management (RM), this thesis studies various problems in sequential decision-making and demand learning.

In the first module, we consider a personalized RM setting, where items with limited inventories are recommended to heterogeneous customers sequentially visiting an e-commerce platform. We take the perspective of worst-case competitive ratio analysis, and aim to develop algorithms whose performance guarantees do not depend on the customer arrival process. We provide the first solution to this problem when there are both multiple items and multiple prices at which they could be sold, framing it as a general online resource allocation problem and developing a system of forecast-independent bid prices (Chapter 2). Second, we study a related assortment planning problem faced by Walmart Online Grocery, where before checkout, customers are recommended "add-on" items that are complementary to their current shopping cart (Chapter 3). Third, we derive inventory-dependent price-skimming policies for the single-leg RM problem, which extends existing competitive ratio results to non-independent demand (Chapter 4). In this module, we test our algorithms using a publicly-available data set from a major hotel chain.

In the second module, we study bundling, which is the practice of selling different items together, and show how to learn and price using bundles. First, we introduce bundling as a new, alternate method for learning the price elasticities of items, which does not require any changing of prices; we validate our method on data from a large online retailer (Chapter 5). Second, we show how to sell bundles of goods profitably even when the goods have high production costs, and derive both distribution-dependent and distribution-free guarantees on the profitability (Chapter 6).

In the final module, we study the Markovian multi-armed bandit problem under an undiscounted finite time horizon (Chapter 7). We improve existing approximation algorithms using LP rounding and random sampling techniques, which result in a (1/2 - eps)-approximation for the correlated stochastic knapsack problem that is tight relative to the LP. In this work, we introduce a framework for designing self-sampling algorithms, which is also used in our chronologically-later-to-appear work on add-on recommendation and single-leg RM.

Thesis Supervisor: David Simchi-Levi
Title: Professor

Acknowledgments

My PhD story at the MIT ORC has been a long-winded adventure: from starting straight out of undergrad in 2010, to leaving for my own start-up in 2013, to being re-admitted in 2015. In 2018, I was able to graduate and pursue my lifelong dream of becoming an academic. This happy ending would not have been possible without the countless people who have supported me over the years, believed in me through the worst of times, and believed in giving people a second chance.

First and foremost, I would like to thank my advisor, David Simchi-Levi. In short, I could not have asked for more from a PhD advisor. David has guided my research at both a macro/strategic level and a micro/tactical level, and really demonstrated to me the joy of solving problems. David has never failed to motivate or inspire me, and has never disappointed me in his support, all the while being completely open and extremely to-the-point during our conversations.

Next, I would like to thank the other members of my thesis committee: Jim Orlin, Chung-Piaw Teo, and Xin Chen. I look up to them as exemplary academics whom I can only dream of becoming like one day. I would also like to thank the other professors at MIT who have guided me over the years, including my former advisors Michel Goemans and Andreas Schulz, as well as Rob Freund.

Next, I would like to acknowledge my co-authors for work in this thesis. Chapters 2, 3, 4, 5, 6 are all joint work with my thesis advisor David Simchi-Levi. Chapter 4 is also joint work with committee member Chung-Piaw Teo. Chapter 3 is also joint work with Xi Chen and Linwei Xin, who have both been not only amazing collaborators, but also exceptionally generous in taking the time to guide me in the field of Operations Research, as well as teach me how to speak Chinese.

I would also like to acknowledge other people I have worked with, on projects that either have appeared elsewhere or are ongoing: Louis Chen, Karthik Natarajan, and Zhenzhen Yan, with whom I worked on a problem in distributionally-robust optimization; Wang Chi Cheung and Xinshang Wang, with whom I worked on a problem that incorporates online learning with competitive ratio analysis; Shalev Ben-David, Elyot Grant, and Malcolm Sharpe, co-authors of mine from publications during the first half of my PhD on computational geometry and topological combinatorics; Hanwei Li and Jinglong Zhao, with whom I worked on promotion planning for a CPG manufacturer as well as theoretical pricing questions; Elaheh Fata, with whom I worked on a problem in two-sided online bipartite matching; Arjun Khandelwal, an MIT undergraduate whom I had the pleasure of co-supervising in a project with my former start-up Prismata; and Xiaoyue Gong, with whom I worked on proposal writing.

Having had my time at the MIT last over eight years, I've had the unique luxury of making lasting friendships with over a dozen generations of ORC graduates. I would like to mention (in chronological order) a handful of upper-classmen (and women) whose advice has significantly influenced my path back to MIT and the second half of my PhD career— Yuan Zhong, Yehua Wei, Ross Anderson, Adam Elmachtoub, Kris Johnson Ferreira, Maxime Cohen, He Wang, Velibor Misic, Peng Shi, Wang Chi Cheung, and Ali Aouad. I would also like to thank my friends and colleagues, from MIT and elsewhere, who have been extremely helpful and supportive throughout my final, graduating year: Andrew, Anran, Antoine, Bradley, Can, Chloe, Clark, Daniel, Hamsa, Irene, Joann, Joey, Kimia, Maria, Negin, Seb, Tarek, and anyone else I may have missed.

I am indebted to professors Dimitris Bertsimas and Paul Mende, who made it possible for me to initiate and teach the MIT poker class over the years, greatly enriching my MIT experience. I am grateful for Elyot Grant and Mike McDonald, two childhood friends who have drastically influenced my approach to problem-solving, decision-making, and life—I would not be here today without them. I am also grateful for the companionship of Thomas and Jackie.

Last but not least, I would like to thank my grandparents, professors Keyao Ma and Yinzeng Geng of Peking University, and my parents, Xiaowu Ma and Jixia Huo, for raising me and instilling in me a lifelong love for knowledge and learning. I dedicate this thesis to them.

Contents

List of Figures

Chapter 1

Introduction

The area of Revenue Management (RM) is concerned with helping firms make more profitable decisions surrounding customer segmentation, product positioning, pricing, inventory allocation, and the like (Phillips, 2005; Talluri and Van Ryzin, 2006). RM rose as a prominent application of Operations Research during the 1980's, where it single-handedly determined the winners in the airline industry (Cross, 2011). Indeed, airlines with sophisticated pricing and yield management capabilities could make much better decisions on their selling mechanisms, fare class pricing and allocation, and assortments of itineraries offered in their promotions.

Like most areas in the field of Operations Research, RM has been greatly affected by the recent advances in computing which have enabled "big data" and "machine learning" capabilities at firms. First, there has been the adoption of *personalization* in RM. When products are purchased through e-commerce, it is possible to collect a cornucopia of data about each individual, and make the best product recommendations tailored specifically to them. Second, there has been the adoption of *integration* in RM. With real-time information available on not just the state of the market, but also on the state of the firm's supply chain, it is possible to make better global decisions which simultaneously consider the up-to-date desires of the consumer, the expected revenues collected by the firm, and the resulting cost of inventory and fulfillment.

This thesis is focused on methodologies which help organizations make these dynamic, data-driven decisions. Leveraging tools from theoretical computer science, our goal is to advance the theory and practice of modern-day RM. At this point, it is fitting to narrow our scope, clearly state the overarching assumptions in our work, and emphasize what this thesis does and does not do.

First, our work is algorithmic in nature and analyzes problems *instance by instance.* An instance consists of all the information treated as exogenously given—for example, this could be the products being sold, the initial stocking decisions, and the demand trajectory for a day. We are optimizing and measuring only the performance on any single instance. We do not analyze how the implementation of our algorithms may affect the distribution of instances faced in the future—in this case, how changing the dynamic pricing algorithm may affect the demand trajectory in future days—nor do we analyze the sum of revenues across multiple instances. Similarly, we do not rigorously study higher-level questions, such as the overall business value gained from investing in infrastructure that enables real-time information gathering and decision making, instead assuming that these controls are already available.

In this sense, our analysis is tactical in nature, and can be seen as short-sighted. Nonetheless, these assumptions are justified in many RM settings. For example, in airline RM, each instance corresponds to a flight (or a set of related flights), whose seat capacity and selling horizon have been determined long in advance. The RM team's Key Performance Index (KPI) is based on only the instances it was responsible for, measured from the start to end of those instances. We refer to Talluri and Van Ryzin (2006) for an in-depth discussion of these assumptions, including also the assumptions about demand/control, and henceforth, we will focus on these tactical models which enable prescriptive algorithms and direct implementation on data instances. Our hope is that insights from our lower-level algorithmic optimizations can also play a part in answering higher-level business questions.

Our emphasis on instances and instance-level optimality also distinguishes our work from similar lines of work in theoretical computer science, where the emphasis tends to be on elegant, universal performance guarantees that are tight on some worst-case instance. For example, we derive the optimal competitive ratio for any given set of airline fare class prices, even though this ratio is 0% (and hence meaningless) on the worst-case set of prices. One consequence of this is that our results can involve convoluted expressions in the instance parameters, and our methods are often more "brute force". However, once again, we want to answer the immediate business question of "given that this is the instance, here is how you optimize performance on it" as opposed to making a general statement of the form "here is an elegant way of guaranteeing 50%-performance on all possible instances".

In light of these discussions on scope, we outline the modules of this thesis, and describe their application in Revenue Management.

1.1 First Module: Competitive Ratio Analysis of Personalized RM Settings (Chapters 2, 3, 4)

Chapters 2–4 of this thesis all take place in a personalized RM setting. A firm is selling products with pre-determined inventories which expire at a known time (e.g. the seats on a flight, or the rooms in a hotel on a given night). Customers sequentially arrive to the firm's e-commerce platform (e.g. travel website) over the course of the selling horizon. The platform observes each customer's characteristics (e.g. computer type) and makes a personalized offer based on this (e.g. recommend business-class flights to Mac users). The firm would like to maximize its revenue earned before the products stock out or expire.

The tradeoff in this problem lies between making offers to customers which maximize immediate revenue, vs. making offers which reserve sufficient inventory for the remaining time horizon. This tradeoff is also encountered in online display advertising, where the inventories correspond to the daily budgets of bidders (advertisers) and the customers represent impressions (website queries); healthcare scheduling, where the inventories correspond to appointment slots and the customers correspond to patients; and one-way trading in financial markets, where the inventories correspond to initial capital and the heterogeneous "customers" capture fluctuations in the stock market. In fact, all of these problems can be abstractly conceived as the same dynamic resource allocation problem, where there are multiple resources, each of which could be converted to reward at multiple known rates, and the controller must sequentially choose between different "conversion options" which provide different rewards at the expense of different (stochastic) resource consumption patterns. Nonetheless, we will use Revenue Management terminology hereafter.

A problem instance $\mathcal{I}$ consists of the firm's initial product, fare class, and stocking decisions, which are exogenously given. The arrival sequence $\mathcal{A}$ of customers visiting the e-commerce platform is also exogenous, but unknown; the heterogeneous characteristics of each customer are only observed upon arrival. We consider *online* algorithms which must sequentially and irrevocably make an offer to each customer, based on her characteristics as well as the inventory state at hand, without knowing what she will choose or who the future customers will be. We would like to guarantee that the total (expected) revenue earned by the online algorithm, denoted as ALG, is at least some fraction α of the optimum, denoted as OPT. Here, the optimum is defined by an *offline* clairvoyant who knows the entire arrival sequence $\mathcal{A}$ in advance.

By designing algorithms to maximize α, we are taking a conservative approach, since these algorithms need to "hedge" evenly against all possible arrival scenarios to ensure that $\mathsf{ALG}/\mathsf{OPT} \geq \alpha$. We are essentially playing a game against an adversary, who is assumed to choose the arrival sequence $\mathcal{A}$ which minimizes $\mathsf{ALG}/\mathsf{OPT}$ after seeing the algorithm's strategy; as a result, it is often beneficial to use randomization to "hide" the algorithm's strategy from the adversary. We say that the ratio α is *optimal* on instance $\mathcal{I}$ if a better guarantee is not possible, i.e. for any $\alpha' > \alpha$ and fixed (but possibly randomized) algorithm, the adversary can always choose an arrival sequence $\mathcal{A}$ such that $\mathsf{ALG}/\mathsf{OPT} < \alpha'$. In this case, α is called the *competitive ratio*, and we denote it using $\mathsf{CR}(\mathcal{I})$. We allow the value of the competitive ratio to depend arbitrarily on the parameters of instance $\mathcal{I}$; $\mathsf{CR}(\mathcal{I})$ is known to be well-defined by Yao's minimax principle (Yao, 1977).

Note that in competitive ratio analysis, the challenge is *conceptual*—we aim to identify some ratio $\mathsf{CR}(\mathcal{I})$ which represents the "value of information" in knowing the arrival sequence $\mathcal{A}$ for instance $\mathcal{I}$, as well as a policy which yields that ratio. This can be contrasted with personalized RM problems where the arrival sequence $\mathcal{A}$ is generated by a given stochastic process (e.g. IID, or non-homogeneous Poisson), and the challenge is *computational*—overcoming the "curse of dimensionality" caused by having exponentially-many inventory states, and identifying a policy which yields optimal or near-optimal revenue in expectation over the randomness in $\mathcal{A}$. *Approximation ratios* have been established for many of these policies, which guarantee that $\mathbb{E}_{\mathcal{A}}[\mathsf{ALG}]/\mathbb{E}_{\mathcal{A}}[\mathsf{OPT}]$ is at least some fraction α. The competitive ratio provides a lower bound for the approximation ratio, since if $\mathsf{ALG} \geq \alpha \cdot \mathsf{OPT}$ for all arrival sequences $\mathcal{A}$, then this implies $\mathsf{ALG} \geq \alpha \cdot \mathsf{OPT}$ in expectation over the arrival sequences $\mathcal{A}$ generated by the stochastic process.

In practice, both approaches have their benefits and drawbacks. Policies designed to maximize the competitive ratio are computationally simple to implement, and do not require a specific stochastic model for the arrival sequence. On the other hand, policies designed to optimize for a given arrival model make much better forward-looking decisions when that model is accurate. Throughout our simulations on the public hotel data set from Bodea et al. (2009), the best performance was obtained when the two approaches were combined using ensemble meta-heuristics. This has also been observed in the works of Mahdian et al. (2012); Golrezaei et al. (2014).

1.1.1 Unified Problem Setting

We formally introduce a setting which allows us to discuss our results from Chapter 2 ("Online Resource Allocation"), Chapter 3 ("Recommendation at Checkout"), and Chapter 4 ("Single-leg RM") in a unified manner. This is a general setting (see Section 2.6) where we can offer an *assortment* of multiple products to each customer.

A firm is selling n different items. Each item i starts with a fixed inventory of b_i units, and could be offered at one of m_i feasible prices $r_i^{(1)}, \ldots, r_i^{(m_i)}$. These prices, representing fare classes, are sorted such that $0 < r_i^{(1)} < \ldots < r_i^{(m_i)}$. (This can be generalized to allow for a continuum of feasible prices; see Sections A.5.1 and C.4.) We refer to each combination of $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m_i\}$ as a fare class *product* which could be shown.

There are T customers who arrive sequentially. Upon the arrival of customer t, the firm is given her *choice function* $p_t(\cdot)$. For every product (i, j) and subset S of products, $p_t(i, j, S)$ specifies the probability of customer t choosing (i, j) if she is offered S. For now, we assume that any subset of products is a feasible assortment to show the customer, and let $\mathcal{S}$ denote the collection of all subsets of $\{(i, j) : i = 1, \ldots, n; j = 1, \ldots, m_i\}$. (This can be easily generalized to incorporate restrictions on offering multiple products/prices corresponding to the same item, restrictions on assortments requiring too much space, restrictions on offering certain products after a point in time, etc.) The firm offers each customer t an assortment S_t in an online fashion, using only products (i, j) for which item i has remaining inventory, after which the customer's purchase decision is immediately realized according to $p_t(\cdot)$. Note that customer t chooses to purchase nothing with probability $1 - \sum_{(i,j) \in S_t} p_t(i, j, S_t)$. Otherwise, if customer t chooses to purchase product (i, j), then the firm earns revenue $r_i^{(j)}$ and decrements the remaining inventory of item i by 1.

The instance $\mathcal{I}$ consists of the information on the number of items n, their starting inventories b_i, and their fare classes $r_i^{(j)}$. The arrival sequence $\mathcal{A}$ consists of the information on the total number of customers, and their choice functions $p_t(\cdot)$. For any fixed (but possibly randomized) online algorithm, problem instance $\mathcal{I}$, and arrival sequence $\mathcal{A}$, we let $\mathsf{ALG}(\mathcal{I}, \mathcal{A})$ denote the revenue earned in expectation, over both the randomness in the customer purchase decisions as well as the randomness in the algorithm. Meanwhile, $\mathsf{OPT}(\mathcal{I}, \mathcal{A})$ is defined as the optimal objective value of the following LP.

$$\begin{aligned}
\max \sum_{t=1}^{T} \sum_{S \in \mathcal{S}} \left(\sum_{(i,j) \in S} r_i^{(j)} p_t(i,j,S) \right) x_t(S) & & \\
\text{s.t.} \sum_{t=1}^{T} \sum_{S \in \mathcal{S}} \left(\sum_{j:(i,j) \in S} p_t(i,j,S) \right) x_t(S) \leq b_i & \qquad & i = 1, \dots, n \\
\sum_{S \in \mathcal{S}} x_t(S) = 1 & & t = 1, \dots, T \\
x_t(S) \geq 0 & & t = 1, \dots, T; S \in \mathcal{S}
\end{aligned}$$

This is the standard definition of the offline optimum in problems with both online arrivals and stochastic purchases, originating from Mehta and Panigrahi (2012); Golrezaei et al. (2014). It can be shown (see Lemmas 2.2.1 and 3.2.7) that $\mathsf{OPT}(\mathcal{I}, \mathcal{A})$ provides an upper bound on the expected revenue earned by any offline policy which knows $\mathcal{A}$ in advance, because $x_t(S)$ encapsulates the unconditional probability of the policy offering assortment S to customer t.

1.1.2 Synthesis of Results

Our results in Chapters 2–4 bound

$$\mathsf{CR}(\mathcal{I}) = \inf_{\mathcal{A}} \frac{\mathsf{ALG}(\mathcal{I}, \mathcal{A})}{\mathsf{OPT}(\mathcal{I}, \mathcal{A})}$$

for instances $\mathcal{I}$ falling under different families. Our work in Chapters 2 and 3 is the first to consider instances with both multiple items, as well as multiple prices at which each item's inventory could be converted into revenue, like in the problem setting defined here. This combines two challenges for the online algorithm which have previously been considered in isolation, as we outline below.

First, when there are multiple items each with a single price, there is the challenge of how to *prioritize* between selling them at their respective prices, where it may be beneficial to offer smaller assortments, to reduce cannibalization across items. This challenge was introduced in the seminal work of Karp et al. (1990) in the form of prioritizing between matching different nodes in a bipartite graph, and has since been studied in the form of online advertising (Mehta et al., 2007) and online retailing (Golrezaei et al., 2014) problems.

On the other hand, when there is a single item with multiple prices, there is the challenge of how to *reserve* the item's inventory, where it may be beneficial to offer nothing at all to customers who are not willing to pay the item's highest feasible price. The reservation challenge has also been studied extensively, in the form of one-way trading (El-Yaniv et al., 2001) and revenue management (Ball and Queyranne, 2009) problems.

In Chapter 2, we provide a general solution which simultaneously considers these two challenges, by introducing a simple online algorithm based on *virtual costs*, and showing that it achieves the best-possible competitive ratio. The idea of using virtual costs coincides with *bid-price controls* in Revenue Management, where in the classical setting the bid prices (virtual costs) are optimized with respect to a forecast, instead of optimized for the competitive ratio. Our algorithm generalizes the penalty-function algorithm of Golrezaei et al. (2014), who introduced the model in Section 1.1.1 but with a single price per item.

Our results in Chapter 2 require the following assumption on the choice models given.

Assumption 1.1.1 (Substitutability). For any customer t and any product (i, j), if $(i, j) \in S \subseteq S'$, then $p_t(i, j, S) \geq p_t(i, j, S')$.

The substitutability assumption, introduced in Golrezaei et al. (2014), is very mild and captures all *rational* choice models, where a customer has a random utility for each product (including the no-purchase option) and chooses her highest utility option from the assortment offered. Substitutability imposes that the items are not complements, i.e. the probability of selling a specific product can only decrease when more products are added to the assortment. It is important because it allows us to consider the reservation challenge separately for each item based on its feasible prices, without having to consider reserving an item's inventory to help sell another item.

In Chapter 3, we go beyond substitutability and consider a problem motivated by Walmart's online grocery, where customers are recommended "add-ons" to the items in their shopping carts before checking out. Currently, Walmart's system does not take inventory into account when making recommendations, as covered in a media article[1] about our paper. Furthermore, the add-ons being recommended are usually complementary to the items

[1] "Retailers: A better algorithm could increase online sales by 76 percent", *Chicago Booth Review*, January 31st 2018.

already in the shopping cart (e.g. butter for bread), and hence introduce additional theoretical challenges beyond our work in Chapter 2 or that in Golrezaei et al. (2014). Indeed, our bid price for each item is determined based on only that item's inventory level, and is *independent* of other items. We remedy this in Chapter 3 by introducing the notion of a *protection level in expectation*, and an algorithm makes decisions based on *expected* inventory levels instead of realized inventory levels. Such an algorithm correctly calibrates for the amount of each item to "protect" (reserve) when there is complementarity, as we explain in Section 3.2.3. The rest of the chapter focuses on overcoming the computational obstacles attached to implementing an online algorithm based on expected inventory state.

In Chapter 4, we study the problem setting from Section 1.1.1 when there is a single item, and seek improved competitive ratios in this special case. The further special case of *independent demand*, where in the choice functions $p_t(\cdot)$ no customer t is interested in more than one fare class, is the problem previously studied by Ball and Queyranne (2009). We extend their results by incorporating the price-skimming technique from Eren and Maglaras (2010), and derive inventory-dependent price-skimming policies which stochastically-increase their price distributions as inventory decreases. A key technical ingredient in this work is a new "valuation tracking" subroutine, which tracks the possible values for the optimum, and follows the most inventory-conservative control which maintains a certain competitive ratio. Using it, we derive the tight $\mathsf{CR}(\mathcal{I})$ for every instance $\mathcal{I}$, and show that it is equal to the competitive ratio from Ball and Queyranne (2009); Eren and Maglaras (2010), which is dependent on only the set of feasible prices and not on the starting inventory.

1.1.3 Comparison with Results in Literature

We illustrate our new contributions from Chapters 2–4 by comparing our competitive ratio bounds with existing ones in the literature. We distinguish instances with multiple items from those with a single item, and instances with multiple prices per item from those with a single price per item. As described in Section 1.1.2, instances with multiple items and instances with multiple prices have both been previously studied, in isolation. For simplicity, here we only distinguish between having one price per item and having *two* (arbitrary) prices per item. We derive the general form of the competitive ratio for arbitrary sets of feasible prices in both Chapters 2 and 4.

We introduce a third dimension, where we distinguish instances in the *large-inventory* *regime* and instances with *deterministic choice.* The large inventory regime is the limiting case where $\min_i b_i \to \infty$, capturing retail settings where all items are plentifully stocked. Deterministic choice assumes that $p_t(i,j,S) \in \{0,1\}$ for all customers t, products (i,j), and assortments S, capturing deterministic "matching" settings where the customer states her preferences and then the platform makes the final assignment decision. It turns out that both of these assumptions lead to the same improvement in competitive ratio, with the intuition being that under large inventories, the law of large numbers causes aggregate customer choice to become deterministic. Therefore, our third dimension distinguishes instances satisfying *either* of these two assumptions from those satisfying *neither* of these two assumptions.

We consider the 8 families of instances defined by these 3 dimensions, where along each dimension, instances in the family could either be restricted (single item; single price per item; large inventories *or* deterministic choice) or unrestricted (multiple items; two prices per item; no assumptions on inventory/choice). For each family, we display the best-known lower and upper bounds on $\mathsf{CR}(\mathcal{I})$ for instances $\mathcal{I}$ in that family, with the tight results being bolded. The results are shown on the "cube" in Fig. 1-1.

As seen in Fig. 1-1, the competitive ratio guarantee increases from 1/4 in the bottom-left corner (where there are multiple prices, two arbitrary prices per item, and no assumptions on inventory/choice) to 1 in the top-right corner (where the problem is trivial). The lower bound of 1/4 first appeared in our chronologically-earlier Chapter 3, although a cleaner solution which does not need to handle complementarity is presented in Chapter 2. When each item has a single price, the improved lower bound of 1/2 was first established by Mehta and Panigrahi (2012) in the special case where the prices are identical. This was later generalized to allow for the price of one item differing from the price of another in Golrezaei et al. (2014), using a clever probabilistic primal-dual argument. The upper bound of .621 comes from Mehta and Panigrahi (2012).

The tight competitive ratio of $1-1/e$ when there are multiple items first appeared in the seminal work of Karp et al. (1990), in the form of the deterministic online bipartite matching problem, where all items (offline nodes) have a fixed weight (price) of 1. This was not generalized to allow for differing fixed weights until Aggarwal et al. (2011). Meanwhile, the same competitive ratio of $1-1/e$ was established for the online b-matching problem in Kalyanasundaram and Pruhs (2000), and generalized to the Adwords problem in Mehta et al. (2007),

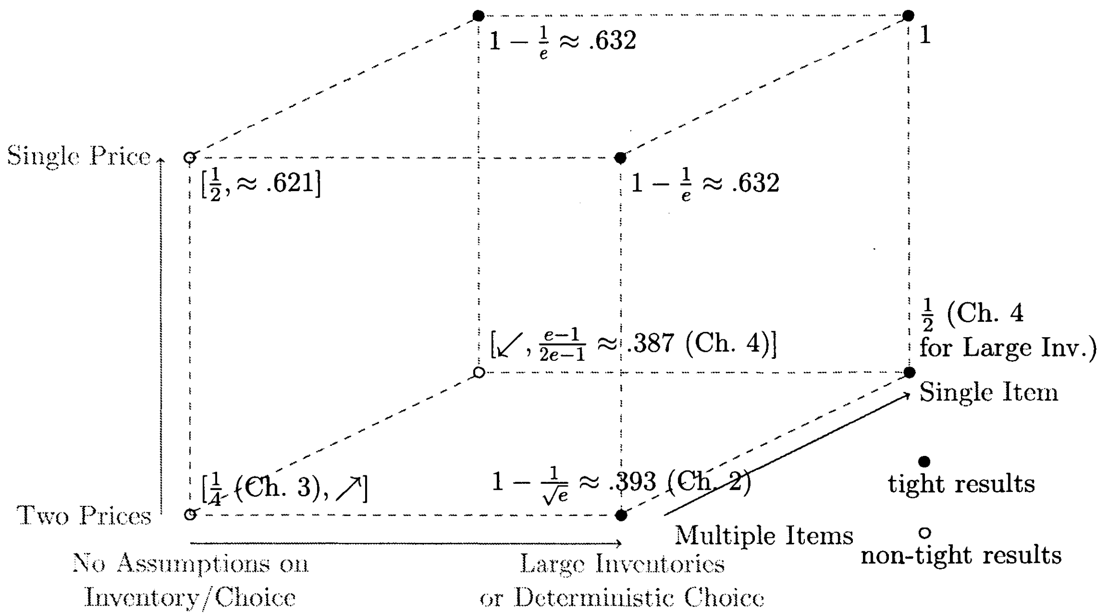


Figure 1-1: Lower and upper bounds on the competitive ratio for the 8 families of instances (our contributions in thesis are indicated; see text for other references). Filled circles represent tight results, while unfilled circles represent non-tight results. For the non-tight results, a range indicating both the lower bound and the upper bound is displayed, where arrows point to lower (resp. upper) bounds that are implied by larger (resp. smaller) families.

when the starting inventories are large. We should point out that in the Adwords problem, the inventory consumption is *fractional*, instead of stochastic. Nonetheless, following the probabilistic primal-dual argument of Golrezaei et al. (2014), we show (see Section 2.6) that the fractional-consumption model is subsumed by the stochastic-consumption model, and hence all of the results on Adwords are captured by our framework. The primal-dual argument of Golrezaei et al. (2014) is rooted in Buchbinder et al. (2007), which contains the state-of-the-art in analyzing the fractional Adwords problem. We should also refer to Devanur et al. (2013), which contains the state-of-the-art in analyzing the deterministic matching problem.

All of these results are generalized to the setting of multiple prices per item in Chapter 2. That is, we derive best-possible multi-price algorithms and competitive ratios in the fractional-consumption, stochastic-consumption, and discrete-matching settings, extending the analyses of Buchbinder et al. (2007), Golrezaei et al. (2014), and Devanur et al. (2013), respectively. The idea of deriving competitive ratios based on prices that are known originated from Ball and Queyranne (2009), where they consider the case of a single item and deterministic choice. When there are multiple items each with two arbitrary prices, our tight competitive ratio is $1 - 1/\sqrt{e} \approx .393$, as shown in Fig. 1-1. Note that this is greater than $(1-1/e)/2 \approx .316$, which would be the naive guess from combining the aforementioned ratio of $1 - 1/e$ for the multi-item, single-price case with the ratio of $1/2$ from Ball and Queyranne (2009) for the single-item, two-price case. Therefore, using our bid-price control policy, which synthesizes the prioritization and reservation challenges, results in a greater competitive ratio than naively combining the existing policies.

Finally, when there is a single item with a single price, the decision in the problem is trivial (offer the same item at the same price to every customer). However, the competitive ratio is again $1 - 1/e$ (this can be seen from, e.g., Gallego and Van Ryzin (1994)). The competitive ratio is not 1 because the optimum is defined using the LP from Section 1.1.1, which is a relaxation of the optimal offline policy that can make "fractional" offers a real-life policy cannot. To avoid this degeneracy in Chapter 4, where we study the single-item case, we define the competitive ratio using the *Hindsight Optimum (HO)* instead of the LP. We defer the definition of the HO and further discussion comparing different optima to Section 4.3.1.

Nonetheless, our results in Chapter 4 are still relevant to the "cube" in Fig. 1-1, where the optimum is based on the LP. First, in Section 4.3.1, we provide an example showing that the ratio of $1 - 1/e$ relative to the LP decreases to $(e-1)/(2e-1)$ when the item has two arbitrary prices. Furthermore, as mentioned earlier, Ball and Queyranne (2009) only analyze the single-item case under deterministic choice. Our work in Chapter 4 shows how to modify their policies to allow for stochastic choice models and obtain the same competitive ratios relative to the HO. And when the inventory is large, the LP value is no greater than the HO (asymptotic optimality relative to the LP is well-known; see e.g. Gallego and Van Ryzin (1994)). Therefore, our work implies a competitive ratio of $1/2$ when there is a single item, two arbitrary prices, and large inventories, for general stochastic customer choice, as shown in Fig. 1-1.

1.2 Second Module: Learning and Pricing using Bundles (Chapters 5, 6)

Chapters 5–6 of this thesis study how *bundling*, the practice of selling different items together, can be used by a firm for both demand learning and profit maximization. It has been known since the seminal paper of Adams and Yellen (1976) that bundling can boost profits, even when there are no complementarity effects between the items being bundled. Indeed, by selling the same items under different pricing options, the firm is achieving a form of price discrimination via bundling, and extracting greater consumer surplus.

1.2.1 Learning Valuation Distributions from Bundle Sales

In Chapter 5, we show that beyond profit maximization, bundling has the added benefit of leading to richer sales data which contains more information about demand. We develop an algorithm that, given bundle sales data, fits the customer valuation model of Adams and Yellen (1976), which has always been used for price optimization but not for demand estimation. In using it for demand estimation, we show that the price elasticities of items can be identified, despite the fact that their prices have never changed in the bundle sales data.

It is particularly interesting to us that earlier work on estimation from bundle sales has focused on fitting discrete-choice models (Chung and Rao, 2003), which are both different than the model used for bundle price optimization, and unable to identify price elasticities without additional information such as a second set of sales numbers under changed prices. (Contrast this with assortment optimization, where the same choice models that are estimated from data are also used to prescribe optimal assortments; see e.g. Farias et al. (2013).) Our hope in our work from Chapter 5 is to bridge this gap in the literature, and provide a complementary problem to both the bundle pricing problem of Adams and Yellen (1976) and the discrete-choice estimation problem of Chung and Rao (2003).

One benefit of fitting the discrete-choice model is that it is designed to enable powerful computational machinery, and high-parameter models can be quickly estimated given large-scale data with many price changes and customer/item covariates. Nonetheless, fitting the model of Adams and Yellen (1976) allows us to measure an item's price elasticity by simply comparing its sales rates inside and outside of bundles. Furthermore, we observe that on a data set from a large online retailer, the price elasticities of items identified by our model are consistent with their real price elasticities, as measured by the magnitudes of their sales spikes under Black Friday discounts. In fact, the best indicator of an item's price elasticity is the number of people who purchased the *other* items in its bundles, as we explain in Chapter 5. This is exactly in line with our model of choice and fitting algorithm.

1.2.2 Pricing Bundles from Learned Valuation Distributions

In Chapter 6, we study the classical bundle pricing problem originating from Adams and Yellen (1976) of how bundles should be sold when given the customer valuation distributions. We show how to sell bundles of goods profitably even when the goods have high production costs, by introducing a new bundling scheme called *Pure Bundling with Disposal for Cost* (PBDC). We relate it to the *two-part tariff*, and derive both distribution-dependent and distribution-free guarantees on its profitability, which improve previous techniques. We also conduct extensive numerical experiments, following the setup of Chu et al. (2011), which demonstrate its profitability over a large range of instances.

One unifying theme of our work in this module is the importance of *simple* bundling schemes, which do not overload the customers with too many choices for bundles. Indeed, in both our learning and pricing problems, we focus on bundling schemes where the customer's surplus-maximizing decision is easy to compute. Furthermore, we develop a notion of equivalence between bundling schemes for both the learning (see Section D.4.1) and pricing (see Section 6.2) problems. The conclusion from these equivalences is that our PBDC bundling scheme provides great opportunity for both demand learning and profit maximization at the same time.

1.3 Third Module: Tight Approximation Algorithm for Stochastic Knapsack and Markovian Multi-Armed Bandits (Chapter 7)

Chapter 7 of this thesis considers the *Markovian* variant of the multi-armed bandit problem. The input consists of multiple *arms*, each of which is a Markov chain with rewards. Arms return reward and evolve to the next state when they are *pulled*, and the objective is to maximize the expected undiscounted reward earned under a finite budget of pulls, which can be sequentially allocated among arms. The multi-armed bandit problem features the tradeoff of how to adaptively switch between the arms, to balance the *exploration* of arms which could potentially transition to high-reward states, and the *exploitation* of arms which could be scheduled within the remaining budget to return high reward. The Markovian formulation captures the stochastic knapsack problem and many reward-maximization stochastic scheduling problems, along with marketing problems (see Ravi and Sun (2016)) in RM where a finite budget of marketing actions can be sequentially spent on customers who return feedback (through clicks, etc.) on the state of their interest.

Under a finite budget, the well-known method of Gittins indices does not apply, and the optimal policy is generally intractable. Our main result is a $(1/2-\varepsilon)$-approximation algorithm for the irrevocable variant of this problem, where the pre-emption of arms is not allowed, using LP rounding. Our algorithm imitates the decisions of a relaxed LP, but needs to sample its own decisions in order to be run in polynomial time, resulting in the loss of ε in the approximation ratio. To the best of our knowledge, this is a novel obstacle—an LP-based algorithm that provably exists, but cannot easily found because there are exponentially-many states, and cannot be easily approximated because there are high-reward states which occur with infinitesimal probabilities. Our algorithm improves previously-known $1/16$- and $1/8$-approximations, and is tight in the sense that we show the integrality gap of our LP to be $1/2$, so it is not possible to improve the performance guarantee without tightening the LP. In the variant where preemption is allowed, we provide a $\frac{1}{12}$-approximation, which improves to a $\frac{4}{27}$-approximation if all of the pulls have the same (unit) cost to the budget.

We encounter a similar obstacle two other times in this thesis, in the chronologically-later-to-appear Chapters 3 and 4, where we again use self-sampling to convert exponential-runtime online algorithms into polynomial-runtime online algorithms that only lose ε in the competitive ratio. For this thesis, we present an abstract view of the sampling technique and analysis framework, which allows us to discuss our three sampling-based algorithms under concordant voicing.

1.3.1 Framework for Designing Self-Sampling Algorithms

Abstractly, our algorithms from Sections 7.3.4, 3.3.3, and 4.3.3 can all be described as follows. Decisions occur over a finite time horizon. At each time t, we let $\mathscr{X}_t$ denote the (random) state of the system, which affects the algorithm's decision. However, the algorithm also needs an instruction $\mathscr{I}_t$ to make the decision at time t. With this instruction, the algorithm is polynomial-time. However, these instructions cannot be computed in polynomial-time.

In Chapters 7 and 3, these instructions are state-independent[2], in that they are based on the *distribution* of $\mathscr{X}_t$. Being state-independent allows the algorithm to "coordinate" its decisions over different sample paths, and achieve some desired reward in expectation. The state-independent instruction $\mathscr{I}_t$ is combined the actual state realization at each time t to make a decision that is feasible in reality.

However, the challenge is that the state $\mathscr{X}_t$ has exponentially-many possibilities, and its distribution is intractable to compute exactly. The algorithm can only generate an empirical distribution by simulating its own execution from[3] time $t' = 1$ to $t' = t - 1$, using the previously-recorded instructions $\mathscr{I}_1, \ldots, \mathscr{I}_{t-1}$ which affect the algorithm's decisions before time t and hence affect the state of the system at time t. By repeating this simulation M_t times, where M_t is a fixed constant for each time t, the algorithm obtains M_t samples for the distribution of $\mathscr{X}_t$. Of course, this empirical distribution could be erroneous and lead to a *faulty* instruction $\mathscr{I}_t$; nonetheless M_t is chosen to be large enough so that the error in $\mathscr{I}_t$ is outside some ε-tolerance with probability at most δ_t.

In our meta-analysis, we treat both the instructions $\{\mathscr{I}_t : t\}$ and the system states $\{\mathscr{X}_t : t\}$ as random. The values of δ_t are chosen to sum to ε, so that the probability of having any faulty instruction cannot exceed ε via the union bound. Furthermore, we show that conditioned on the instructions $\{\mathscr{I}_t : t\}$ being non-faulty, the expected reward of the

[2]Alternatively, the instructions can be computed ahead of time, before observing any state realizations. However, when this framework is used in Chapter 3, online information arrives at the start of each time step t, so the pre-computation of $\mathscr{I}_t$ is not possible. Therefore, we always interpret the instructions $\mathscr{I}_t$ as being generated on-the-fly, for consistency in the analysis framework.

[3]It may be tempting to maintain a "cache" of sample paths used for simulation, so that the algorithm does not have to start simulating from $t' = 1$ at each time t. However, this would result in biased sample paths, because *conditioned* on the previously-recorded instructions $\mathscr{I}_1, \ldots, \mathscr{I}_{t-1}$, which affect the simulation at time t, the cache of sample paths would not be uniformly random.

algorithm is at least some $(\alpha - \varepsilon)$-fraction (e.g., $\alpha = 1/2$) of the optimum. That is,

$$\mathbb{E}[\mathsf{ALG}|\{\mathscr{I}_t : t\} \text{ non-faulty}] \geq (\alpha - \varepsilon)\mathsf{OPT}. \tag{1.1}$$

Statement (1.1) may seem peculiar. The algorithm's reward ALG depends on the system states incurred $\{\mathscr{X}_t : t\}$. By conditioning on a subset of possible realizations for $\{\mathscr{I}_t : t\}$, how can the distribution of $\{\mathscr{X}_t : t\}$ remain unaffected? This is where it becomes of utmost importance for the analysis that each instruction $\mathscr{I}_t$ was determined independent of any realized system states $\mathscr{X}_1, \ldots, \mathscr{X}_t$, and only dependent on the previously-recorded instructions $\mathscr{I}_1, \ldots, \mathscr{I}_{t-1}$. The event of $\{\mathscr{I}_t : t\}$ being non-faulty only affects the conditional distribution of $\{\mathscr{X}_t : t\}$ *through its effect on the algorithm's decisions.* And by the non-faulty assumption, the algorithm's decisions lose at most an ε-fraction of reward due to sampling error.

Combining statement (1.1) with the fact that the instructions are non-faulty with probability at least $1 - \varepsilon$, the following can be derived:

$$\begin{aligned}\mathbb{E}[\mathsf{ALG}] &\geq \Pr[\{\mathscr{I}_t : t\} \text{ non-faulty}] \cdot \mathbb{E}[\mathsf{ALG}|\{\mathscr{I}_t : t\} \text{ non-faulty}] \\ &\geq (1 - \varepsilon)(\alpha - \varepsilon)\mathsf{OPT}.\end{aligned}$$

Therefore, this framework allows us to establish constant-factor approximation algorithms which are made polynomial-time by sampling.

1.3.2 Details of Sampling Framework pertaining to Chapter 7

We now illustrate this framework, by elaborating on how it is used to obtain a $(1/2 - \varepsilon)$-approximation for the stochastic knapsack problem in Chapter 7.

In the stochastic knapsack problem, n denotes the number of items and B denotes the number of time steps. For each time step $t \in [B]$, the instruction $\mathscr{I}_t$ consists of, for each item $i \in [n]$, an estimate of the probability that it has not been played before time t, under the previously-recorded instructions $\mathscr{I}_1, \ldots, \mathscr{I}_{t-1}$. (Again, the algorithm cares about these *probabilities* and not just the realization of whether item i was played before time t, since its aim is to satisfy a bound on expected reward.)

The algorithm would like all of its estimates to be within a *multiplicative* error bound of

ε with high probability. However, an additional challenge is faced when the error tolerance required is multiplicative, caused by rare[4] events.

To overcome this challenge, instead of fixing a constant number of samples M_t, we imagine the following sampling process. Let p denote the probability of interest and suppose we can sample from a Bernoulli(p) distribution until H "hits" (occurrences of probability p) are encountered. Then it is possible to guarantee with high probability that our estimate would be in $[(1-\varepsilon)p, (1+\varepsilon)p]$, because smaller probabilities would have required more samples before encountering a constant number of hits H. This is formalized in the following proposition, for which we provide a proof at the end of this section.

Proposition 1.3.1 (Hit Bound for Sampling). *Suppose we draw IID samples from a Bernoulli distribution with unknown probability $p > 0$ until we encounter H "hits" (occurrences of probability p). Let N be the random variable for the number of trials required. For all $\varepsilon \in (0, 1/4]$ and $\delta \in (0, 1]$, if $H \geq 4\ln(2/\delta)/\varepsilon^2$, then*

$$\Pr[(1-\varepsilon)p \leq \frac{H}{N} \leq (1+\varepsilon)p] \geq 1-\delta.$$

The caveat is that this sampling process could run indefinitely, and we have to cut off the sampling after some number of trials M if we want a polynomial run-time. Nonetheless, interpreting the sampling in terms of hits allows the algorithm to *detect* infinitesimal and zero probabilities, and respond accordingly when the cutoff of M is reached.

We now formally outline our proof of the $(1/2 - \varepsilon)$-approximation for the stochastic knapsack problem and how it uses the analysis framework from Section 1.3.1.

Definition 1.3.2. Define the following constants.

- ε: the multiplicative error bound within which we would like all of the algorithm's estimates to be, with high probability.
- H: an integer which achieves the multiplicative error bound of ε by Proposition 1.3.1, in the hypothetical scenario where samples can be drawn indefinitely.

[4]For any fixed number of times to sample M_t, if the probability of interest p is $o(1/M_t)$, then it is not possible for the sample average approximation to be in $[(1-\varepsilon)p, (1+\varepsilon)p]$ with high probability. Moreover, approximating these $o(1/M_t)$ probabilities is still relevant, because the reward when they occur could be e.g. $\Omega(M_t^2)$.

- τ: a threshold (equal to $\frac{\varepsilon}{4|S|B}$ in Chapter 7), where we will not attempt to accurately estimate probabilities smaller than this threshold.

- M: the constant number of trials within which all probabilities greater than τ will produce at least H hits, with high probability.

- def_t: the default value for each time t (equal to $\sum_{i=1}^{n} x_{\rho_i,t}/2$ in Chapter 7) used to approximate probabilities which do not produce H hits within M trials.

Definition 1.3.3. Define the following random variables for each item $i \in [n]$ and time step $t \in [B]$.

- $\mathsf{Free}(i,t)$: the probability of item i not having been played before time t, which is a random variable whose value is determined by the realizations of the instructions $\mathscr{I}_1, \ldots, \mathscr{I}_{t-1}$.

- $N_{i,t}$: the number of trials required when a Bernoulli($\mathsf{Free}(i,t)$) distribution is sampled until H hits are encountered. Defined to be ∞ if $\mathsf{Free}(i,t) = 0$.

- $\mathsf{Free}^{\mathsf{emp}}(i,t)$: the algorithm's recorded estimate of $\mathsf{Free}(i,t)$ based on $N_{i,t}$, where the experiment is cut off after M trials. $\mathsf{Free}^{\mathsf{emp}}(i,t)$ is set to $H/N_{i,t}$ if the experiment for item i and time t terminates before the cutoff of M, and set to the default value def_t otherwise.

We now define $H = \lceil 4\ln(2nB/\varepsilon)/\varepsilon^2 \rceil$ and substitute $\delta = \frac{\varepsilon}{nB}$ into Proposition 1.3.1 to obtain that for any $i \in [n]$ and $t \in [B]$,

$$\Pr\left[\frac{H}{N_{i,t}} \notin [(1-\varepsilon)\mathsf{Free}(i,t), (1+\varepsilon)\mathsf{Free}(i,t)]\right] \leq \frac{\varepsilon}{nB}$$

(where H/∞ is defined to be 0). By the union bound, the event

$$(1-\varepsilon)\mathsf{Free}(i,t) \leq \frac{H}{N_{i,t}} \leq (1+\varepsilon)\mathsf{Free}(i,t), \qquad \forall\, i \in [n], t \in [B] \tag{1.2}$$

occurs with probability at least $1-\varepsilon$. We define the instructions to be *non-faulty* if the realizations of random variables $N_{i,t}$ and $\mathsf{Free}(i,t)$ satisfy (1.2). Note that the values of $N_{i,t}$ uniquely determine the values of $\mathsf{Free}^{\mathsf{emp}}(i,t)$, which constitute the instructions $\{\mathscr{I}_t : t \in [B]\}$.

Conditioning on non-faultiness allows for the following classification. For each time t and item i, if $\mathsf{Free}(i,t) > \tau$, then $N_{i,t} < \frac{H}{(1-\varepsilon)\tau}$ by (1.2). Thus, if we set the cutoff M to be $\lceil \frac{H}{(1-\varepsilon)\tau} \rceil$, then all probabilities $\mathsf{Free}(i,t)$ above the threshold of τ are large enough for the sampling to terminate without being cut off, and $\mathsf{Free}^{\mathsf{emp}}(i,t)$ would be set to $H/N_{i,t}$. On the other hand, if $\mathsf{Free}(i,t) \leq \tau$, then either the sampling is cut off and $\mathsf{Free}^{\mathsf{emp}}(i,t) = \mathsf{def}_t$, or it is also possible that $N_{i,t}$ is still no greater than M and $\mathsf{Free}^{\mathsf{emp}}(i,t) = H/N_{i,t}$.

The important consequence of this classification is that either $\mathsf{Free}^{\mathsf{emp}}(i,t)$ is set to $H/N_{i,t}$ and $\mathsf{Free}(i,t)$ is within the range $[\frac{1}{1+\varepsilon}\mathsf{Free}^{\mathsf{emp}}(i,t), \frac{1}{1-\varepsilon}\mathsf{Free}^{\mathsf{emp}}(i,t)]$, or $\mathsf{Free}^{\mathsf{emp}}(i,t)$ is set to def_t and $\mathsf{Free}(i,t)$ is at most τ. This establishes that the algorithm's instructions cause it to earn expected reward at least

$$\frac{(1-\varepsilon)^2}{(1+\varepsilon)} \cdot \frac{\mathsf{OPT}}{2}. \tag{1.3}$$

(The details of (1.3) are specific to the stochastic knapsack problem from Chapter 7 and unimportant in this discussion about sampling. It is possible to prove such a statement because we bound OPT from above by an LP-relaxation, and show that an algorithm with non-faulty instructions can imitate the LP solution to earn $\frac{(1-\varepsilon)^2}{2(1+\varepsilon)}$ of its reward.)

Combining (1.3) with the fact that the event (1.2) occurs (i.e. the instructions are non-faulty) with probability at least $1-\varepsilon$, a $\frac{(1-\varepsilon)^3}{2(1+\varepsilon)}$-approximation results. Redefining ε yields a $(1/2-\varepsilon)$-approximation, which is the main result of Section 7.3.

Finally, as promised, we provide the proof of Proposition 1.3.1 based on the multiplicative Chernoff bound.

Proof of Proposition 1.3.1. N is the sum of H geometric random variables of probability p, so its expectation if H/p (although note that H/N is actually a biased estimate of p). We separately show that both "bad" events $N > \frac{H}{(1-\varepsilon)p}$ and $N < \frac{H}{(1+\varepsilon)p}$ occur with probability at most $\delta/2$. The result will then follow from the union bound. We need to apply the following standard inequality (Mitzenmacher and Upfal, 2005, thm. 4.4–4.5) in our proof.

Proposition 1.3.4 (Multiplicative Chernoff Bound). *Let* $\mathsf{Bin}(n,p)$ *be a random variable* *for the sum of n IID Bernoulli random variables of probability p. Then*

$$\Pr[\tfrac{\mathsf{Bin}(n,p)}{n} \geq (1+\varepsilon)p] \leq \exp(-np\varepsilon^2/3) \qquad \forall\, 0 < \varepsilon \leq 1;$$
$$\Pr[\tfrac{\mathsf{Bin}(n,p)}{n} \leq (1-\varepsilon)p] \leq \exp(-np\varepsilon^2/2) \qquad \forall\, 0 < \varepsilon < 1.$$

The event $N < \frac{H}{(1+\varepsilon)p}$ occurs if and only if we have already encountered H hits after trial $\lceil\frac{H}{(1+\varepsilon)p}\rceil - 1$. The probability of this occurring can be expressed as

$$\begin{aligned}
\Pr[\mathsf{Bin}(\lceil\tfrac{H}{(1+\varepsilon)p}\rceil - 1, p) \geq H] &= \Pr\left[\frac{\mathsf{Bin}(\lceil\frac{H}{(1+\varepsilon)p}\rceil - 1, p)}{\lceil\frac{H}{(1+\varepsilon)p}\rceil - 1} \geq \frac{H}{\lceil\frac{H}{(1+\varepsilon)p}\rceil - 1}\right] \\
&\leq \Pr\left[\frac{\mathsf{Bin}(\lceil\frac{H}{(1+\varepsilon)p}\rceil - 1, p)}{\lceil\frac{H}{(1+\varepsilon)p}\rceil - 1} \geq (1+\varepsilon)p\right] \\
&\leq \exp\left(-\frac{(\lceil\frac{H}{(1+\varepsilon)p}\rceil - 1)p\varepsilon^2}{3}\right) \qquad (1.4)
\end{aligned}$$

where the first inequality holds because $\lceil\frac{H}{(1+\varepsilon)p}\rceil - 1 \leq \frac{H}{(1+\varepsilon)p}$, and the second inequality holds by Proposition 1.3.4.

Meanwhile, the event $N > \frac{H}{(1-\varepsilon)p}$ occurs if and only if we have encountered strictly less than H hits after trial $\lfloor\frac{H}{(1-\varepsilon)p}\rfloor$. The probability of this occurring can be expressed as

$$\begin{aligned}
\Pr[\mathsf{Bin}(\lfloor\tfrac{H}{(1-\varepsilon)p}\rfloor, p) \leq H - 1] &\leq \Pr\left[\frac{\mathsf{Bin}(\lfloor\frac{H}{(1-\varepsilon)p}\rfloor, p)}{\lfloor\frac{H}{(1-\varepsilon)p}\rfloor} \leq \frac{H-1}{\frac{H}{(1-\varepsilon)p} - 1}\right] \\
&\leq \Pr\left[\frac{\mathsf{Bin}(\lfloor\frac{H}{(1-\varepsilon)p}\rfloor, p)}{\lfloor\frac{H}{(1-\varepsilon)p}\rfloor} \leq (1-\varepsilon)p\right] \\
&\leq \exp\left(-\frac{\lfloor\frac{H}{(1-\varepsilon)p}\rfloor p\varepsilon^2}{2}\right) \\
&\leq \exp\left(-\frac{(\lceil\frac{H}{(1+\varepsilon)p}\rceil - 1)p\varepsilon^2}{3}\right). \qquad (1.5)
\end{aligned}$$

The first inequality holds because $\lfloor\frac{H}{(1-\varepsilon)p}\rfloor \geq \frac{H}{(1-\varepsilon)p} - 1$. The second inequality holds because adding 1 to both the numerator and denominator of a fraction less than 1 can only increase the fraction. The third inequality holds by Proposition 1.3.4. The fourth inequality holds because $\lfloor\frac{H}{(1-\varepsilon)p}\rfloor \geq \lceil\frac{H}{(1+\varepsilon)p}\rceil - 1$ and $2 \leq 3$.

Now, it is easy to calculate that given $H \geq 4\ln(2/(1))/(1/4)^2 = 64\ln 2$, $\varepsilon \leq 1/4$, and

$p \leq 1$, the expression $(\lceil \frac{H}{(1+\varepsilon)p} \rceil - 1)p$ arising in both (1.4) and (1.5) is bounded from below by $3H/4$. Therefore, both (1.4) and (1.5) are bounded from above by

$$\begin{aligned} \exp(-\frac{H\varepsilon^2}{4}) &\leq \exp(-\frac{4\ln(2/\delta)}{\varepsilon^2} \cdot \frac{\varepsilon^2}{4}) \\ &= \frac{\delta}{2}. \end{aligned}$$

This completes the proof of the proposition. □

1.3.3 Application of Sampling Framework in Chronologically-later-to-appear Chapter 3

We explain how the sampling framework was used in the chronologically-later-to-appear work on personalized recommendation in Chapter 3.

In the recommendation at checkout problem, n denotes the number of items, and T denotes the number of time steps. At each time step $t \in [T]$, the algorithm constructs for each item $i \in [n]$ a sample average estimate $\hat{d}_i(t)$ of the number of times that i has been sold as a recommended add-on by the end of time t, given the previously-recorded instructions. The instruction $\mathscr{I}_t$ at each time t is a *protection list* which prevents items i for which $\hat{d}_i(t)$ is too large, relative to its starting inventory b_i, from being recommended as add-ons.

For this algorithm we only require an additive ε error bound in its estimates, so we are not concerned about small probabilities. The number of samples required M is fixed for each time step, and can be computed directly from the Chernoff-Hoeffding inequality.

One intricacy of this problem, however, is that the type the customer at time t is *online* information which is not revealed until the start of time t, for all t. We are interested in online algorithms which are guaranteed to earn a constant fraction of an offline benchmark, with this benchmark knowing all of the customer types in advance. In this setting, the algorithm being polynomial-time is in some sense less important, since the comparison focuses on online vs. offline algorithms, not polynomial-time vs. optimal algorithms.

Nonetheless, in the work in Chapter 3, we show how to convert an exponential-time 1/4-approximation ("1/4-competitive algorithm") to a polynomial-time $(1/4-\varepsilon)$-approximation using the same sampling framework. We outline the proof here.

Definition 1.3.5. For each time t and item i, define $\bar{d}_i(t)$ as the true expected number of units of i that have been sold as an add-on by the end of time t.

$\bar{d}_i(t)$ is treated as a random variable whose value is determined by the realizations of the instructions $\mathscr{I}_1, \ldots, \mathscr{I}_{t-1}$. Meanwhile, recall that $\hat{d}_i(t)$ is the random variable for the algorithm's estimate of $\bar{d}_i(t)$, and instruction $\mathscr{I}_t$ is determined by $\hat{d}_1(t), \ldots, \hat{d}_n(t)$. For the realized instruction set $\{\mathscr{I}_t : t \in [T]\}$ to be non-faulty, we only require that they were determined by $\{\hat{d}_i(t) : t \in [T], i \in [n]\}$ such that the *final* estimates satisfy

$$\hat{d}_i(T) - \varepsilon \leq \bar{d}_i(T) \leq \frac{b_i}{2}, \qquad \forall\, i \in [n]. \tag{1.6}$$

We would like event (1.6) to occur with probability at least $1-\varepsilon$. In the recommendation problem, it is not possible to define constants δ_t bounding the sampling error at each time t, because randomly-occurring events over the time horizon may require the algorithm to estimate additional probabilistic quantities using sampling. Nevertheless, it is possible to show the total number of quantities estimated cannot exceed $n(T+n)$, and hence if we set M such that each estimate is faulty with probability at most $\frac{\varepsilon}{n(T+n)}$, then by the union bound, the probability of having any fault cannot exceed ε.

The non-faultiness condition (1.6) ensures that at the end of the time horizon, for any item i, it was not recommended beyond its protection level of $b_i/2$ in actuality, and also that the algorithm did not overestimate its add-on sales by more than ε and thus overprotect it. By comparing to an LP-relaxation of the offline problem, it is possible to show that conditioned on non-faulty instructions, the online algorithm's expected revenue is at least $(1/4-\varepsilon)\mathsf{OPT}$. Therefore, the online algorithm's overall expected revenue is at least $(1-\varepsilon)(1/4-\varepsilon)\mathsf{OPT}$, which is the main result of Chapter 3.

1.3.4 Application of Sampling Framework in Chronologically-later-to-appear Chapter 4

We explain how the sampling framework was used in the chronologically-later-to-appear work on single-leg RM in Chapter 4.

Unlike Chapters 7 and 3, in Chapter 4, the instruction $\mathscr{I}_t$ at each time t is dependent on the realized state $\mathscr{X}_t$ (the remaining inventory). The sampling algorithm is trying to mimic the random decision (price) that a hypothetical algorithm would choose from the same state $\mathscr{X}_t$. The hypothetical algorithm's random decision from state $\mathscr{X}_t$ depends on its history and hence cannot be directly computed. Instead, we must sample runs of the hypothetical algorithm from the start until hitting a run where state $\mathscr{X}_t$ is reached, and then record the hypothetical algorithm's decision from state $\mathscr{X}_t$ as instruction $\mathscr{I}_t$. Some states are only reached with infinitesimal probabilities, and therefore the sampling must be cut off after a constant M_t number of trials, in which case instruction $\mathscr{I}_t$ is empty and thus faulty.

Using the fact that each instruction $\mathscr{I}_t$ is state-dependent, it is possible to bound the unconditional probability of $\mathscr{I}_t$ being faulty. Intuitively, this is because low-probability states that are likely to cause the sampling to be cut off are only reached with low probabilities in the first place. Indeed, we show that if $M_t = \Omega(t^2)$, then the probability δ_t of instruction $\mathscr{I}_t$ being faulty is $O(t^{-2})$. Since $\sum_t t^{-2}$ converges, after taking the union bound, we establish for an arbitrarily small ε that

$$\Pr[\{\mathscr{I}_t : t\} \text{ non-faulty}] \geq 1 - \varepsilon.$$

However, recall from the discussion in Section 1.3.1 that we also need to establish

$$\mathbb{E}[\mathsf{ALG}|\{\mathscr{I}_t : t\} \text{ non-faulty}] \geq (\alpha - \varepsilon)\mathsf{OPT}, \tag{1.7}$$

which becomes a lot more difficult when the instructions are state-dependent. In fact, inequality (1.7) as stated is false, because conditioning on non-faultiness greatly down-weighs low-probability states, which may be the high-revenue states that comprise most of the expectation of ALG.

To remedy this, in Chapter 4, we mark the first point of failure (if any) on each sample path consisting of states and instructions. We couple the execution of our algorithm with the hypothetical algorithm *before* these points of failure, and show that the revenue before these points is still collected. This allows us to establish that $(\alpha - \varepsilon)\mathsf{OPT}$ revenue is still obtained when the instructions are non-faulty, which yields a competitive ratio of $(\alpha - \varepsilon)$ with a polynomial-runtime algorithm (for details, see Section 4.3.3).

Chapter 2

Online Resource Allocation under Arbitrary Arrivals: Optimal Algorithms and Tight Competitive Ratios

We consider the problem of allocating fixed resources to heterogeneous customers arriving sequentially. We study this problem under the framework of competitive analysis, which does not assume any predictability in the sequence of customer arrivals. Previous work has culminated in optimal algorithms under two scenarios: (i) there are multiple resources, each of which yields reward at a constant rate when allocated; or (ii) there is a single resource, which yields reward at different rates when allocated to different customers.

In this chapter, we derive optimal allocation algorithms when there are multiple resources, each with multiple reward rates. Our algorithms are simple, intuitive, and robust against forecast error. Their tight competitive ratio cannot be achieved by combining existing algorithms, which consider the tradeoffs between multiple resources and multiple reward rates separately.

By showing how to integrate these tradeoffs while making allocation decisions, we expand the applicability of competitive analysis in many areas, such as online advertising, matching markets, and personalized e-commerce. We test our methodological contribution on the hotel data set of Bodea et al. (2009), where there are multiple resources (hotel rooms), each with multiple reward rates (fares at which the room could be sold). We find that applying our algorithms, in conjunction with algorithms which attempt to forecast and learn the future transactions, results in the best performance.

2.1 Introduction

In this chapter we study a general online resource allocation problem, stated in revenue management terminology. A firm has multiple items, each with an unreplenishable starting inventory, and a set of feasible prices at which its units of inventory could be sold. Heterogeneous customers arrive sequentially over time. Upon a customer's arrival, the probability that she would buy each item at each price is revealed; these probabilities can be 0 for items she is not interested in, or prices that are too high. The firm then chooses an available item and feasible price to offer her, after which her purchase decision is immediately realized according to the probability given. The firm's goal is to maximize its expected revenue before the inventories run out, or there are no more customers.

A special case of our problem is the *deterministic case*, where all purchase probabilities are 0 or 1. In this case, the firm knows the maximum a customer is willing to pay for each item, possibly 0. Therefore, the firm's decision can be reduced to choosing an item to *assign* to the customer (charging her maximum willingness-to-pay for that item), or *rejecting* the customer if her willingness-to-pay is low for every item.

We study these problems under the framework of competitive analysis. In competitive analysis, no information is given about the sequence of customers, nor are they assumed to follow any observable pattern. The algorithm's performance is expressed as a fraction of an optimum which knows the complete customer sequence in advance. For $c \leq 1$, if an algorithm can guarantee that this fraction is at least c for every problem instance (and customer sequence), then it is said to *achieve a competitive ratio of c*. The goal is to develop robust algorithms which achieve the *optimal* competitive ratio, i.e. a ratio c^* such that no algorithm, without knowing the customer sequence in advance, can do better.

2.1.1 Previous Work in Competitive Analysis

Our model involves multiple items, as well as multiple feasible prices for each item. This combines two challenges in competitive analysis, which have previously been studied separately.

1. **Multiple Items**: The challenge of how to prioritize between multiple items, when a customer can only be offered (or assigned) one of them, has been considered in the online b-matching problem (Kalyanasundaram and Pruhs, 2000), Adwords problem (Mehta et al., 2007; Buchbinder et al., 2007), and online assortment problem (Golrezaei et al., 2014). The optimal algorithms for these problems all perform some kind of *inventory balancing*, placing lower priority on selling items with lower remaining inventory. Inventory balancing algorithms are also related to the *randomized ranking* algorithms used in the online bipartite matching problem (Karp et al., 1990; Aggarwal et al., 2011).

2. **Multiple Prices**: The challenge of when to reject a customer only willing to pay a low price, to preserve inventory for customers willing to pay higher prices, has been considered in the single-item, deterministic case of our problem (Ball and Queyranne, 2009; Lan et al., 2008). The optimal algorithm employs *booking limits*, rejecting customers with low willingness-to-pay once a threshold amount of the item has been sold.

Our model studies the challenges introduced when multiple prices are incorporated into the aforementioned problems with multiple items. In Section 2.6, we explain how our techniques can be extended to allow for fractional inventory consumption, like in the Adwords problem; or multiple items to be offered to each customer, like in the online assortment problem. We now discuss two additional ways to view our model, which emphasize the increase in modeling power from allowing for multiple prices:

- First, one can think of each of our (item, price)-combinations as an independent *product*. By allowing for multiple prices, we have allowed the multiple products corresponding to each item to draw from the same inventory, or *resource*. The different products can also consume different amounts of that resource, under the extension with fractional inventory consumption.

- Second, in some applications, the customers are classified under a finite number of *types*, and instead of a pricing decision, there is a different reward (corresponding to "match quality") for allocating each item to each customer type. This can be reduced to our problem, with the feasible prices for an item being that item's "match qualities" over all types. By allowing for multiple prices, we have allowed each item to yield different rewards when allocated to different types, as opposed to yielding the same reward for all types.

2.1.2 Integrating the Challenges

We introduce a *bid-price control* policy which achieves the optimal competitive ratio under both multiple items and multiple prices. Our algorithm maintains for each item a bid price, which is the value placed on one unit of its inventory. The *pseudorevenue* associated with an (item, price)-combination is then the price minus the value of the item. The algorithm offers to each customer the (item, price)-combination with the highest expected pseudorevenue, never offering combinations with non-positive pseudorevenue.

Bid-price control is a classical idea in revenue management (see Talluri and Van Ryzin (2006); Liu and Van Ryzin (2008)), where the bid prices are computed using an LP, based on the remaining inventory and forecasted distribution of remaining customers. However, since we make no assumptions about future customers, our bid prices are based on only the remaining inventory. Our bid prices are very simple—they are computed separately for each item i, like the multiplicative *penalties* in Golrezaei et al. (2014). Let w_i be the fraction of the starting inventory of i which has already been sold. At each point in time, the bid price of item i is set to $\Phi_i(w_i)$, where Φ_i is a *value function* dependent on the set of feasible prices for item i.

To illustrate our algorithm, we display the form of Φ_i for an example item i which could be sold at fares \$150 or \$450, in Figure 2-1. Note the following:

- As the fraction of item i sold increases over time, the value of one unit of inventory increases, hence the pseudorevenues associated with the feasible prices of item i decrease, and the bid-price algorithm places lower priority on offering/assigning item i. This captures the "inventory balancing" used to address the challenge of multiple items.

- Let α_i be the value at which $\Phi_i(\alpha_i) = 150$. The algorithm stops selling item i at the lower price of 150 once its fraction sold reaches α_i, because the pseudorevenue associated with the lower price is $150 - \Phi_i(w_i)$, which is non-positive for $w_i \geq \alpha_i$.

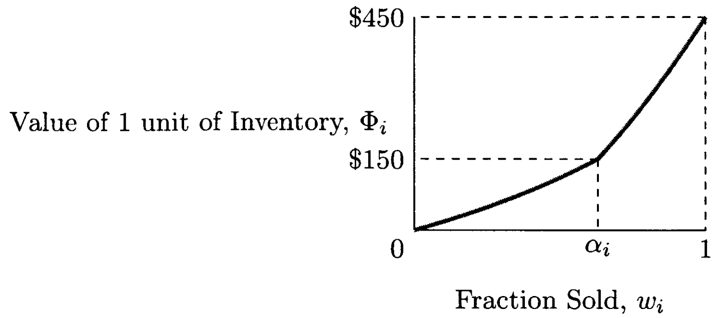


Figure 2-1: The value function for an item with feasible price set $\{150, 450\}$.

Therefore, our algorithm captures the "booking limits" used to address the challenge of multiple prices. The specific value of $\Phi_i(w_i)$ also tells the algorithm how to choose between a lower price which may have higher expected revenue, versus a higher price which has lower expected inventory consumption.

- Φ_i increases from 0 to the maximum price of 450 over [0,1], and is piecewise-convex.

In general, each value function Φ_i is designed to maximize the competitive ratio CR_i associated with it. As we will explain, the exact function Φ_i is defined as the solution to a differential equation arising from a primal-dual analysis.

The booking limits implied by such a Φ_i are different than the booking limits derived by Ball and Queyranne (2009) which are optimal when i is the single item being sold. For example, if item i has two prices, with r_i being the ratio of high to low price, then the value of α_i is $\alpha(r_i)$, where

$$\alpha(r) = \ln \frac{2(r-1)}{\sqrt{1+4r(r-1)/e}-1}. \tag{2.1}$$

Meanwhile, the optimal booking limit from the single-item case is $\frac{r_i}{2r_i-1}$. α_i is greater than $\frac{r_i}{2r_i-1}$, with the intuition being that with multiple items, there is less upside to reserving inventory for higher prices, because the reserved units may have to compete with other items to be sold. Indeed, when there are both multiple items and multiple prices, the optimal algorithm must integrate inventory balancing when setting booking limits, instead of using the single-item booking limits.

2.1.3 Competitive Ratio Results

The overall competitive ratio associated with our algorithm is $\min_i \mathsf{CR}_i$, being limited by the item i with the smallest value of CR_i. While this competitive ratio is not achieved by the exact bid-price algorithm specified in the previous subsection, we prove the following results in this chapter:

1. A variant of the bid-price algorithm, which we call MULTI-PRICE BALANCE, achieves a competitive ratio of $\min_i \mathsf{CR}_i$ in the *asymptotic regime*, where all starting inventories go to ∞.

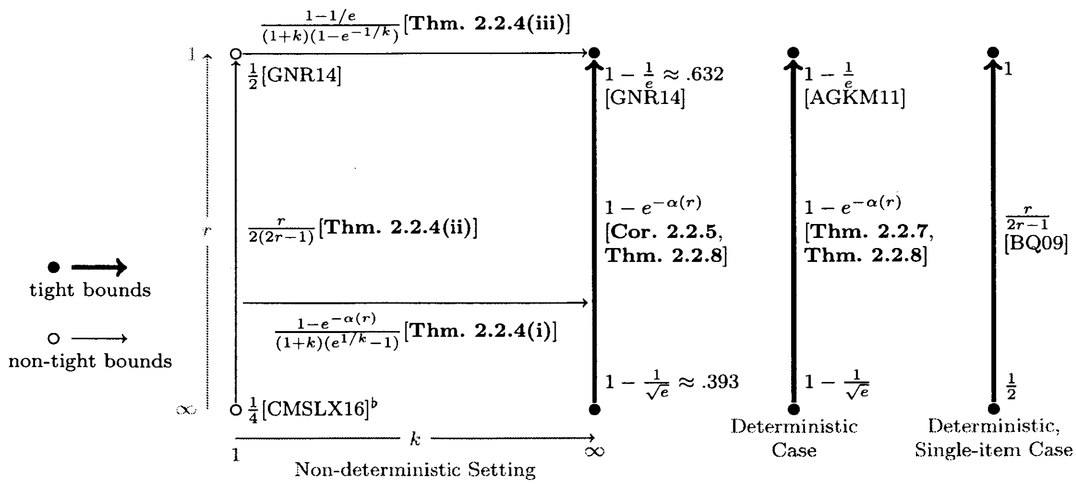
2. A different variant of the bid-price algorithm, which we call MULTI-PRICE RANKING, achieves a competitive ratio of $\min_i \mathsf{CR}_i$ in the *deterministic* case of our problem.

3. A counterexample, which can be made to fall under both the asymptotic regime and the deterministic case, shows that the competitive ratio of any algorithm cannot exceed $\min \mathsf{CR}_i$.

When there is a single feasible price for an item i, $\mathsf{CR}_i = 1 - \frac{1}{e}$. Our statements 1–3 are generalizations of results that exist when every item has only one price. Statement 1 corresponds to the inventory balancing algorithm of Golrezaei et al. (2014) achieving a competitive ratio of $1 - \frac{1}{e}$. Statement 2 corresponds to the ranking-based algorithm of Aggarwal et al. (2011) achieving the same competitive ratio. Statement 3 shows that both of these results are tight.

These results may not be tight in the non-asymptotic, non-deterministic setting, which is an important open problem (Devanur et al., 2013) in the single-price case as well. Nonetheless, we establish lower bounds on the competitive ratio achieved which hold in the non-deterministic setting, and are parametrized by k, the minimum starting inventory of an item. As k increases, these bounds sharply approach the tight guarantee of $\min_i \mathsf{CR}_i$ from the asymptotic regime. In the single-price case, our bounds show that the multiplicative gap from $1 - \frac{1}{e}$ is at most $(1 + k)(1 - e^{-1/k})$, which improves the previously-best-known gap from Golrezaei et al. (2014).

We illustrate our bounds on the case where every item has two feasible prices, in Figure 2-2. The competitive ratio CR_i associated with an item i is $1 - e^{-\alpha(r_i)}$, where r_i is its ratio of high to low price, and α is defined in (2.1). Thus the overall competitive ratio $\min_i \mathsf{CR}_i$



$\flat$ The smallest guarantee of $\frac{1}{4}$ in this diagram is also implied by the results of Chapter 3.

Figure 2-2: Competitive ratios achieved in the two-price case, where r denotes the maximum ratio of an item's high to low price, and k denotes the minimum starting inventory of an item. The guarantees improve from bottom to top (as r decreases), and from left to right (as k increases).

can be written as

$$1 - e^{-\alpha(r)}, \tag{2.2}$$

where $r = \max_i r_i$. (2.2) is decreasing in r. As $r \to 1$, $\alpha(r) \to 1$ and (2.2) approaches the known value of $1 - \frac{1}{e} \approx .632$. The smallest competitive ratio occurs as $r \to \infty$, with (2.2) approaching $1 - \frac{1}{\sqrt{e}} \approx .393$.

The formal statements of our theorems, which allow each item to have an arbitrary set of feasible prices, are deferred to Section 2.2. We analyze MULTI-PRICE BALANCE in Section 2.3 and MULTI-PRICE RANKING in Section 2.4. Descriptions of our techniques are also deferred to these sections.

In general, the tight competitive ratio of CR_i can approach 0 is the feasible price set for item i contains both a large number of prices and a large ratio from highest to lowest price, which is a known negative result (Aggarwal et al., 2011). Nonetheless, in many applications, one can enumerate the price points (e.g., an item which could only be sold at \$19.99 or \$24.99), or bound the ratio between the highest and lowest prices (e.g., an advertiser who bids between .1 and .2).

2.1.4 Application on Hotel Data Set of Bodea et al. (2009)

We first summarize the general benefits of applying competitive analysis, and the competitive algorithms derived from this research. In contrast to traditional algorithms, which optimize based on a forecast of future demand, or attempt to learn the demand, competitive algorithms hedge against some worst case, and operate without any demand information. Most immediately, they are useful for products with highly unpredictable demand (Ball and Queyranne, 2009; Lan et al., 2008), or for initializing new products with no historical sales data (Van Ryzin and McGill, 2000). Second, by eschewing stochastic processes for generating demand, competitive algorithms are usually simple and flexible, leading to clean insights about the problem (Borodin and El-Yaniv, 2005). Third, past research has reported on cases where competitive algorithms perform well in practice (Feldman et al., 2010), or on average in numerical experiments (Golrezaei et al., 2014).

In Section 2.7, we run simulations on the publicly-accessible hotel data set of Bodea et al. (2009). We use the product availability information to estimate customer choice models, and the transactional data as the sequence of arrivals. This leads to an online assortment problem like in Golrezaei et al. (2014), with multiple prices (advance-purchase rate, rack rate, etc.) for each item (King room, Two-double room, etc.). We compare the performance of our MULTI-PRICE BALANCE algorithm, using the extension discussed in Section 2.6 which can offer assortments, to various benchmarks and forecasting algorithms.

The main conclusion from our simulations is that the best performance is achieved by *hybrid* algorithms (see Golrezaei et al. (2014)). These are forecasting-based algorithms which continuously reference our forecast-independent value functions $\Phi_1, \ldots, \Phi_n$, and adjust their decisions accordingly. Although this only changes a small fraction ($\approx 5\%$) of decisions, these tend to be the decisions where the forecast is being most overconfident. Therefore, not only does this boost average performance, it drastically reduces the variance in performance caused when the forecast is wrong.

2.1.5 Other Related Work

We briefly discuss some related papers which has not been mentioned until now.

Alternate Approaches to Online Matching. Our problem captures the *online edge-weighted bipartite matching* problem, which has been studied under various settings designed to get around a basic impossibility result (see Aggarwal et al. (2011)). One such setting is *free disposal* (Feldman et al., 2009). Alternatively, one could assume that the arrivals appear in a *random order*, which allows for some form of learning (Kesselheim et al., 2013); this approach is very general and has been extended to online linear programming (Agrawal et al., 2014). However, to our knowledge, we are the first to focus on the *weight-dependent* competitive ratio for the online edge-weighted bipartite matching problem, instead of making assumptions such as free disposal or randomly-ordered arrivals. For a survey of online matching, we refer to Mehta (2013).

Known Stochastic Processes. When the stochastic process generating the arrivals in our problem is given, the resulting optimization problem is still computationally intractable. Nonetheless, many effective heuristics have been proposed, under different variations of the model (Zhang and Cooper, 2005; Jasin and Kumar, 2012; Ciocan and Farias, 2012; Chen and Farias, 2013). These heuristics can earn $\frac{1}{2}$ of the LP optimum in general settings (Chan and Farias, 2009; Wang et al., 2015; Gallego et al., 2015). Manshadi et al. (2012) derive an improved performance ratio when the given stochastic process is IID. From a modeling perspective, our problem with multiple items and multiple prices is similar to the multi-fare, parallel flights problem of Zhang and Cooper (2005), and the appointment scheduling with customer preferences problem of Wang et al. (2015).

Alternate Metrics. Competitive/approximation ratio both consider the algorithm's expected reward as a fraction of an LP optimum. Our problem has been analyzed under other metrics as well. When the arrival process is unknown but assumed to be IID, one popular metric is regret, which measures the *additive* loss from optimum (see Ferreira et al. (2016)). When the arrival process is known, the *fluid* and *diffusion* analysis approaches have also been used (see Reiman and Wang (2008)). However, unlike competitive ratio, these metrics all tend to focus on asymptotic performance as the number of customers grows to infinity. Finally, a recent metric which has been studied is *regret ratio* (Zhang et al., 2016). For a comprehensive review of different metrics to use under different models of demand (for a single item), we refer the reader to Araman and Caldentey (2011).

2.2 Problem Definition, Algorithm Sketch, and Theorem Statements

A firm is selling $n \in \mathbb{N}$ different items. Each item $i \in [n]$[1] starts with a fixed inventory of $k_i \in \mathbb{N}$ units, and could be offered at one of $m_i \in \mathbb{N}$ feasible prices, with corresponding fares $r_i^{(1)}, \ldots, r_i^{(m_i)} \in \mathbb{R}$ satisfying $0 < r_i^{(1)} < \ldots < r_i^{(m_i)}$. For convenience, we let $r_i^{(0)} = 0$ for each i. In Section A.5.1, we allow for a continuum of feasible prices in some range $[r^{\min}, r^{\max}]$.

There are $T \in \mathbb{N}$ customers arriving sequentially. Upon the arrival of customer $t \in [T]$, the firm observes $p_{t,i}^{(j)}$, the probability that customer t would buy item i at price j, for all $i \in [n]$ and $j \in [m_i]$.[2] The firm chooses up to one of the items i with inventory remaining, and offers it to customer t, at any price j. The customer accepts the offer with probability $p_{t,i}^{(j)}$, in which case the firm earns revenue $r_i^{(j)}$, and the inventory of item i is decremented by 1. In Section 2.6, we discuss models where multiple items can be offered or multiple units of inventory can be consumed at a time.

We define an *instance* $\mathcal{I}$ of the problem to consist of all of the following:

1. Initial information—$n, \{k_i, m_i, r_i^{(1)}, \ldots, r_i^{(m_i)} : i \in [n]\}$;

2. Arrival information—$T, \{p_{t,i}^{(j)} : t \in [T], i \in [n], j \in [m_i]\}$.

An *online algorithm* prescribes, based on the initial information, how to make the offering decision at each time t, without knowing $\{p_{t',i}^{(j)} : i \in [n], j \in [m_i]\}$ for future customers $t' > t$ nor the length of the time horizon T. For an online algorithm, let $\mathsf{ALG}(\mathcal{I})$ denote the revenue earned on a run on instance $\mathcal{I}$, which is a random variable with respect to the customers' purchase decisions as well as any coin flips in the algorithm.

[1] For a general positive integer b, let $[b]$ denote the set $\{1, \ldots, b\}$.

[2] These probabilities can be 0 for items the customer is not interested in, or prices that are too high. A rational customer would have $p_{t,i}^{(1)} \geq \ldots \geq p_{t,i}^{(m_i)}$, although we do not need this assumption.

Meanwhile, we can write the following LP based on instance $\mathcal{I}$:

$$\max \sum_{t=1}^{T}\sum_{i=1}^{n}\sum_{j=1}^{m_i} p_{t,i}^{(j)} r_i^{(j)} x_{t,i}^{(j)} \tag{2.3a}$$

$$\sum_{t=1}^{T}\sum_{j=1}^{m_i} p_{t,i}^{(j)} x_{t,i}^{(j)} \leq k_i \qquad i \in [n] \tag{2.3b}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{m_i} x_{t,i}^{(j)} \leq 1 \qquad t \in [T] \tag{2.3c}$$

$$x_{t,i}^{(j)} \geq 0 \qquad t \in [T], i \in [n], j \in [m_i] \tag{2.3d}$$

LP (2.3) encapsulates the execution of any algorithm, which could make full use of the arrival information at the start, on instance $\mathcal{I}$. $x_{t,i}^{(j)}$ represents the unconditional probability of the algorithm offering item i at price j to customer t. (2.3b) enforces that starting inventories are respected, while (2.3c) enforces that at most one combination of item and price is offered to each customer. Objective function (2.3a) represents the expected revenue earned by the algorithm. Let $\mathsf{OPT}(\mathcal{I})$ denote the optimal objective value.

The *competitive ratio* of the online algorithm is then defined to be

$$\inf_{\mathcal{I}} \frac{\mathbb{E}[\mathsf{ALG}(\mathcal{I})]}{\mathsf{OPT}(\mathcal{I})}. \tag{2.4}$$

We say that an algorithm *achieves a competitive ratio of* c if (2.4) is lower-bounded by c.

Given any fixed online algorithm, (2.4) considers the worst-case instance, including the worst-case arrival sequence. The goal for the algorithm is to hedge against the worst-case arrival sequence, possibly by using randomness. Definition (2.4) provides a guarantee on $\mathbb{E}[\mathsf{ALG}(\mathcal{I})]$ relative to any algorithm which could have been possible, due to the following result.

Lemma 2.2.1. $\mathsf{OPT}(\mathcal{I})$ *is an upper bound on the expected revenue of any algorithm, which could make full use of the arrival information at the start, on instance* $\mathcal{I}$.

The proof of Lemma 2.2.1 is deferred to Section A.1. The definition of OPT based on the LP is standard in problems with stochastic purchase realizations and arbitrary customer arrivals—we leave its justification to Mehta and Panigrahi (2012); Golrezaei et al. (2014).

In the *deterministic* case of our problem, every $p_{t,i}^{(j)}$ is 0 or 1. The problem can be simplified by letting $j_{t,i} = \max\{j \in [m_i] : p_{t,i}^{(j)} = 1\}$, with $j_{t,i} = 0$ if the set is empty, for all

$t \in [T]$ and $i \in [n]$. We say that item i is *assigned* to customer t to indicate that i is offered to customer t at price $j_{t,i}$, which results in a sale; there is no reason to offer any other price. Customer t can also be *rejected*, e.g. if $j_{t,i}$ is low for every i. In the deterministic case, the LP (2.3) is integral, so $\mathsf{OPT}(\mathcal{I})$ is *equal* to the revenue of the best algorithm knowing the arrival sequence at the start.

2.2.1 The Multi-price Value Function Φ_i

For an arbitrary item i, we specify its value function Φ_i, which is dependent on its feasible prices $r_i^{(1)}, \ldots, r_i^{(m_i)}$. Recall that Φ_i is a function of w_i, the fraction of item i sold. For $w_i \in [0,1]$, $\Phi_i(w_i)$ is the value the algorithm currently places on one unit of inventory of i.

First we define booking limits $\alpha_i^{(1)}, \ldots, \alpha_i^{(m_i)}$, which are the fractions of starting inventory "reserved" for the respective fares $r_i^{(1)}, \ldots, r_i^{(m_i)}$, via the following proposition.

Proposition 2.2.2. *Consider any item i, with an arbitrary number of discrete prices satisfying $0 < r_i^{(1)} < \ldots < r_i^{(m_i)}$. There are unique positive values $\alpha_i^{(1)}, \ldots, \alpha_i^{(m_i)}$ which sum to 1 and satisfy*

$$1 - e^{-\alpha_i^{(1)}} = \frac{1}{1 - r_i^{(1)}/r_i^{(2)}} \cdot (1 - e^{-\alpha_i^{(2)}}) = \ldots = \frac{1}{1 - r_i^{(m_i-1)}/r_i^{(m_i)}} \cdot (1 - e^{-\alpha_i^{(m_i)}}). \tag{2.5}$$

There are also unique positive values $\sigma_i^{(1)}, \ldots, \sigma_i^{(m_i)}$ which sum to 1 and satisfy

$$\sigma_i^{(1)} = \frac{1}{1 - r_i^{(1)}/r_i^{(2)}} \cdot \sigma_i^{(2)} = \ldots = \frac{1}{1 - r_i^{(m_i-1)}/r_i^{(m_i)}} \cdot \sigma_i^{(m_i)}. \tag{2.6}$$

Furthermore,

$$\alpha_i^{(1)} \geq \frac{1}{m_i}. \tag{2.7}$$

The proof of Proposition 2.2.2 is deferred to Section A.1. While finding the exact solution to (2.5) requires finding the roots of a degree-m polynomial, a numerical solution can easily be found via bisection search.

Proposition 2.2.2 contrasts $\alpha_i^{(1)}, \ldots, \alpha_i^{(m_i)}$ in (2.5) with the booking limits $\sigma_i^{(1)}, \ldots, \sigma_i^{(m_i)}$ in (2.6) originally derived by Ball and Queyranne (2009), which are optimal when i is the single item being sold. Inequality (2.7) says that the fraction of starting inventory reserved for the lowest fare is at least its "fair share", i.e. $1/m_i$ where m_i is the number of fares.

Booking limits $\alpha_i^{(1)}, \ldots, \alpha_i^{(m_i)}$ and value function Φ_i are a by-product of our analysis, and maximize the competitive ratio. Our method for deriving them is deferred to Section A.5. For now, we complete the definition of Φ_i:

Definition 2.2.3. For each item i, define the following based on $\alpha_i^{(1)}, \ldots, \alpha_i^{(m_i)}$:

- $L_i^{(j)}$: the sum $\sum_{j'=1}^{j} \alpha_i^{(j')}$, defined for all $j = 0, \ldots, m_i$ (note that $L_i^{(0)} = 0$ and $L_i^{(m_i)} = 1$);

- $\ell_i(\cdot)$: a function on $[0,1]$, where $\ell_i(w)$ is the unique $j \in [m_i]$ for which $w \in [L_i^{(j-1)}, L_i^{(j)})$ (note that $\ell_i(L_i^{(j)}) = j+1$ for $j = 0, \ldots, m_i - 1$; we define $\ell_i(L_i^{(m_i)})$ to be m_i).

The value function Φ_i is then defined over $w_i \in [0,1]$ by

$$\Phi_i(w_i) = r_i^{(\ell_i(w_i)-1)} + (r_i^{(\ell_i(w_i))} - r_i^{(\ell_i(w_i)-1)}) \frac{\exp(w_i - L_i^{(\ell_i(w_i)-1)}) - 1}{\exp(\alpha_i^{(\ell_i(w))}) - 1}. \tag{2.8}$$

An example of a value function Φ_i with 2 prices was plotted in Figure 2-1. In general, Φ_i is continuously increasing and piecewise-convex over m_i *segments* of lengths $\alpha_i^{(1)}, \ldots, \alpha_i^{(m_i)}$, separated by *segment borders* $L_i^{(0)}, \ldots, L_i^{(m_i)}$. For each j, Φ_i reaches the value of $r_i^{(j)}$ at $L_i^{(j)}$, hence price j stops being offered once the fraction sold w_i reaches $L_i^{(j)}$.

We will see that the competitive ratio CR_i associated with Φ_i is $1 - e^{-\alpha_i^{(1)}}$, which is optimal. When $m_i = 1$, it can be seen that $\alpha_i^{(1)} = 1$, $\Phi_i(w_i) = r_i^{(1)} \cdot \frac{e^{w_i} - 1}{e - 1}$, and $\mathsf{CR}_i = 1 - \frac{1}{e}$, which correspond to known results. The functions $\Phi_1, \ldots, \Phi_n$ facilitate the tradeoff between immediate reward and future inventory. We develop two algorithms, which use them in different ways.

2.2.2 Sketch of our MULTI-PRICE BALANCE and MULTI-PRICE RANKING Algorithms

We first sketch MULTI-PRICE RANKING, which is simpler. It assumes that $k_i = 1$ for all i, which does not lose generality since an item which starts with multiple units of inventory can be transformed into multiple disparate items. At the start, the algorithm fixes for each item i a random seed W_i, drawn independently and uniformly from $[0,1]$. It then treats $\Phi_i(W_i)$ as the bid price for the single unit of item i: it offers to each customer t the available item i and price j maximizing the expected pseudorevenue, $p_{t,i}^{(j)}(r_i^{(j)} - \Phi_i(W_i))$.

MULTI-PRICE RANKING hedges against the ambiguity in customer arrivals by using randomness, which is standard in competitive analysis. The random seed W_i determines the random minimum price at which the algorithm is willing to sell item i, as well as a random priority for selling i when the algorithm is choosing between multiple items.

We now sketch MULTI-PRICE BALANCE, which updates the bid price of each item i based on the fraction w_i of its k_i units which has been sold. However, the algorithm does not directly use $\Phi_i(w_i)$ as the bid price of item i, because w_i would always be a multiple of $\frac{1}{k_i}$, while the booking limits and segment borders which Φ_i is based on may not be multiples of $\frac{1}{k_i}$. Instead, the algorithm first uses a randomized scheme for rounding the booking limits to multiples of $\frac{1}{k_i}$.

Specifically, at the start, the algorithm fixes for each item i *random* segment borders $\tilde{L}_i^{(0)}, \ldots, \tilde{L}_i^{(m_i)}$, which are multiples of $\frac{1}{k_i}$ satisfying $0 = \tilde{L}_i^{(0)} \leq \ldots \leq \tilde{L}_i^{(m_i)} = 1$. These realizations imply a *random, perturbed* value function $\tilde{\Phi}_i$. Function $\tilde{\Phi}_i$ is defined on $\{0, \frac{1}{k_i}, \ldots, 1\}$, since the fraction sold w_i is always a multiple of $\frac{1}{k_i}$. Function $\tilde{\Phi}_i$ still satisfies $0 = \tilde{\Phi}_i(0) \leq \tilde{\Phi}_i(\frac{1}{k_i}) \leq \ldots \leq \tilde{\Phi}_i(1)$. The algorithm treats $\tilde{\Phi}_i(w_i)$ as the bid price for item i: it offers to each customer t the item i and price j maximizing the expected pseudorevenue

$$p_{t,i}^{(j)}\big(\tilde{\Phi}_i(\tilde{L}_i^{(j)}) - \tilde{\Phi}_i(w_i)\big). \tag{2.9}$$

In expression (2.9), the definition of pseudorevenue at price j is $\tilde{\Phi}_i(\tilde{L}_i^{(j)}) - \tilde{\Phi}_i(w_i)$, i.e. we have used $\tilde{\Phi}_i(\tilde{L}_i^{(j)})$ in place of $r_i^{(j)}$, so that the pseudorevenue is exactly 0 when $w_i = \tilde{L}_i^{(j)}$. However, the realized $\tilde{\Phi}_i$ will generally be close to Φ_i, so that $\tilde{\Phi}_i(\tilde{L}_i^{(j)}) \approx r_i^{(j)}$. In the asymptotic regime with $k_i \to \infty$, $\tilde{\Phi}_i = \Phi_i$ deterministically. However, for small k_i, optimizing a *randomized procedure* for initializing $\tilde{\Phi}_i$ (based on $r_i^{(1)}, \ldots, r_i^{(m_i)}$ as well as k_i) instead of having a deterministic Φ_i (based on only $r_i^{(1)}, \ldots, r_i^{(m_i)}$) allows us to achieve a greater competitive ratio.

2.2.3 Statements of Our Results

Theorem 2.2.4. MULTI-PRICE BALANCE *achieves a competitive ratio of* $\min_i \widetilde{\mathsf{CR}}_i$, *where for all* i, $\widetilde{\mathsf{CR}}_i$ *is lower-bounded by all of: (i)* $\frac{1-e^{-\alpha_i^{(1)}}}{(1+k_i)(e^{1/k_i}-1)}$*; (ii)* $\frac{\sigma_i^{(1)}}{2}$*; and (iii)* $\frac{1-e^{-1}}{(1+k_i)(1-e^{-1/k_i})}$*, if* $m_i = 1$.

Corollary 2.2.5. MULTI-PRICE BALANCE *achieves a competitive ratio approaching* $1 -$

$\exp(-\min_i \alpha_i^{(1)})$ *as each starting inventory* k_i *approaches* ∞.

Corollary 2.2.6. *Suppose that each item has at most* m *discrete prices and at least* k *units of starting inventory. Then the competitive ratio achieved by* MULTI-PRICE BALANCE *is lower-bounded by* $\frac{1-e^{-1/m}}{(1+k)(e^{1/k}-1)}$*, which approaches* $1-e^{-1/m}$ *as* k *approaches* ∞.

Theorem 2.2.4 is our general result, where for each i, $\widetilde{\mathsf{CR}}_i$ is the competitive ratio associated with the optimal randomized procedure for initializing $\tilde{\Phi}_i$, based on $r_i^{(1)}, \dots, r_i^{(m_i)}$ and k_i.

Lower bound (i) on $\widetilde{\mathsf{CR}}_i$ is attained by a randomized procedure which defines $\tilde{\Phi}_i$ based on the fixed function Φ_i. The numerator in (i) is a function of the feasible prices $r_i^{(1)}, \dots, r_i^{(m_i)}$, while the denominator is a function of the starting inventory k_i. Note that the denominator $(1+k_i)(e^{1/k_i}-1)$ decreases toward 1 as $k_i \to \infty$, resulting in Corollary 2.2.5. Corollary 2.2.6 is a further simplification of the bound presented, using inequality (2.7). Lower bound (ii) is attained from solving an optimization problem for the best randomized procedure, which is tractable when $k_i = 1$. Interestingly, the bound turns out to be based on the single-item booking limits $\sigma_i^{(1)}, \dots, \sigma_i^{(m_i)}$, instead of $\alpha_i^{(1)}, \dots, \alpha_i^{(m_i)}$. Lower bound (iii) is the improvement of (i) in the single-price case, where we have gained a factor of e^{1/k_i}. It simplifies and improves the bound from Golrezaei et al. (2014).

MULTI-PRICE BALANCE is formalized and Theorem 2.2.4 is proven in Section 2.3. We explain the ideas behind our primal-dual analysis, why we need *random* value functions, and how to overcome the resulting analytical challenges.

Theorem 2.2.7. MULTI-PRICE RANKING *achieves a competitive ratio of* $1 - \exp(-\min_i \alpha_i^{(1)})$ *in the deterministic case of our problem.*

MULTI-PRICE RANKING is formalized and Theorem 2.2.7 is proven in Section 2.4. While we described MULTI-PRICE RANKING as an algorithm for our general problem in Section 2.2.2, it is most amenable to analysis in the deterministic case. This is also true in the single-price case, as our analysis uses the framework of Devanur et al. (2013) and extends it to handle to multiple prices.

Theorem 2.2.8. *Consider a set of* m *prices satisfying* $0 < r^{(1)} < \dots < r^{(m)}$*, from which* $\alpha^{(1)}$ *and* $\sigma^{(1)}$ *are defined according to Proposition 2.2.2. Then there exists a distribution over instances* $\mathcal{I}$ *(a "randomized instance") with* $m_i = m$ *and* $r_i^{(1)} = r^{(1)}, \dots, r_i^{(m)} = r^{(m)}$

for each item i, on which no online algorithm can have expected revenue greater than $(1 - e^{-\alpha^{(1)}})\mathbb{E}[\mathsf{OPT}(\mathcal{I})]$. *Furthermore, for every instance $\mathcal{I}$ in the support of the distribution:*

1. *the starting inventories k_i can be made arbitrarily large;*

2. *$\mathcal{I}$ falls under the deterministic case of our problem.*

Theorem 2.2.8 is proven in Section 2.5. It implies that no online algorithm can achieve a competitive ratio greater than $1 - e^{-\alpha^{(1)}}$, via Yao's minimax principle (Yao, 1977). The counterexample can be made to satisfy the conditions of both Corollary 2.2.5 and Theorem 2.2.7, showing that these results are tight.

In our counterexample, a large number of customers arrive according to a random permutation, like in Karp et al. (1990); Mehta et al. (2007); Golrezaei et al. (2014). In our case, the customers are further split into m "phases", where the customers in phase j are willing to pay $r^{(j)}$ for any of the items they are interested in. The lengths of the phases are optimized by an adversary to minimize the competitive ratio.

Interestingly, on the related counterexamples from the literature (Karp et al., 1990; Mehta et al., 2007; Ball and Queyranne, 2009; Golrezaei et al., 2014), all (reasonable) algorithms have the same performance. On our counterexample, with the adversarially-optimized phase lengths, the unique optimal algorithm turns out to be our two algorithms. We say *unique* because MULTI-PRICE BALANCE and MULTI-PRICE RANKING converge toward the same algorithm as the starting inventories go to ∞; this phenomenon has also been noted in the single-price case by Aggarwal et al. (2011).

Proposition 2.2.9. *For $m \geq 2$ prices satisfying $0 < r^{(1)} < \ldots < r^{(m)}$, from which $\alpha^{(1)}$ and $\sigma^{(1)}$ are defined according to Proposition 2.2.2, the following inequalities hold:*

$$(1 - \frac{1}{e}) \cdot \sigma^{(1)} < 1 - e^{-\sigma^{(1)}} < 1 - e^{-\alpha^{(1)}}; \tag{2.10}$$

$$\frac{1}{1 + \ln \frac{r^{(m)}}{r^{(1)}}} < \sigma^{(1)}; \tag{2.11}$$

$$1 - e^{-\alpha} < 1 - e^{-\alpha^{(1)}}, \text{ where } \alpha \text{ is the unique solution to } 1 - e^{-\alpha} = \frac{1 - \alpha}{\ln \frac{r^{(m)}}{r^{(1)}}}. \tag{2.12}$$

Finally, Proposition 2.2.9, which is proven in Section A.1, puts our tight competitive ratio of $1 - e^{-\alpha^{(1)}}$ into perspective. $\sigma^{(1)}$ is the existing tight competitive ratio for a single item, while $1 - \frac{1}{e}$ is the existing tight competitive ratio for multiple items with one price each. (2.10) shows that our competitive ratio for multiple items with multiple prices is not a naive combination of the existing competitive ratios, and hence our algorithms cannot be obtained by combining existing algorithms.

With a single item whose price can take any value in the continuum $[r^{(1)}, r^{(m)}]$, the tight competitive ratio is $\frac{1}{1+\ln(r^{(m)}/r^{(1)})}$ (Ball and Queyranne, 2009). $1-e^{-\alpha}$, with α as defined in (2.12), is our corresponding competitive ratio when there are multiple items (α can be solved to equal $1 - W(Re^{R-1})/R$, where W is the inverse of the function $f(x) = xe^x$, and $R = \ln(r^{\max}/r^{\min})$—see Section A.5.1). (2.11) and (2.12) say that when the prices vary within a discrete subset of $[r^{(1)}, r^{(m)}]$, the competitive ratios can only be greater. (2.11) combined with (2.10) shows that our competitive ratio of $1 - e^{-\alpha^{(1)}}$ is $\Omega(\frac{1}{\log(r^{(m)}/r^{(1)})})$.

2.3 MULTI-PRICE BALANCE and the Proof of Theorem 2.2.4

MULTI-PRICE BALANCE, as sketched in Section 2.2.2, is formalized in Algorithm 1. For now, we consider a generic randomized procedure for initializing $\tilde{L}_i^{(0)}, \ldots, \tilde{L}_i^{(m_i)}$ and $\tilde{\Phi}_i$ in Step 1, which deterministically satisfies the following monotonicity conditions:

$$\tilde{L}_i^{(0)}, \ldots, \tilde{L}_i^{(m_i)} \in \{0, \frac{1}{k_i}, \ldots, 1\},\ 0 = \tilde{L}_i^{(0)} \leq \ldots \leq \tilde{L}_i^{(m_i)} = 1; \tag{2.13}$$

$$\tilde{\Phi}_i(0), \tilde{\Phi}_i(\frac{1}{k_i}), \ldots, \tilde{\Phi}_i(1) \in \mathbb{R},\ 0 = \tilde{\Phi}_i(0) \leq \tilde{\Phi}_i(\frac{1}{k_i}) \leq \ldots \leq \tilde{\Phi}_i(1). \tag{2.14}$$

Since $\tilde{\Phi}_i$ is non-decreasing, the expression $\tilde{\Phi}_i(\tilde{L}_i^{(j)}) - \tilde{\Phi}_i(\frac{N_i}{k_i})$ in (2.15) is non-positive once the number sold N_i reaches k_i. Therefore, Algorithm 1 never tries to offer an item i which has stocked out.

Theorem 2.3.1. *Suppose in Line 1 of Algorithm 1, for each $i \in [n]$, the segment borders $\tilde{L}_i^{(1)}, \ldots, \tilde{L}_i^{(m_i)}$ and value function $\tilde{\Phi}_i$ are randomly initialized in a way such that*

$$k_i(\tilde{\Phi}_i(\frac{N+1}{k_i}) - \tilde{\Phi}_i(\frac{N}{k_i})) + \tilde{\Phi}_i(\tilde{L}_i^{(j)}) - \tilde{\Phi}_i(\frac{N}{k_i}) \leq \frac{r_i^{(j)}}{F}, \quad j \in [m_i], N \in \{0, \ldots, \tilde{L}_i^{(j)}k_i - 1\}; \tag{2.16}$$

$$\mathbb{E}[\tilde{\Phi}_i(\tilde{L}_i^{(j)})] \geq r_i^{(j)}, \qquad j \in [m_i]. \tag{2.17}$$

Algorithm 1 MULTI-PRICE BALANCE

```
1: Initialize L̃_i^(0), ..., L̃_i^(m_i), Φ̃_i randomly and independently for each i ∈ [n]
2: N_i ← 0 for all i ∈ [n] (N_i tracks the total number of copies of item i sold, at any price)
3: for t = 1, 2, ... do
4:     Compute
           max_{i∈[n], j∈[m_i]} p_{t,i}^(j) (Φ̃_i(L̃_i^(j)) − Φ̃_i(N_i / k_i))        (2.15)
5:     if the value of (2.15) is strictly positive then
6:         Offer any item i_t^* and price j_t^* maximizing (2.15) to customer t
7:         if customer t accepts (occurring with probability p_{t,i_t^*}^(j_t^*)) then
8:             Z_t ← Φ̃_{i_t^*}(L̃_{i_t^*}^(j_t^*)) − Φ̃_{i_t^*}(N_{i_t^*} / k_{i_t^*}) (this is the pseudorevenue earned)
9:             N_{i_t^*} ← N_{i_t^*} + 1
10:        end if
11:    end if
12: end for
```

Where line 4 computes

$$\max_{i\in[n],j\in[m_i]} p_{t,i}^{(j)}(\tilde{\Phi}_i(\tilde{L}_i^{(j)}) - \tilde{\Phi}_i(\frac{N_i}{k_i})) \tag{2.15}$$

and line 8 sets $Z_t \leftarrow \tilde{\Phi}_{i_t^*}(\tilde{L}_{i_t^*}^{(j_t^*)}) - \tilde{\Phi}_{i_t^*}(N_{i_t^*}/k_{i_t^*})$.

Then Algorithm 1 achieves a competitive ratio of F.

Theorem 2.3.1 identifies conditions which, when satisfied by the randomized procedure for each i, yields a competitive ratio of F. Note that (2.16) needs to hold for every potential instantiation of $\tilde{\Phi}_i$, while (2.17) only needs to hold in expectation over the instantiations. We prove Theorem 2.3.1 in Section A.2, but outline its proof here and provide some intuition.

First, we take the dual of the LP (2.3):

$$\min \sum_{i=1}^{n} k_i y_i + \sum_{t=1}^{T} z_t \tag{2.18a}$$

$$p_{t,i}^{(j)} y_i + z_t \geq p_{t,i}^{(j)} r_i^{(j)} \qquad t \in [T], i \in [n], j \in [m_i] \tag{2.18b}$$

$$y_i, z_t \geq 0 \qquad i \in [n], t \in [T] \tag{2.18c}$$

By weak duality, $\mathsf{OPT}(\mathcal{I})$ is bounded from above by the objective value of any feasible dual solution.

During the (random) execution of Algorithm 1, it maintains a dual variable $y_i = \tilde{\Phi}_i(\frac{N_i}{k_i})$ for each i. At each time t, only if a sale is realized, does the algorithm set z_t to a non-zero value Z_t (Line 8) and increment the y_i-variables by incrementing $N_{i_t^*}$ (Line 9). We prove three claims:

1. During each time $t \in [T]$, the gain in the dual objective is at most some multiple $\frac{1}{F}$ of the revenue earned by the algorithm;

2. During each time $t \in [T]$, the conditional expectation of Z_t over the random purchase decision of customer t, combined with the current value of y_i, make the LHS of (2.18b) at least $p_{t,i}^{(j)} \cdot \tilde{\Phi}_i(\tilde{L}_i^{(j)})$, for all $i \in [n]$ and $j \in [m_i]$;

3. The expectation of $\tilde{\Phi}_i(\tilde{L}_i^{(j)})$, over the random segment borders and value function initially chosen by the algorithm, is at least $r_i^{(j)}$, for all $i \in [n]$ and $j \in [m_i]$.

Claim 1 follows from condition (2.16), while Claim 3 follows from condition (2.17). Claims 2 and 3 can be combined to show that the dual variables y_i and z_t maintained by the algorithm are feasible, after taking an expectation over all sample paths.

We explain the intuition behind our idea of a *random* value function, and the resulting analysis. Even for a single item, with a small starting inventory and a large ratio r from its highest to lowest price, in order to achieve a constant competitive ratio which does not scale with r, one must use *random* booking limits (Ball and Queyranne, 2009). With multiple items, our equivalent is to have the configuration of segment borders $\tilde{L}_i^{(0)}, \ldots, \tilde{L}_i^{(m_i)}$ be random, and define an arbitrary value function $\tilde{\Phi}_i$ corresponding to each one. In order to "average" over these configurations in the analysis, we relax dual feasibility to only hold in expectation. The idea of feasibility in expectation has been previously seen, but in different contexts: in Devanur et al. (2013), over a random seed, and in Golrezaei et al. (2014), over a random purchase decision (similar to our Claim 2).

2.3.1 Optimizing the Randomized Procedures

Theorem 2.3.1 reduces the problem of deriving a competitive algorithm to that of finding a randomized procedure for initializing $\tilde{\Phi}_1, \ldots, \tilde{\Phi}_n$ satisfying (2.16)–(2.17). We can consider this problem separately for each i, based on $r_i^{(1)}, \ldots, r_i^{(m_i)}$ and k_i, and omit the subscript i.

A randomized procedure consists of a distribution over the all of the configurations satisfying (2.13), and for each configuration, values for $\tilde{\Phi}(\frac{1}{k}), \tilde{\Phi}(\frac{2}{k}), \ldots, \tilde{\Phi}(1)$ satisfying (2.14). We would like to find a randomized procedure which satisfies (2.16)–(2.17) with a maximal value of F. While this optimization problem is intractable in general, we can use the intuition behind the definitions of $L^{(0)}, \ldots, L^{(m)}$ and Φ from Section 2.2.1 to specify a near-optimal randomized procedure.

Definition 2.3.2. Define the following randomized procedure for initializing $\tilde{\Phi}$:

1. Draw a random seed W uniformly from $[0, 1]$;

2. For each j, set $\tilde{L}^{(j)} = \frac{\lfloor L^{(j)}k \rfloor + 1}{k}$ if $W < L^{(j)}k - \lfloor L^{(j)}k \rfloor$, and $\tilde{L}^{(j)} = \frac{\lfloor L^{(j)}k \rfloor}{k}$ otherwise;

3. For $q \in \{0, \frac{1}{k}, \ldots, 1\}$, let $\tilde{\ell}(q)$ be the unique $j \in [m]$ such that $\tilde{L}^{(j-1)} \leq q < \tilde{L}^{(j)}$ (note that $\tilde{\ell}(\tilde{L}^{(j)}) = j + 1$ for $j = 0, \ldots, m - 1$; we define $\tilde{\ell}(\tilde{L}^{(m)})$ to be m).

The value function $\tilde{\Phi}$ is then defined over $q \in \{0, \frac{1}{k}, \ldots, 1\}$ by

$$\tilde{\Phi}(q) = \sum_{j=1}^{\tilde{\ell}(q)-1} (r^{(j)} - r^{(j-1)}) \frac{\exp(\tilde{L}^{(j)} - \tilde{L}^{(j-1)}) - 1}{\exp(\alpha^{(j)}) - 1} + (r^{(\tilde{\ell}(q))} - r^{(\tilde{\ell}(q)-1)}) \frac{\exp(q - \tilde{L}^{(\tilde{\ell}(q)-1)}) - 1}{\exp(\alpha^{(\tilde{\ell}(q))}) - 1}. \tag{2.19}$$

It is important that the random segment borders $\tilde{L}^{(0)}, \ldots, \tilde{L}^{(m)}$ are rounded *comonotonically* (in a perfectly positively correlated fashion) using a single seed, both to ensure that they satisfy the monotonicity condition in (2.13), and to reduce the number of potential configurations on which (2.16) needs to hold. $\tilde{\Phi}$ increases over the m (possibly empty) "segments" of its domain $\{0, \frac{1}{k}, \ldots, 1\}$, which are "bordered" by $\tilde{L}^{(0)}, \ldots, \tilde{L}^{(m)}$. (2.19) is similar to definition (2.8) for Φ, except the sum in (2.19) does not telescope, since $\tilde{L}^{(j)} - \tilde{L}^{(j-1)}$ equals $\alpha^{(j)}$ only in expectation.

Theorem 2.3.3. *The randomized procedure for initializing $\tilde{\Phi}$ from Definition 2.3.2 satisfies (2.16)-(2.17) with $F = \frac{1-e^{-\alpha^{(1)}}}{(1+k)(e^{1/k}-1)}$. Furthermore, if $m = 1$, then the value of F can be improved to $\frac{1-e^{-\alpha^{(1)}}}{(1+k)(1-e^{-1/k})}$.*

Theorem 2.3.3 is proven in Section A.2. It, in conjunction with Theorem 2.3.1, establishes bounds (i) and (iii) from our main result for MULTI-PRICE BALANCE, Theorem 2.2.4. In Section A.2, we state the complete proof of Theorem 2.2.4, including bound (ii), which involves explicitly formulating the optimization problem over randomized procedures and solving it when $k = 1$.

2.4 MULTI-PRICE RANKING and the Proof of Theorem 2.2.7

In Section 2.2.2, we sketched MULTI-PRICE RANKING, for our general problem. In Algorithm 2, we formalize it in the deterministic case, which is the case analyzed in Theorem 2.2.7. Recall that we have assumed, without loss of generality, that $k_i = 1$ for each item i.

Algorithm 2 MULTI-PRICE RANKING in the Deterministic Case

```
Initialize W_i uniformly at random from [0,1], independently for each i ∈ [n]
available_i ← true for all i ∈ [n]
for t = 1, 2, ... do
    Compute
```

$$\max_{i\in[n], j\in[m_i]:\mathsf{available}_i=\mathbf{true}} (r_i^{(j_{t,i})} - \Phi_i(W_i)) \tag{2.20}$$

```
    if the value of (2.20) is strictly positive then
        Offer any item i*_t maximizing (2.20) to customer t, at price j_{t,i*_t}
        available_{i*_t} ← false
    end if
end for
```

Our analysis extends the framework of Devanur et al. (2013) to incorporate multiple prices. It uses the dual LP defined in (2.18), where every $p_{t,i}^{(j)}$ is 0 or 1.

If Algorithm 2 assigns item i to customer t (charging price $j_{t,i}$), then we set dual variables $Z_t = r_i^{(j_{t,i})} - \Phi_i(W_i)$ and $Y_i = \Phi_i'(W_i)$, where Φ_i is the fixed function defined in Section 2.2.1 (we ignore the measure-zero set where Φ_i' is undefined). All dual variables not set during a time period are defined to be zero. The following lemmas are proven in Section A.3:

Lemma 2.4.1. *If Algorithm 2 assigns item i to customer t, then $(1-e^{-\alpha_i^{(1)}})(Y_i+Z_t) \le r_i^{(j_{t,i})}$ w.p.1.*

Lemma 2.4.2. *Setting $y_i = \mathbb{E}[Y_i], z_t = \mathbb{E}[Z_t]$ for all i, t forms a feasible solution to the dual LP (2.18).*

The proof of Theorem 2.2.7 is straight-forward given these lemmas:

Proof. Proof of Theorem 2.2.7. Lemma 2.4.2 implies $\mathsf{OPT}(\mathcal{I}) \le \sum_{i=1}^n \mathbb{E}[Y_i] + \sum_{t=1}^T \mathbb{E}[Z_t]$, via weak duality. However, by Lemma 2.4.1, the revenue earned by Algorithm 2, or ALG, is at least $\min_{i\in[n]}\{1 - e^{-\alpha_i^{(1)}}\} \cdot \left(\sum_{i=1}^n Y_i + \sum_{t=1}^T Z_t\right)$, with probability 1. Thus, $\mathbb{E}[\mathsf{ALG}] \ge (1 - \exp(-\min_{i\in[n]} \alpha_i^{(1)})) \cdot \mathsf{OPT}(\mathcal{I})$. □

2.5 Randomized Instance and the Proof of Theorem 2.2.8

We formalize the randomized instance described in Section 2.2.3 and use it to prove Theorem 2.2.8.

The $n \in \mathbb{N}$ items i all have $m_i = m$, $r_i^{(j)} = r^{(j)}$ for all j, and $k_i = k$ for some $k \in \mathbb{N}$. We think of n as going to ∞, while k is arbitrary. Throughout this example, we often express quantities as portions τ of n. We abuse notation and write τn to refer to an integer, even if τ is irrational, since the error from rounding τn to the nearest integer is negligible as $n \to \infty$.

The arrival sequence is randomized following the classical construction of Karp et al. (1990). There are $T = nk$ customers, split into n "groups" of k identical customers each. Uniformly draw a random permutation $\pi = (\pi_1, \ldots, \pi_n)$ of $(1, \ldots, n)$ from the $n!$ possibilities. For $i \in [n]$, all k customers in group i would deterministically buy any item in $\{\pi_i, \ldots, \pi_n\}$. Our construction differs from existing ones in that the n groups of customers are further split into m "phases". Let $\beta_1, \ldots, \beta_m$ be positive numbers summing to 1, corresponding to the fraction of groups in each phase, whose values we specify later. For all $j \in [m]$, the customers in groups $(\beta_1 + \ldots + \beta_{j-1})n + 1, \ldots, (\beta_1 + \ldots + \beta_j)n$ are willing to pay $r^{(j)}$ for any of the items in their interest set.

Definition 2.5.1. Define the following shorthand notation for all $j = 1, \ldots, m+1$:

- $A_j := \sum_{\ell=j}^{m} \alpha^{(\ell)}$ (note that $A_1 = 1$ and $A_{m+1} = 0$);
- $B_j := \sum_{\ell=j}^{m} \beta_\ell$ (note that $B_1 = 1$ and $B_{m+1} = 0$).

Proposition 2.5.2. *Given $m \in \mathbb{N}$, $0 < r^{(1)} < \ldots < r^{(m)}$, and $\alpha^{(1)}, \ldots, \alpha^{(m)}$ as defined in Proposition 2.2.2, there exists a unique solution to the following system of equations in variables $B_2, \ldots, B_m$:*

$$B_m r^{(m)} e^{-\alpha^{(m)}} = \ldots = B_2 r^{(2)} e^{-\alpha^{(2)}} = r^{(1)} e^{-\alpha^{(1)}}, \tag{2.21}$$

with $0 < B_m < \ldots < B_2 < B_1 = 1$.

We define $B_2, \ldots, B_m$ according to Proposition 2.5.2. This implies definitions for $\beta_1, \ldots, \beta_m$, which are strictly positive and sum to 1.

Now, regardless of the permutation π, the optimal algorithm allocates the k copies of item π_i to the customers in group i, for each $i \in [n]$, successfully serving all $T = nk$ customers and earning revenue $\sum_{j=1}^{m} r^{(j)}(\beta_j n)k$. This is also the optimal objective value of the LP (2.3). Therefore, $\mathsf{OPT}(\mathcal{I}) = \sum_{j=1}^{m} r^{(j)}(\beta_j n)k$ deterministically, which we can rewrite as

$$\sum_{j=1}^{m} (r^{(j)} - r^{(j-1)}) B_j nk. \tag{2.22}$$

2.5.1 Upper Bound on Performance of Online Algorithms

Lemma 2.5.3. *The expected revenue of an online algorithm on this randomized instance is upper-bounded by the maximum value of*

$$\sum_{j=1}^{m} r^{(j)} B_j n(1-e^{-\lambda_j})k \tag{2.23}$$

subject to $0 \leq \lambda_j \leq \ln \frac{B_j}{B_{j+1}}$ *for* $j \in [m-1]$, $0 \leq \lambda_m$, *and* $\sum_{j=1}^{m} \lambda_j \leq 1$.

Lemma 2.5.3 drastically simplifies the analysis of the online algorithm, because it restricts to algorithms which are *indifferent* to the realized permutation π, allowing for a deterministic analysis. However, our analysis differs from existing ones (e.g. (Golrezaei et al., 2014, Lem. 6)) in that despite the item symmetry, the online algorithm has a decision—how many customers in each phase to serve, as opposed to reserving inventory for customers in future phases.

This is controlled by the λ-variables, where λ_j denotes the expected fraction of item π_n's inventory sold to phase-j customers. The expected number of groups served during phase j is then at most $B_j n(1-e^{-\lambda_j})$, resulting in the upper bound (2.23). Constraint $\lambda_j \leq \ln \frac{B_j}{B_{j+1}}$ comes from the fact that $B_j n(1-e^{-\lambda_j})$ must not exceed the total number of groups in phase j, $\beta_j n$.

Lemma 2.5.4. *Let* $j \in [m]$ *and* $\tau \in [0,1]$. *The maximum value of*

$$\sum_{\ell=j}^{m} r^{(\ell)} B_\ell n(1-e^{-\lambda_\ell})k \tag{2.24}$$

subject to $\lambda_\ell \geq 0$ *for all* $\ell = j, \ldots, m$ *as well as* $\sum_{\ell=j}^{m} \lambda_\ell \leq \tau$ *is*

$$nk \sum_{\ell=j}^{m} r^{(\ell)} B_\ell \Big(1 - \exp\big(-\alpha^{(\ell)} + \frac{A_j - \tau}{m-j+1}\big)\Big). \tag{2.25}$$

Lemma 2.5.4 establishes the optimal objective value of the optimization problem from Lemma 2.5.3. The upper bound of $\ln \frac{B_j}{B_{j+1}}$ on λ_j for $j \in [m-1]$ turns out to not be binding. With both lemmas, the proof of Theorem 2.2.8 is easy.

Proof. Proof of Theorem 2.2.8. The value of (2.25) with $j = 1$ and $\tau = 1$ is

$$nk \sum_{\ell=1}^{m} r^{(\ell)} B_\ell (1 - e^{-\alpha^{(\ell)}}) = (1 - e^{-\alpha^{(1)}}) \sum_{\ell=1}^{m} (r^{(\ell)} - r^{(\ell-1)}) B_\ell n k, \tag{2.26}$$

where we have used (2.5) to derive the equality. Combining Lemmas 2.5.3–2.5.4, we get that the RHS of (2.26) is an upper bound on $\mathbb{E}[\mathsf{ALG}(\mathcal{I})]$, for any online algorithm. Meanwhile, $\mathsf{OPT}(\mathcal{I})$ on this randomized instance is always equal to (2.22), which is exactly the RHS of (2.26) divided by $(1-e^{-\alpha^{(1)}})$. We have established that $\mathbb{E}[\mathsf{ALG}(\mathcal{I})] \leq (1-e^{-\alpha^{(1)}})\mathbb{E}[\mathsf{OPT}(\mathcal{I})]$, which is the desired result. Finally, the second condition of Theorem 2.2.8 is clearly satisfied; the first condition is also satisfied because our analysis holds for any value of $k \in \mathbb{N}$, hence k can be made arbitrarily large. $\square$

Remark 2.5.5. Suppose $k \to \infty$. It can be seen that our algorithm (either MULTI-PRICE BALANCE or MULTI-PRICE RANKING, which behave identically on this instance—see Aggarwal et al. (2011)), with booking limits $\alpha^{(1)}, \ldots, \alpha^{(m)}$, is the unique optimal algorithm on this instance. The proof of Lemma 2.5.3 shows that given $\lambda_1, \ldots, \lambda_m$, the dominant strategy for the online algorithm is to deplete the inventories of items evenly (which is possible since $k \to \infty$), in which case upper bound (2.23) is attained. The proof of Lemma 2.5.4 shows that the unique optimal values for $\lambda_1, \ldots, \lambda_m$ are $\alpha^{(1)}, \ldots, \alpha^{(m)}$.

It only remains to show that $\lambda_j = \alpha^{(j)}$ is feasible, namely $\alpha^{(j)} \leq \ln \frac{B_j}{B_{j+1}}$ for $j < m$. Applying (2.21), this is equivalent to showing $e^{-\alpha^{(j)}} \geq \frac{r^{(j)} e^{-\alpha^{(j)}}}{r^{(j+1)} e^{-\alpha^{(j+1)}}}$, or $e^{-\alpha^{(j+1)}} \geq \frac{r^{(j)}}{r^{(j+1)}}$, which follows from (2.5) since $1 - e^{-\alpha^{(1)}} \leq 1$.

2.6 Extending our Techniques

We explain how our techniques can be extended to allow for fractional inventory consumption like in the Adwords problem (Mehta et al., 2007), or offering multiple items like in the online assortment problem (Golrezaei et al., 2014). The extension to continuous price sets in deferred to Section A.5.1.

Consider the following modification of our problem from Section 2.2: when customer t is offered item i at price j, she *deterministically* pays $p_{t,i}^{(j)} r_i^{(j)}$ and consumes a fractional amount $p_{t,i}^{(j)} \leq 1$ of item i's inventory, instead of paying $r_i^{(j)}$ and consuming 1 unit with probability $p_{t,i}^{(j)}$. We assume that $\min_i k_i \to \infty$. This generalizes the Adwords problem under the small bids assumption, by allowing each budget i to be depleted at m_i different rates $r_i^{(1)}, \ldots, r_i^{(m_i)}$.

For this problem, we use MULTI-PRICE BALANCE, except since we are taking $\min_i k_i \to \infty$, we can deterministically set each $\tilde{\Phi}_i = \Phi_i$. The three claims used to establish Theorem 2.3.1 are simpler: Claim 2 now holds deterministically instead of requiring a conditional expectation over Z_t, while Claim 3 also holds deterministically since $\tilde{\Phi}_i$ is always Φ_i. In Theorem 2.3.3, condition (2.16) is now only satisfied under an additional error term ε, since N is no longer a discrete integer. Nonetheless, the rounding error ε approaches 0 as $k_i \to \infty$, so the optimal competitive ratio is still achieved.

For online assortment, we use the term *product* to refer to an (item, price)-combination (i, j). Consider the following modification of our problem from Section 2.2: upon the arrival of customer t, for any subset (assortment) S of products and $(i, j) \in S$, we are given $p_{t,i}^{(j)}(S)$, the probability that customer t would pick product (i, j) when offered the choice from S. After being given these probabilities, we must offer an assortment S to customer t. This generalizes the original online assortment problem, by allowing each item to have multiple feasible prices. The execution of an algorithm can be encapsulated by the following modification of the LP (2.3):

$$\max \sum_{t=1}^{T} \sum_{S} x_t(S) \sum_{(i,j)\in S} r_i^{(j)} p_{t,i}^{(j)}(S) \tag{2.27a}$$

$$\sum_{t=1}^{T} \sum_{S} x_t(S) \sum_{j:(i,j)\in S} p_{t,i}^{(j)}(S) \leq k_i \qquad i \in [n] \tag{2.27b}$$

$$\sum_{S} x_t(S) = 1 \qquad t \in [T] \tag{2.27c}$$

$$x_t(S) \geq 0 \qquad t \in [T], S \subseteq \{(i,j) : i \in [n], j \in [m_i]\} \tag{2.27d}$$

MULTI-PRICE BALANCE can be directly applied to this problem, with the change that it offers the *assortment* S maximizing expected pseudorevenue, $\sum_{(i,j)\in S} p_{t,i}^{(j)}(S)\big(\tilde{\Phi}_i(\tilde{L}_i^{(j)}) - \tilde{\Phi}_i(w_i)\big)$, to each customer t. In the analysis, dual constraints (2.18b) now require $z_t \geq \sum_{(i,j)\in S} p_{t,i}^{(j)}(S)(r_i^{(j)} - y_i)$ for all t and S, which is still implied by the conditions of Theorem 2.3.1 so long as the choice probabilities for customers satisfy a mild *substitutability* assumption (see Golrezaei et al. (2014) for details).

2.7 Simulations on Hotel Data Set of Bodea et al. (2009)

We test our algorithms on the publicly-accessible hotel data set collected by Bodea et al. (2009). Based on the data, we consider a multi-price online assortment problem, as defined in Section 2.6. In general, we aim to follow the experimental setup of Golrezaei et al. (2014).

2.7.1 Experimental Setup

We consider Hotel 1 from the data set, which has more transactions than the other four hotels. For each transaction, we use *booking* to refer to the date the transaction occurred, and *occupancy* to refer to the dates the customer will stay in the hotel. We consider occupancies spanning the 5-week period from Sunday, March 11th, 2007 to Sunday, April 15th, 2007. Although the data contains occupancies for a couple of weeks outside this range, such transactions are sparse.

We merge the different rooms into 4 categories: King rooms, Queen rooms, Suites, and Two-double rooms. Rooms under the same category draw from the same inventory. We merge the different fare classes into two: discounted advance-purchase fares and regular rack rates. We use *product* to refer to any of the 8 combinations formed by the 4 room categories and 2 fares.

We estimate a Multinomial Logit (MNL) choice model on these 8 products, for each of 8 customer types. The customer types are based on the booking channel, party size, and VIP status (if any) associated with a transaction. These types capture preference heterogeneity (for example, party sizes greater than 1 tend to prefer Suites and Two-double rooms). The details of our choice estimation are deferred to Section A.6.

We should point out that more sophisticated segmentation and estimation techniques have been employed on this data set (van Ryzin and Vulcano, 2014; Newman et al., 2014). Nonetheless, MNL has been reported to perform relatively well (van Ryzin and Vulcano, 2014, sec. 5.2). The MNL choice model is convenient for our purposes because under it, both the assortment optimization problem, as well as the choice-based LP (2.27) with exponentially many variables, can be solved efficiently (Talluri and Van Ryzin, 2004; Liu and Van Ryzin, 2008; Cheung and Simchi-Levi, 2016).

We treat each occupancy date as a separate instance of the problem, for which we define a sequence of arrivals, with one arrival for each transaction which occupies that date. The choice probabilities for each arrival are determined by the customer type associated with the corresponding transaction.[3] The number of days in advance of occupancy that each arrival occurred is also recorded, but this information is only relevant for algorithms which attempt to forecast the remaining number of arrivals based on the remaining length of time.

Before we proceed, we discuss the limitations of our analysis and the data set:

1. In the data set, 55% of the transactions occupy multiple, *consecutive* days. However, we treat such a transaction as a separate arrival in the instances for each of those occupancy dates. While this is a simplifying assumption, the focus of our paper is on the basic allocation problem without complementarity effects across consecutive days, and our goal in using the data set is to extract an arrival pattern over time.

2. It is not possible to deduce from the data the fixed capacity for each category of room. Nonetheless, we consider a wide range of starting capacities in our tests.

3. Estimating the number of customers who do not make a purchase is a standard challenge in choice modeling, which is exacerbated in this data set by the fact that the arrivals are rather non-stationary. We test various assumptions on the weight of the no-purchase option in the MNL model for each customer type. In general, we assume that this weight is large, which causes the revenue-maximizing assortments to be large, allowing for tension between offering large assortments which maximize immediate revenue, and offering small assortments which regulate inventory consumption (details in Section A.6).

2.7.2 Instance Definition

An instance consists of a fixed capacity for each room category, corresponding to a specific occupancy date. Each customer interested in that occupancy date arrives in sequence, after which her characteristics (channel, party size, VIP status) are revealed. The problem is to show a *personalized* assortment of (room, fare)-options to each customer. The instances we test are defined below.

- Arrival sequence: 35 possibilities, one for each day in the 5-week occupancy period. We multiply the arrivals by 10 (i.e. instead of a type-1 customer followed by a type-

[3]The choice realized in that transaction was used for choice model estimation, but is not used in defining the arrival.

Table 2.1: Details on Room Categories and Fares

Room Category	Low Fare	High Fare	Fraction of Rooms
King	\$307	\$361	52%
Queen	\$304	\$361	15%
Suite	\$384	\$496	13%
Two Double	\$306	\$342	20%

2 customer, we have 10 type-1 customers followed by 10 type-2 customers), being interested in the high-inventory regime. After multiplication, the average number of arrivals per day is 1340, peaking on Sundays and Mondays, although the number and breakdown of customers varies every day.

- Number of products: 8 (room, fare)-combinations, identical for all instances.
- Prices of products: displayed in Table 2.1, identical for all instances. These prices were determined by taking the average price of that (room, fare)-combination over all transactions.
- Starting inventories: 3 possibilities, defined by the *loading factor*, which is the average number of customers per unit of starting inventory. We use the same loading factors (1.4, 1.6, 1.8) as Golrezaei et al. (2014). The breakdown of starting inventory is fixed, based on the relative frequency with which each room type is booked over all transactions (see Table 2.1).

We test additional synthetic instances, with greater differentiation between high and low fares and a greater range of loading factors, in Section 2.7.5.

2.7.3 Algorithms Compared

We compare the performances of 9 algorithms on each instance.

First we describe the forecast-independent algorithms we test.

1. **Myopic**: offer each customer the assortment maximizing immediate expected revenue, from the items that have not stocked out.
2. **Conservative**: only offer items at their maximum prices, using the optimal algorithm of Golrezaei et al. (2014) to choose assortments.

3. **Multi-price Balance**: offer each customer t the assortment S maximizing

$$\sum_{(i,j)\in S} p_{t,i}^{(j)}(S)(r_i^{(j)} - \Phi_i(w_i)), \tag{2.28}$$

   where w_i is the fraction of item i sold. Expression (2.28) is the expected *pseudorevenue* of assortment S. Since we are in the high-inventory regime, for simplicity we have used the fixed value function Φ_i, instead of the random $\tilde{\Phi}_i$, to define the bid price of each item i.

The Myopic and Conservative algorithms represent two extremes, where the former extracts the maximum in expectation from every customer and is optimal as the loading factor approaches 0, while the latter extracts the maximum from every unit of inventory and is optimal as the loading factor approaches ∞. In-between these extremes, our algorithm attempts to balance revenue-per-customer and revenue-per-item, as it chooses items and prices to put in the assortment.

Next we describe the forecasting-based algorithms. These algorithms all estimate the number of each type of customer yet to arrive, and then incorporate this information into the LP (2.27) to set bid prices. They differ in how they perform the forecasting, and how frequently they update the bid prices by re-solving the LP. Further details about these algorithms, as well as discussion of alternative algorithms, are deferred to Section A.6.1.

4. **One-shot LP**: solve the LP only once, at the start, using the average number of customers of each type to appear on a given day.

5. **LP Resolving**: re-solve the LP every 100 arrivals, using updated forecasts and inventory counts. During each re-solve, the estimated number of remaining customers is updated, taking into account the length of time remaining until occupancy, and the number of customers that have arrived. The estimated type breakdown is fixed, based on the aggregate distribution.

6. **LP Learning**: same as LP Resolving, except the estimated type breakdown is also updated, based on the empirical distribution observed thus far.

7. **LP Clairvoyant**: same as LP Resolving, but given the true number of customers of each type remaining.

Finally, we describe the hybrid algorithms we test. These algorithms combine a forecasting algorithm with MULTI-PRICE BALANCE, based on a parameter $\gamma > 1$. For each customer t, the hybrid algorithm considers the expected pseudorevenue (as defined in (2.28)) of the assortment S^{fcst} suggested by the forecasting algorithm. If this is at least $\frac{1}{\gamma}$ of the maximum value of (2.28) over all assortments S, then the hybrid algorithm offers S^{fcst}. Otherwise, the hybrid algorithm offers the assortment suggested by MULTI-PRICE BALANCE, which maximizes (2.28).

8. **Resolve-1.5**: hybrid algorithm based on LP Resolving and parameter $\gamma = 1.5$.

9. **Learn-1.5**: hybrid algorithm based on LP Learning and parameter $\gamma = 1.5$.

2.7.4 Results

On every instance, we express the performance of each algorithm as a percentage of the LP upper bound. That is, we take the expected revenue of the algorithm (approximated over 10 runs), and divide it by the optimal objective value of the LP (2.27) with the true arrival sequence. In Table 2.2, we report the mean and standard deviation of each algorithm's percentages over the 35 arrival sequences, for each loading factor.

In general, MULTI-PRICE BALANCE is the most profitable and robust among the forecast-independent algorithms. The forecast-dependent algorithms have much greater fluctuation in their performance for different occupancy days, dependent on how accurate their forecasts were for that day. LP Learning is slightly better than the others, but is most prone to overfitting in its forecasts.

Nonetheless, by combining these algorithms with MULTI-PRICE BALANCE, the hybrid algorithms are able to correct for forecast overconfidence and achieve the best performance overall (aside from the Clairvoyant algorithm, which has a perfect forecast of the future). We find that although the hybrid algorithm only changes a small fraction ($\approx 5\%$) of the forecasting algorithm's decisions, this drastically improves the profitability and robustness.

2.7.5 Results under Greater Fare Differentiation

The instances tested in Section 2.7.4 were "easy" in that there was not so much difference between selling rooms at their low or high fares. In this subsection, we synthetically modify the higher fare for each room category to be *twice* its lower fare. We also increase the utility

Table 2.2: The percentages of optimum achieved by different algorithms. The 3 highest percentages in each row are **bolded**. The 3 lowest standard deviations in each row are *italicized.*

Loading Factor		Forecast-independent			Forecast-dependent				Hybrid	
		Myopic	Conservative	Balance	One-shot	Resolve	Learn	Clairvoyant	Resolve-1.5	Learn-1.5
1.4	Mean	0.974	0.940	**0.976**	0.973	0.962	0.958	0.991	**0.977**	**0.977**
	Stdev	0.023	0.034	*0.013*	*0.016*	0.039	0.041	0.008	*0.018*	0.020
1.6	Mean	0.965	0.960	**0.971**	0.964	0.961	0.963	0.990	**0.977**	**0.978**
	Stdev	0.025	0.036	*0.014*	0.021	0.031	0.030	0.008	*0.008*	*0.010*
1.8	Mean	0.957	**0.972**	0.968	0.808	0.962	0.968	0.990	**0.977**	**0.977**
	Stdev	0.020	0.036	*0.012*	0.100	0.029	0.023	0.009	*0.008*	*0.007*

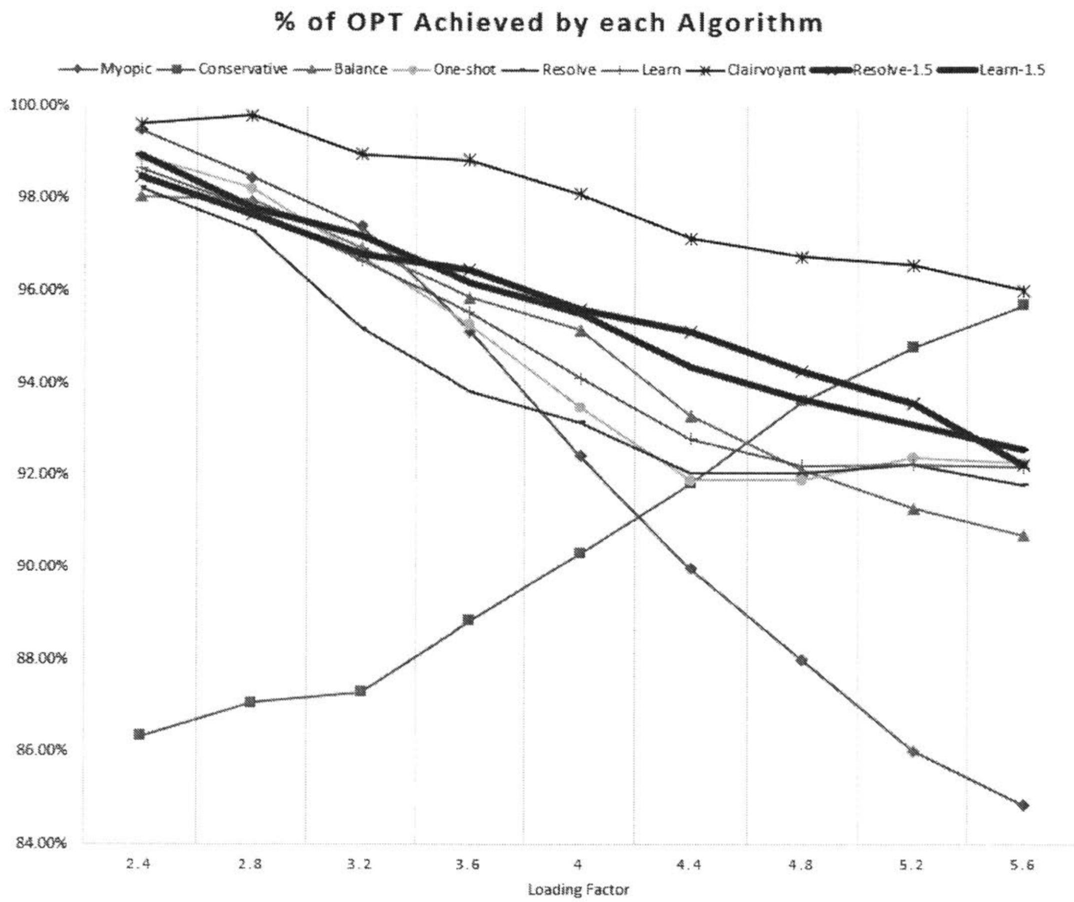


Figure 2-3: Algorithm performances in the setting with greater fare differentiation. The lines corresponding to the two hybrid algorithms, which perform the best overall, have been **bolded**.

of the no-purchase option in the MNL model for each customer type (see Section A.6), to maintain the tension between lower fares which maximize expected revenue, and higher fares which limit inventory consumption.

Furthermore, we test the complete range of loading factors, including both the extreme where the Myopic algorithm is optimal, and the extreme where the Conservative algorithm is optimal. In Figure 2-3, we plot the average percentages of optimum attained by each algorithm over the 35 arrival sequences, for each loading factor.

The conclusion again is that the two hybrid algorithms, which use forecasts but continuously reference our forecast-independent value functions, are the most profitable and robust, with MULTI-PRICE BALANCE coming third. However, it is important to note that our methodology is only relevant in-between the extremes, where there is a non-trivial tradeoff between immediate revenue and future inventory. If a firm knew that its inventory constraints tend to not be binding, then it would be better off using the Myopic algorithm. Similarly, if a firm knew that it has too much demand for its inventory, then it would be better off always offering the maximum prices, using the Conservative algorithm.

2.8 Conclusion

Competitive analysis is a well-established methodology in sequential decision-making problems, providing a baseline decision in the absence of a reliable forecast of the future. Previously, optimal algorithms have been derived for allocating a single unreplenishable resource to customers from different fare classes, or allocating multiple resources which each have a fixed price. In this chapter, we derive optimal allocation algorithms which jointly consider the tradeoffs between different fares and different resources. This broadly expands the applicability of competitive analysis, in areas such as online advertising, matching markets, personalized e-commerce, and appointment scheduling.

Acknowledgments

The authors would like to thank Rong Jin of Alibaba for pointing out a technical error in an earlier version of the appendix. The authors would also like to thank Ozan Candogan and James Orlin for asking questions which led to the simpler bound presented in Corollary 2.2.6.

Chapter 3

Dynamic Recommendation at Checkout under Inventory Constraints

This work is motivated by a new checkout recommendation system at Walmart's online grocery, which offers a customer an assortment of up to 8 items that can be added to an existing order, at potentially discounted prices. We formalize this as an online assortment planning problem under limited inventory, with customer types defined by the items initially selected in the order. Multiple item prices, combined with customer withdrawal when their initially selected items stock out, pose additional challenges for the development of an online policy. We overcome these challenges by introducing the notion of an inventory protection level in expectation, and derive an algorithm with a constant-factor competitive ratio guarantee under adversarial arrivals.

3.1 Introduction

The past decade has seen a tremendous advent in online recommendation systems. Since different customers arriving to the website have different preferences, instead of recommending the same products to everyone, *personalized* recommendation has been widely adopted for both enhancing customer experience and boosting revenue. For example, Amazon.com makes recommendations based on a customer's past purchases, the ratings she has given to products, and the purchasing behavior of similar customers; according to Linden et al. (2003) this has dramatically increased click-through and conversion rates. Another example is Netflix, which tailors recommendations of movies to customer taste. A recent article by Wood (2014) highlights the popularity of personalized recommendation, reviewing several start-ups that are adopting customer-specific operational decisions, including Stitch Fix, Trunk Club, Birchbox, and Club W.

3.1.1 Motivation

Despite the success of personalized recommendation in online retailing, it is often difficult to accomplish: many customers are transient and make a one-time purchase without registering an account. Even when historical purchasing information is available, it can be irrelevant or misleading, because a customer can return to the website planning to purchase products in a different category. One way to overcome these obstacles is to make recommendations based on the products currently in the customer's shopping cart.

Walmart has begun to implement this idea in the form of a new *recommendation at checkout* system for its online grocery. As described in Yuan et al. (2016): "When a customer finishes shopping and clicks checkout, she would see a 'stock up' page with at most 8 additional items recommended, generated by our system."

In contrast to traditional recommendation systems, Walmart's recommendation at checkout system is based on the products the customer is already purchasing. The recommended products are usually complementary to the products already in the cart (e.g. cereal to go with milk), and can even be offered at discounted prices and be seen as a bundle deal.

3.1.2 Problem Formulation and Main Result

Motivated by Walmart's recommendation at checkout system, we consider the following online assortment planning problem. A firm is selling n items each with a starting inventory, and there is no replenishment. Customers arrive sequentially, and each customer t plans on buying her *primary item* i_t for its *full price*. So long as there is inventory of i_t available, the customer arrives at the checkout page. At this point, the recommendation algorithm can offer the customer an assortment of *add-on items*, at potentially *discounted prices*, which can be purchased with i_t. Our model can incorporate constraints on the assortments offered—for example, Walmart's cardinality constraint of 8 add-ons.

The customer proceeds to purchase a subset of the add-ons offered (possibly none) with item i_t, according to known choice probabilities. These choice probabilities are estimated by aggregating the decisions of past customers who checked out with item i_t in their basket. In this vein, i_t can be seen as the *type* of customer t, and our model allows for general, different choice functions for customers with different primary items. That is, we allow for n different choice functions, each of which is associated with a different item.

In this work, we do not consider the learning problem, and focus on the optimization problem when the choice functions are given. Our goal is to develop a recommendation algorithm, which sequentially selects the add-on assortment for each arriving customer, to maximize the expected revenue earned before the selling season is over or the inventories run out. To highlight the main idea of our work, we assume that each customer clicks checkout with a *single* item in her basket. We treat the sequence of primary items ("arrivals") as exogenously given, and our aim is to increase revenue with the ability to offer add-on assortments while controlling inventory.

Online personalized assortment planning is a challenging problem due to the combination of dynamic assortment selection, heterogeneous customer types, and inventory constraints. A few recent papers pioneer the study of this problem under different settings: Golrezaei et al. (2014); Bernstein et al. (2015); Gallego et al. (2016). Compared to existing models, the following aspects of the recommendation at checkout problem introduce additional challenges for the development of an online policy:

1. In the previously-considered choice models, if a specific item is not shown in the assortment, it does not decrease the customer's interest in other items. In our problem, since the customer wanted to check out a specific primary item, she leaves the online retailing system if the product she desires is unavailable; she never clicks checkout and there is no opportunity to offer add-ons.

2. Most existing work on assortment optimization assumes that each item has a fixed price. Our setting allows for an item to be marked down when it is offered as an add-on: we allow a fixed, distinct discount price for each item. Therefore, in our assortment planning problem, the total revenue cannot be determined from only tallying the inventory depleted, because we would not know how much of that inventory the firm

managed to sell at the full price.

All in all, in existing models, an item is excluded from the assortment to reduce cannibalization of other sales, while in our model, an item could be excluded also to *protect* that item for (i) enabling add-on sales to future customers with that item as their primary item, and (ii) selling it at its full price. As a result, the recommendation at checkout problem requires new insights on the trade-off between assortment optimization and inventory protection.

To evaluate online algorithms, we adopt worst-case *competitive ratio* as our performance measure (see the book by Borodin and El-Yaniv (2005) for background on this measure). No assumptions are made about the types of future customers, or the total time horizon remaining; the sequence of customer types can be thought of as chosen *adversarially.* This is useful in practical settings where the demand is highly uncertain. We defer further discussion about the choice of performance measure to the aforementioned papers.

To define the competitive ratio, let $\mathsf{OPT}(i_1, \ldots, i_T)$ denote an LP-based upper bound (details deferred to Section 3.2.2) to the expected revenue of an optimal clairvoyant offline algorithm, which knows in advance both the sequence of primary items $i_1, \ldots, i_T$ (types of customer arrivals) and problem instance $\mathcal{I}$ (including choice functions, revenues, and initial inventories). Let $\mathsf{ALG}(i_1, \ldots, i_T)$ denote the expected revenue of an online algorithm, which knows $\mathcal{I}$ but does not know in advance the sequence $i_1, \ldots, i_T$. Note that neither algorithm knows in advance the outcomes of the randomness in the customers' purchase decisions. We then define the competitive ratio of the algorithm to be

$$\inf_{\mathcal{I}} \inf_{i_1,\ldots,i_T} \frac{\mathsf{ALG}(i_1, \ldots, i_T)}{\mathsf{OPT}(i_1, \ldots, i_T)}. \tag{3.1}$$

Our main result is an algorithm whose competitive ratio is at least $1/4 - \varepsilon$, where ε is an error that can be made arbitrarily small, being caused by the necessity of sampling to make the algorithm polynomial-time.

3.1.3 Directly-related Existing Results

We explain our new algorithmic techniques in Section 3.1.4, but first motivate them by discussing the existing algorithms and bounds that are closely related.

Consider the special case of our problem where the primary item for each customer is a *different* dummy item with price 0 and starting inventory 1. In this case, there is no concern about primary items stocking out, and an assortment of non-dummy items (with fixed add-on prices) can be offered to each of the arriving customers. This is the personalized assortment problem introduced by Golrezaei et al. (2014). They prove that for any starting inventory amounts and *substitutable* choice models (where a customer's interest in one item cannot decrease if a different item is unavailable), the competitive ratio is at least $1/2$. This can be achieved by the greedy algorithm, which maximizes the immediate revenue from each customer without considering future inventory.

In our general problem, there is the concern about *protecting* items, because each unit of inventory can sold in one of two ways—the "good option", where it is sold as a primary item, and the "bad option", where it is sold as an add-on—with the good option being better than the bad option by a potentially unbounded factor. This tradeoff has been analyzed in the single-leg booking problem studied by Ball and Queyranne (2009); Lan et al. (2008). For a single item with two selling options, a competitive ratio of $1/2$ can be achieved by a simple rule: flip a coin of probability $1/2$; if heads, protect every unit of the item from being sold under the bad option; if tails, allow the item to be sold under either option.

It may appear that our competitive ratio of $1/4$ is easily obtained by combining these two existing results. Indeed, one could decide whether or not to protect each item with probability $1/2$, losing a factor of $1/2$, and otherwise maximize the immediate revenue from each customer, losing another $(1/2)$-factor due to being greedy. However, our problem has the additional feature that the majority of the revenue could require a specific *combination* of items. For example, if the majority of the revenue occurs from selling item 2 as an add-on to item 1, then the algorithm could only obtain this revenue if it both: (i) decided to protect item 1 from selling out as an add-on, and (ii) decided *not to protect* item 2 from being sold as an add-on to item 1.

If we use the strategy of protecting each item with probability $1/2$ independently, then in the preceding example where the revenue comes from selling item 2 as an add-on to item 1, we have already lost a factor of $1/2 \times 1/2 = 1/4$ before getting to any additional factor lost due to being greedy. Therefore, this does not achieve a competitive ratio of $1/4$. In fact, it appears that any algorithm which fixes the items and levels of inventory to protect beforehand, even if it determines these using some coordinated randomness, has competitive ratio strictly less than $1/4$—we defer the detailed calculations to Section 3.2.3.

3.1.4 New Algorithmic Techniques

This brings us to our main algorithmic innovation, which is a *protection level in expectation.* Our algorithm ensures that the *expected* number of units of each item sold under the "bad option" does not exceed half of its starting inventory, where the expectation is over the purchase decisions of the past customers (and this can be computed because their types have already been revealed to the online algorithm). By contrast, the existing algorithms fix the inventory to protect beforehand, and ensure that on any *realization* of customer purchase decisions the protected inventory is not sold under the "bad option".

Our algorithm is defined by a simple greedy rule:

> "Offer the revenue-maximizing add-on assortment to each customer, while ensuring that the expected units of each item sold as an add-on never exceeds half of its starting inventory."

Of course, while this constraint is related to expected remaining inventory, the algorithm is also physically constrained from offering any items whose realized remaining inventory is 0.

Implementing such a rule requires many considerations. For example, suppose both items i and j have remaining inventory in reality, and the algorithm is deliberating whether to maximize immediate revenue by offering item j as an add-on to primary item i. Let d_j^{curr} denote the expected number of units of item j that has already been sold as an add-on. To evaluate whether offering item j would push d_j^{curr} above its allowable threshold, we must know:

- If I offer item j given the current inventory state, how much do I add to d_j^{curr}? In other words, what is the measure of the present inventory state that has been realized?
- What are the other possible inventory states for the present time step, and what decisions am I making on those?

We answer these questions by envisioning time as passing fluidly over a unit interval [0,1] for each time step. We consider the entire distribution for the inventory state $(I_1, \ldots, I_n)$ at the start of the time step, and evaluate how the values of $(d_1^{\mathrm{curr}}, \ldots, d_n^{\mathrm{curr}})$ increase if we offer from each state the revenue-maximizing assortment from the items j such that both $I_j > 0$ and d_j^{curr} has not reached its threshold. If this causes any add-on thresholds to be violated by the end of the time step, then we consider the first *breakpoint* s^* in

[0,1] where d_j^{curr} has reached its threshold for some item j^*. Over the time interval $[0, s^*]$, we offer the revenue-maximizing assortments for each inventory state which have already been computed. In the time interval after s^*, we re-optimize the assortments for each state, where j^* is no longer eligible to be an add-on (and this could completely change the optimal assortments). The algorithm repeats this process, potentially adding multiple breakpoints in a time step, until the end of the continuous interval [0,1] is reached with no threshold violations. Finally, the algorithm returns to reality, and makes the corresponding decision given the inventory state at hand, which is in the form of a random assortment if there were breakpoints during the time step.

There is a final computational challenge, caused by the fact that there are exponentially-many inventory states. At the start of each time step, we generate a new empirical distribution for the inventory state, using the known customer types $i_1, \ldots, i_{t-1}$ and simulating the random assortments offered by the algorithm and the random purchase decisions of those customers. This simulation technique is similar in spirit to the one from Chapter 7 In our final primal-dual analysis of the competitive ratio, we show how to define dual variables such that the ε-error from simulation does not propagate, to achieve a final competitive ratio of $1/4 - \varepsilon$.

3.1.5 Comparison with Bounds from Literature

Our recommendation at checkout problem is closely related to the aforementioned personalized assortment (Golrezaei et al., 2014) and single-leg booking (Ball and Queyranne, 2009; Lan et al., 2008) problems from the literature, as well as the online matching with stochastic rewards problem studied in Mehta and Panigrahi (2012); Mehta et al. (2014), which is the special case of the personalized assortment problem when the items have the same price and the assortment offered is constrained to consist of at most one item. Our problem generalizes all of these problems, and in the special case of single-leg booking, our algorithm coincides with that of Ball and Queyranne (2009). The competitive ratio achieved by our analysis, however, is smaller. We do provide an upper bound showing that the existing competitive ratios cannot be achieved for our problem in general. Table 3.1 presents a detailed comparison of our model with existing models, and our bounds with existing bounds.

While our lower bound of $1/4$ on the competitive ratio does not match our upper bound

Table 3.1: Comparison of problems and competitive ratios. For each row, **bold** font indicates greatest generality.

Problem	Real-time Personalized Assortment	Online Matching w/ Stochastic Rewards	Two-Fare Single-Leg Booking Problem	Recommendation at Checkout
Reference	Golrezaei et al. '14	Mehta/Panigrahi '12	Ball/Queyranne '09	[this chapter]
# of Items	**multiple**	**multiple**	single	**multiple**
Revenue of Items	one price per item	same price for all items	**two** prices per item	**two** prices per item
Choice Function	arbitrary; assume *substitutability*$^\sharp$	purchase probabilities for single items	deterministic	**arbitrary; assume *weak substitutability*$^\sharp$**
Lower Bd	$\frac{1}{2}^\natural$	$\frac{1}{2}^\flat$	$\frac{1}{2}$	$\frac{1}{4}-\varepsilon$
Upper Bd	0.621		$\frac{1}{2}$	0.43

$\sharp$ Golrezaei et al. (2014) impose a necessary mild *substitutability* assumption on the choice model (i.e., the purchase probability of an item can only decrease as additional items are added into the assortment). We call our assumption *weak substitutability* because, while we assume substitutability between the add-ons, we allow the purchase probability of an add-on to decrease (to zero) if the primary product requested is unavailable.
$\natural$ The competitive ratio improves from $\frac{1}{2}$ to $1-\frac{1}{e}\approx 0.632$ as the minimum starting inventory increases from 1 to ∞, and in this case, matches the upper bound of $1-\frac{1}{e}$ (the upper bound of 0.621 is from Mehta and Panigrahi (2012) and uses unit starting inventories).
$\flat$ The competitive ratio improves to 0.534 if the purchase probabilities approach 0, and 0.567 if they are also equal (the first statement is due to Mehta et al. (2014)).

of .43, we are not aware of any tight competitive ratio results under both: (i) uncertainty in purchasing, and (ii) small starting inventories. To highlight the difficulty of the problem, even in the special case of online matching with stochastic rewards, with the further assumption that every purchase probability is equal to a single infinitesimal value ρ, there is a gap between the lower bound 0.567 and the upper bound of 0.621 (see Mehta and Panigrahi (2012) and the later work by Mehta et al. (2014)). When (i) is relaxed and we have the deterministic online *matching* problem, a "random ranking" online algorithm can achieve a tight competitive ratio of $1-1/e$ (see Karp et al. (1990), its generalization by Aggarwal et al. (2011), and the unified analysis by Devanur et al. (2013)). Similarly, when (ii) is relaxed and the starting inventories approach ∞, an "inventory balancing" online algorithm can achieve a tight competitive ratio, also of $1-1/e$ (see Golrezaei et al. (2014); this result is closely related to the *Adwords* stream of research initiated in Kalyanasundaram and Pruhs (2000); Mehta et al. (2007); Buchbinder et al. (2007)). In Section B.1, we discuss why large starting inventories do not appear to improve our competitive ratio, including an example under this regime which shows that the competitive ratio must be strictly less than 1/2.

We see our guarantee of 1/4 as the natural baseline competitive ratio which combines the loss of 1/2 from the online matching with stochastic rewards problem, with the loss of 1/2 from items having the "good option" of being sold as a primary item and "bad option" of being sold as an add-on. It is particularly interesting to us that the competitive ratio of

1/4 is not immediately achieved by flipping a coin for each item to decide whether to sell it under its bad option; instead, it is achieved by constraining the *expected* units of each item sold under its bad option.

Finally, we would like to mention that unlike in the classical online matching problem, where it is immediate that greedy is (1/2)-competitive (Karp et al., 1990; Aggarwal et al., 2011), showing that greedy is (1/2)-competitive in the online matching with stochastic rewards problem requires an intricate probabilistic dual-balancing argument (Mehta and Panigrahi, 2012; Golrezaei et al., 2014). Extending this to a (1/4)-competitive algorithm for our recommendation at checkout problem requires new insights on how to define dual variables when the algorithm is based on expected add-on sales, instead of realized add-on sales.

3.1.6 Other Related Work

The papers which are most related to our work from a technical standpoint have already been discussed in Sections 3.1.3–3.1.5. In this section we emphasize several papers which have been previously unmentioned.

The retailing problem of planning a sequence of assortments to offer over a finite horizon subject to various operational constraints (e.g. assortment cardinality, inventory considerations) has been studied extensively since the seminal papers van Ryzin and Mahajan (1999); Mahajan and van Ryzin (2001). The model we study here, where the tradeoff is between immediate revenue vs. future inventory, originally focused on the multinomial-logit choice model in Talluri and Van Ryzin (2004); Rusmevichientong et al. (2010).

In this chapter, instead of focusing on a specific choice model, we view the assortment decision as a generic lever for managing the tradeoff between different depletion options for the inventories which have different revenue rates (see Maglaras and Meissner (2006); Chan and Farias (2009)). Conceptually, our analysis holds as long as the choice model satisfies a *weak substitutability* condition, which essentially says that the add-ons cannot be complements to each other (even though they can be complements to the primary item). Computationally, our algorithm is polynomial-time as long as the single-period assortment optimization problem corresponding to any of the choice models given can be solved efficiently. We refer to Cheung and Simchi-Levi (2016) for a recent summary on the state of the art in single-period assortment optimization.

This chapter provides a worst-case approximation guarantee which makes no assumptions on the arrival sequence. This benchmark is referred to as the competitive ratio under adversarial arrivals; see the book by Borodin and El-Yaniv (2005), or Buchbinder et al. (2016) which discusses some recent results on the competitive ratio. In the context of the recent assortment planning literature, Golrezaei et al. (2014) focus on the worst-case competitive ratio, while Bernstein et al. (2015); Gallego et al. (2016) consider improved approximation guarantees under additional assumptions about the arrival sequence.

3.2 Model Specification

Let $\mathbb{R}$ denote the set of real numbers, and $\mathbb{N}$ denote the set of positive integers. For a general $m \in \mathbb{N}$, let $[m]$ denote the set $\{1, \ldots, m\}$.

A firm is selling $n \in \mathbb{N}$ different items. Each item $j \in [n]$ starts with an initial inventory of $b_j \in \mathbb{N}$ units, and there is no replenishment. Each item has two prices: the full price $r_j \in \mathbb{R}$, and a discounted price of $r_j^{\mathsf{disc}} \in \mathbb{R}$, satisfying $r_j \geq r_j^{\mathsf{disc}} \geq 0$ (we allow the single-price case where $r_j^{\mathsf{disc}} = r_j$). There are n customer types, one for each item $i \in [n]$, characterized by the population of customers who arrive planning to purchase item i, at the full price of r_i. Any items that type-i customers purchase other than i are sold at their discounted prices. For a customer of type i, we assume that the probabilities of her purchasing different subsets of items are given in the form of a choice function, as defined below.

3.2.1 Choice Functions for Add-ons

Definition 3.2.1 (Customer Choice Functions). Consider any customer type $i \in [n]$. We let S denote *all* of the items (including i) shown to the customer.

- For each $S' \subseteq S$, let $\phi_i(S', S)$ denote the probability that a customer of type i purchases subset S' when shown S.

- For every $j \in [n]$, define
$$p_{ij}(S) = \sum_{S' \subseteq S : j \in S'} \phi_i(S', S), \tag{3.2}$$
which is the total probability that item j is sold when a customer of type i is shown subset S.

The execution of the algorithm will depend on the probabilities of the customer choosing each subset of S given by ϕ_i; for an example, see Section B.2. However, the analysis will only depend on the aggregate probabilities of purchasing each item, defined as $p_{ij}(S)$ in (3.2).

Definition 3.2.2 (Model assumptions). For all $i \in [n]$, we make the following assumptions on the values of p_{ij} implied by function ϕ_i.

1. $i \in S$ implies $p_{ii}(S) = 1$.
2. $i \notin S$ implies $p_{ij} = 0$ for all $j \in [n]$.
3. (Weak Substitutability) $\{i, j\} \subseteq S_1 \subseteq S_2$ implies $p_{ij}(S_1) \geq p_{ij}(S_2)$.

Assumption 1 says that a customer of type i *always* purchases item i when it is available, since by definition they were already planning to check out item i. Assumption 2 says that a customer of type i purchases *nothing* when item i is unavailable, since they never get to the checkout page to see the add-ons. Assumption 3, Weak Substitutability, says that conditional on i being in the set shown, the probability of successfully selling another item j can only decrease as more items are added to the set. We use the word *weak* to emphasize that substitutability need not hold if $i \in S_2 \setminus S_1$, in which case $p_{ij}(S_1)$ would be zero while $p_{ij}(S_2)$ may be non-zero.

Definition 3.2.3 (Terminology). Hereafter we use the following terminology in describing the interactions between the customers and the firm:

- We say that a customer of type i *requests* item i, and that i is her *primary item*;
- We call the set S shown to a customer the *assortment* which is *offered* to her;
- If the customer's primary item is in the set S offered to her, then we say that she is *served*;
- We call the items in S other than i *add-ons.*

Definition 3.2.4 (Feasible assortments). We let $\mathcal{S}_i$ denote the family of feasible non-empty assortments that can be offered to a customer of type i; the empty assortment $\emptyset$ is always feasible and will be considered separately. This allows us to incorporate constraints on the assortment offered from the firm's side, such as Walmart's cardinality constraint of 8 items.

Definition 3.2.5 (Assumptions on $\mathcal{S}_i$)**.** We make the following assumptions on $\mathcal{S}_i$, which follow from the earlier assumptions in Definition 3.2.2.

- All sets in $\mathcal{S}_i$ contain i. This is without loss of generality, since Assumption 2 states that offering any S without i is equivalent to offering $\emptyset$.
- If any S containing i is not in $\mathcal{S}_i$, then that means $p_{ij}(S) = 0$ for all $j \neq i$. By Assumption 3, for any superset S_2 of S, $p_{ij}(S_2) = 0$ for all $j \neq i$, so we can eliminate S_2 from $\mathcal{S}_i$ as well. Thus we can without loss of generality assume that $\mathcal{S}_i$ is downward-closed, i.e. $i \in S_1 \subseteq S_2 \in \mathcal{S}_i$ implies $S_1 \in \mathcal{S}_i$.
- $\mathcal{S}_i$ includes the singleton assortment $\{i\}$, by Assumption 1.

Example 3.2.6. We provide an example to clarify our notation. Throughout the examples in this chapter, we use two conventions, for brevity:

1. We only list the maximal subsets in $\mathcal{S}_i$ (this is enough since $\mathcal{S}_i$ is downward-closed);
2. We don't list choice probabilities that can be inferred from other choice probabilities.

For this example, the situation is the following. There are 4 items. Customers of type 1 will always buy item 1 (leaving with nothing if it is not offered), and (deterministically) buy at most one add-on using preference order $2 \succ 3 \succ 4$, with the exception that she is willing to buy the two add-ons 2 and 4 together. Following the conventions defined in bullets 1–2 above, this can be formalized as follows:

- $\mathcal{S}_1 = \{\{1,2,3,4\}\}$ (i.e. all subsets of $\{1,2,3,4\}$ containing 1 are feasible, by the conventions);
- $\phi_1(\{1,2,4\},\{1,2,3,4\}) = 1$ (i.e. $\phi_1(S',\{1,2,3,4\}) = 0$ for all $S' \neq \{1,2,4\}$, by the conventions);
- $\phi_1(\{1,2\},\{1,2,3\}) = 1$; $\phi_1(\{1,3\},\{1,3,4\}) = 1$; $\phi_1(\{1,4\},\{1,4\}) = 1$.

We do not specify $\phi_1(\cdot,\{1,2,4\})$, $\phi_1(\cdot,\{1,2\})$, and $\phi_1(\cdot,\{1,3\})$ because they can be inferred using weak substitutability from $\phi_1(\cdot,\{1,2,3,4\})$, $\phi_1(\cdot,\{1,2,3\})$, and $\phi_1(\cdot,\{1,3,4\})$, respectively. As a result, it could be convenient to shrink $\mathcal{S}_1$ down to $\{\{1,2,4\},\{1,3\}\}$, since all offerings are equivalent to $\{1,2,4\}$, $\{1,2\}$, $\{1,4\}$, or $\{1,3\}$.

3.2.2 Problem Definition and Main Result

Customers arrive sequentially over $T \in \mathbb{N}$ time periods. After customer $t \in [T]$ arrives, the firm observes her type, i_t, and then chooses an assortment $S^t \in \mathcal{S}_{i_t} \cup \{\emptyset\}$ to offer her. The firm knows the choice probabilities, but does not know S'_t, the realization of which items she will actually buy. Furthermore, the firm does not know the types of future customers $i_{t+1}, \ldots, i_T$, and does not know T. The firm must offer an assortment to each arriving customer in an online fashion with the objective of maximizing expected revenue before inventory runs out.

The types of the arriving customers are chosen adversarially and the performance of an online algorithm is measured by its *competitive ratio.* To define this term, we first present the following LP, whose optimal objective value is an upper bound on the expected revenue of any online algorithm:

$$
\begin{array}{llrcll}
\max & \displaystyle\sum_{t=1}^{T}\sum_{S\in\mathcal{S}_{i_t}} \Big(r_{i_t} + \sum_{j\neq i_t} r_j^{\mathsf{disc}} p_{i_t j}(S)\Big) y^t(S) & & & & \\
\text{s.t.} & & \displaystyle\sum_{t=1}^{T}\sum_{S\in\mathcal{S}_{i_t}} p_{i_t j}(S) y^t(S) & \leq & b_j & j \in [n] \\
& & \displaystyle\sum_{S\in\mathcal{S}_{i_t}} y^t(S) & \leq & 1 & t \in [T] \\
& & y^t(S) & \geq & 0 & t \in [T], S \in \mathcal{S}_{i_t}.
\end{array}
\tag{3.3}
$$

Let $\mathsf{OPT}(i_1, \ldots, i_T)$ denote the optimal objective value of this LP; note that it is a function of the types of arriving customers. For all t, $y^t(S)$ corresponds to the probability that assortment S is offered during time t (over all sample paths with these fixed customer types). The objective function is the expected revenue obtained as a result of these probabilistic offerings. The first set of constraints impose that the expected units of item j sold does not exceed its starting inventory of b_j, while the next set of constraints impose that the total probability customer t is offered a non-empty subset does not exceed 1. Note that we do not need a variable for $y^t(\emptyset)$, since no revenue is earned and no inventory is consumed.

The following is a standard in both revenue management (see Golrezaei et al. (2014)) and discrete mathematics (see Dean et al. (2008)):

Lemma 3.2.7. $\mathsf{OPT}(i_1, \ldots, i_T)$ *is an upper bound on the expected revenue obtainable by any* offline *algorithm, which knows* $i_1, \ldots, i_T$ *at the start of the time horizon.*

Nonetheless, we provide a self-contained proof for our specific problem in Section B.3. By Lemma 3.2.7, $\mathsf{OPT}(i_1, \ldots, i_T)$ is also an upper bound on the expected revenue obtainable by any *online* algorithm, when the arrivals are $i_1, \ldots, i_T$. Having established this benchmark, we define competitive ratio as follows:

Definition 3.2.8. Fix an online algorithm. For a given instance $\mathcal{I}$ (consisting of choice functions, revenues, and starting inventories), let $\mathsf{ALG}(i_1, \ldots, i_T)$ denote the expected revenue earned by the algorithm when the sequence of arrival types is $i_1, \ldots, i_T$. Then the *competitive ratio* of the algorithm is

$$\inf_{\mathcal{I}} \inf_{i_1,\ldots,i_T \in [n]} \frac{\mathsf{ALG}(i_1, \ldots, i_T)}{\mathsf{OPT}(i_1, \ldots, i_T)}.$$

In other words, an adversary, who knows the algorithm beforehand (but knows neither the outcomes of any randomness in the algorithm nor the customers' purchase decisions), chooses an instance and arrival sequence to minimize the online algorithm's expected revenue as a fraction of the offline optimum. Note that for the offline optimum, we are using a value which could be greater than the expected revenue of the best algorithm which knows the arrival types in advance. However, this can only decrease the declared ratio, and is typical in the definition of competitive ratio for online problems where there is randomness in the offline variant (see, e.g., Mehta and Panigrahi (2012); Golrezaei et al. (2014)).

We prove the following bound on the competitive ratio:

Theorem 3.2.9. *For any $\varepsilon > 0$, our online algorithm (whose runtime is polynomial in $\frac{1}{\varepsilon}$ and the instance parameters) has competitive ratio at least $\frac{1}{4} - \varepsilon$.*

Our algorithm needs to estimate probabilities by sampling virtual outcomes for the decisions of past customers, which explains the error of ε. Before we describe our algorithm, we present an example which provides intuition by illustrating the main "difficulty" in the problem.

3.2.3 Motivation for our Algorithm

We present an example which demonstrates why the fixed-protection-level algorithm, which would follow naturally from the existing literature discussed in Section 3.1.3, does not suffice for our problem.

Consider the following instance, where N and M are large integers:

- $n = 3$; $r_1 = r_1^{\mathsf{disc}} = 0$, $b_1 = N$; $r_2 = r_2^{\mathsf{disc}} = \frac{1}{N}$, $b_2 = N$; $r_3 = M^2$, $r_3^{\mathsf{disc}} = M$, $b_3 = 1$;
- $\mathcal{S}_1 = \{\{1,2\}\}$; $\phi_1(\{1,2\},\{1,2\}) = 1$; $\mathcal{S}_2 = \{\{2,3\}\}$; $\phi_2(\{2,3\},\{2,3\}) = \frac{1}{N}$.

In words, there are 3 items, where items 1 and 2 start with N units of inventory each, and item 3 starts with 1 unit of inventory. Item 1 is a dummy item of zero revenue. Customers of type 1 can be recommended item 2 as an add-on, earning a small but non-zero revenue of $1/N$ with probability 1. However, item 2 is better used as a primary item for selling item 3, because customers of type 2 can be recommended item 3 as an add-on to earn a large revenue of M with probability $1/N$.

The arrival sequence begins with N customers of type 1, and the firm must decide how many units of item 2 to sell as an add-on at the modest price of $1/N$. If the firm is myopic and sells all of it, then this is suboptimal in the scenario where the arrival sequence continues with N customers of type 2, because the firm would be forgoing N opportunities to earn M/N expected revenue. On the other hand, if the firm sells none of item 2 as an add-on, then it earns zero revenue in the scenario where no customers of type 2 arrive.

Following the discussion of the literature in Section 3.1.3, the firm can balance between the two extreme scenarios by randomly deciding whether it is going to "protect" items from being sold as an add-on, with probability $1/2$ for each item. Suppose it made this random decision *independently* for each item, and consider the scenario where type-2 customers do arrive. In order to realize the opportunities of selling item 3 as an add-on, the firm must both: (i) have inventory of item 2 remaining to serve the type-2 customers; and (ii) not be protecting[1] item 3. By independence, this only occurs with probability $1/4$, so the firm's expected revenue would be

$$M \cdot \frac{1}{4}(1 - (1 - \frac{1}{N})^N) + o(M) \approx M \cdot \frac{1 - 1/e}{4}. \tag{3.4}$$

This is less than $1/4$ of the optimum defined in Section 3.2.2 using the LP, which equals $M \cdot (1/N)N = M$.

To our knowledge, there is no immediate cure to the problem demonstrated by equation (3.4). Even if the algorithm made the improved decision of deterministically protecting half

[1]The firm would also need to flip a coin to decide whether to sell as an add-on for item 3, since its full price is M^2, which is yet another order of magnitude greater in revenue.

of the units of item 2 (instead of randomly protecting all of the units with probability 1/2), its expected revenue would only increase to

$$M \cdot \frac{1}{2}(1-(1-\frac{1}{N})^{N/2}) + o(M) \approx M \cdot \frac{1-1/\sqrt{e}}{2}$$

which is still less than 1/4 of the optimum. In fact, for any fixed fraction α of starting inventory to protect, the expected revenue would be

$$M(1-\alpha)(1-e^{-\alpha}) \tag{3.5}$$

which strictly less than 1/4 for any $\alpha \in [0,1]$. There also does not appear[2] to exist a simple way to improve upon (3.5) and achieve the 1/4 ratio by *correlating* the randomized rounding across items.

This example brings us to the main idea behind our algorithm: for each item, instead of protecting based on the *realized* number of units sold, we can protect based on the *expected* number of units sold, to improve the guarantee on expected revenue. We defer the difficulty of specifying such an algorithm to Section 3.3, and first formally modify this example to establish an upper bound on the competitive ratio.

3.2.4 Upper Bound on Competitive Ratio

We use the example from Section 3.2.3 to generate an upper bound on the competitive ratio, by choosing the probability distribution between the two scenarios (type-2 customers arrive; type-2 customers do not arrive) which generates the best bound. Note that we have changed the full price of item 3 from M^2 to M, so that our upper bound still holds under the restriction that $r_j^{\mathsf{disc}} = r_j$ for all j.

Example 3.2.10 (Upper Bound Construction). Let C denote the constant $\frac{1}{w}$, where w is the unique real number satisfying $we^w = 1$. Consider the following randomized instance:

- $n = 3$; $r_1 = r_1^{\mathsf{disc}} = 0$, $b_1 = N$; $r_2 = r_2^{\mathsf{disc}} = \frac{1}{N}$, $b_2 = N$; $r_3 = r_3^{\mathsf{disc}} = M$, $b_3 = 1$;
- $\mathcal{S}_1 = \{\{1,2\}\}$; $\phi_1(\{1,2\},\{1,2\}) = 1$; $\mathcal{S}_2 = \{\{2,3\}\}$; $\phi_2(\{2,3\},\{2,3\}) = \frac{1}{N}$.

[2] This example shows why positive correlation (e.g., comonotonicity) does not cure the problem in general. While negative correlation does cure the problem on this example, it is difficult to define negative correlation when there are a large number of items and the majority of revenue could come from selling multiple add-ons.

- Arrivals are $T = N, i_1 = \ldots = i_N = 1$, w.p. $1 - \frac{C}{M}$;
- Otherwise, arrivals are $T = 2N, i_1 = \ldots = i_N = 1, i_{N+1} = \ldots = i_{2N} = 2$ (w.p. $\frac{C}{M}$).

Our construction is similar in spirit to that of Mehta and Panigrahi (2012), who also exploit the fact that the optimum can use a fractional LP solution. This is used in the following theorem.

Theorem 3.2.11. *Consider the randomized instance defined in Example 3.2.10, and let $M, N \to \infty$. The expected value of $\mathsf{OPT}(i_1, \ldots, i_T)$ is at least $C+1$. On the other hand, for any online algorithm, the expected value of $\mathsf{ALG}(i_1, \ldots, i_T)$ is at most $C - \ln C$. By Yao's minimax principle, the competitive ratio of any online algorithm cannot exceed*

$$\frac{C - \ln C}{C + 1} \approx 0.43.$$

Note that Yao's minimax principle (cf. Yao (1977)) states that the competitive ratio of any (deterministic or randomized) online algorithm that has a lack of knowledge about the arrival sequence is bounded from above by the competitive ratio of any deterministic algorithm that knows the probability distribution of the arrival sequence. The value of $C = \frac{1}{w} = e^w \approx 1.76$ has been chosen to minimize the competitive ratio. We defer the calculations required to prove Theorem 3.2.11 to Section B.3.

3.3 Algorithm

In this section we describe our algorithm, which offers the revenue-maximizing assortment during each time period, but ensures for every item that the expected number of units sold as an add-on never exceeds its *add-on threshold*, which is half of its starting inventory. In the following subsections we specify how our algorithm accomplishes this.

3.3.1 Subroutine for Offering Assortments

Definition 3.3.1 (Protection List and Forbidden Set)**.** For each time period $t = 1, \ldots, T$, let $\mathcal{L}^t$ be a list of tuples of the form (item, probability). $\mathcal{L}^t$ is called a *protection list.* Formally,

$$\mathcal{L}^t = \big((j_k^t, \rho_k^t) : k \in [K_t]\big),$$

where $K_t \geq 0$ is its length, $j_k^t \in [n]$, and $\rho_k^t \in [0,1]$.

For all $t \in [T]$, $\mathcal{L}^1, \ldots, \mathcal{L}^t$, and $k = 0, \ldots, K_t$, let $\mathcal{F}_k^t$ be a set of items, called a *forbidden set*. $\mathcal{F}_k^t$ is a function of $\mathcal{L}^1, \ldots, \mathcal{L}^t$ and is defined as follows:

$$\mathcal{F}_k^t(\mathcal{L}^1, \ldots, \mathcal{L}^t) := \Big(\bigcup_{s<t}\{j_{k'}^s : k' \in [K_s]\}\Big) \cup \{j_{k'}^t : k' \leq k\}.$$

We will usually write $\mathcal{F}_k^t$ for short, when the context is clear.

Example 3.3.2 (Running Example). We will maintain this example throughout, to help explain our algorithm. Similarly to Section 3.2.1, we only list maximal subsets for each $\mathcal{S}_i$, and don't list choice probabilities that can be inferred.

- $b_1 = \infty$, $b_2 = b_3 = b_4 = 1$;
- $r_2 \gg r_3 \gg r_4 \gg r_1 = 0$; $r_j = r_j^{\mathsf{disc}}$ for all j;
- $\mathcal{S}_1 = \big\{\{1,2\},\{1,3\}\big\}$; $\phi_1(\{1,2\},\{1,2\}) = \frac{3}{4}$; $\phi_1(\{1,3\},\{1,3\}) = 1$;
- $\mathcal{S}_3 = \big\{\{3,4\}\big\}$; $\phi_3(\{3,4\},\{3,4\}) = \frac{3}{5}$;
- $(i_1, i_2) = (1,3)$.

The first customer arrives requesting item 1, and is willing to buy either item 2 as an add-on with probability $\frac{3}{4}$, or item 3 as an add-on with probability 1. We would like to choose item 2, since it has much greater revenue. However, we do not want the only unit of it to be sold as an add-on with probability exceeding $\frac{1}{2}$. Therefore, we flip a coin and only offer it with probability $\frac{2}{3}$, which would result in a sale probability of $\frac{2}{3} \cdot \frac{3}{4} = \frac{1}{2}$. With probability $\frac{1}{3}$, we offer item 3 instead.

Following this reasoning, the protection list $\mathcal{L}^1$ is set $\big((2, \frac{2}{3})\big)$, which tells the subroutine to only offer item 2 with probability $\frac{2}{3}$ during time period 1. By the second time period, the forbidden set $\mathcal{F}_k^t$ will always contain 2, telling the subroutine to not offer item 2 as an add-on. We formalize the subroutine in Pseudocode 3. It takes as input a time period t, customer types $i_1, \ldots, i_t$ (which have been observed by time t), and protection lists $\mathcal{L}^1, \ldots, \mathcal{L}^t$ (which we assume are given, for now).

Algorithm 3 $\mathsf{SellTo}(t, i_1, \ldots, i_t, \mathcal{L}^1, \ldots, \mathcal{L}^t)$

1: $I_j \leftarrow b_j$ for all $j \in [n]$ (initialize inventories)
2: **for** $s = 1, \ldots, t$ **do**
3:     **if** $I_{i_s} > 0$ **then**
4:         $J \leftarrow \{j \neq i_s : I_j = 0\}$
5:         $k \leftarrow 0$
6:         **while** $k < K_s$ **do**
7:             flip a coin which is Heads with probability ρ^s_{k+1}
8:             **if** Heads **then break** out of **while** loop
9:             $k \leftarrow k + 1$
10:         **end while**
11:         $F = \mathcal{F}^s_k \setminus \{i_s\}$
12:         $\tilde{S} \leftarrow \operatorname{argmax}_{S \in \mathcal{S}_{i_s} \setminus (J \cup F)} \sum_{j \neq i_s} r^{\mathsf{disc}}_j p_{i_s j}(S)$
13:     **else**
14:         $\tilde{S} \leftarrow \emptyset$
15:     **end if**
16:     offer assortment $\tilde{S}$ to customer s
17:     observe (or virtually generate) her decision and update inventories $I_1, \ldots, I_n$ accordingly
18: **end for**

We make the assumption that the single-period assortment problem in Step 12 can be solved efficiently; we refer the reader to the references mentioned in Section 3.1.6 for details on the single-period problem. We also assume that a fixed tie-breaking rule for the revenue maximization problem has been established at the start; this rule can be arbitrary.

Example 3.3.3 (Running Example). To help with understanding Pseudocode 3, we demonstrate its execution on an expanded version of Example 3.3.2.

- $b_1 = \infty$, $b_2 = \ldots = b_6 = 1$;
- $r_2 \gg r_3 \gg r_4 \gg r_5 \gg r_6 \gg r_1 = 0$; $r_j = r^{\mathsf{disc}}_j$ for all j;
- $\mathcal{S}_1 = \big\{\{1,2\}, \{1,3\}, \{1,4,6\}, \{1,5\}\big\}$:
    - $\phi_1(\{1,2\}, \{1,2\}) = \frac{3}{4}$; $\phi_1(\{1,3\}, \{1,3\}) = 1$;

- $\phi_1(\{1,4,6\},\{1,4,6\}) = 1$; $\phi_1(\{1,5\},\{1,5\}) = 1$ (this is the same as Example 3.2.6 from Section 3.2.1);

- $\mathcal{S}_3 = \{\{3,4\}\}$; $\phi_3(\{3,4\},\{3,4\}) = \frac{3}{5}$;

- $(i_1, i_2, i_3) = (1, 3, 1)$.

Assume that the protection lists have been given as $\mathcal{L}^1 = ((2, \frac{2}{3})), \mathcal{L}^2 = (), \mathcal{L}^3 = ((4, \frac{1}{6}), (5, \frac{13}{25}))$.

Consider time $s = 1$. All inventories are full, so $I_{i_1} = I_1 > 0$ and $J = \emptyset$. With probability $\frac{2}{3}$, $k = 0$ and $F = \emptyset$, so $\tilde{S} = \{1,2\}$. With probability $\frac{1}{3}$, $k = 1$ and $F = \{2\}$, which prevents item 2 from being offered as an add-on, so $\tilde{S} = \{1,3\}$. After time period 1, there are three possibilities for the state of the inventory:

$$(I_1, \ldots, I_6) = \begin{cases} (\infty, 0, 1, 1, 1, 1) \text{ w.p. } \frac{2}{3} \cdot \frac{3}{4} = \frac{1}{2}; \\ (\infty, 1, 1, 1, 1, 1) \text{ w.p. } \frac{2}{3} \cdot \frac{1}{4} = \frac{1}{6}; \\ (\infty, 1, 0, 1, 1, 1) \text{ w.p. } \frac{1}{3} \cdot 1 = \frac{1}{3}. \end{cases}$$

Now consider $s = 2$. If $I_{i_2} = I_3 = 0$, then we have no choice but to offer $\tilde{S} = \emptyset$. Otherwise, $k = 0$, $F = \{2\}$, and J is either $\{2\}$ or $\emptyset$, so $\tilde{S} = \{3,4\}$. Although the customer buys item 4 as an add-on with probability $\frac{3}{5}$ (greater than $\frac{1}{2}$), we only get to this state with probability $\frac{1}{2} + \frac{1}{6} = \frac{2}{3}$, so the add-on threshold for item 4 is not violated. After time 2, these are the possibilities for the state:

$$(I_1, \ldots, I_6) = \begin{cases} (\infty, ?, 0, 0, 1, 1) \text{ w.p. } \frac{2}{3} \cdot \frac{3}{5} = \frac{2}{5}; \\ (\infty, ?, 0, 1, 1, 1) \text{ w.p. } \frac{1}{3} + \frac{2}{3} \cdot \frac{2}{5} = \frac{3}{5}. \end{cases}$$

Note that we have merged some states since it will be irrelevant for the remainder of the execution whether $I_2 = 0$ or $I_2 = 1$ (2 will always be in F and forbidden from being sold as an add-on).

Now consider $s = 3$. J always contains 3, and contains 4 with probability $\frac{2}{5}$. Independently, $k = 0$ w.p. $\frac{1}{6}$, $k = 1$ w.p. $(1 - \frac{1}{6})\frac{13}{25} = \frac{13}{30}$, and $k = 2$ w.p. $(1 - \frac{1}{6})(1 - \frac{13}{25}) = \frac{2}{5}$. The respective forbidden sets are $\{2\}$, $\{2,4\}$, and $\{2,4,5\}$. When $4 \notin J$ and $F = \{2\}$, the revenue-maximizing assortment is $\{1,4,6\}$ (guaranteeing the sales of items 4 and 6). Alternatively, if $\{4,5,6\} \cap (J \cup F) = \{4\}$, then $\tilde{S} = \{1,5\}$. Finally, if $\{4,5,6\} \cap (J \cup F) = \{4,5\}$, then $\tilde{S} = \{1,6\}$. Note that since $\tilde{S}$ only depends on $J \cup F$, if $4 \in J$, then it doesn't matter whether $k = 0$ ($F = \{2\}$) or $k = 1$ ($F = \{2,4\}$). The final possibilities for the state at the end of time $t = 3$ can be computed to be:

$$(I_1, \dots, I_6) = \begin{cases} (\infty, ?, 0, 0, 0, 1) \text{ w.p. } \frac{2}{5} \cdot (\frac{1}{6} + \frac{13}{30}) = \frac{6}{25}; \\ (\infty, ?, 0, 0, 1, 0) \text{ w.p. } \frac{2}{5} \cdot \frac{2}{5} + \frac{3}{5} \cdot \frac{1}{6} = \frac{13}{50}; \\ (\infty, ?, 0, 1, 0, 1) \text{ w.p. } \frac{3}{5} \cdot \frac{13}{30} = \frac{13}{50}; \\ (\infty, ?, 0, 1, 1, 0) \text{ w.p. } \frac{3}{5} \cdot \frac{2}{5} = \frac{6}{25}. \end{cases}$$

It can be seen that $\Pr[I_4 = 0] = \Pr[I_5 = 0] = \Pr[I_6 = 0] = \frac{13}{50} + \frac{6}{25} = \frac{1}{2}$, i.e. the add-on thresholds for items 4, 5, and 6 are respected. Now we explain how we came up with protection lists that accomplish this.

3.3.2 Computing Protection Lists

Definition 3.3.4. For all $t \in [T]$, given arrival types $i_1, \dots, i_t$ and protection lists $\mathcal{L}^1, \dots, \mathcal{L}^t$, consider a run of $\mathsf{SellTo}(t, i_1, \dots, i_t, \mathcal{L}^1, \dots, \mathcal{L}^t)$. We use the phrase *final sub-iteration* to refer to sales made by SellTo from an $\tilde{S}$ chosen when $s = t, k = K_t$ (K_t is the length of list $\mathcal{L}^t$). Define the following functions for each $j \in [n]$:

- let $d_j(t, i_1, \dots, i_t, \mathcal{L}^1, \dots, \mathcal{L}^t) \in [0, b_j]$ be the expected units of item j sold as an add-on *before* the final sub-iteration, when SellTo is run on input $(t, i_1, \dots, i_t, \mathcal{L}^1, \dots, \mathcal{L}^t)$;
- let $h_j(t, i_1, \dots, i_t, \mathcal{L}^1, \dots, \mathcal{L}^t) \in [0, 1]$ be the probability that item j is sold as an add-on *during* the final sub-iteration, when SellTo is run on input $(t, i_1, \dots, i_t, \mathcal{L}^1, \dots, \mathcal{L}^t)$.

The algorithm maintains d_j and computes h_j during its execution. In general, h_j cannot be computed in polynomial time because the state space is exponential in n, but later we show that it can be approximated in polynomial time via sampling, with error ε as small as necessary. For now, it may aid comprehension to assume oracle access to the function h_j and set $\varepsilon = 0$.

The protection lists are constructed inductively. Pseudocode 4 explains how to construct list $\mathcal{L}^t$ (after observing customer type i_t), given previous lists $\mathcal{L}^1, \dots, \mathcal{L}^{t-1}$. For all $j \in [n]$, d_j^{curr} is the value maintained by the algorithm for $d_j(t, i_1, \dots, i_t, \mathcal{L}^1, \dots, \mathcal{L}^{t-1}, ())$, where

() is the empty list. $d_1^{\text{curr}}, \ldots, d_n^{\text{curr}}$ are required as input to Pseudocode 4; they are also updated in Pseudocode 4.

Algorithm 4 $\mathsf{ConstructList}(t, i_1, \ldots, i_t, \mathcal{L}^1, \ldots, \mathcal{L}^{t-1}, d_1^{\text{curr}}, \ldots, d_n^{\text{curr}}, \varepsilon) \rightarrow (\mathcal{L}^t, d_1^{\text{curr}}, \ldots, d_n^{\text{curr}})$

```
𝓛 ← (), K = 0
loop
    G ← [n] \ (𝓕_K^t(𝓛^1, …, 𝓛^{t−1}, 𝓛) ∪ {i_t})
    ĥ_j ← h_j(t, i_1, …, i_t, 𝓛^1, …, 𝓛^{t−1}, 𝓛) (or an approximation) for all j ∈ G
    if d_j^curr + ĥ_j ≤ b_j/2 − ε for all j ∈ G then
        d_j^curr ← d_j^curr + ĥ_j for all j ∈ G
        return (𝓛, d_1^curr, …, d_n^curr)
    end if
    ℓ ← argmin_{j∈G} ((b_j/2 − ε) − d_j^curr)/ĥ_j , ρ ← ((b_ℓ/2 − ε) − d_ℓ^curr)/ĥ_ℓ
    append (ℓ, ρ) to list 𝓛 and increment K by 1
    d_j^curr ← d_j^curr + ρ · ĥ_j for all j ∈ G
end loop
```

Example 3.3.5 (Running Example). To help with understanding Pseudocode 4, we demonstrate its execution (with $\varepsilon = 0$) on a further expanded version of the running example.

- $b_1 = \infty$, $b_2 = \ldots = b_8 = 1$;
- $r_2 \gg \ldots \gg r_8 \gg r_1 = 0$; $r_j = r_j^{\text{disc}}$ for all j;
- $\mathcal{S}_1 = \big\{\{1,2\}, \{1,3\}, \{1,4,6\}, \{1,5\}, \{1,7,8\}\big\}$:
  - $\phi_1(\{1,2\}, \{1,2\}) = \frac{3}{4}$; $\phi_1(\{1,3\}, \{1,3\}) = 1$;
  - $\phi_1(\{1,4,6\}, \{1,4,6\}) = 1$; $\phi_1(\{1,5\}, \{1,5\}) = 1$;
  - $\phi_1(\{1,7,8\}, \{1,7,8\}) = 1$;
- $\mathcal{S}_3 = \big\{\{3,4\}\big\}$; $\phi_3(\{3,4\}, \{3,4\}) = \frac{3}{5}$;
- $(i_1, i_2, i_3, i_4) = (1, 3, 1, 1)$.

Consider time $t = 1$. At the very start, d_j^{curr} has been initialized to $d_j(1, i_1, ()) = 0$ for all $j \in [8]$. During the first run through the loop, $\mathcal{L}$ is (), and G, the set of items which have not reached their add-on thresholds, is $\{2,\ldots,8\}$. With an empty protection list, SellTo deterministically offers assortment $\{1,2\}$ during time 1, resulting in item 2 being sold as an add-on with probability $h_2(1,i_1,\mathcal{L})=\frac{3}{4}$. Therefore, $\hat{h}_2>\frac{1}{2}$, and the add-on threshold in Step 5 is violated for $j=2$. In Step 9, $\ell=2$ and $\rho=\dfrac{\frac{1}{2}-0}{\frac{3}{4}}=\frac{2}{3}$. ρ is computed so that had the algorithm offered $\{1,2\}$ with probability ρ (instead of probability 1), then the expected units of item 2 sold as an add-on would be exactly $\frac{b_2}{2}$, as indicated by the updates in Step 11.

The algorithm tries again with $\mathcal{L}=\big((2,\frac{2}{3})\big)$. Now, $\hat{h}_3=\frac{1}{3}$, since the probability of getting to the final sub-iteration is $1-\frac{2}{3}=\frac{1}{3}$, after which item 3 is deterministically sold as an add-on. The add-on thresholds in Step 5 are upheld and the algorithm updates $d_3^{\mathrm{curr}}\leftarrow\frac{1}{3},\mathcal{L}^1\leftarrow\mathcal{L}$. Note that $2\notin G$ and d_2^{curr} will never again be updated, remaining at $\frac{1}{2}$ until the end of the selling season.

Now consider time $t=2$. It is straight-forward to see that $\hat{h}_4$ is the only non-zero $\hat{h}_j$, with $\hat{h}_4=\frac{2}{3}\cdot\frac{3}{5}=\frac{2}{5}$. The algorithm updates $d_4^{\mathrm{curr}}\leftarrow\frac{2}{5}$ and proceeds to time 3 with $\mathcal{L}^2=()$.

From the analysis in the previous subsection, the state of the inventory $(I_1,\ldots,I_8)$ at the start of time 3 is $(\infty,?,0,0,1,1,1,1)$ with probability $\frac{2}{5}$, and $(\infty,?,0,1,1,1,1,1)$ with probability $\frac{3}{5}$. The expectations $(d_1^{\mathrm{curr}},\ldots,d_8^{\mathrm{curr}})$ are at $(0,\frac{1}{2},\frac{1}{3},\frac{2}{5},0,0,0,0)$. Now consider SellTo$(3,i_1,i_2,i_3,\mathcal{L}^1,\mathcal{L}^2,\mathcal{L})$, where $\mathcal{L}=()$. With probability $\frac{2}{5}$ (when $I_4=0$), the subroutine will offer $\{1,5\}$ (since $4\in J$), resulting in the deterministic sale of item 5 as an add-on. With probability $\frac{3}{5}$, the subroutine will offer $\{1,4,6\}$ and sell items 4 and 6 as add-ons. Therefore, $\hat{h}_4=\hat{h}_6=\frac{3}{5}$, while $\hat{h}_5=\frac{2}{5}$. The add-on threshold in Step 5 is violated for both $j=4$ and $j=6$. In Step 9, the expression $\dfrac{\frac{b_j}{2}-d_j^{\mathrm{curr}}}{\hat{h}_j}$ is equal to $\dfrac{\frac{1}{2}-\frac{2}{5}}{\frac{3}{5}}=\frac{1}{6}$ when $j=4$, and $\dfrac{\frac{1}{2}-0}{\frac{3}{5}}=\frac{5}{6}$ when $j=6$. Therefore, we choose $\ell=4$ and $\rho=\frac{1}{6}$. In Step 11, $d_4^{\mathrm{curr}}\leftarrow\frac{2}{5}+\frac{1}{6}\cdot\frac{3}{5}=\frac{1}{2}$, $d_5^{\mathrm{curr}}\leftarrow\frac{1}{6}\cdot\frac{2}{5}=\frac{1}{15}$, and $d_6^{\mathrm{curr}}\leftarrow\frac{1}{6}\cdot\frac{3}{5}=\frac{1}{10}$. Note that choosing an ℓ that *minimizes* ρ is important for respecting the add-on thresholds; had we chosen $\ell=6$ and $\rho=\frac{5}{6}$, then d_4^{curr} would be updated to $\frac{2}{5}+\frac{5}{6}\cdot\frac{3}{5}>\frac{1}{2}$.

The algorithm tries again with updated list $\mathcal{L}=\big((4,\frac{1}{6})\big)$ and updated expectations $(d_1^{\mathrm{curr}},\ldots,d_8^{\mathrm{curr}})=(0,\frac{1}{2},\frac{1}{3},\frac{1}{2},\frac{1}{15},\frac{1}{10},0,0)$. The inventory possibilities are still the same, and the probability that SellTo gets to the final sub-iteration is $1-\frac{1}{6}=\frac{5}{6}$. Regardless of whether $I_4=1$, it will offer (and successfully sell) item 5 as an add-on, so $\hat{h}_5=\frac{5}{6}$, which exceeds $\frac{1}{2}$. Therefore, $\ell = 5$ and $\rho = \frac{\frac{1}{2} - \frac{1}{15}}{\frac{5}{6}} = \frac{13}{25}$. Note that item 6 violated its add-on threshold with an earlier list, but no longer violates its add-on threshold even though d_6^{curr} increased with the earlier iteration. This is why adding items to the list one at a time, even when multiple items violate their add-on thresholds, is important.

The algorithm tries again having appended $(5, \frac{13}{25})$ to $\mathcal{L}$ and updated d_5^{curr} to $\frac{1}{2}$. Now SellTo gets to the final sub-iteration with probability $(1 - \frac{1}{6})(1 - \frac{13}{25}) = \frac{2}{5}$, in which case it will deterministically sell item 6 as an add-on. Therefore, $\hat{h}_6 = \frac{2}{5}$, which when added to $d_6^{\mathrm{curr}} = \frac{1}{10}$ is exactly $\frac{1}{2}$. The algorithm proceeds to the next time period with $\mathcal{L}^3 \leftarrow \mathcal{L}$ even though d_6^{curr} will be $\frac{1}{2}$; item 6 will immediately get added to $\mathcal{L}$ if SellTo attempts to offer it as an add-on again.

In $t = 4$, the expected units of each item sold as an add-on, $(d_1^{\mathrm{curr}}, \ldots, d_8^{\mathrm{curr}})$, is $(0, \frac{1}{2}, \frac{1}{3}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 0, 0)$. Item 6 has inventory remaining with probability $\frac{1}{2}$, since $d_6^{\mathrm{curr}} = \frac{1}{2}$. Note that 6 is in G during the first iteration where $\mathcal{L} = ()$. However, the algorithm will compute $\hat{h}_6 = \frac{1}{2}$, $\rho = 0$, and try again with $\mathcal{L} = \big((6, 0)\big)$. In this case, SellTo deterministically gets to the final sub-iteration and offers $\{1, 7, 8\}$, resulting in $\hat{h}_7 = \hat{h}_8 = 1$. ρ will be $\frac{1}{2}$, resulting in $d_7^{\mathrm{curr}} \leftarrow \frac{1}{2}, d_8^{\mathrm{curr}} \leftarrow \frac{1}{2}$, and the algorithm can break ties arbitrarily and choose $\ell = 7$. With $\mathcal{L} = \big((6, 0), (7, \frac{1}{2})\big)$, SellTo gets to the final sub-iteration with probability $\frac{1}{2}$, causing $\hat{h}_8$ to be $\frac{1}{2}$. The algorithm appends $(8, 0)$ to $\mathcal{L}$, after which SellTo is finally exhausted of add-ons to offer (even though $d_3^{\mathrm{curr}} < \frac{b_3}{2}$, I_3 must be 0, since a customer of type 3 arrived), so it simply offers $\{1\}$ in the final sub-iteration. The final $\mathcal{L}^4$ is $\big((6, 0), (7, \frac{1}{2}), (8, 0)\big)$.

Proposition 3.3.6. *Consider an iteration of the loop in Pseudocode 4, with starting values for $\mathcal{L}, K$ and $d_1^{\mathrm{curr}}, \ldots, d_n^{\mathrm{curr}}$. Let $\varepsilon_1, \varepsilon_2 \geq 0$. Suppose for all $j \in [n]$, d_j^{curr} is within ε_1 of the expected units of item j sold as an add-on before the iteration, i.e. $\big|d_j^{\mathrm{curr}} - d_j(t, i_1, \ldots, i_t, \mathcal{L}^1, \ldots, \mathcal{L}^{t-1}, \mathcal{L})\big| \leq \varepsilon_1$. Furthermore, suppose that for $j \in G$, the $\hat{h}_j$ computed in Step 4 satisfies $\big|\hat{h}_j - h_j(t, i_1, \ldots, i_t, \mathcal{L}^1, \ldots, \mathcal{L}^{t-1}, \mathcal{L})\big| \leq \varepsilon_2$.*

Then for $j \in G$, in Step 6 or Step 11, Pseudocode 4 updates d_j^{curr} to be within $\varepsilon_1 + \varepsilon_2$ of the expected units of item j sold as an add-on after the iteration. That is, when the if statement in Step 5 is entered,

$$\big|(d_j^{\mathrm{curr}} + \hat{h}_j) - d_j(t+1, i_1, \ldots, i_{t+1}, \mathcal{L}^1, \ldots, \mathcal{L}^{t-1}, \mathcal{L}, ())\big| \leq \varepsilon_1 + \varepsilon_2,$$

and otherwise,

$$\left|(d_j^{\mathrm{curr}} + \rho \cdot \hat{h}_j) - d_j(t, i_1, \ldots, i_t, \mathcal{L}^1, \ldots, \mathcal{L}^{t-1}, \mathcal{L} + (\ell, \rho))\right| \leq \varepsilon_1 + \varepsilon_2.$$

All proofs in this section are straight-forward from the construction of the algorithm and deferred to Section B.3. If we are assuming oracle access to h_j, then no error is ever accumulated in $d_1^{\mathrm{curr}}, \ldots, d_n^{\mathrm{curr}}$, i.e. $\varepsilon_1 = \varepsilon_2 = 0$, in which case Proposition 3.3.6 simply says that Pseudocode 4 correctly updates the probabilities $d_1^{\mathrm{curr}}, \ldots, d_n^{\mathrm{curr}}$.

The following propositions also help conclude that Pseudocodes 3–4 accomplishes their goals:

Proposition 3.3.7. *Consider an iteration of the loop in Pseudocode 4. Suppose for all $j \in [n]$, d_j^{curr} does not violate its add-on threshold, i.e. $d_j^{\mathrm{curr}} \leq \frac{b_j}{2} - \varepsilon$. Then for these $j \in G$, the updated value for d_j^{curr} is still at most $\frac{b_j}{2} - \varepsilon$.*

Proposition 3.3.8. *Any item $j \in [n]$ can get added to the protection lists at most once. That is, in Step 9 of Pseudocode 4, ℓ does not appear in $\mathcal{L}^1, \ldots, \mathcal{L}^{t-1}, \mathcal{L}$. Furthermore, ℓ is never again offered as an add-on, and d_j^{curr} is not increased, after sub-iteration K of iteration t.*

3.3.3 The Combined Algorithm, with Sampling

We now address how to compute the probability $h_j(t, i_1, \ldots, i_t, \mathcal{L}^1, \ldots, \mathcal{L}^{t-1}, \mathcal{L})$ in Step 4 of Pseudocode 4. Fix some large integer M whose value is to be determined later. We simultaneously estimate $\hat{h}_j$ for all $j \in G$ using the naive sampling algorithm in Pseudocode 5.

Algorithm 5 $\mathsf{SampleAddon}(t, i_1, \ldots, i_t, \mathcal{L}^1, \ldots, \mathcal{L}^t, M) \to \{\hat{h}_j : j \in G\}$

1: $G \leftarrow [n] \setminus \left(\mathcal{F}^t_{K_t}(\mathcal{L}^1, \ldots, \mathcal{L}^t) \cup \{i_t\}\right)$, where K_t is the length of list $\mathcal{L}^t$

2: $C_j \leftarrow 0$ for all $j \in G$

3: **repeat**

4: run $\mathsf{SellTo}(t, i_1, \ldots, i_t, \mathcal{L}^1, \ldots, \mathcal{L}^t)$, virtually generating the purchase decisions in Step 17

5: $C_j \leftarrow C_j + 1$ for all $j \in G$ that was sold as an add-on in sub-iteration K_t of iteration t

6: **until** M runs have passed

7: **return** $\{\frac{C_j}{M} : j \in G\}$

Its accuracy can be bounded immediately by Chernoff-Hoeffding (see Lugosi (2009)):

Lemma 3.3.9. *Recall that $h_j(t, i_1, \ldots, i_t, \mathcal{L}^1, \ldots, \mathcal{L}^t)$, or h_j for short, is the probability that j is sold as an add-on during the final sub-iteration of* $\mathsf{SellTo}(t, i_1, \ldots, i_t, \mathcal{L}^1, \ldots, \mathcal{L}^t)$. *Therefore, for all $j \in G$ and any $\varepsilon_2 > 0$, the estimate $\frac{C_j}{M}$ provided in Step 7 of Pseudocode 5 satisfies*

$$\Pr\left[\left|\frac{C_j}{M} - h_j\right| > \varepsilon_2\right] \le 2e^{-2\varepsilon_2^2 M}.$$

We are finally ready to put the online, sampling personalized assortment algorithm together.

Algorithm 6 $\mathsf{OnlineSamplingPersonalizedAssortment}(\varepsilon)$

$d_j^{\text{curr}} \leftarrow 0$ for all $j \in [n]$

for $t = 1, \ldots, T$ **do**

observe customer type i_t

run $\mathsf{ConstructList}(t, i_1, \ldots, i_t, \mathcal{L}^1, \ldots, \mathcal{L}^{t-1}, d_1^{\text{curr}}, \ldots, d_n^{\text{curr}}, \varepsilon)$, using $\mathsf{SampleAddon}$ with $M = \frac{1}{2}(\frac{T+n}{\varepsilon})^2 \ln \frac{2n(T+n)}{\varepsilon}$ to estimate the functions $h_1, \ldots, h_n$

set $\mathcal{L}^t$ and update $d_1^{\text{curr}}, \ldots, d_n^{\text{curr}}$ according to the results of the previous step

offer an assortment to customer t in reality, using Steps 3 to 17 of $\mathsf{SellTo}(t, i_1, \ldots, i_t, \mathcal{L}^1, \ldots, \mathcal{L}^t)$ with $s = t$, and update inventories according to her purchase decision

end for

When a new customer arrives, the algorithm first runs itself from the start using the known arrivals, repeating this process a large number of times to estimate the expectations. Then the algorithm uses these estimates to offer the best assortment in reality. This iterative sampling over time periods is similar to what is done in Chapter 7, where an offline algorithm that makes decisions based on expectations over all sample paths, instead of realizations on the current sample path, is presented. To our knowledge, we present the first online algorithm of this type, and the example from Section B.2 demonstrates the necessity of making decisions based on expectations.

Theorem 3.3.10. *Fix some small $\varepsilon > 0$ and consider* $\mathsf{OnlineSamplingPersonalizedAssortment}(\varepsilon)$*. For all $j \in [n]$, let $\bar{d}_j$ denote the total expected units of item j sold as an add-on, and let $\hat{d}_j$ denote the recorded value of d_j^{curr}, at the end of the selling season.*

Then with probability at least $1-\varepsilon$, $\hat{d}_j - \varepsilon \leq \bar{d}_j \leq \frac{b_j}{2}$ for all $j \in [n]$. Furthermore, $\mathsf{OnlineSamplingPersonalizedAssortment}(\varepsilon)$ *terminates in time polynomial in T, n, and $\frac{1}{\varepsilon}$, assuming that we can solve the single-period assortment optimization problem in constant time.*

It may be tempting to make the sampling more efficient by storing the states of the M sample paths as t progresses from 1 to T, so that we do not have to run SellTo from period 1 every time. However, this would make the state of the sample paths before the final sub-iteration of $\mathsf{SellTo}(t, i_1, \ldots, i_t, \mathcal{L}^1, \ldots, \mathcal{L}^t)$ depend on $\mathcal{L}^t$, resulting in a biased sample.

3.4 Analysis

In this section we prove Theorem 3.2.9, which bounds the competitive ratio of our algorithm $\mathsf{OnlineSamplingPersonalizedAssortment}(\varepsilon)$. We reuse the definitions from the statement of Theorem 3.3.10: $\hat{d}_j$ is our algorithm's value of d_j^{curr} at the end of the selling season, while $\bar{d}_j$ is the true expected units of item j sold as an add-on. Again, it may aid comprehension to assume that $\varepsilon = 0$ and $\hat{d}_j = \bar{d}_j$ for all $j \in [n]$; i.e., our algorithm had oracle access to the sales probabilities instead of having to estimate them.

We provide a general roadmap of the proof:

1. We define each θ_j, the dual variable for the inventory constraint on item j, based on

$\bar{d}_j$ and $\hat{d}_j$, providing shadow prices for each unit of starting inventory;

2. To ensure feasibility, we also need to have dual variables λ^t which provide shadow prices for each customer t (Lemma 3.4.3);

3. We bound from above the value of the LP defined in Section 3.2.2 using weak duality (Lemma 3.4.4);

4. To bound the revenue our algorithm earns from selling add-ons to customer t, we need to bound the probability of customer t being served and the probabilities of add-ons not stocking out, which we accomplish given the protection levels in expectation (Lemma 3.4.7);

5. This allows us to obtain a lower bound on the expected revenue of our algorithm (Lemma 3.4.8);

6. Finally, using a variant of Markov's inequality (Proposition 3.4.9), we show that the lower bound from Step 5 is at least $\frac{1}{4} - \varepsilon$ of the upper bound from Step 3, and that the ε-error in sampling translates to an ε-loss in revenue (Section 3.4.3).

3.4.1 Bounding OPT via Duality

Definition 3.4.1. We define the following notation, for all $j \in [n]$:

- Let a_j denote the number of arriving customers of type j. That is, $a_j = \big|\{t \in [T] : i_t = j\}\big|$.
- Set d_j as follows:

$$d_j = \begin{cases} \max\{\bar{d}_j, \hat{d}_j + \varepsilon\} \text{ if } \hat{d}_j \geq \frac{1}{2} - \varepsilon; \\ \bar{d}_j \text{ otherwise.} \end{cases}$$

Note that if $\varepsilon = 0$, then $d_j = \bar{d}_j = \hat{d}_j$.

It simplifies the analysis to assume that $a_j \leq b_j$ for all $j \in [n]$. This can be justified by the fact that it is strictly dominant to serve all customers if possible, i.e. offer to each customer t an assortment containing at least item i_t as long as i_t has remaining inventory. Since customers of type j always buy item j when offered, the $b_j + 1$'st customer requesting each item would never get served.

Definition 3.4.2. For each $i \in [n]$, introduce a fixed

$$S_i^* \in \text{argmax}_{S \in \mathcal{S}_i} \sum_{j \neq i} r_j^{\mathsf{disc}}(1 - \frac{a_j + 2d_j}{b_j})p_{ij}(S).$$

We assume that S_i^* is chosen to not contain any $j \neq i$ such that $a_j + 2d_j \geq b_j$. This assumption is WOLOG: by removing all such elements from S, we have not removed any positive terms from the sum, and for the remaining $j' \in S'$ (where $S' := S \setminus \{j : j \neq i, a_j + 2d_j \geq b_j\}$), $p_{ij'}(S') \geq p_{ij'}(S)$ holds by weak substitutability.

With this assumption, $d_j < \frac{b_j}{2}$ for all $j \in S_i^* \setminus \{i\}$. By definition, either $d_j \geq \hat{d}_j + \varepsilon$ which would imply $\hat{d}_j < \frac{b_j}{2} - \varepsilon$, or $\hat{d}_j < \frac{1}{2} - \varepsilon$. In both cases, the conclusion is that $\hat{d}_j < \frac{b_j}{2} - \varepsilon$, since b_j is always at least 1. This means that at any point in our algorithm, the value of $d_j^{\mathtt{curr}}$ never reached the threshold of $\frac{b_j}{2} - \varepsilon$. As a result, items in $S_i^* \setminus \{i\}$ do not appear in any protection list and are never in the forbidden set, for all $i \in [n]$.

Now we are ready to specify the bound on $\mathsf{OPT}(i_1, \ldots, i_T)$, which is the optimal objective value of the LP defined in Section 3.2.2. The dual of this LP can be written as follows, where $\{\theta_j : j \in [n]\}$ are the variables corresponding to the first n primal constraints, and $\{\lambda^t : t \in [T]\}$ are the variables corresponding to the next T primal constraints:

$$\begin{array}{llcll}
\min & \sum_{j=1}^{n} b_j\theta_j + \sum_{t=1}^{T} \lambda^t & & & \\
\text{s.t.} & \sum_{j=1}^{n} p_{i_t j}(S)\theta_j + \lambda^t & \geq & r_{i_t} + \sum_{j \neq i_t} r_j^{\mathsf{disc}} p_{i_t j}(S) & t \in [T], S \in \mathcal{S}_{i_t} \\
& \theta_j & \geq & 0 & j \in [n] \\
& \lambda^t & \geq & 0 & t \in [T].
\end{array}$$

We propose the following values for the dual variables:

$$\begin{aligned}
\theta_j &= \frac{r_j a_j + 2r_j^{\mathsf{disc}} d_j}{b_j} && j \in [n], \\
\lambda^t &= r_{i_t}(1 - \frac{a_{i_t}}{b_{i_t}}) + \sum_{j \neq i_t} r_j^{\mathsf{disc}}(1 - \frac{a_j + 2d_j}{b_j})p_{i_t j}(S_{i_t}^*) && t \in [T].
\end{aligned} \tag{3.6}$$

Golrezaei et al. (2014) were able to define dual variables that are sample-path dependent, and prove feasibility on every sample path. We need to define our dual variables based on the d_j's, which can be seen as aggregate statistics over all the sample paths, in order to prove feasibility. Note that the value of λ^t is fully determined by the type of the customer, i_t.

Lemma 3.4.3 (Dual Feasibility). *The solution of the dual LP defined in (3.6) is feasible.*

Proof. Proof. We know that $a_{i_t} \leq b_{i_t}$ for all $t \in [T]$, by assumption. Also, for $j \neq i_t$, $p_{i_t j}(S^*_{i_t})$ can only be non-zero for $j \in S^*_{i_t}$, which implies $a_j + 2d_j \leq b_j$. Therefore, $\lambda^t \geq 0$. It is also easy to see that $\theta_j \geq 0$ for all $j \in [n]$.

The remaining constraints can be rearranged as

$$\lambda^t \geq r_{i_t} - \theta_{i_t} + \sum_{j \neq i_t} p_{i_t j}(S)(r_j^{\mathsf{disc}} - \theta_j), \tag{3.7}$$

where we have used the fact that $p_{ii}(S) = 1$ for all i and $S \in \mathcal{S}_i$.

Take an arbitrary $t \in [T]$ and $S \in \mathcal{S}_{i_t}$. By our definition of the dual variables, $\theta_j \geq r_j^{\mathsf{disc}} \cdot \frac{a_j + 2d_j}{b_j}$ for all $j \neq i_t$, since $r_j \geq r_j^{\mathsf{disc}}$. Also, it is immediate that $\theta_{i_t} \geq r_{i_t} \cdot \frac{a_{i_t}}{b_{i_t}}$. Therefore, the RHS of (3.7) is at most

$$r_{i_t}(1 - \frac{a_{i_t}}{b_{i_t}}) + \sum_{j \neq i_t} r_j^{\mathsf{disc}}(1 - \frac{a_j + 2d_j}{b_j}) p_{i_t j}(S).$$

Now, $S^*_{i_t}$ was specifically chosen to maximize the second part of the preceding expression among all $S \in \mathcal{S}_{i_t}$. Therefore, the preceding expression is no greater than our definition of λ^t, completing the proof of dual feasibility. □

Lemma 3.4.4 (Dual Objective Value). *The objective value of the dual solution in (3.6) is at most*

$$\sum_{i=1}^{n} 2r_i a_i + \sum_{j=1}^{n} 2r_j^{\mathsf{disc}} d_j + \sum_{i=1}^{n} a_i \sum_{j \neq i} r_j^{\mathsf{disc}}(1 - \frac{a_j + 2d_j}{b_j}) p_{ij}(S^*_i).$$

Proof. Proof. Substituting our values of θ_j and λ^t into the objective function of the dual LP, we get

$$\begin{aligned}
\sum_{j=1}^{n} b_j \theta_j + \sum_{t=1}^{T} \lambda^t &= \sum_{j=1}^{n} (r_j a_j + 2r_j^{\mathsf{disc}} d_j) + \sum_{t=1}^{T} \Big(r_{i_t}(1 - \frac{a_{i_t}}{b_{i_t}}) + \sum_{j \neq i_t} r_j^{\mathsf{disc}}(1 - \frac{a_j + 2d_j}{b_j}) p_{i_t j}(S^*_{i_t}) \Big) \\
&= \sum_{j=1}^{n} r_j a_j + \sum_{j=1}^{n} 2r_j^{\mathsf{disc}} d_j + \sum_{i=1}^{n} \sum_{t : i_t = i} \Big(r_i(1 - \frac{a_i}{b_i}) + \sum_{j \neq i} r_j^{\mathsf{disc}}(1 - \frac{a_j + 2d_j}{b_j}) p_{ij}(S^*_i) \Big).
\end{aligned}$$

Noting the fact a_i is defined to be the cardinality of the set $\{t \in [T] : i_t = i\}$ and rearranging the terms, we have

$$\sum_{j=1}^{n} b_j \theta_j + \sum_{t=1}^{T} \lambda^t \leq \sum_{i=1}^{n} 2r_i a_i + \sum_{j=1}^{n} 2r_j^{\mathsf{disc}} d_j + \sum_{i=1}^{n} a_i \sum_{j \neq i} r_j^{\mathsf{disc}} (1 - \frac{a_j + 2d_j}{b_j}) p_{ij}(S_i^*),$$

which completes the proof. □

Combining Lemmas 3.4.3–3.4.4 and applying weak duality, we have established in this subsection that the expression from Lemma 3.4.4 is an upper bound on $\mathsf{OPT}(i_1, \ldots, i_T)$.

3.4.2 Bounding the Algorithm's Revenue

Definition 3.4.5. For a run of our algorithm, define the following random variables:

- For all $t \in [T]$ and $j \in [n]$, let I_j^t denote the inventory of item j remaining at the *end* of time period t. It is understood that $I_j^0 = b_j$, its starting inventory, for all items $j \in [n]$.
- For all $t \in [T]$, let $S^t \in \mathcal{S}_{i_t} \cup \{\emptyset\}$ denote the assortment offered at time t.
- For all $t \in [T]$ and $j \in [n]$, let P_j^t be the indicator random variable for whether customer t bought item j. P_j^t can only potentially be 1 for $j \in S^t$.
- For all $j \in [n]$, let D_j denote the units of item j sold as an add-on by the end of the selling season. Note that $D_j = \sum_{t: i_t \neq j} P_j^t$, and $\mathbb{E}[D_j] = \bar{d}_j$.
- For all $t \in [T]$, let V^t denote the items in $S_{i_t}^*$ with inventory available at time t. That is, $V^t = \{j \in S_{i_t}^* : I_j^{t-1} > 0\}$.

The following proposition is immediate:

Proposition 3.4.6. *For all $j \in [n]$, the units of item j sold for its full price by the end of the selling season is* $\min\{b_j - D_j, a_j\}$.

Proof. Proof. For any $j \in [n]$, consider a single sample path with a fixed value for D_j. The units of item j remaining to sell at full price is $b_j - D_j$. Since our algorithm offers item j to every customer requesting it as long as inventory is available, the number of these customers served is $\min\{b_j - D_j, a_j\}$. □

Now we analyze the revenue earned by our algorithm in expectation and compare it to Lemma 3.4.4. The following lemma is crucial in bounding the revenue earned from selling add-ons.

Lemma 3.4.7 (Bound on Add-on Revenue). *For all $t \in [T]$, the revenue earned by our algorithm from selling add-ons during time t, $\sum_{j \neq i_t} r_j^{\mathsf{disc}} P_j^t$, is in expectation at least*

$$\sum_{j \in S^*_{i_t}, j \neq i_t} r_j^{\mathsf{disc}} p_{i_t j}(S^*_{i_t}) \big(\Pr[I_{i_t}^{t-1} > 0] - \frac{\bar{d}_j}{b_j - a_j} \big).$$

(Recall that for all $j \in S^*_{i_t} \setminus \{i_t\}$, we have $a_j + 2d_j < b_j$, which implies $b_j - a_j \neq 0$.)

Proof. Proof. Fix an arbitrary $t \in [T]$. We have

$$\begin{aligned} \mathbb{E}\Big[\sum_{j \neq i_t} r_j^{\mathsf{disc}} P_j^t\Big] &= \mathbb{E}\Big[\sum_{j \neq i_t} r_j^{\mathsf{disc}} p_{i_t j}(S^t)\Big] \\ &= \mathbb{E}\Big[\sum_{j \neq i_t} r_j^{\mathsf{disc}} p_{i_t j}(S^t) \Big| I_{i_t}^{t-1} > 0\Big] \Pr[I_{i_t}^{t-1} > 0]. \end{aligned} \tag{3.8}$$

Both equalities use the law of total expectation. The first equality holds because $\mathbb{E}\Big[\sum_{j \neq i_t} r_j^{\mathsf{disc}} P_j^t\Big] = \mathbb{E}_{S^t}\Big[\mathbb{E}\big[\sum_{j \neq i_t} r_j^{\mathsf{disc}} P_j^t | S^t\big]\Big]$, and the inner expectation is equal to $\sum_{j \neq i_t} r_j^{\mathsf{disc}} p_{i_t j}(S^t)$, since for a fixed S^t, P_j^t is an independent binary random variable which is 1 with probability $p_{i_t j}(S^t)$. The second equality holds because if $I_{i_t}^{t-1} = 0$, then $S^t = \emptyset$, and $p_{i_t j}(\emptyset) = 0$ for all j.

Now, we claim that on every sample path where $I_{i_t}^{t-1} > 0$, the following holds:

$$\begin{aligned} \sum_{j \neq i_t} r_j^{\mathsf{disc}} p_{i_t j}(S^t) &\geq \sum_{j \neq i_t} r_j^{\mathsf{disc}} p_{i_t j}(V^t) \\ &= \sum_{j \in V^t, j \neq i_t} r_j^{\mathsf{disc}} p_{i_t j}(V^t) \\ &\geq \sum_{j \in V^t, j \neq i_t} r_j^{\mathsf{disc}} p_{i_t j}(S^*_{i_t}). \end{aligned}$$

The first inequality is true because conditioned on $I_{i_t}^{t-1} > 0$, V^t was a feasible choice of assortment during the maximization step (Step 12) of Pseudocode 3 (add-ons in V^t are never in the forbidden set, and have inventory available). The second inequality is true because conditioned on $I_{i_t}^{t-1} > 0$, V^t is always a subset of $S^*_{i_t}$ containing i_t, and by weak substitutability, $p_{i_t j}(V^t) \geq p_{i_t j}(S^*_{i_t})$ for all $j \in V^t$. Thus (3.8) is at least

$$\begin{aligned}
&\mathbb{E}\Big[\sum_{j \in V^t, j \neq i_t} r_j^{\mathsf{disc}} p_{i_t j}(S^*_{i_t}) \Big| I_{i_t}^{t-1} > 0\Big] \Pr[I_{i_t}^{t-1} > 0] \\
= &\sum_{j \in S^*_{i_t}, j \neq i_t} r_j^{\mathsf{disc}} p_{i_t j}(S^*_{i_t}) \Pr\left[I_j^{t-1} > 0 \middle| I_{i_t}^{t-1} > 0\right] \Pr[I_{i_t}^{t-1} > 0] \\
= &\sum_{j \in S^*_{i_t}, j \neq i_t} r_j^{\mathsf{disc}} p_{i_t j}(S^*_{i_t}) \Pr\left[(I_j^{t-1} > 0) \cap (I_{i_t}^{t-1} > 0)\right] \\
\geq &\sum_{j \in S^*_{i_t}, j \neq i_t} r_j^{\mathsf{disc}} p_{i_t j}(S^*_{i_t}) \big(\Pr[I_{i_t}^{t-1} > 0] - \Pr[I_j^{t-1} = 0]\big),
\end{aligned}$$

where in the first equality we have used the definition that V^t is the subset of $S^*_{i_t}$ with inventory remaining.

Finally, $I_j^{t-1} = 0$ implies $I_j^T = 0$. By Proposition 3.4.6, this event is equivalent to $D_j + \min\{b_j - D_j, a_j\} = b_j$, which in turn is equivalent to $\min\{b_j - a_j, D_j\} = b_j - a_j$, or $D_j \geq b_j - a_j$. By Markov's inequality, this is at most $\frac{\mathbb{E}[D_j]}{b_j - a_j} = \frac{\bar{d}_j}{b_j - a_j}$, completing the proof of Lemma 3.4.7. □

Lemma 3.4.8 (Algorithm's Expected Revenue). *The expected revenue of our algorithm is at least*

$$\sum_{i=1}^{n} r_i \cdot \mathbb{E}[\min\{b_i - D_i, a_i\}] + \frac{1}{2}\sum_{j=1}^{n} r_j^{\mathsf{disc}} \bar{d}_j + \frac{1}{2}\sum_{i=1}^{n}\sum_{j \neq i} r_j^{\mathsf{disc}}(\mathbb{E}[\min\{b_i - D_i, a_i\}] - \frac{a_i}{2} \cdot \frac{a_j + 2d_j}{b_j}) p_{ij}(S^*_i).$$

Proof. Proof. Recall that for every customer $t \in [T]$, our algorithm serves her as long as item i_t is available. Therefore, the revenue earned on a run of our algorithm is

$$\sum_{t=1}^{T} \Big(r_{i_t} \cdot \mathbb{1}(I_{i_t}^{t-1} > 0) + \sum_{j \neq i_t} r_j^{\mathsf{disc}} P_j^t\Big).$$

Its expected value is equal to

$$\mathbb{E}\Big[\sum_{t=1}^{T} r_{i_t} \cdot \mathbb{1}(I_{i_t}^{t-1} > 0)\Big] + \mathbb{E}\Big[\sum_{t=1}^{T}\sum_{j \neq i_t} r_j^{\mathsf{disc}} P_j^t\Big].$$

The first term is the expected revenue earned from selling items i to customers requesting it (at the full price of r_i), while the second term is the expected revenue earned from selling items $j \neq i_t$ as add-ons (at the discounted price of r_j^{disc}) summed over all time periods t.

Applying Proposition 3.4.6, the first term is equal to

$$\sum_{i=1}^{n} r_i \cdot \mathbb{E}\Big[\sum_{t:i_t=i} \mathbb{1}(I_i^{t-1} > 0)\Big] = \sum_{i=1}^{n} r_i \cdot \mathbb{E}[\min\{b_i - D_i, a_i\}]. \tag{3.9}$$

Applying the definitions that $D_j = \sum_{t:i_t \neq j} P_j^t$ and $\mathbb{E}[D_j] = \bar{d}_j$, the second term is equal to

$$\begin{aligned}\mathbb{E}\Big[\sum_{t=1}^{T}\sum_{j\neq i_t} r_j^{\mathsf{disc}} P_j^t\Big] &= \sum_{j=1}^{n} r_j^{\mathsf{disc}} \cdot \mathbb{E}\Big[\sum_{t:i_t\neq j} P_j^t\Big] \\ &= \sum_{j=1}^{n} r_j^{\mathsf{disc}} \bar{d}_j. \end{aligned} \tag{3.10}$$

We also analyze the second term in a different way. It can be re-arranged as follows:

$$\mathbb{E}\Big[\sum_{t=1}^{T}\sum_{j\neq i_t} r_j^{\mathsf{disc}} P_j^t\Big] = \sum_{i=1}^{n}\sum_{t:i_t=i} \mathbb{E}\Big[\sum_{j\neq i} r_j^{\mathsf{disc}} P_j^t\Big]. \tag{3.11}$$

Fix an arbitrary i and apply Lemma 3.4.7 for all t such that $i_t = i$. We obtain

$$\begin{aligned}\sum_{t:i_t=i} \mathbb{E}\Big[\sum_{j\neq i} r_j^{\mathsf{disc}} P_j^t\Big] &\geq \sum_{t:i_t=i}\sum_{j\in S_i^*, j\neq i} r_j^{\mathsf{disc}} p_{ij}(S_i^*)\big(\Pr[I_i^{t-1} > 0] - \frac{\bar{d}_j}{b_j - a_j}\big) \\ &= \sum_{j\in S_i^*, j\neq i} r_j^{\mathsf{disc}} p_{ij}(S_i^*)\Big(\sum_{t:i_t=i}\Pr[I_i^{t-1} > 0] - a_i \cdot \frac{\bar{d}_j}{b_j - a_j}\Big) \\ &= \sum_{j\in S_i^*, j\neq i} r_j^{\mathsf{disc}} p_{ij}(S_i^*)\Big(\mathbb{E}[\min\{b_i - D_i, a_i\}] - \frac{a_i}{2} \cdot \frac{2\bar{d}_j}{b_j - a_j}\Big),\end{aligned}$$

where the final equality follows after the same derivation as (3.9). Now, by definition, $\bar{d}_j \leq d_j$. Therefore, we can replace $\bar{d}_j$ with d_j and the expression would be no greater. Furthermore, since $2d_j < b_j - a_j$ for all $j \in S_i^* \setminus \{i\}$, $\frac{2d_j}{b_j - a_j} \leq \frac{2d_j + a_j}{b_j}$. Substituting back into (3.11), we conclude that

$$\mathbb{E}\Big[\sum_{t=1}^{T}\sum_{j\neq i_t} r_j^{\mathsf{disc}} P_j^t\Big] \geq \sum_{i=1}^{n}\sum_{j\in S_i^*, j\neq i} r_j^{\mathsf{disc}}(\mathbb{E}[\min\{b_i - D_i, a_i\}] - \frac{a_i}{2} \cdot \frac{2d_j + a_j}{b_j}) p_{ij}(S_i^*). \tag{3.12}$$

Taking (3.9)+$\frac{1}{2}$·(3.10)+$\frac{1}{2}$·(3.12) completes the proof of Lemma 3.4.8. $\square$

3.4.3 Proof of Theorem 3.2.9

In this subsection we aim to prove our main result Theorem 3.2.9, by using Theorem 3.3.10 to bound the expression from Lemma 3.4.4 relative to the expression from Lemma 3.4.8. Both expressions can be seen as a sum of three terms, and we compare the respective terms separately. Namely, we compare

$$\sum_{i=1}^{n} r_i \cdot \mathbb{E}[\min\{b_i - D_i, a_i\}] \quad \text{to} \quad \sum_{i=1}^{n} 2r_i a_i,$$

$$\frac{1}{2}\sum_{j=1}^{n} r_j^{\mathsf{disc}} \bar{d}_j \quad \text{to} \quad \sum_{j=1}^{n} 2r_j^{\mathsf{disc}} d_j,$$

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j\neq i} r_j^{\mathsf{disc}}(\mathbb{E}[\min\{b_i - D_i, a_i\}] - \frac{a_i}{2}\cdot\frac{a_j + 2d_j}{b_j})p_{ij}(S_i^*) \quad \text{to} \quad \sum_{i=1}^{n} a_i \sum_{j\neq i} r_j^{\mathsf{disc}}(1 - \frac{a_j + 2d_j}{b_j})p_{ij}(S_i^*).$$

Fix a small ε. We will condition on our sampling algorithm not failing, i.e. $\hat{d}_j - \varepsilon \leq \bar{d}_j \leq \frac{b_j}{2}$ for all $j \in [n]$, which occurs with probability at least $1 - \varepsilon$. We will scale our total revenue by $(1 - \varepsilon)$ at the end, and treat a failed run as a run with 0 revenue.

To bound the value $\min\{b_j - D_j, a_j\}$, we use the following modification of Markov's inequality:

Proposition 3.4.9. *For some $b \geq 0$, let X be a random variable distributed over $[0, b]$ with $\mathbb{E}[X] \geq \frac{b}{2}$. Then for any a such that $0 \leq a \leq b$, $\mathbb{E}[\min\{X, a\}] \geq \frac{a}{2}$.*

Proof. Proof. Consider the random variable $X - \min\{X, a\}$. When $X \geq a$, its value is at most $b - a$, since $X \leq b$. When $X < a$, its value is 0. Therefore, $\mathbb{E}[X - \min\{X, a\}] \leq (b - a)\Pr[X \geq a]$, which in conjunction with $\mathbb{E}[X] \geq \frac{b}{2}$ implies

$$\mathbb{E}[\min\{X, a\}] \geq \frac{b}{2} - (b - a)\Pr[X \geq a]. \tag{3.13}$$

If $\Pr[X \geq a] \leq \frac{1}{2}$, then (3.13) implies $\mathbb{E}[\min\{X, a\}] \geq \frac{b}{2} - \frac{b-a}{2} = \frac{a}{2}$, as desired.

Otherwise, if $\Pr[X \geq a] > \frac{1}{2}$, we can immediately conclude from the non-negativity of X and a that $\mathbb{E}[\min\{X, a\}] \geq a \cdot \Pr[X \geq a]$, which combined with $\Pr[X \geq a] > \frac{1}{2}$ yields the desired result. □

Now, to compare the first terms, consider any $i \in [n]$. $\mathbb{E}[D_i] = \bar{d}_i \leq \frac{b_i}{2}$, which implies $\mathbb{E}[b_i - D_i] \geq \frac{b_i}{2}$. Therefore, we can apply Proposition 3.4.9 with $X = b_i - D_i$ to obtain

$\mathbb{E}[\min\{b_i - D_i, a_i\}] \geq \frac{a_i}{2}$ for all $i \in [n]$. This establishes that

$$\sum_{i=1}^{n} r_i \cdot \mathbb{E}[\min\{b_i - D_i, a_i\}] \geq \frac{1}{4} \sum_{i=1}^{n} 2r_i a_i. \tag{3.14}$$

To compare the second terms, consider any $j \in [n]$. If $d_j \neq \bar{d}_j$, then $d_j = \hat{d}_j + \varepsilon$ and $\hat{d}_j \geq \frac{1}{2} - \varepsilon$. Also, $\bar{d}_j \geq \hat{d}_j - \varepsilon$. Therefore, for these j, $\frac{\frac{\bar{d}_j}{2}}{2d_j} \geq \frac{\hat{d}_j - \varepsilon}{4\hat{d}_j + 4\varepsilon} = \frac{1}{4} - \frac{2\varepsilon}{4\hat{d}_j + 4\varepsilon} \geq \frac{1}{4} - \frac{2\varepsilon}{4(1/2 - \varepsilon) + 4\varepsilon} = \frac{1}{4} - \varepsilon$. This establishes that

$$\frac{1}{2} \sum_{j=1}^{n} r_j^{\mathsf{disc}} \bar{d}_j \geq (\frac{1}{4} - \varepsilon) \sum_{j=1}^{n} 2r_j^{\mathsf{disc}} d_j. \tag{3.15}$$

Finally, to compare the third terms, we can again apply Proposition 3.4.9 for all $i \in [n]$ to obtain

$$\frac{1}{2} \sum_{i=1}^{n} \sum_{j \neq i} r_j^{\mathsf{disc}} (\mathbb{E}[\min\{b_i - D_i, a_i\}] - \frac{a_i}{2} \cdot \frac{a_j + 2d_j}{b_j}) p_{ij}(S_i^*) \geq \frac{1}{4} \sum_{i=1}^{n} a_i \sum_{j \neq i} r_j^{\mathsf{disc}} (1 - \frac{a_j + 2d_j}{b_j}) p_{ij}(S_i^*). \tag{3.16}$$

Adding (3.14), (3.15), and (3.16), we conclude that with probability at least $1 - \varepsilon$, our algorithm earns revenue at least $\frac{1}{4} - \varepsilon$ of the dual objective. Therefore, using $\mathsf{OnlineSamplingPersonalizedAssortment}(\varepsilon)$, for any instance and sequence of arrivals, $\mathsf{ALG}(i_1, \ldots, i_T) \geq (1 - \varepsilon)(\frac{1}{4} - \varepsilon) \cdot \mathsf{OPT}(i_1, \ldots, i_T)$, completing the proof of Theorem 3.2.9.

Chapter 4

On Policies for Single-leg Revenue Management with Limited Demand Information

In this chapter we study the single-leg revenue management problem, with no information given about the demand trajectory over time. The competitive ratio for this problem has been established by Ball and Queyranne (2009) under the assumption of independent demand. We extend their results to general non-independent demand, by incorporating the price-skimming technique from Eren and Maglaras (2010). That is, we derive price-skimming policies which stochastically-increase their price distributions as inventory decreases, in a way that yields the best-possible competitive ratio. Furthermore, our policies have the benefit that they can be easily adapted to exploit available demand information, such as the personal characteristics of an incoming online customer, while maintaining the competitive ratio guarantee. A key technical ingredient in our chapter is a new "valuation tracking" subroutine, which tracks the possible values for the optimum, and follows the most inventory-conservative control which maintains the desired competitive ratio.

4.1 Introduction

In this chapter we consider the single-leg revenue management problem, where a firm is selling multiple products that share a single capacity over a finite time horizon. The price of each product and the unreplenishable starting capacity are exogenously determined. The firm's objective is to maximize its total revenue earned, by dynamically controlling the availability of different products over time. There is a tradeoff between controls which maximize immediate revenue, and controls which reserve sufficient capacity for the remaining time horizon.

The motivating application for this problem lies in airlines, where each flight leg has a limited seat capacity, and the products correspond to different "fare classes" (e.g. economy, business) which offer seats at different prices. The seat capacity and fare classes have been determined long in advance, through factors such as business strategy, positioning, competition, etc. The time horizon is finite, ending upon the flight's departure.

We study this problem in the setting where very limited information is given or can be learned about demand. This setting was introduced by Ball and Queyranne (2009); Lan et al. (2008), where meaningful controls can be derived based on only the knowledge of the fare class prices. Ball and Queyranne (2009) consider *booking limit* policies, which can be described as follows. Initially, all of the fare classes are made available to customers. Once a critical threshold on the total seat sales is reached, the lowest fare becomes unavailable; progressively higher fares are made unavailable until the flight either becomes full or takes off.

An important assumption made in the model of Ball and Queyranne (2009) is that demand for the different fare classes is *independent*. That is, although the lower fares are made available until their booking thresholds are reached, there is no risk of *cannibalizing* the sales of higher fares. The justification for this assumption in the literature is twofold. First, the fare classes have been designed to segment customers and achieve price discrimination, i.e. the perks provided by business class dissuade price-insensitive business travelers from having interest in lower-class fares. Second, these business travelers tend to book last-minute, i.e. by the time they book the lower-class fares have usually become unavailable anyway.

However, the advent of e-commerce has brought both an increase in customer sophistication, and an increased opportunity for more sophisticated booking controls. The goal of this chapter is to derive such controls, which perform well without the assumption of independent demand.

4.1.1 Model and Results

Throughout this chapter, we analyze the extreme case where customers always substitute to the lowest available fare, delaying the extension to general *rational* choice models to the Conclusion. We can thus describe each customer using a *valuation*, or maximum willingness-to-pay. The control reduces to *dynamic pricing*, where at each point in time a single price is offered, and customers who encounter that price make a purchase if and only if it does not exceed their valuation. (Offering multiple prices is redundant, because the lower price would always be chosen over the higher price.)

We let $\mathcal{P}$ denote the set of fare class prices, which are the feasible prices to charge. The starting capacity comes in the form of k discrete units of *inventory*. The selling horizon consists of T discrete time steps, which are sufficiently granular such that at most one customer arrives during each time step. We let V_t denote the maximum price in $\mathcal{P}$ that the customer in time t is willing to pay, which is 0 if no customer arrived. An *online algorithm* must sequentially choose a price P_t for each time t, and if $V_t \geq P_t$, then revenue P_t is earned and one unit of inventory is depleted.

In settings where no information is given about the sequence of valuations, an online algorithm is evaluated by comparing its total revenue earned on different sequences to that of a clairvoyant optimum. For any sequence $V_1, \ldots, V_T$, the *offline optimum* $\mathsf{OPT}(V_1, \ldots, V_T)$ is defined as the maximum revenue that could have been earned from knowing all the valuations in advance, equal to the k largest values in $V_1, \ldots, V_T$. For $c \leq 1$, if an online algorithm can guarantee that its revenue is at least $c \cdot \mathsf{OPT}(V_1, \ldots, V_T)$ on every sequence $V_1, \ldots, V_T$, then it is said to be *c-competitive.* If c is best-possible in that any (potentially randomized) online algorithm cannot simultaneously guarantee greater than $c \cdot \mathsf{OPT}(V_1, \ldots, V_T)$ revenue on all sequences $V_1, \ldots, V_T$, then c is called the *competitive ratio.*

In the model of Ball and Queyranne (2009), the competitive ratio for any problem instance is shown to be a function of only the price set $\mathcal{P}$, which we will denote using $\mathsf{CR}(\mathcal{P})$. Their model corresponds to ours when each V_t is *deterministic* and given at the *start* of time t. The decision at time t then reduces to an accept-reject decision, where accepting customer t corresponds to charging her maximum willingness-to-pay of V_t, and rejecting customer t corresponds to charging any price above V_t (possibly ∞ if the inventory has run out). Ball and Queyranne (2009) derive $\mathsf{CR}(\mathcal{P})$-competitive booking limit policies which specify when to reject customers paying low prices.

In this chapter, we derive $\mathsf{CR}(\mathcal{P})$-competitive online algorithms under the following models with progressively less information:

1. V_t is *deterministic* and given at the *end* of time t (Section 4.2);

2. V_t is *stochastic* and its distribution is given at the *end* of time t (in this case, we define the offline benchmark as the *Hindsight Optimum* $\mathbb{E}[\mathsf{OPT}(V_1, \ldots, V_T)]$, instead of based on the *Deterministic Linear Program*—see Section 4.3);

3. No information on V_t is ever given (Section 4.4).

All of these algorithms are best-possible, in that if an online algorithm cannot be better than $\mathsf{CR}(\mathcal{P})$-competitive under the model of Ball and Queyranne (2009), then an online algorithm also cannot be better than $\mathsf{CR}(\mathcal{P})$-competitive under our models with less information.

Our algorithms use the *price-skimming* technique of Eren and Maglaras (2010), who analyze how the price of an item should be distributed (e.g. across stores, across time) when the price set $\mathcal{P}$ is known but the demand is completely unknown. Their model corresponds to ours when the inventory constraint is irrelevant (e.g. when $k \geq T$), in which case they show that the competitive ratio is also $\mathsf{CR}(\mathcal{P})$. Our work shows how the price-skimming distribution should depend on inventory when it is relevant: at any time step, the price distribution which maximizes the competitive ratio is strictly stochastically-decreasing in the amount of remaining inventory (see Section 4.2.4). In fact, this is analogous to a classical structural property when the demand sequence is known or distributionally-known: at any time step, the price which maximizes the expected revenue is strictly decreasing in the amount of remaining inventory (Gallego and Van Ryzin, 1994; Zhao and Zheng, 2000).

Finally, in Section 4.5, we discuss the model where each valuation V_t is *stochastic* and distributionally-given at the *start* of time t. This is the personalized online revenue management setup introduced by Golrezaei et al. (2014), with a very practical motivation of e-commerce, where personalized recommendations (e.g. business-class flight) can be made to each customer based on her characteristics (e.g. Mac user). Our algorithms in Sections 4.2–4.3 are designed for e-tailers who choose not to engage in personalized pricing, and we instead use the estimated distributions of V_t to refine the pricing strategy for future customers. Nevertheless, our algorithms can be naturally adapted into the personalized pricing setting, and in Section 4.5 we show how to exploit personalization without losing the worst-case performance ratio of $\mathsf{CR}(\mathcal{P})$.

4.1.2 Sketch of Techniques, and Comparison with Existing Techniques

The main technical contribution behind our results is a new "valuation tracking" procedure which incorporates both booking limits and price-skimming. We motivate it using the following example, under the model where each valuation V_t is deterministic and revealed at the end of time t. The price set is $\mathcal{P} = \{1, 2, 4\}$, and we will refer to customers with these valuations as being of type-L (Low), type-M (Medium), and type-H (High), respectively.

The competitive ratio for this price set derived by Ball and Queyranne (2009) and Eren and Maglaras (2010) is $\mathsf{CR}(\mathcal{P}) = 1/2$, and we describe below their respective 1/2-competitive policies.

- Booking Limits (Ball and Queyranne, 2009): Initially charge \$1; increase the price to \$2 after 1/2 of the starting inventory has been sold; further increase the price to \$4 after 3/4 of the starting inventory has been sold. (This is the variant of booking limits with "theft nesting".)

- Price-skimming (Eren and Maglaras, 2010): Charge \$1 for 1/2 of the time steps; charge \$2 for 1/4 of the time steps; charge \$4 for 1/4 of the time steps.

We now discuss what happens if we try to implement these policies under our model.

Attempt 1: Direct implementation of booking limits. It is easy to see that this would not be 1/2-competitive—suppose just one type-H customer arrived at the start. The algorithm would be charging the low price of \$1, while the offline optimum would be the customer's valuation of \$4.

Any direct implementation of price-skimming would suffer similarly, since there could be a single type-H customer who arrives during a time when the price is set to \$1.

Attempt 2: Price-skimming as a randomized price. It appears that the problem with Attempt 1 can be solved using the "random price" interpretation of price-skimming—instead of deterministically partitioning the time horizon according to ratios $\frac{1}{2}, \frac{1}{4}, \frac{1}{4}$ and offering prices 1,2,4 respectively, one could at each time step choose the prices randomly with respective probabilities $\frac{1}{2}, \frac{1}{4}, \frac{1}{4}$. Then, if a single type-H customer arrives, the expected revenue would be

$$\frac{1}{2}\cdot 1+\frac{1}{4}\cdot 2+\frac{1}{4}\cdot 4=2$$

which is $1/2$ of the customer's valuation of \$4. It can be checked that $1/2$ of the customer's valuation is also earned when it is \$1 or \$2; this is by construction of the price-skimming distribution.

However, having a fixed price-skimming distribution is no longer effective under inventory constraints. Indeed, if a long sequence of type-L customers arrive, then this would deplete the inventory with high probability, and type-H customers who arrive last-minute would not be served, and the ratio of optimum earned would again be $1/4$.

Attempt 3: Naive incorporation of booking limits into price-skimming. It appears that the problem with Attempt 2 can be solved by respecting the booking limits, i.e. forbidding price-skimming from randomly choosing the price of \$1 after $1/2$ of the starting inventory has been sold. However, this still fails to be $1/2$-competitive, as shown by the following example. Suppose the starting inventory is 4, and that 2 type-H customers arrive followed by a type-L customer, with no customers arriving after that. The optimum would be \$9. However, the algorithm's revenue would only equal \$4: it would earn \$2 in expectation from each of the type-H customers, depleting 2 units of inventory, and then earn \$0 from the type-L customer due to the booking limit.

Our procedure: Valuation tracking. The problem with Attempt 3 leads to the following observation—the optimum is guaranteed to increase from the first 4 customers (since there are 4 units of inventory), so in order to be $1/2$-competitive, the algorithm must maintain the initial price-skimming distribution for the first 4 customers. After that, the algorithm can respect booking limits as long as customers rejected in this way would not increase the optimum, and in fact should do so, to avoid the problem in Attempt 2 of stocking out. This motivates our procedure below.

- <u>Valuation Tracking</u>: At each time t, let ℓ_t denote the smallest value (possibly 0) in the 4 largest valuations to have arrived before time t (since 4 is the starting inventory). Then, randomly choose the price in a way such that the algorithm's expected revenue during time t is *equal to*

$$\frac{1}{2}(V_t - \ell_t), \qquad \text{for any } V_t > \ell_t. \tag{4.1}$$

$V_t - \ell_t$ is the gain in the offline optimum should the valuation of customer t be V_t, and $1/2$ is the desired competitive ratio. The constraint that the algorithm's revenue is exactly equal to (4.1) effectively forces it to use the most *inventory-conservative* controls which maintains $1/2$-competitiveness, thereby hedging against a stockout. The price distribution used at each time t depends on the inventory state, and in fact, the calculation for the algorithm's expected revenue must account for the probability of stocking out before time t. The surprising fact is that it is *possible* to choose price distributions which collectively guarantee (4.1).

We illustrate our basic valuation tracking procedure in Section 4.2.1 by "stacking" the past valuations in a geometric configuration, and formalize it in Section 4.2.2. The basic procedure is designed to have a clean analysis and we consider variants in Sections 4.2.3–4.2.4, as well as generalize it to the stochastic-valuation model in Section 4.3.2. A unique challenge arises in Section 4.3.3, where we use sampling to convert a $\mathsf{CR}(\mathcal{P})$-competitive valuation tracking procedure with exponential runtime into a $(\mathsf{CR}(\mathcal{P}) - \varepsilon)$-competitive procedure with polynomial runtime.

4.1.3 Related Work

Single-leg revenue management is a cornerstone problem in the area of revenue management and pricing, as outlined in the book by Talluri and Van Ryzin (2006). Many different approaches for modeling demand have been considered over the years, as surveyed in Araman and Caldentey (2011). Our work falls under the stream of literature which analyzes the competitive ratio, and extends the booking limits of Ball and Queyranne (2009) to general non-independent demand by incorporating the price-skimming of Eren and Maglaras (2010). We should point out that our "random price" interpretation of price-skimming originated from Bergemann and Schlag (2008). Randomization is a powerful technique for improving the competitive ratio of online algorithms; for further background we refer to the book by Borodin and El-Yaniv (2005).

Motivated by e-commerce, the competitive ratio has been studied in many "personalized recommendation" settings (see Golrezaei et al. (2014), as well as Chapters 2 and 3). In this other work, there are multiple commodities ("legs") and interchangeability in which commodities are sold to each customer. Our work differs from this other work in two ways. First, we do not rely on personalized controls, and instead derive a global price-skimming distribution, which is useful for online retailers who choose not to engage in personalized pricing. Also, even when personalization is desired, we derive in Section 4.5 how to exploit personalized information in the single-leg case, and our competitive ratio guarantee $\mathsf{CR}(\mathcal{P})$ is greater than the one from Chapter 2 for multiple legs.

Finally, while we analyze the problem of inventory-constrained dynamic pricing throughout this chapter, our results easily generalize to the dynamic assortment setting, as explained in the Conclusion of this chapter. We view our techniques as a general way to trade off between revenue earning and inventory consumption (see Maglaras and Meissner (2006)) when there is a single leg, known fare classes, but general unknown demand.

4.2 Deterministic Valuations

In this section we consider the problem defined in Section 4.1.1 under the model where each customer's valuation is deterministic and revealed immediately after she leaves.

First we define some additional notation to that defined in Section 4.1.1. For any positive integer n, let $[n]$ denote the set $\{1, \ldots, n\}$. For notational convenience, we will assume that $\mathcal{P}$ consists of m discrete prices, i.e. $\mathcal{P} = \{r^{(j)} : j \in [m]\}$, sorted $0 < r^{(1)} < \ldots < r^{(m)}$. All of our results can be generalized to the case where $\mathcal{P}$ is a continuum of prices taking the form $[r^{\min}, r^{\max}]$, as we discuss in Section C.4. We define $r^{(0)}$ to be 0, and then the valuation V_t at any time t lies in $r^{(0)}, \ldots, r^{(m)}$, with $V_t = r^{(0)}$ representing the lack of a customer during time t. Similarly, we define $r^{(m+1)}$ to be ∞, and then the price P_t at any time t lies in $r^{(1)}, \ldots, r^{(m+1)}$, with $P_t = r^{(\infty)}$ representing the firm shutting off demand during time t, which is the only option if its inventory is out of stock. Let X_t be the indicator variable for making a sale during time t, i.e. it is 1 if $V_t \geq P_t$, and 0 otherwise.

Let T denote the number of time steps. None of the algorithms in this chapter assume any knowledge of T; note that T can always be made arbitrarily large by inserting customers with valuation 0. We will hereafter treat T as the unknown total number of customers, and use the phrase "customer t" to refer to valuation V_t (even if it is 0).

An *online algorithm* must choose each P_t based on the history of past prices and valuations, $P_1, V_1, \ldots, P_{t-1}, V_{t-1}$. This history also determines the values of $X_1, \ldots, X_{t-1}$. The online algorithm does not know T, and has no information about $V_t, V_{t+1}, \ldots, V_T$, when choosing P_t. In contrast, the *offline optimum* knows the entire sequence $V_1, \ldots, V_T$ before having to choose any prices. Given any valuation sequence $V_1, \ldots, V_T$, we use the P_t and X_t variables to refer to the execution of an online algorithm on the valuation sequence. Since the online algorithm may be randomized, we treat P_t and X_t as random variables. Let $\mathsf{ALG}(V_1, \ldots, V_T)$ denote the total revenue earned by the online algorithm, equal to $\sum_{t=1}^{T} P_t X_t$. Then $\mathbb{E}[\mathsf{ALG}(V_1, \ldots, V_T)]$ is its expected revenue. Meanwhile, let $\mathsf{OPT}(V_1, \ldots, V_T)$ denote the offline optimum for sequence $V_1, \ldots, V_T$, equal to the $\min\{k, T\}$ largest valuations from $V_1, \ldots, V_T$. Formally, an online algorithm is said to be c-competitive if

$$\mathbb{E}[\mathsf{ALG}(V_1, \ldots, V_T)] \geq c \cdot \mathsf{OPT}(V_1, \ldots, V_T), \quad \forall\, T \geq 1, (V_1, \ldots, V_T) \in (\mathcal{P} \cup \{0\})^T. \tag{4.2}$$

We will omit the arguments $(V_1, \ldots, V_T)$ in ALG and OPT when the context is clear.

As explained in Section 4.1.1, since our problem captures the problems of both Ball and Queyranne (2009) and Eren and Maglaras (2010), an upper bound for the value of c in (4.2) is given by $\mathsf{CR}(\mathcal{P})$, as defined below.

Definition 4.2.1. For any $m \geq 1$, $0 < r^{(1)} < \ldots < r^{(m)}$, and $\mathcal{P} = \{r^{(1)}, \ldots, r^{(m)}\}$, define:

- $q^{(j)} = 1 - \frac{r^{(j-1)}}{r^{(j)}}$ for all $j \in [m]$ (recall that $r^{(0)} = 0$);
- $q = \sum_{j=1}^{m} q^{(j)}$;
- $\mathsf{CR}(\mathcal{P}) = \frac{1}{q}$.

The interpretation of $q^{(j)}$ in Ball and Queyranne (2009) is the fraction of initial inventory "set aside" for prices j and higher. The interpretation of $q^{(j)}$ in Eren and Maglaras (2010) is the fraction of time that price j should be charged. Both of the papers establish that the competitive ratio cannot be better than $\mathsf{CR}(\mathcal{P})$ via Yao's minimax principle (Yao, 1977). Therefore, for any fixed (but possibly randomized) online algorithm in our problem, there exists a sequence $V_1, \ldots, V_T$ such $\mathbb{E}[\mathsf{ALG}(V_1, \ldots, V_T)] \leq \mathsf{CR}(\mathcal{P}) \cdot \mathsf{OPT}(V_1, \ldots, V_T)$. In this chapter we derive various $\mathsf{CR}(\mathcal{P})$-competitive algorithms, using our valuation tracking procedure as the core subroutine.

4.2.1 Intuition behind Valuation Tracking Procedure

The goal of our basic procedure is to, for each customer, earn a constant fraction $\mathsf{CR}(\mathcal{P})$ of the gain in OPT from that customer arriving, which would imply being $\mathsf{CR}(\mathcal{P})$-competitive. To accomplish this, it tracks the current value of OPT, i.e. the sum of the k largest valuations to have arrived thus far, which are assumed to be known.

Consider an example where the feasible price set is $\mathcal{P} = \{1, 2, 4\}$, in which case $\mathsf{CR}(\mathcal{P}) = \frac{1}{2}$. Suppose the starting inventory is $k = 5$, and that 5 customers, with valuations $4, 1, 4, 1, 2$, have already arrived. The current value of OPT is then the sum of these 5 valuations, $4 + 1 + 4 + 1 + 2 = 12$.

The procedure considers the possibilities for the increase in OPT from the next customer, which we denote as $\Delta\mathsf{OPT}$. Since the smallest valuation currently counted toward OPT is 1, if the valuation of the next customer if 4, then $\Delta\mathsf{OPT} = 3$; if it is 2, then $\Delta\mathsf{OPT} = 1$. If the next customer has valuation not exceeding 1, then $\Delta\mathsf{OPT} = 0$. The procedure wants to guarantee that its *expected* revenue on the next customer is at least $\frac{1}{2} \cdot \Delta\mathsf{OPT}$, for all of these possible valuations. To accomplish this, it has to consider the probability that it has stocked out at this point; on those sample paths its revenue is 0.

Our procedure cleanly accounts for the probability of stocking out using the following approach. Each customer is assigned to a *specific* unit of inventory $i \in [k]$ upon arrival. Each inventory unit i maintains a variable $\texttt{level}[i]$, which is the maximum valuation of a customer previously assigned to it. The next customer is always assigned to an unit i^* with the smallest value of $\texttt{level}[i^*]$, *regardless of whether that unit i^* has already been sold.* In this way, the assignment procedure is deterministic, and allows us to maintain an invariant: the probability a unit i has been sold is dependent on only the (deterministic) value of $\texttt{level}[i]$.

For each customer, the procedure makes an offer to her *only if unit i^* has not been sold,* at a random price exceeding $\texttt{level}[i^*]$. The higher $\texttt{level}[i^*]$ is, the more likely it is that unit i^* has been sold, and the lower the expected revenue from that customer. However, if $\texttt{level}[i^*]$ is high, then the potential increase in OPT from that customer is also lower; if the valuation of the customer does not exceed $\texttt{level}[i^*]$, then both the procedure's revenue and $\Delta\mathsf{OPT}$ are 0. By properly choosing the distributions for the random prices, our procedure is able to maintain the invariant on the probability of each unit being sold, while earning

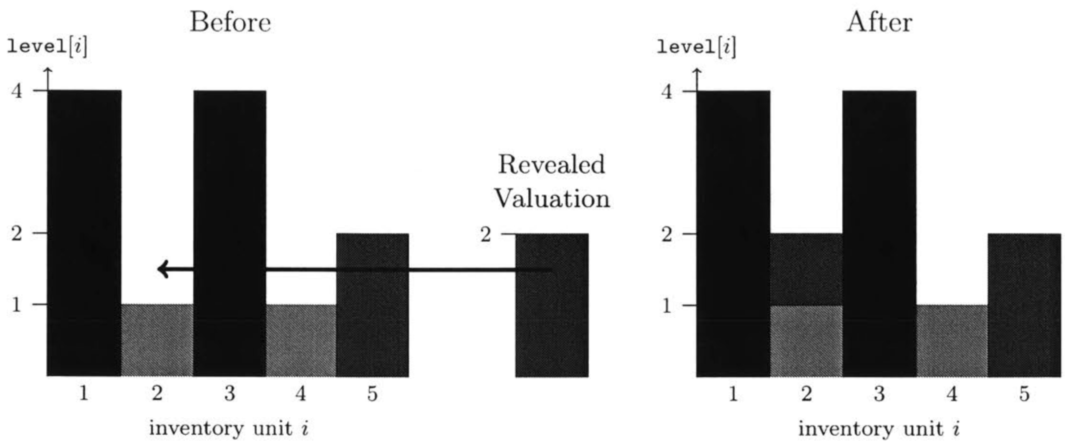


Figure 4-1: The configuration of valuations, before and after a customer with valuation 2 arrives.

$\frac{1}{2} \cdot \Delta\mathsf{OPT}$ in expectation from each customer.

Returning to the example, given that the first 5 customers had valuations $4, 1, 4, 1, 2$, the values of $\texttt{level}[i]$ for $i = 1, \ldots, k$ are shown in the LHS of Fig. 4-1. The next customer, "customer #6", is assigned to inventory unit 2. After her valuation is revealed to be 2, the updated configuration is shown on the RHS of Fig. 4-1, regardless of whether she was rejected.

Customer #6 would have been rejected if unit 2 was sold before her arrival, even if other units were available. When $\mathcal{P} = \{1, 2, 4\}$, the probability that a unit i has been sold equals $0, \frac{1}{2}, \frac{3}{4}, 1$ if $\texttt{level}[i]$ is $0, 1, 2, 4$, respectively. These probabilities correspond to the values of $q^{(j)}$ from Definition 4.2.1. Since $\texttt{level}[2]$ was 1 before customer #6 arrived, she is made an offer with probability $\frac{1}{2}$, at a random price exceeding 1. The price is 2 with probability proportional to $\frac{1}{4}$, and 4 with probability proportional to $\frac{1}{4}$ (again using the values of $q^{(j)}$), hence each price would be offered with probability $\frac{1}{2}$. The customer's valuation is 2, so she will only buy the item if offered price 2, which occurs with total probability $\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$. Note that:

1. Customer #6 increases the probability of unit 2 being sold from $\frac{1}{2}$ to $\frac{3}{4}$, which is consistent with her increasing $\texttt{level}[2]$ from 1 to 2;

2. Customer #6 increases the value of OPT, equal to $\sum_{i=1}^{k} \texttt{level}[i]$, by 1 (from 12 to 13);

Algorithm 7 Weakly Randomized Online Algorithm

1: `level`$[i] = 0$, `sold`$[i] =$ `false` for $i = 1, \ldots, k$
2: $t = 1$
3: **while** customer t arrives **do**
4: $\quad \ell = \min_{i'}\{$`level`$[i']\}$
5: $\quad i = \min\{i' :$ `level`$[i'] = \ell\}$
6: $\quad$ **if** `sold`$[i] =$ `false` **then**
7: $\quad\quad$ offer price $r^{(j)}$ with probability $\frac{q^{(j)}}{\sum_{j'=\ell+1}^{m} q^{(j')}}$, for all $j = \ell + 1, \ldots, m$
8: $\quad$ **else**
9: $\quad\quad$ reject the customer by choosing price ∞
10: $\quad$ **end if**
11: $\quad$ observe valuation V_t and purchase decision X_t
12: $\quad$ **if** $V_t = r^{(j)}$ for some $j = \ell + 1, \ldots, m$ **then**
13: $\quad\quad$ `level`$[i] = j$
14: $\quad\quad$ **if** $X_t = 1$ **then** `sold`$[i] =$ `true`
15: $\quad\quad$ **end if**
16: $\quad$ **end if**$t = t + 1$
17: **end while**

Input: Customers $t = 1, 2, \ldots$ arriving online, with each valuation V_t revealed after the price P_t is chosen.
Output: For each customer t, a (possibly random) price P_t for her.

3. Customer #6 brings in expected revenue $\frac{1}{4} \cdot 2 = \frac{1}{2}$.

Therefore, during time step 6, our procedure has earned expected revenue $\frac{1}{2} \cdot \Delta\mathsf{OPT}$. We will show that it achieves this for a general $\mathcal{P}$, and all time steps t, regardless of the valuation of customer t.

4.2.2 Valuation Tracking Procedure and Analysis

We now formalize our valuation tracking procedure, in Algorithm 7.

In line 7, the procedure offers exactly one of the prices $r^{(\ell_t+1)}, \ldots, r^{(m)}$, with the offering probabilities summing to unity. Note that it cannot branch to line 7 if $\ell_t = m$. This can be seen in the following way. If $\ell_t = m$, then i_t^* must have been assigned to some past customer t' with $V_{t'} = r^{(m)}$. At time t', either inventory unit i_t^* was already sold, or customer t' was offered a price at most $r^{(m)}$, which she would have accepted. In either case, `sold`$[i_t^*]$ must be `true`.

The analysis of Algorithm 7 is conceptually simple. Let $b_{i,t}$ be the index in $0, \ldots, m$ such that `level`$[i] = r^{(b_{i,t})}$ at the end of time t, and j_t be the index such that $V_t = r^{(j_t)}$. We show that the following claims are maintained:

1. At the end of each time step t, the probability that any inventory unit i has been sold is $(\sum_{j=1}^{b_{i,t}} q^{(j)})/q$;

2. During a time step t, if the valuation of the customer exceeds the level of the inventory unit she is assigned to, i.e. $j_t > \ell_t$, then:

   (a) The expected revenue earned by Algorithm 7 is $\frac{1}{q}(r^{(j_t)} - r^{(\ell_t)})$;

   (b) The increase in the offline optimum is $r^{(j_t)} - r^{(\ell_t)}$.

If $j_t \leq \ell_t$ during a time step t, then both the revenue of Algorithm 7 and the gain in OPT are 0.

These claims establish the following theorem, whose full proof is deferred to Section C.1.

Theorem 4.2.2. *Algorithm 7 is* $\mathsf{CR}(\mathcal{P})$*-competitive.*

4.2.3 Modified Algorithm based on Valuation Tracking Procedure

In this section we present a modified version of Algorithm 7 which is useful for the subsequent developments under the stochastic-valuation model in Section 4.3.

First we show how to modify Algorithm 7 so that its decision at each time t depends on only the remaining inventory, instead of the entire history of purchase decisions $X_1, \ldots, X_{t-1}$.

Definition 4.2.3. For all $t = 0, \ldots, T$, let I_t denote the random variable for the amount of remaining inventory at the end of time t, which is equal to $k - \sum_{t'=1}^{t} X_{t'}$.

Our modified algorithm makes an offer to customer t according to the *probability* that unit i_t^* hasn't been sold, conditioned on the realized value of I_{t-1}. In this way, its decisions depend on only the inventory state, instead of the exact decisions of past customers.

Definition 4.2.4 (Algorithm 7'). Define the following algorithm for choosing the price at each time t, based on the past valuations $V_1, \ldots, V_{t-1}$ and the amount of remaining inventory I_{t-1}.

1. Consider the indices i_t^* and ℓ_t during iteration t of Algorithm 7, which are deterministic based on $V_1, \ldots, V_{t-1}$.

2. Compute the probability that $\texttt{sold}[i_t^*] = \texttt{true}$ on a run of Algorithm 7, *conditioned on I_{t-1} units of inventory remaining after time $t-1$* in that run. Let γ_t denote this probability.

3. With probability γ_t, make an offer to customer t with the same price distribution as Algorithm 7 (line 7); with probability $1-\gamma_t$, reject customer t.

Algorithm 7' chooses the distribution for P_t by "averaging" over all runs of Algorithm 7 which have the same value of I_{t-1}. We first remark that this can be done in polynomial time, despite there being exponentially many sample paths for Algorithm 7. We prove the following in Section C.2.

Lemma 4.2.5. *The value of γ_t in Step 2 of Algorithm 7' can be computed in polynomial time.*

We now introduce some notation to disambiguate between random variables depicting the runs of different algorithms.

Definition 4.2.6. For an algorithm $\mathcal{A}$, let $P_t^{\mathcal{A}}$, $X_t^{\mathcal{A}}$, and $I_t^{\mathcal{A}}$ be the random variables for the price at time t, purchase decision at time t, and inventory remaining at the end of time t, respectively. Let $\mathsf{ALG}^{\mathcal{A}}$ be the random variable for the total revenue earned by algorithm $\mathcal{A}$. We will omit the superscripts $\mathcal{A}$ when the context is clear.

Let $\mathcal{A} = \mathsf{A1}$ refer to Algorithm 7 and $\mathcal{A} = \mathsf{A1}'$ refer to Algorithm 7'.

We show that Algorithms 7 and 7' are virtually the same in that they have identical distributions for the remaining inventory at each time step, as well as the random price at each time step conditioned on any value of remaining inventory. This also establishes that Algorithm 7' is feasible, in that it does not try to make a sale with zero remaining inventory.

Lemma 4.2.7. *For all $t \in [T]$, $k' \in \{0, \dots, k\}$ such that $\Pr[I_{t-1}^{\mathsf{A1}'} = k'] > 0$, and $j \in \{1, \dots, m, m+1\}$, $\Pr[P_t^{\mathsf{A1}'} = r^{(j)} | I_{t-1}^{\mathsf{A1}'} = k'] = \Pr[P_t^{\mathsf{A1}} = r^{(j)} | I_{t-1}^{\mathsf{A1}} = k']$.*

Also, for all $t = 0, \dots, T$ and $k' \in \{0, \dots, k\}$, $\Pr[I_t^{\mathsf{A1}'} = k'] = \Pr[I_t^{\mathsf{A1}} = k']$.

Lemma 4.2.7, proven in Section C.2, is a consequence of the design of Algorithm 7'. For all t, the random price $P_t^{\mathsf{A1}'}$ is identically distributed as P_t^{A1}, conditional on any value for the amount of remaining inventory at the end of time $t-1$. Hence if $I_{t-1}^{\mathsf{A1}'}$ and I_{t-1}^{A1} are identically distributed, then so are $I_t^{\mathsf{A1}'}$ and I_t^{A1}. This allows us to inductively establish that the two algorithms have the same aggregate behavior after combining all sample paths, even though their behavior may differ given a specific history of purchase decisions. This also makes it easy to see that the expected revenues of the two algorithms are the same. Lemma 4.2.7 directly implies the following theorem.

Theorem 4.2.8. *Algorithm 7' is* $\mathsf{CR}(\mathcal{P})$*-competitive.*

4.2.4 Further Modified Algorithm and Structural Properties

In this section we present a further-modified version of Algorithm 7' which satisfies two structural properties: (i) it never rejects a customer if it has remaining inventory, offering the maximum price instead; (ii) the distribution of prices offered to a customer is strictly stochastically-decreasing in the amount of remaining inventory. Property (ii) is consistent with the classical structural result from Gallego and Van Ryzin (1994, Thm. 1) and its generalization to non-homogeneous demand in Zhao and Zheng (2000, Thm. 3): at any time step, if the firm has more inventory, then it is strictly more willing to sell at lower prices.

Definition 4.2.9 (Algorithm 7"). Define the following modification to Algorithm 7': in Step 3, offer price the maximum price $r^{(m)}$ to customer t, instead of rejecting her, with probability $1-\gamma_t$.

We prove the following general lemma, which is intuitively easy to see, in Section C.2.

Lemma 4.2.10. *Let $\mathcal{A}$ be any pricing algorithm. Let $\mathcal{A}'$ be the modified algorithm which: whenever $\mathcal{A}$ would reject a customer while there is remaining inventory, $\mathcal{A}'$ offers price $r^{(m)}$ instead. Then on any valuation sequence $V_1, \ldots, V_T$,* $\mathbb{E}[\mathsf{ALG}^{\mathcal{A}'}] \geq \mathbb{E}[\mathsf{ALG}^{\mathcal{A}}]$.

Lemma 4.2.10 shows that Algorithm 7" is $\mathsf{CR}(\mathcal{P})$-competitive. Now, we would like to further show that the probability of Algorithm 7' rejecting, or correspondingly the probability of Algorithm 7" offering the maximum price, is smaller when conditioned on larger values of remaining inventory.

Theorem 4.2.11. *Suppose that the unconditional probability of Algorithm 7' rejecting customer t,* $\Pr[\mathtt{sold}[i_t^*] = \mathtt{true}]$*, lies in (0,1). Then for any $k_1 < k_2$ with* $\Pr[I_{t-1} = k_1] > 0$

and $\Pr[I_{t-1} = k_2] > 0$,

$$\Pr[\mathtt{sold}[i_t^*] = \mathtt{true} | I_{t-1} = k_1] > \Pr[\mathtt{sold}[i_t^*] = \mathtt{true} | I_{t-1} = k_2].$$

That is, Algorithm 7" chooses strictly stochastically-lower prices when it has more inventory.

This structural property is intuitive, and we defer its proof to Section C.2.

4.3 Stochastic Valuations

The model with stochastic valuations differs from the model with deterministic valuations studied in the previous section in the following ways. The valuation of each arriving customer is now *randomly* drawn from some probability distribution. The valuations of different customers are independent, but not necessarily identically distributed. An online algorithm is given the valuation *distribution* for each customer after the price for that customer has been chosen.

Definition 4.3.1. We use the following notation, defined for all t:

- V_t: the valuation of customer t, a random variable taking values in $\{r^{(0)}, r^{(1)}, \ldots, r^{(m)}\}$;
- $\mathbf{v}_t$: the probability vector $(v_t^{(0)}, v_t^{(1)}, \ldots, v_t^{(m)})$ for the distribution of V_t, with $v_t^{(j)} = \Pr[V_t = r^{(j)}]$ and $\sum_{j=0}^{m} v_t^{(j)} = 1$;
- $\mathbf{V}_t$: $(V_1, \ldots, V_t)$, the vector of realized valuations up to time t;
- $\mathbf{P}_t^{\mathcal{A}}$: $(P_1^{\mathcal{A}}, \ldots, P_t^{\mathcal{A}})$, the vector of prices up to time t chosen by algorithm $\mathcal{A}$.

We now provide a justification for our choice of offline optimum in our definition of competitiveness and competitive ratio, where in (4.2) we have replaced $\mathsf{OPT}(V_1, \ldots, V_T)$ with its expected value (and the values of V_t are realized independently).

4.3.1 Discussion of the Offline Optimum with Stochastic Valuations

The weakest (least clairvoyant) offline benchmark one could compare against with stochastic valuations is the following. Consider an offline algorithm which is given the sequence of valuation distributions, $\mathbf{v}_1, \ldots, \mathbf{v}_T$, in advance. Given this sequence, it can solve for the policy which maximizes expected revenue, using dynamic programming. Clearly, the expected revenue of such a policy is an upper bound on the expected revenue obtainable by an online algorithm.

However, such a benchmark is difficult to compare against, because the optimal policy knowing $\mathbf{v}_1, \ldots, \mathbf{v}_T$, while computable in polynomial time, may not admit any structure. Therefore, we relax the offline optimum by allowing the offline algorithm to know the *realizations* of all the valuations $V_1, \ldots, V_T$ in advance. The optimal algorithm knowing such information then has a trivial structure (sell to the k largest valuations).

In line with the definition from Section 4.2, let $\mathsf{OPT}(V_1, \ldots, V_T)$ denote the sum of the k largest valuations in $V_1, \ldots, V_T$. We define the competitive ratio with stochastic valuations to be

$$\inf_{\mathbf{v}_1,\ldots,\mathbf{v}_T} \frac{\mathbb{E}[\mathsf{ALG}(\mathbf{v}_1, \ldots, \mathbf{v}_T)]}{\mathbb{E}_{V_1\sim\mathbf{v}_1,\ldots,V_T\sim\mathbf{v}_T}[\mathsf{OPT}(V_1, \ldots, V_T)]}. \tag{4.3}$$

In (4.3), an adversary chooses $\mathbf{v}_1, \ldots, \mathbf{v}_T$, which determines the expected revenue of the online algorithm; the offline optimum is defined to be the expected value of $\mathsf{OPT}(V_1, \ldots, V_T)$ where $V_1, \ldots, V_T$ are realized according to $\mathbf{v}_1, \ldots, \mathbf{v}_T$. We should note that such an expected value cannot be computed in polynomial time; it is related to computing the expected project duration in a PERT network with independent task durations, which is #P-hard (Hagstrom, 1988).

Finally, we should point out that a different relaxation in the offline optimum is possible, originating from Gallego and Van Ryzin (1994); see also Talluri and Van Ryzin (2006). Given the valuation distributions, $\mathbf{v}_1, \ldots, \mathbf{v}_T$, one can write the following *Deterministic Linear Program (DLP)*:

$$\max \sum_{j=1}^{m} r^{(j)} \sum_{t=1}^{T} x_t^{(j)} \Pr[V_t \geq r^{(j)}] \tag{4.4}$$

$$\sum_{j=1}^{m} \sum_{t=1}^{T} x_t^{(j)} \Pr[V_t \geq r^{(j)}] \leq k$$

$$\sum_{j=1}^{m} x_t^{(j)} \leq 1 \qquad t \in [T]$$

$$x_t^{(j)} \geq 0 \qquad j \in [m], t \in [T]$$

Let $\mathsf{OPT}_{\mathsf{LP}}(\mathbf{v}_1, \ldots, \mathbf{v}_T)$ denote the optimal objective value of the LP (4.4) where the value of each $\Pr[V_t \geq r^{(j)}]$ is computed based on $\mathbf{v}_t$. It can be shown that $\mathsf{OPT}_{\mathsf{LP}}(\mathbf{v}_1, \dots, \mathbf{v}_T)$ is an upper bound on the expected revenue of the optimal dynamic programming policy knowing $\mathbf{v}_1, \dots, \mathbf{v}_T$, since $x_t^{(j)}$ encapsulates the unconditional probability of the policy offering price j to customer t.

Nonetheless, in this chapter we don't compare against $\mathsf{OPT}_{\mathsf{LP}}(\mathbf{v}_1, \dots, \mathbf{v}_T)$, which appears to be too strong of an offline benchmark. We show that the competitive ratio of $\mathsf{CR}(\mathcal{P})$, which was optimal in our deterministic setting, can still be achieved with stochastic valuations under definition (4.3). In other words, allowing the offline optimum to know the realizations of $V_1, \dots, V_T$ does not decrease the competitive ratio. On the other hand, allowing the offline optimum to use the fractionality of the DLP (4.4) does decrease the competitive ratio; we provide some examples in Section C.5.

4.3.2 Optimally-Competitive Algorithm with Exponential Runtime

Having established our offline benchmark, we now derive $\mathsf{CR}(\mathcal{P})$-competitive algorithms in the stochastic-valuation model. We do so by using our valuation tracking procedure as a subroutine, in a similar way to the development in Section 4.2.3, which may be helpful to reference.

Conceptually, our algorithm is a generalization of Algorithm 7' to stochastic valuations. However, since the assignment procedure in Algorithm 7 is no longer deterministic, we describe the algorithm in a different way. At a time step t, given $\mathbf{v}_1, \dots, \mathbf{v}_{t-1}$ and k':

1. Consider a run of Algorithm 7 to the end of time t, where $V_1, \dots, V_{t-1}$ are randomly drawn according to $\mathbf{v}_1, \dots, \mathbf{v}_{t-1}$. For all $j \in \{1, \dots, m, m+1\}$, compute $\Pr[P_t^{\mathsf{A1}} = r^{(j)} | I_{t-1}^{\mathsf{A1}} = k']$, where the probability is over both the random valuations and the random prices chosen by the algorithm. (If $\Pr[I_{t-1}^{\mathsf{A1}} = k']$ has measure 0, then choose price $r^{(m+1)}$.)

2. For each $j \in \{1, \dots, m, m+1\}$, choose price $r^{(j)}$ with probability $\Pr[P_t^{\mathsf{A1}} = r^{(j)} | I_{t-1}^{\mathsf{A1}} = k']$.

Let Exp denote this algorithm, and we will use the corresponding notation from Definition 4.2.6. It will be seen that Exp is a feasible policy when we establish that I_t^{Exp} and I_t^{A1}

are identically distributed. First expand the expression $\Pr[P_t^{\mathsf{A1}} = r^{(j)} | I_{t-1}^{\mathsf{A1}} = k']$ as follows:

$$\frac{\Pr[P_t^{\mathsf{A1}} = r^{(j)} \cap I_{t-1}^{\mathsf{A1}} = k']}{\Pr[I_{t-1}^{\mathsf{A1}} = k']} = \frac{\sum\limits_{\mathbf{P}_{t-1}^{\mathsf{A1}}, \mathbf{V}_{t-1}: I_{t-1}^{\mathsf{A1}} = k'} \Pr[P_t^{\mathsf{A1}} = r^{(j)} | \mathbf{P}_{t-1}^{\mathsf{A1}}, \mathbf{V}_{t-1}] \cdot \Pr[\mathbf{P}_{t-1}^{\mathsf{A1}} \cap \mathbf{V}_{t-1}]}{\sum\limits_{\mathbf{P}_{t-1}^{\mathsf{A1}}, \mathbf{V}_{t-1}: I_{t-1}^{\mathsf{A1}} = k'} \Pr[\mathbf{P}_{t-1}^{\mathsf{A1}} \cap \mathbf{V}_{t-1}]}. \tag{4.5}$$

In (4.5), the probability $\Pr[P_t^{\mathsf{A1}} = r^{(j)} | \mathbf{P}_{t-1}^{\mathsf{A1}}, \mathbf{V}_{t-1}]$, which conditions on a fixed history $P_1^{\mathsf{A1}}, V_1, \ldots, P_{t-1}^{\mathsf{A1}}, V_{t-1}$, is defined by lines 6–10 of Algorithm 7. Thus, calculating (4.5) requires enumerating all histories that result in $I_{t-1}^{\mathsf{A1}} = k'$.

Unfortunately, at each time step t, this takes time exponential in t. The computational difficulty arises because the assignment procedure in Algorithm 7 is no longer deterministic, as it was throughout Section 4.2. For now, we ignore computational constraints and focus on obtaining an $\mathsf{CR}(\mathcal{P})$-competitive online algorithm; in Section 4.3.3 we show how to use sampling to achieve polynomial runtime while only losing ε in the competitiveness.

The following lemma is analogous to Lemma 4.2.7 and proved in Section C.3.

Lemma 4.3.2. *For all $t \in [T]$, $k' \in \{0, \ldots, k\}$ such that $\Pr[I_{t-1}^{\mathsf{Exp}} = k'] > 0$, and $j \in \{1, \ldots, m, m+1\}$,*

$$\Pr[P_t^{\mathsf{Exp}} = r^{(j)} | I_{t-1}^{\mathsf{Exp}} = k'] = \Pr[P_t^{\mathsf{A1}} = r^{(j)} | I_{t-1}^{\mathsf{A1}} = k']. \tag{4.6}$$

Also, for all $t = 0, \ldots, T$ and $k' \in \{0, \ldots, k\}$,

$$\Pr[I_t^{\mathsf{Exp}} = k'] = \Pr[I_t^{\mathsf{A1}} = k']. \tag{4.7}$$

Lemma 4.3.2 establishes that Exp is a feasible policy, i.e. it does not try to make a sale with no inventory remaining. Having established this, it remains to prove that Exp is optimally competitive. Theorem 4.3.3 is proved in Section C.3.

Theorem 4.3.3. $\mathbb{E}[\mathsf{ALG}^{\mathsf{Exp}}(V_1, \ldots, V_T)] = \mathbb{E}[\mathsf{ALG}^{\mathsf{A1}}(V_1, \ldots, V_T)]$. *By Theorem 4.2.2, $\mathsf{ALG}^{\mathsf{A1}}(V_1, \ldots, V_T) = \frac{1}{q}\mathsf{OPT}(V_1, \ldots, V_T)$ for all realizations $(V_1, \ldots, V_T)$. Therefore,* Exp *is $\mathsf{CR}(\mathcal{P})$-competitive.*

We should point out that although Exp does not inherit the polynomial-time property from Algorithm 7', it does inherit the structural property of the price at any time being stochastically-decreasing in the amount of remaining inventory. This is immediate from Theorem 4.2.11, which holds conditioned on any realization of $V_1, \dots, V_T$.

Also, note the following. P_t^{Exp} and P_t^{A1} are only guaranteed to be identically distributed when averaged over *all* the sample paths up to time $t-1$ such that the total remaining inventory is k'. They may not be identically distributed when conditioned on a specific purchase sequence $X_1, \dots, X_{t-1}$ such that $\sum_{t'=1}^{t-1} X_{t'} = k - k'$, or a specific valuation sequence $V_1, \dots, V_{t-1}$. Nonetheless, our method works in general. For example, if valuations were correlated, then we would condition on both I_{t-1} and $V_1, \dots, V_{t-1}$. One benefit of conditioning on only I_{t-1} in the independent case is to limit the state space, which is necessary for our polynomial-time sampling algorithm in Section 4.3.3.

4.3.3 Emulating the Exponential-runtime Algorithm using Sampling

In this section we show how to "emulate" Exp, using sampling, to achieve a polynomial runtime. First we provide a high-level overview of the challenges and the techniques used to overcome them. The intrinsic difficulty is that our original procedure is based on tracking the value of the offline optimum, but this becomes a #P-hard problem when the optimum equals the expected value of the k largest elements from independent realizations (see Hagstrom (1988)).

To overcome this using sampling, suppose we are at the start of time t, with inventory k' remaining. If we randomly sample a run of Algorithm 7 (drawing valuations randomly) such that $I_{t-1}^{\mathsf{A1}} = k'$, and copy price P_t^{A1} for time t, then we would match the probabilities prescribed in (4.5). This motivates the following algorithm: sample runs of Algorithm 7 to the end of $t-1$ until hitting one where $I_{t-1}^{\mathsf{A1}} = k'$, and then choose the price for time t according to lines 6–10 of Algorithm 7. Such an algorithm is equivalent to Exp, and thus would be $\mathsf{CR}(\mathcal{P})$-competitive.

However, on sample paths where $\Pr[I_{t-1}^{\mathsf{A1}} = k']$ is small, the sampling could take arbitrarily long. We limit the number of sampling tries so that the algorithm deterministically finishes in polynomial time, and show that the total measure of sample paths which fail at any point is $O(\varepsilon)$. Unfortunately, there could be correlation between the sampling failing, and having high revenue on a sample path. Nonetheless, we can *couple* the sample paths of the sampling algorithm to those of the exponential-time algorithm, mark the first point of failure on each sample path, and bound the difference in revenue after that point.

Algorithm 8 Weakly Randomized Online Algorithm based on Inventory Remaining

1: `inventory` $= k$
2: $t = 1$
3: **while** customer t arrives **do**
4: **repeat**
5: run Algorithm 7 to the start of time t, with valuations $V_1, \ldots, V_{t-1}$ drawn according to $\mathbf{v}_1, \ldots, \mathbf{v}_{t-1}$, and prices $P_1^{\mathsf{A1}}, \ldots, P_{t-1}^{\mathsf{A1}}$ realized according to the random prices chosen by Algorithm 7
6: **if** $I_{t-1}^{\mathsf{A1}} =$ `inventory` **then**
7: choose each price $r^{(1)}, \ldots, r^{(m)}, r^{(m+1)}$ according to the probability that Algorithm 7 (on this run) would choose that price for customer t
8: observe $\mathbf{v}_t$
9: observe purchase decision of customer t and update `inventory` accordingly
10: $t = t + 1$ and **continue** to next iteration of **while** loop
11: **end if**
12: **until** $C(k+1)t^2$ runs elapse
13: choose price ∞
14: $t = t + 1$
15: **end while**

Input: Customers $t = 1, 2, \ldots$ arriving online, with each valuation distribution $\mathbf{v}_t$ revealed after the price P_t is chosen.
Output: For each customer t, a (possibly random) price P_t for her.

The details of the sampling algorithm, which we will call Samp, are specified in Algorithm 8.

In line 12, C is a positive integer to be chosen later. The decision of what to do when the sampling *fails*, i.e. defaults to line 13, is inconsequential, since in our analysis we do not expect any revenue from a sample path *after* the first point of failure.

To bound the revenue of Algorithm 8, we consider an algorithm which behaves identically to Algorithm 8, except even when it defaults to line 13, it is able to behave as if the sampling succeeded and makes the same decisions as lines 7–10. Such an algorithm is equivalent to Exp, and hereafter we will refer to it as Exp. The results of the sample runs do not affect the outcome of the algorithm, but help with bookkeeping.

Definition 4.3.4. Let F_t^{Samp} be the indicator random variable for the sampling in Algorithm 8 failing at time t, defined for all $t \in [T+1]$. Let $F_{T+1}^{\mathsf{Samp}} = 1$ deterministically. Analogously, let F_t^{Exp} be the indicator random variable for the sampling in Exp "failing" at time t, $\forall t \in [T+1]$.

For convenience, here we will use different random variables to denote the valuations in the runs of Samp and Exp: V_t^{Samp} and V_t^{Exp}, respectively. We will also use the notation from Definition 4.2.6.

Definition 4.3.5. Define the *history up to time* t to consist of realizations up to and including the sampling at time t. Formally, for all $t \in [T+1]$, let $\mathbf{h}_t = (f_1, p_1, v_1, \ldots, f_{t-1}, p_{t-1}, v_{t-1}, f_t)$, where:

- $f_{t'} \in \{0,1\}$, for all $t' \in [t]$;
- $p_{t'}$ is a price in $\{r^{(1)}, \ldots, r^{(m)}, r^{(m+1)}\}$, for all $t' \in [t-1]$;
- $v_{t'}$ is a valuation in $\{r^{(0)}, r^{(1)}, \ldots, r^{(m)}\}$, for all $t \in [t-1]$.

Furthermore, define the following vectors of random variables for all $t \in [T+1]$:

- $\mathbf{H}_t^{\mathsf{Samp}} = (F_1^{\mathsf{Samp}}, P_1^{\mathsf{Samp}}, V_1^{\mathsf{Samp}}, \ldots, F_{t-1}^{\mathsf{Samp}}, P_{t-1}^{\mathsf{Samp}}, V_{t-1}^{\mathsf{Samp}}, F_t^{\mathsf{Samp}})$;
- $\mathbf{H}_t^{\mathsf{Exp}} = (F_1^{\mathsf{Exp}}, P_1^{\mathsf{Exp}}, V_1^{\mathsf{Exp}}, \ldots, F_{t-1}^{\mathsf{Exp}}, P_{t-1}^{\mathsf{Exp}}, V_{t-1}^{\mathsf{Exp}}, F_t^{\mathsf{Exp}})$.

Now, we would like to partition the sample paths by the history up to the first point of failure, and prove that the two algorithms behave identically up to this point.

Definition 4.3.6. Let $\mathcal{F}_t$ denote the histories up to time t such that the first failure in the sampling occurs at time t. Formally, for all $t \in [T+1]$, $\mathcal{F}_t$ is the set of $\mathbf{h}_t = (f_1, p_1, v_1, \ldots, f_{t-1}, p_{t-1}, v_{t-1}, f_t)$ such that $f_1 = \ldots = f_{t-1} = 0$ and $f_t = 1$. ($p_1, \ldots, p_{t-1}$ and $v_1, \ldots, v_{t-1}$ are arbitrary, and thus $|\mathcal{F}_t| = (m+1)^{2(t-1)}$.)

Lemma 4.3.7. *For a run of Algorithm 8,* $\bigcup_{t=1}^{T+1} \bigcup_{\mathbf{h}_t \in \mathcal{F}_t} \{\mathbf{H}_t^{\mathsf{Samp}} = \mathbf{h}_t\}$ *is a set of mutually exclusive and collectively exhaustive events. Analogously, for a run of* Exp, $\bigcup_{t=1}^{T+1} \bigcup_{\mathbf{h}_t \in \mathcal{F}_t} \{\mathbf{H}_t^{\mathsf{Exp}} = \mathbf{h}_t\}$ *is a set of mutually exclusive and collectively exhaustive events.*

Furthermore, $\Pr[\mathbf{H}_t^{\mathsf{Samp}} = \mathbf{h}_t] = \Pr[\mathbf{H}_t^{\mathsf{Exp}} = \mathbf{h}_t]$, *for all* $t \in [T+1]$ *and* $\mathbf{h}_t \in \mathcal{F}_t$.

Lemma 4.3.7 is straight-forward, so we defer its proof to Section C.3. Having proved it, we can write:

$$\mathbb{E}[\mathsf{ALG}^{\mathsf{Samp}}] = \sum_{t=1}^{T+1} \sum_{\mathbf{h}_t \in \mathcal{F}_t} \mathbb{E}[\mathsf{ALG}^{\mathsf{Samp}} | \mathbf{H}_t^{\mathsf{Samp}} = \mathbf{h}_t] \Pr[\mathbf{H}_t^{\mathsf{Samp}} = \mathbf{h}_t] \tag{4.8}$$

$$\mathbb{E}[\mathsf{ALG}^{\mathsf{Exp}}] = \sum_{t=1}^{T+1} \sum_{\mathbf{h}_t \in \mathcal{F}_t} \mathbb{E}[\mathsf{ALG}^{\mathsf{Exp}} | \mathbf{H}_t^{\mathsf{Exp}} = \mathbf{h}_t] \Pr[\mathbf{H}_t^{\mathsf{Exp}} = \mathbf{h}_t]. \tag{4.9}$$

Since we also know that $\Pr[\mathbf{H}_t^{\mathsf{Samp}} = \mathbf{h}_t] = \Pr[\mathbf{H}_t^{\mathsf{Exp}} = \mathbf{h}_t]$, our goal is to compare the expected revenues of the two algorithms conditional on each history $\mathbf{h}_t \in \mathcal{F}_t$.

When $t = T+1$, i.e. the sampling never fails, it is easy to see that the two revenues are equal. Indeed, for any $\mathbf{h}_{T+1} \in \mathcal{F}_{T+1}$:

$$\begin{aligned}
\mathbb{E}[\mathsf{ALG}^{\mathsf{Samp}}|\mathbf{H}_{T+1}^{\mathsf{Samp}} = \mathbf{h}_{T+1}] &= \mathbb{E}\Big[\sum_{t=1}^{T} P_t^{\mathsf{Samp}} \cdot \mathbb{1}(V_t^{\mathsf{Samp}} \geq P_t^{\mathsf{Samp}})\Big|\mathbf{H}_{T+1}^{\mathsf{Samp}} = \mathbf{h}_{T+1}\Big] \\
&= \sum_{t=1}^{T} p_t \cdot \mathbb{1}(v_t \geq p_t) \\
&= \mathbb{E}[\mathsf{ALG}^{\mathsf{Exp}}|\mathbf{H}_{T+1}^{\mathsf{Exp}} = \mathbf{h}_{T+1}]. \qquad (4.10)
\end{aligned}$$

Lemma 4.3.8. *Recall that* $\mathbb{E}[\mathsf{OPT}(V_1, \ldots, V_T)]$ *is the expected value of the offline optimum with* $V_1, \ldots, V_T$ *drawn independently according to* $\mathbf{v}_1, \ldots, \mathbf{v}_T$*. For* $t \leq T$ *and* $\mathbf{h}_t \in \mathcal{F}_t$*,*

$$\mathbb{E}[\mathsf{ALG}^{\mathsf{Exp}}|\mathbf{H}_t^{\mathsf{Exp}} = \mathbf{h}_t] - \mathbb{E}[\mathsf{ALG}^{\mathsf{Samp}}|\mathbf{H}_t^{\mathsf{Samp}} = \mathbf{h}_t] \leq \mathbb{E}[\mathsf{OPT}(V_1, \ldots, V_T)], \qquad (4.11)$$

Proof. Proof. Consider any $t \in [T]$ and $\mathbf{h}_t \in \mathcal{F}_t$. We have

$$\begin{aligned}
\mathbb{E}[\mathsf{ALG}^{\mathsf{Samp}}|\mathbf{H}_t^{\mathsf{Samp}} = \mathbf{h}_t] &\geq \mathbb{E}\Big[\sum_{t'=1}^{t-1} P_{t'}^{\mathsf{Samp}} \cdot \mathbb{1}(V_{t'}^{\mathsf{Samp}} \geq P_{t'}^{\mathsf{Samp}})\Big|\mathbf{H}_t^{\mathsf{Samp}} = \mathbf{h}_t\Big] \\
&= \sum_{t'=1}^{t-1} p_{t'} \cdot \mathbb{1}(v_{t'} \geq p_{t'}). \qquad (4.12)
\end{aligned}$$

Meanwhile, $\mathbb{E}[\mathsf{ALG}^{\mathsf{Exp}}|\mathbf{H}_t^{\mathsf{Exp}} = \mathbf{h}_t]$ can be decomposed into

$$\sum_{t'=1}^{t-1} p_{t'} \cdot \mathbb{1}(v_{t'} \geq p_{t'}) + \mathbb{E}\Big[\sum_{t'=t}^{T} P_{t'}^{\mathsf{Exp}} \cdot \mathbb{1}(V_{t'}^{\mathsf{Exp}} \geq P_{t'}^{\mathsf{Exp}})\Big|\mathbf{H}_t^{\mathsf{Exp}} = \mathbf{h}_t\Big]. \qquad (4.13)$$

We elaborate on the second term in (4.13). Clearly, $\sum_{t'=t}^{T} P_t^{\mathsf{Exp}} \cdot \mathbb{1}(V_t^{\mathsf{Exp}} \geq P_t^{\mathsf{Exp}})$ cannot exceed the sum of the $\min\{k, T-t+1\}$ largest valuations to appear during $t, \ldots, T$, which we denote by $\mathsf{OPT}(V_t^{\mathsf{Exp}}, \ldots, V_T^{\mathsf{Exp}})$. Furthermore, the random valuations $V_t^{\mathsf{Exp}}, \ldots, V_T^{\mathsf{Exp}}$ are independent of the history $\mathbf{H}_t^{\mathsf{Exp}}$ up to time t, so we can remove the conditioning and upper-bound (4.13) with

$$\sum_{t'=1}^{t-1} p_{t'} \cdot \mathbb{1}(v_{t'} \geq p_{t'}) + \mathbb{E}[\mathsf{OPT}(V_t^{\mathsf{Exp}}, \ldots, V_T^{\mathsf{Exp}})]. \qquad (4.14)$$

The expectation in (4.14) is with respect to $V_t^{\mathsf{Exp}}, \ldots, V_T^{\mathsf{Exp}}$ being drawn independently according to $\mathbf{v}_t, \ldots, \mathbf{v}_T$. (4.14) in turn is no greater than $\sum_{t'=1}^{t-1} p_{t'} \cdot \mathbb{1}(v_{t'} \geq p_{t'}) + \mathbb{E}[\mathsf{OPT}(V_1^{\mathsf{Exp}}, \ldots, V_T^{\mathsf{Exp}})]$, where the random variables $V_1^{\mathsf{Exp}}, \ldots, V_{t-1}^{\mathsf{Exp}}$ are *not* conditioned on the event $\mathbf{H}_t^{\mathsf{Exp}} = \mathbf{h}_t$. The proof of the lemma concludes by comparing this expression with (4.12). $\square$

Substituting (4.10), for $\mathbf{h}_{T+1} \in \mathcal{F}_{T+1}$, and (4.11), for $\mathbf{h}_1, \ldots, \mathbf{h}_T \in \mathcal{F}_1, \ldots, \mathcal{F}_T$, into (4.8) and (4.9), we conclude that

$$\mathbb{E}[\mathsf{ALG}^{\mathsf{Exp}}] - \mathbb{E}[\mathsf{ALG}^{\mathsf{Samp}}] \leq \mathbb{E}[\mathsf{OPT}] \cdot \Big(\sum_{t=1}^{T} \sum_{\mathbf{h}_t \in \mathcal{F}_t} \Pr[\mathbf{H}_t^{\mathsf{Samp}} = \mathbf{h}_t] \Big). \tag{4.15}$$

By Definition 4.3.6, the expression in parentheses is the total probability of the sampling failing at any point, before choosing the final price P_T^{Samp}. We bound the term for each $t \in [T]$ separately. As t increases, the number of samples increases, so the probability of failure decreases:

Lemma 4.3.9. *For all $t \in [T]$, $\sum_{\mathbf{h}_t \in \mathcal{F}_t} \Pr[\mathbf{H}_t^{\mathsf{Samp}} = \mathbf{h}_t] \leq \frac{1}{eCt^2}$.*

Proof. Proof. Consider any $t \in [T]$. For all $\mathbf{h}_t \in \mathcal{F}_t$, let $G(\mathbf{h}_t) = (f_1, p_1, v_1, \ldots, f_{t-1}, p_{t-1}, v_{t-1})$, which is the vector of the first $3(t-1)$ entries in $\mathbf{h}_t$. Let $\mathbf{G}_{t-1}^{\mathsf{Exp}} = (F_1^{\mathsf{Exp}}, P_1^{\mathsf{Exp}}, V_1^{\mathsf{Exp}}, \ldots, F_{t-1}^{\mathsf{Exp}}, P_{t-1}^{\mathsf{Exp}}, V_{t-1}^{\mathsf{Exp}})$, which is a vector of $3(t-1)$ random variables.

We can write $\sum_{\mathbf{h}_t \in \mathcal{F}_t} \Pr[\mathbf{H}_t^{\mathsf{Exp}} = \mathbf{h}_t]$ as

$$\sum_{\mathbf{h}_t \in \mathcal{F}_t} \Pr[F_t^{\mathsf{Exp}} = 1 | \mathbf{G}_{t-1}^{\mathsf{Exp}} = G(\mathbf{h}_t)] \Pr[\mathbf{G}_{t-1}^{\mathsf{Exp}} = G(\mathbf{h}_t)]. \tag{4.16}$$

Now, for each $\mathbf{h}_t \in \mathcal{F}_t$, $\Pr[F_t^{\mathsf{Exp}} = 1 | \mathbf{G}_{t-1}^{\mathsf{Exp}} = G(\mathbf{h}_t)]$ is the probability that all $C(k+1)t^2$ independent runs of Algorithm 7 fail to match the inventory remaining at the start of time t according to $\mathbf{h}_t$. For convenience, define $I(\mathbf{h}_t) = k - \sum_{t'=1}^{t-1} \mathbb{1}(v_{t'} \geq p_{t'})$. Then

$$\Pr[F_t^{\mathsf{Exp}} = 1 | \mathbf{G}_{t-1}^{\mathsf{Exp}} = G(\mathbf{h}_t)] = (1 - \Pr[I_{t-1}^{\mathsf{A1}} = I(\mathbf{h}_t)])^{C(k+1)t^2}, \tag{4.17}$$

where I_{t-1}^{A1} is the total inventory remaining at the start of time t in a run of Algorithm 7.

Therefore, we can partition the $\mathbf{h}_t$ in $\mathcal{F}_t$ by $I(\mathbf{h}_t)$. For all $k' \in \{0, \ldots, k\}$, define $\rho_{t,k'} = \Pr[I^{\mathsf{A1}}_{t-1} = k']$. The following can be derived by substituting (4.17) into (4.16):

$$\begin{aligned}\sum_{\mathbf{h}_t \in \mathcal{F}_t} \Pr[\mathbf{H}^{\mathsf{Exp}}_t = \mathbf{h}_t] &= \sum_{k'=0}^{k} (1-\rho_{t,k'})^{C(k+1)t^2} \sum_{\mathbf{h}_t \in \mathcal{F}_t : I(\mathbf{h}_t) = k'} \Pr[\mathbf{G}^{\mathsf{Exp}}_{t-1} = G(\mathbf{h}_t)] \\ &\leq \sum_{k'=0}^{k} \exp(-\rho_{t,k'} C(k+1)t^2) \sum_{\mathbf{h}_t \in \mathcal{F}_t : I(\mathbf{h}_t) = k'} \Pr[\mathbf{G}^{\mathsf{Exp}}_{t-1} = G(\mathbf{h}_t)] \quad (4.18)\end{aligned}$$

At this point, we would like to argue that $\sum_{\mathbf{h}_t \in \mathcal{F}_t : I(\mathbf{h}_t) = k'} \Pr[\mathbf{G}^{\mathsf{Exp}}_{t-1} = G(\mathbf{h}_t)] \leq \Pr[I^{\mathsf{Exp}}_{t-1} = k']$. To see this, note that $\sum_{\mathbf{h}_t \in \mathcal{F}_t : I(\mathbf{h}_t) = k'} \Pr[\mathbf{G}^{\mathsf{Exp}}_{t-1} = G(\mathbf{h}_t)] = \Pr[I^{\mathsf{Exp}}_{t-1} = k' \cap (F^{\mathsf{Exp}}_1 = \ldots = F^{\mathsf{Exp}}_{t-1} = 0)]$.

Applying the second statement of Lemma 4.3.2, we see that $\Pr[I^{\mathsf{Exp}}_{t-1} = k'] = \rho_{t,k'}$. Substituting into (4.18), the following can be derived:

$$\begin{aligned}\sum_{\mathbf{h}_t \in \mathcal{F}_t} \Pr[\mathbf{H}^{\mathsf{Exp}}_t = \mathbf{h}_t] &\leq \sum_{k'=0}^{k} \rho_{t,k'} \exp(-\rho_{t,k'} C(k+1)t^2) \\ &\leq \sum_{k'=0}^{k} \frac{1}{C(k+1)t^2} \exp(-1) \\ &= \frac{1}{eCt^2}.\end{aligned}$$

The second inequality holds because for a single $\rho_{t,k'} \in [0,1]$, the function $\rho_{t,k'} e^{-\rho_{t,k'} C(k+1)t^2}$ is maximized at $\rho_{t,k'} = \frac{1}{C(k+1)t^2}$. The proof of the lemma is now complete. □

It now follows easily that the sampling algorithm is within ε of being optimally competitive.

Theorem 4.3.10. *For all $\varepsilon > 0$, if we set $C = \lceil \frac{6}{e\pi^2\varepsilon} \rceil$ in line 12 of Algorithm 8, then it is $(\frac{1}{q} - \varepsilon)$-competitive, and has runtime polynomial in $\frac{1}{\varepsilon}$, k, T, and m.*

Theorem 4.3.10 is straight-forward and proved in Section C.3.

4.4 No Information on Valuations

In this section we discuss whether it is possible for an online algorithm to be $\mathsf{CR}(\mathcal{P})$-competitive without any information (before or after, deterministic or distributional) on the valuations.

First we show that this is impossible for any online algorithm which price-skims independently, i.e. realizes its random price at each time step using an independent source of random bits.

Proposition 4.4.1. *Suppose that either: (i) $m \geq 2$ and valuations can be 0 (as usual); or (ii) $m \geq 3$ and valuations cannot be 0. (Recall that m is the number of prices.) Then for any online algorithm where each P_t chosen independently based on the sales history $X_1, \ldots, X_{t-1}$, there exists a sequence $V_1, \ldots, V_T$ such that*

$$\frac{\mathbb{E}[\mathsf{ALG}(V_1, \ldots, V_T)]}{\mathsf{OPT}(V_1, \ldots, V_T)} < \mathsf{CR}(\mathcal{P}).$$

Proof. Proof of Proposition 4.4.1. Let the starting inventory $k = 1$.

First, it is easy to see that if the distribution of P_1 is not such that $\Pr[P_1 = r^{(j)}] = \frac{q^{(j)}}{q}$ for all $j \in [m]$, then for some deterministic instance consisting of a single valuation in $\{r^{(1)}, \ldots, r^{(m)}\}$, $\frac{\mathbb{E}[\mathsf{ALG}]}{\mathsf{OPT}}$ will be strictly less than $\frac{1}{q}$. Therefore we can without loss of generality assume that $\Pr[P_1 = r^{(j)}] = \frac{q^{(j)}}{q}$ for all $j \in [m]$ (regardless of whether valuations can be 0).

Now suppose $m \geq 3$. Consider the distribution of P_2 conditioned on $X_1 = 0$. If $\Pr[P_2 \geq r^{(m)} | X_1 = 0] = 1$, then consider the instance $T = 2, V_1 = 1, V_2 = r^{(m-1)}$. $\mathsf{OPT} = r^{(m-1)}$, which exceeds 1, since $m \geq 3$. Meanwhile, $q\mathbb{E}[\mathsf{ALG}] = q\frac{1}{q} < \mathsf{OPT}$. On the other hand, if $\Pr[P_2 \geq r^{(m)} | X_1 = 0] < 1$, then consider the instance $T = 3, V_1 = V_2 = r^{(m-1)}, V_3 = r^{(m)}$. $\mathsf{OPT} = r^{(m)}$. $q\mathbb{E}[P_1 X_1] = r^{(m-1)}$, while $\mathbb{E}[X_1] = 1 - \frac{q^{(m)}}{q}$. The best case for the algorithm, given that $V_2 = r^{(m-1)}$, is $P_2 = r^{(m-1)}$ when $P_2 < r^{(m)}$. Let $\Pr[P_2 = r^{(m-1)} | X_1 = 0] = \alpha$, which we know is positive. In this case, $q\mathbb{E}[P_2 X_2] = r^{(m-1)} \alpha q^{(m)}$ and $\mathbb{E}[X_2] = \alpha \frac{q^{(m)}}{q}$. Hence $q\mathbb{E}[P_3 X_3]$ is at most $qr^{(m)}(1 - \mathbb{E}[X_1 + X_2]) = r^{(m)}(1 - \alpha) q^{(m)}$. All in all, $q\mathbb{E}[\mathsf{ALG}]$ is at most

$$\begin{aligned}
&r^{(m-1)} + r^{(m-1)}\alpha(1 - \frac{r^{(m-1)}}{r^{(m)}}) + r^{(m)}(1-\alpha)(1 - \frac{r^{(m-1)}}{r^{(m)}}) \\
=&r^{(m)} + \alpha(2r^{(m-1)} - \frac{(r^{(m-1)})^2}{r^{(m)}} - r^{(m)}) \\
=&r^{(m)} - \frac{\alpha}{r^{(m)}}(r^{(m)} - r^{(m-1)})^2
\end{aligned}$$

The term getting subtracted is non-zero since $\alpha > 0$ and $r^{(m)} > r^{(m-1)}$. Therefore, $q\mathbb{E}[\mathsf{ALG}] < \mathsf{OPT}$. This completes the proof when $m \geq 3$, since $\mathsf{CR}(\mathcal{P}) = 1/q$.

The case where $m = 2$ and valuations can be 0 is argued analogously. If $\Pr[P_2 \geq$

$r^{(2)}] = 1$, then consider the instance $T = 2, V_1 = 0, V_2 = r^{(1)}$. If $\Pr[P_2 < r^{(2)}] = 1$, then consider the instance $T = 3, V_1 = V_2 = r^{(1)}, V_3 = r^{(2)}$. In both cases, it can be seen that $q\mathbb{E}[\mathsf{ALG}] < \mathsf{OPT}$, completing the proof of Proposition 4.4.1. □

However, we show that it is possible to be $\mathsf{CR}(\mathcal{P})$-competitive if the online algorithm can price-skim in a "coordinated" fashion, with the same probabilities as in Eren and Maglaras (2010).

Proposition 4.4.2. *Consider the following random-fixed-price policy:*

1. *Initially, choose a random price P which is equal to each $r^{(j)}$ with probability $q^{(j)}/q$;*

2. *Offer price P as long as there is remaining inventory.*

This policy is $\mathsf{CR}(\mathcal{P})$-competitive.

Proof. Proof of Proposition 4.4.2. Consider any realization of the valuations, $V_1, \ldots, V_T$. Iteratively define the following quantities, for j from m down to 1:

$$n^{(j)} = \min\Big\{\sum_{t=1}^{T} \mathbb{1}(V_t = r^{(j)}), k - \sum_{j'=j+1}^{m} n^{(j')}\Big\}. \tag{4.19}$$

Essentially, for each j, $n^{(j)}$ denotes the number of valuations equal to $r^{(j)}$ that should be picked out when picking out the $\min\{k, T\}$ largest valuations. OPT is then equal to $\sum_{j=1}^{m} r^{(j)} n^{(j)}$.

Now consider the execution of the policy on this instance. For all $j \in [m]$, if the random fixed price P is equal to $r^{(j)}$, then the number of sales will be equal to $\min\{\sum_{t=1}^{T} \mathbb{1}(V_t \geq r^{(j)}), k\}$, which by definition is equal to $\sum_{j'=j}^{m} n^{(j')}$. Therefore,

$$\begin{aligned}
\mathbb{E}[\mathsf{ALG}] &= \frac{1}{q}\sum_{j=1}^{m} q^{(j)} r^{(j)} \sum_{j'=j}^{m} n^{(j')} \\
&= \frac{1}{q}\sum_{j'=1}^{m} n^{(j')} \sum_{j=1}^{j'} \Big(1 - \frac{r^{(j-1)}}{r^{(j)}}\Big) r^{(j)} \\
&= \frac{1}{q}\sum_{j'=1}^{m} n^{(j')} r^{(j')}
\end{aligned}$$

which equals $\frac{1}{q}\mathsf{OPT}$, completing the proof that the random fixed price is $\mathsf{CR}(\mathcal{P})$-competitive.

□

It is known that correlated randomness is very powerful in the design of online algorithms (see, e.g., Karp et al. (1990), who derive an extremely elegant solution to the online matching problem using correlated randomness). Indeed, we can use our policy from Proposition 4.4.2 under our previous models with more information on the valuations and still have a $\mathsf{CR}(\mathcal{P})$-competitive algorithm. However, this is impractical for several reasons. First, the fact that the random price must be fixed makes it impossible to make use of additional information that may be available on the valuations (this issue was also raised in Eren and Maglaras (2010)). Second, the random-fixed-price policy does not show how the price should evolve as inventory is depleted, and does not satisfy the intuitive structural property in dynamic pricing that the price is greater if the remaining inventory is less (see Section 4.2.4). In Section 4.5, we show how our algorithms from Sections 4.2–4.3 can be adapted into the setting where the personal information of each customer can be used in determining her price.

4.5 Personalization Revenue Management Model

In this section we consider the personalized online revenue management setup introduced by Golrezaei et al. (2014), where:

- the stochastic decision of each customer can be modeled accurately upon her arrival to the e-commerce platform (by using her characteristics);
- however, the overall intensity and characteristics of customers to arrive over time is difficult to model (and treated as unknown/arbitrary).

In our model, this would correspond to the stochastic-valuation model in Section 4.3, with the change that the distribution of each V_t is given *before* the algorithm has to set a price, instead of after. The algorithms from Section 4.3 can still be applied, and will be $\mathsf{CR}(\mathcal{P})$-competitive. Furthermore, it is not possible to be better than $\mathsf{CR}(\mathcal{P})$-competitive even with this personalized information, as discussed in Section 4.1.1.

Nonetheless, in this section we specify how our online algorithms can exploit personalized information to strictly improve their decisions while remaining $\mathsf{CR}(\mathcal{P})$-competitive. Take any $\mathsf{CR}(\mathcal{P})$-competitive algorithm $\mathcal{A}$ for the stochastic-valuation model (e.g. the algorithm from Section 4.3.2, or a modification following Section 4.2.4 which never rejects customers before stocking out). For each time step t and inventory level $k' > 0$ such that $\Pr[I_{t-1}^{\mathcal{A}} = k'] > 0$, consider the distribution for the price P_t chosen by algorithm $\mathcal{A}$ conditioned on $I_{t-1}^{\mathcal{A}} = k'$ (this depends on the previously-observed valuation distributions $\mathbf{v}_1, \ldots, \mathbf{v}_{t-1}$). Since now we also know the distribution $\mathbf{v}_t$ of valuation V_t, we can compute the probability of algorithm $\mathcal{A}$ making a sale during time t,

$$\sum_{j=1}^{m} \Pr[P_t^{\mathcal{A}} = r^{(j)} | I_{t-1}^{\mathcal{A}} = k'] \Pr[V_t \geq r^{(j)}], \tag{4.20}$$

as well as its expected revenue,

$$\sum_{j=1}^{m} r^{(j)} \Pr[P_t^{\mathcal{A}} = r^{(j)} | I_{t-1}^{\mathcal{A}} = k'] \Pr[V_t \geq r^{(j)}]. \tag{4.21}$$

We can interpret (4.21) as the reward given to the algorithm during time t in exchange for the probability (4.20) of consuming inventory. The price distribution used by the algorithm to obtain such an exchange was chosen without knowing the distribution of V_t. However, since now we do know the distribution of V_t, we can potentially make a decision which achieves *more* expected reward under the same consumption probability. Specifically, we solve the following LP:

$$\begin{aligned}
\max \sum_{j=1}^{m} & r^{(j)} \Pr[V_t \geq r^{(j)}] p_j(t,k') \\
\text{s.t.} \sum_{j=1}^{m} & \Pr[V_t \geq r^{(j)}] p_j(t,k') = \sum_{j=1}^{m} \Pr[P_t^{\mathcal{A}} = r^{(j)} | I_{t-1}^{\mathcal{A}} = k'] \Pr[V_t \geq r^{(j)}] \\
& \sum_{j=1}^{m} p_j(t,k') \leq 1 \\
& p_j(t,k') \geq 0 \qquad \forall\, j = 1, \ldots, m
\end{aligned} \tag{4.22}$$

$p_j(t,k')$ represents the probability that we should offer price j at time t, conditioned on the remaining inventory being k'. We know that setting each $p_j(t,k') = \Pr[P_t^{\mathcal{A}} = r^{(j)} | I_{t-1}^{\mathcal{A}} = k']$ is a feasible solution, and hence the optimal objective value of the optimization problem is at least (4.21). Let $\{p_j^*(t,k') : j = 1, \ldots, m\}$ denote an optimal solution to the optimization problem, for all t and k'.

Proposition 4.5.1. *Consider the online algorithm which, at each time step t, sets the price randomly according to probabilities $\{p_j^*(t,k') : j = 1, \ldots, m\}$, where k' is the remaining* *inventory at the start of time t. Then for any sequence of valuation distributions* $\mathbf{v}_1, \ldots, \mathbf{v}_T$, *the algorithm's total expected revenue is at least* $\mathbb{E}_{V_1 \sim \mathbf{v}_1, \ldots, V_T, \sim \mathbf{v}_T}[\mathsf{OPT}(V_1, \ldots, V_T)]$.

Proposition 4.5.1 is established in the same way as Theorems 4.2.8 and 4.3.3—for $t = 1, \ldots, T$, we can inductively ensure from constraint (4.22) that the distribution for the starting inventory level matches that of $\mathcal{A}$. Since the algorithm has the same distribution for inventory state at each time t and earns at least as much revenue as $\mathcal{A}$ in expectation on every inventory state, its total revenue must be at least the offline optimum (and in fact is often much better since it exploits personalization).

4.6 Conclusion

In this chapter we have provided a general solution to the single-leg revenue management problem which yields the best-possible competitive ratio with limited demand information. Our policies unify the inventory-dependent booking policies in Ball and Queyranne (2009) with the random price-skimming policies in Eren and Maglaras (2010). An important feature of our policies is that they show at each time step how the *price distribution* should depend on inventory when the future is *unknown*, complementing classical results which show how the *optimal price* should depend on inventory when the future is *known*. Our policies were derived using a new "valuation tracking" technique, which geometrically tracks the optimum and hedges against the arrival sequence immediately ending *in the most inventory-conservative fashion.* We believe this to be of general interest for competitive ratio analysis.

Finally, we explain why our analysis of the pricing case, where each customer has a valuation and chooses the lowest fare not exceeding it, captures all *rational* choice models. Suppose instead that the firm could offer an *assortment* of fare classes, and that each customer has a ranked list of fare classes she is willing to purchase, and chooses the highest-ranked fare class that is offered to her. We can define V_t to be the maximum fare in the list that customer t is willing to purchase, and then the offline optimum would still be the k largest values from $V_1, \ldots, V_T$. Meanwhile, we can modify the online algorithm so that whenever it would have offered price P_t, it now shows all fares greater than or equal to P_t. This algorithm would still make a sale whenever $V_t \geq P_t$, except now it has the opportunity to earn revenue greater than P_t, if customer t does not choose the lowest offered fare. As a result, our $\mathsf{CR}(\mathcal{P})$-competitive algorithms under the pricing model imply corresponding $\mathsf{CR}(\mathcal{P})$-competitive algorithms under the assortment model.

Nevertheless, we would like to end on two open questions related to the assortment generalization. First, our argument above assumes rational choice models; however in practice certain fare classes could be designed as "decoys" for other fare classes. Second, our algorithms imply an "assortment-skimming" distribution over revenue-ordered assortments, but this assumes there is no limit on the number of fare classes offered. We believe that assortment skimming under cardinality constraints is an interesting problem.

Acknowledgments

The authors would like to thank He Wang and Costis Maglaras for insightful discussions.

Chapter 5

Learning Valuation Distributions from Bundle Sales

Bundling has long been used as a form of price discrimination, allowing for items to be sold at a discount when they are purchased together with others. In this chapter, we show that bundling has the added benefit of leading to richer sales data, providing more information about customer demand than individual sales numbers.

We introduce the "reverse" bundling problem of reconstructing the valuation distributions which would fit a set of bundle sales numbers. We show how to solve this fitting problem for a basic class of customer valuation models from the bundling literature, via an iterative algorithm with guaranteed convergence. An important insight from fitting this simple model is that an item's *price elasticity* can be measured by comparing its sales rates inside and outside of bundles.

Finally, we validate this insight on data from a large online retailer, where indeed, the price elasticities of items, as indicated by their sales spikes after a Black Friday markdown, are consistent with their bundle sales before Black Friday. We believe our method to be especially useful in designer fashion, where new products with bundle discounts are often introduced, and little would be known about price elasticity otherwise.

5.1 Introduction

Consider a firm offering an item at a fixed price of \$10. Arriving customers purchase the item if and only if their *valuation*, or maximum willingness-to-pay, for the item is at least

\$10. The firm would like to learn about its customers' valuations. For example, if half of the customers bought the item, then the firm could deduce that half of the valuations must be higher than \$10. However, this reveals nothing about how far away from \$10 the customer valuations spread. That is, the firm cannot deduce the item's *price elasticity*, e.g. how a price drop to \$8 would increase the proportion of customers with valuations exceeding price, from a single sales observation at \$10.

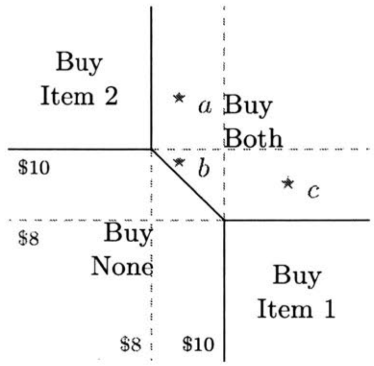
Now suppose instead that the firm offered two different items at \$10 each, and also offered them as a *bundle* at the discounted price of \$18. Under this pricing scheme, a customer who values the items at \$9 and \$11 respectively would purchase the bundle, despite valuing the 1st item below \$10, because given that she is certainly going to buy the 2nd item, she can add the 1st item for \$8 by buying the bundle. This customer is labeled as "a" in Fig. 5-1a. Similarly, customers b and c, who would have purchased nothing and item 1 respectively without the discount, would also be swayed into purchasing the bundle.

By observing sales numbers from this bundle pricing scheme (\$10 for each item; \$18 for both), the firm can decode information about the price elasticities. For example, suppose the firm observed that all customers purchase either both items or none, with 50% selecting each option, as in Fig. 5-1b. This suggests that the valuations below \$10 are tightly concentrated between \$8 and \$10, since whenever a customer would have purchased just one item, she was always willing to add the other for the discounted price of \$8. By contrast, suppose the firm observed that each of the four decision options (buy none, buy item 1, buy item 2, buy both) occur similarly often, around 25% of the time, as in Fig. 5-1c. This suggests that the valuations spread far away from \$10, since the bundle discount is not significant enough to discourage customers from buying just one item.

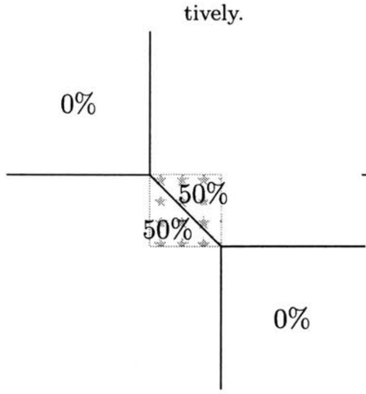
Of course, the deductions in Fig. 5-1 are based on specific modeling assumptions, namely that customers: (i) want at most one of each item, (ii) value the bundle at the sum of individual valuations, and (iii) have independent item valuations. Nonetheless, the key insight is that bundle sales numbers generally reveal more information about customer valuations, and most importantly, *reveal information about price elasticities that individual sales numbers cannot.*

In this chapter, we explore this phenomenon and introduce a new Learning from Bundle Sales problem, which complements the classical Bundle Pricing problem.

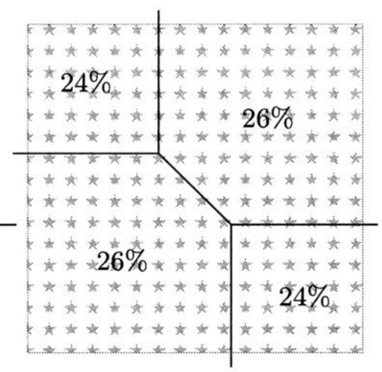
- **Bundle Pricing**: given the distribution of customer valuations, and knowing that



(a) Customer valuations divided into regions based on purchase decision. The x-axis and y-axis represent a customer's valuations for items 1 and 2, respectively.



(b) Sales numbers show that the bundle discount is extremely significant. Indicator of tightly concentrated valuations, and hence high price elasticities.



(c) Sales numbers show that the bundle discount is relatively insignificant. Indicator of widely spread valuations, and hence low price elasticities.

Figure 5-1: How to use bundle sales to speculate on price elasticities

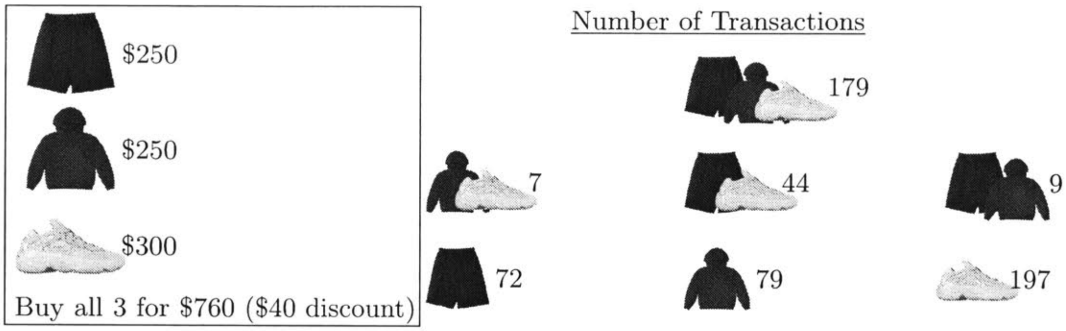


Figure 5-2: The pricing scheme used in the debut of Adidas Yeezy Season 6, with synthetic sales numbers
This data is compiled by counting, for each subset of items, the number of transactions that (out of the three items in question) contain exactly that subset.

each customer buys the subset of items maximizing her *surplus* (valuation minus price), how do we systematically price the subsets to maximize expected revenue?

- **Learning from Bundle Sales**: given for each non-empty subset of items the price and the number of customers who bought it, what can we deduce about the distribution of valuations?

The Learning from Bundle Sales problem requires only aggregate sales numbers, as shown in an example in Fig. 5-2, which are naturally collected through a retailer's operations. Therefore, it is applicable in any industry where discounted bundles are sold, like designer fashion, budget airlines (where ancillary services are bundled), and fast food (where combo meals are offered).

Motivated by the data in Fig. 5-2, we fit a basic class of valuation models where customers are *unit-demand* with *additive* and *independent* valuations, as in assumptions (i)–(iii) above. These assumptions allow for identifiability from the limited data in Fig. 5-2, and coincidentally, are consistent with the bundle pricing literature (see the section on Related Work). Under these assumptions, we separately divide each item's valuations into three regions: above individual price, below individual price but above "discounted price" (individual price minus the discount of $40), and below discounted price. We then identify the distribution of customers over these regions for each item, by fitting to the bundle sales numbers.

The solution to our fitting problem is shown in Fig. 5-3. It can be thought of as

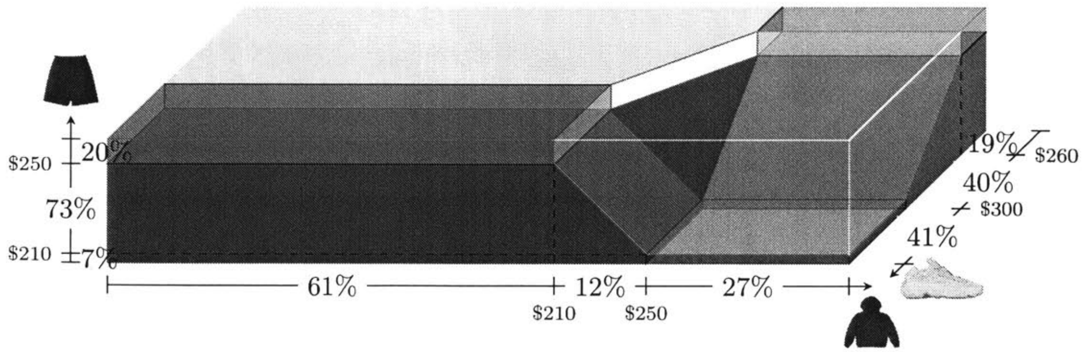


Figure 5-3: Estimated breakdown of customer valuations, if our model and fitting method are used
Each of the 7 non-black regions corresponds to one of the 7 sales numbers from Fig. 5-2. For example, the transparent white region corresponds to the 179 customers who bought the entire bundle.

the rectangular prism with 7 parameters (start and end for each axis, plus one density parameter) which best fits the 7 sales numbers in Fig. 5-2. This rectangular prism is narrow along both the shorts and sneakers axes, implying that these items have concentrated valuations and high price elasticities. It was fit into this shape because only 7 (resp. 9) people bought the *other* two items in the bundle without the shorts (resp. sneakers), i.e. very few people had valuations below \$210 (resp. \$260) which would cause them to decline a near-complete bundle in this case. By contrast, 44 people declined the hoodie despite buying the shorts and sneakers, implying that it has a low price elasticity.

In Section 5.2, we formalize our fitting problem, present our iterative algorithm for solving it, establish guaranteed convergence, and test numerical accuracy under violated model assumptions.

In Section 5.3, we test our insights on data from an industry collaboration with a large Latin American online retailer. We consider their home and kitchen items, which are sold in many discounted bundles, and measure each item's price elasticity by the magnitude of its sales spike after a Black Friday markdown. We find that indeed, these price elasticities are consistent with those indicated by the bundle sales before Black Friday. Furthermore, the best indicator of an item's price elasticity is the number of people who purchased the *other* items in its bundles, exactly in alignment with the intuition from Fig. 5-3.

In the Conclusion, we discuss the limitations in using our restricted class of valuation distributions to model price elasticities. But even with these limitations, the data from our industry collaborator suggests, that while our assumptions may be violated, our insights are still useful. Therefore, they should be incorporated even when additional information beyond Fig. 5-2 is available and richer classes of valuations can be identified—this is a vast direction for future research. The limited data described in Fig. 5-2 is commonly encountered by online retailers (like our industry collaborator) and designer fashion brands (like Yeezy) who cycle through new products and offer package discounts on them. In these situations, we have provided the first scientifically-rigorous method for computing price elasticities, allowing managers to make better product segmentation and pricing decisions in the future.

5.1.1 Related Work

To the best of our knowledge, we are the first to consider the learning from bundle sales problem that we formulated. The classical bundling problem focuses on price optimization, and various questions have been studied under a plethora of customer valuation models over the past decades—we refer to some recent papers (Abdallah, 2016; Chen et al., 2017) as well as Chapter 6 for a comprehensive literature review. The primary model of analytical interest in this chapter (single bundle discount, additive and independent valuations) is parsimonious in that it has few enough parameters to be identifiable from a single set of sales numbers, yet captures some notion of price elasticity in these parameters. It originated from Adams and Yellen (1976), and is a pioneering model in the bundling literature.

The general problem of modeling multi-item and bundle valuations has been studied in many areas, including discrete-choice modeling (Chung and Rao, 2003), conjoint analysis (Jedidi et al., 2003; Toubia et al., 2003), learning in auctions (Buchbinder and Naor, 2009), and copula inference (Letham et al., 2014), to list a few references. The models considered are richer than ours, but all require additional data to be identifiable, such as sales observations under different prices, item covariates, personalized sales records, survey details, or repeated queries to the same customer.

5.2 Model for Customer Valuations

A firm has n different items for sale, referenced by the set $[n] = \{1, \dots, n\}$. Each customer wants at most one of each item, and knows her valuation $v(S)$ for every subset S of items. The firm posts a price $P(S) \geq 0$ for every subset $S \subseteq [n]$, with $P(\emptyset) = 0$. A customer buys the subset S which maximizes[1] her *surplus*, defined as $v(s) - P(S)$. v is treated as random and drawn from a distribution D, which represents the valuation functions of the population. The firm observes the purchase decisions of customers with valuations drawn IID from D and counts the number who selected each subset.

Throughout the analysis in this chapter, we will assume that valuations are:

- Additive, i.e. each customer has atomic item valuations $x_1, \dots, x_n \geq 0$ and $v(S) = \sum_{i \in S} x_i \ \forall S$;
- Independent, i.e. D is a product distribution $D_1 \times \dots \times D_n$ and x_i is drawn independently from the marginal distribution D_i for all $i \in [n]$.

As discussed in the Introduction, combinatorial or correlated valuations introduce too many degrees of freedom and such a model cannot be identified when the input consists of merely 2^n sales counts.

In this section, we will further assume that:

- There is a single bundle discount, given for buying *all* the items in question, i.e.

$$P(S) = \begin{cases} \sum_{i \in S} P_i, & S \neq [n] \\ \sum_{i=1}^{n} P_i - d, & S = [n] \end{cases}$$

  where $P_1, \dots, P_n \geq 0$ are individual prices and $d \in (0, \sum_{i=1}^{n} P_i]$ is the discount;

- The firm observes the number of no-purchases;
- $n \geq 3$, i.e. the firm has at least $2^3 = 8$ sales counts to work with.

We relax these assumptions in Section D.4, where we also allow P to take the form of a *two-part tariff*, develop the setting where no-purchases are unobserved, and solve the $n = 2$ case by brute force. The proofs of Lemmas in this section are straight-forward and deferred to Section D.1.

[1] For convenience, we break ties arbitrarily; this can be achieved by small perturbations in P.

5.2.1 Formulation of Fitting Problem

Definition 5.2.1. For all $S \subseteq [n]$, let $\hat{p}_S$ denote the empirical probability of subset S being selected by a customer (these can be calculated by the firm assuming that no-purchases are observed).

Our goal is to compute empirical estimates of

$$\Big\{ \Pr[x_i < P_i - d],\ \Pr[P_i - d \leq x_i < P_i],\ \Pr[P_i \leq x_i] :\ i \in [n] \Big\} \tag{5.1}$$

which best fit $\{\hat{p}_S : S \subseteq [n]\}$ under the data generating process defined above. These estimates, which sum to unity for each i, would provide two quantiles of each item's valuation distribution D_i, which correspond to immediate sales rate and price elasticity. They can also be used to, e.g., reconstruct a Uniform or Normal demand curve with two parameters.

Definition 5.2.2. For all $i \in [n]$, let:

- $q_i^* = \Pr[x_i \geq P_i]$, the probability of a customer being willing to buy item i at individual price P_i;
- $a_i^* = \Pr[x_i \geq P_i - d | x_i < P_i]$, the probability of a customer being willing to buy item i at price $P_i - d$, conditioned on her *not* being willing to buy item i at price P_i.

It is easy to see that given q_i^* and a_i^* for all i, the probabilities in (5.1) can be derived. Henceforth, we will focus on the problem of estimating q_i^* and a_i^*.

Lemma 5.2.3. *For a customer with valuations $x_1, \ldots, x_n$, the surplus-maximizing subset is $[n]$ iff*

$$\sum_{i=1}^{n} \max\{P_i - x_i, 0\} \leq d. \tag{5.2}$$

We can interpret $\max\{P_i - x_i, 0\}$ as the *deficit* incurred by the customer had she been forced to buy item i at price P_i, which is 0 if $x_i \geq P_i$. This allows us to characterize the customer's purchase decision. Indeed, she first checks inequality (5.2), i.e. whether her total deficit from buying all the items is covered by the bundle discount. If so, then she buys all the items. Otherwise, she individually selects the items i for which her valuation is at least the individual price P_i, and buys this subset.

Lemma 5.2.4. *For all $S \subsetneq [n]$, the probability of a customer selecting subset S is*

$$\Big(\prod_{i\in S} q_i^*\Big)\Big(\prod_{i\notin S}(1-q_i^*)\Big)\Big(1-\Pr\big[\sum_{i\notin S}(P_i-x_i)\le d\big|P_i-d\le x_i<P_i\ \forall i\notin S\big]\prod_{i\notin S}a_i^*\Big). \quad (5.3)$$

Intuitively, the three parentheses in expression (5.3) can be interpreted as follows. For a subset $S \neq [n]$ to be selected, the customer has to: (i) value the items $i \in S$ above their individual prices; (ii) value the items $i \notin S$ below their individual prices; and (iii) not prefer the full bundle over S, i.e. incur greater than d deficit from buying the items $i \notin S$.

With Lemma 5.2.4, our fitting problem can be defined as solving a system based on (5.3).

Definition 5.2.5. [Fitting Problem] For all $S \neq \emptyset$, let F_S denote an approximation used for the value of $\Pr\big[\sum_{i\in S}(P_i-x_i)\le d\big|P_i-d\le x_i<P_i\ \forall i\in S\big]$. Then the fitting problem is to solve for $q_1,\dots,q_n$ and $a_1,\dots,a_n$ from the following system of equations:

$$\Big(\prod_{i\in S} q_i\Big)\Big(\prod_{i\notin S}(1-q_i)\Big)\Big(1-F_{[n]\setminus S}\prod_{i\notin S}a_i\Big)=\hat{p}_S \qquad \forall S\neq[n]. \quad (5.4)$$

For now, we focus on solving (5.4) given $\{F_S : S \neq \emptyset\}$, treating the values of F_S as fixed input parameters. The following lemma derives some baseline parameter values in $F_S = \frac{1}{|S|!}$ using the *uniform* distribution; we will see that they generally provide a good approximation.

Lemma 5.2.6. *Let $S \neq \emptyset$ and suppose that for all $i \in S$, the conditional distribution of D_i on $[P_i - d, P_i)$ is uniform. Then* $\Pr\big[\sum_{i\in S}(P_i-x_i)\le d\big|P_i-d\le x_i<P_i\ \forall i\in S\big]=\frac{1}{|S|!}$.

5.2.2 Iterative Fitting Algorithm

The problem from Definition 5.2.5 involves an intractable system of equations with degree-$(2n)$ polynomials; even if $n = 3$, there appears to be no closed-form solution. Furthermore, the given values of $\hat{p}_S$ are noisy observations and the given values of F_S are approximations, potentially leading to an inconsistent set of equations. To cope, we develop a method which fits the equations iteratively, and bound its error relative to the error in $\hat{p}_S$ and F_S. Its error approaches 0, i.e. our method recovers the true values of q_i^* and a_i^*, as the errors in both $\hat{p}_S$ and F_S approach 0.

For $i = 1, \ldots, n$ initialize

$$q_i^{(0)} \leftarrow \Big(1 + \Big(\prod_{S \in \mathcal{S}_i} \frac{\hat{p}_{S \cup \{i\}}}{\hat{p}_S}\Big)^{-1/|\mathcal{S}_i|}\Big)^{-1}. \tag{5.6}$$

for $k = 0, 1, \ldots$ **do**

For $i = 1, \ldots, n$ set

$$a_i^{(k)} \leftarrow \max\Big\{1 - \frac{\hat{p}_{[n]\setminus\{i\}}}{(\prod_{j \neq i} q_j^{(k)})(1 - q_i^{(k)})}, 0\Big\}. \tag{5.7}$$

For $i = 1, \ldots, n$ set

$$q_i^{(k+1)} \leftarrow \Big(1 + \Big(\prod_{S \in \mathcal{S}_i} \big(\frac{\hat{p}_{S \cup \{i\}}}{\hat{p}_S} \cdot \frac{1 - F_{[n]\setminus S} \prod_{j \notin S} a_j^{(k)}}{1 - F_{[n]\setminus S \setminus \{i\}} \prod_{j \notin S, j \neq i} a_j^{(k)}}\big)\Big)^{-1/|\mathcal{S}_i|}\Big)^{-1}. \tag{5.8}$$

end for

Figure 5-4: Iterative algorithm, on input $\{\hat{p}_S : S \neq [n]\}$, $\{F_S : S \neq \emptyset\}$, and $\{\mathcal{S}_i : i \in [n]\}$

Our iterative algorithm is motivated by the following. If we take any subset S, and compare equation (5.4) for S and $S \cup \{i\}$, then supposing that $i \notin S$ and $S \cup \{i\} \neq [n]$, we have

$$\frac{\hat{p}_{S \cup \{i\}}}{\hat{p}_S} = \frac{q_i}{1 - q_i}\Big(\frac{1 - F_{[n]\setminus S \setminus \{i\}} \prod_{j \notin S, j \neq i} a_j}{1 - F_{[n]\setminus S} \prod_{j \notin S} a_j}\Big). \tag{5.5}$$

That is, comparing $\hat{p}_{S \cup \{i\}}$ with $\hat{p}_S$ almost lets us compute q_i (the sales rate of item i at individual price P_i), except for the term in parentheses which is a bias caused by the bundle discount.

Nonetheless, we can ignore this term to get an initial solution for $(q_1, \ldots, q_n)$. Substituting these values into the equations in (5.4), we can solve for $(a_1, \ldots, a_n)$. We then use these values of a_i to update our solution for $(q_1, \ldots, q_n)$, according to (5.5). By repeating this procedure which alternates between updating $(q_1, \ldots, q_n)$ and updating $(a_1, \ldots, a_n)$, the algorithm is able to account for the bundle discount and correct the bias in its initial estimates—see Fig. 5-4 for its pseudocode.

Instructions (5.6) and (5.8) take an additional input parameter $\mathcal{S}_i$ which can be set. $\mathcal{S}_i$ denotes the collection of subsets S for which the algorithm will combine equation (5.5) to solve for q_i. One useful insight is that the bias term in (5.5) is least significant when

$S = \emptyset$, because this is when the customer is furthest from being affected by the bundle discount.[2] However, setting $\mathcal{S}_i$ to contain more sets than $\emptyset$ can reduce the error from the noise in $\hat{p}_S$. Meanwhile, instruction (5.7) computes a_i, the term corresponding to the price elasticity of item i, using the purchase probability of subset $[n] \setminus \{i\}$. This provides another useful insight: item i's price elasticity is determined by the number of people who bought the other items in the bundle, yet *refused* to add item i to unlock the discount.

5.2.3 Convergence Guarantee

Our algorithm outputs a sequence of estimates $\left(q_i^{(k)}\right)_{k=0}^{\infty}$ and $\left(a_i^{(k)}\right)_{k=0}^{\infty}$ for each q_i^* and a_i^*, respectively. The error in $q_i^{(k)}$ and $a_i^{(k)}$ depends on the error in the values of $\hat{p}_S$ and F_S that were input to the algorithm. Here we state our convergence result in the noise-free, uniform special case, where there is no error in either $\hat{p}_S$ or F_S. Our general analytical results, with noisy observations and arbitrary distributions, are deferred to Section D.2.

Theorem 5.2.7. *Suppose that for all $S \neq [n]$, $\hat{p}_S$ equals the true probability (5.3) of subset S being selected. Furthermore suppose that for all $i \in [n]$, D_i is a uniform distribution and probabilities q_i^* and a_i^* lie in $(0,1)$. Then our algorithm in Fig. 5-4, with parameters $F_S = \frac{1}{|S|!}$ $\forall S$ and $\mathcal{S}_i = \{\emptyset\}$ $\forall i$, does not divides by zero and correctly solves system (5.4), identifying the model parameters with*

$$\lim_{k\to\infty} q_i^{(k)} = q_i^* \qquad \textit{and} \qquad \lim_{k\to\infty} a_i^{(k)} = a_i^* \qquad \forall i \in [n].$$

Theorem 5.2.7 is a corollary of the more general Theorem D.2.16 (resp. Theorem D.2.15) from Section D.2, which bounds for all i the multiplicative error between $q_i^{(k)}$ and q_i^* (resp. $a_i^{(k)}$ and a_i^*) as a function of k, the maximum multiplicative error in the values of $\hat{p}_S$, and the maximum multiplicative error in the values of F_S. This error bound approaches 0 as $k \to \infty$ (at the rate of exponential decay) and the errors in $\hat{p}_S$ and F_S both approach 0.

5.2.4 Testing Algorithm on Synthetic Data

In Section D.3, we test our algorithm in settings with error in the values of $\hat{p}_S$ and non-uniform distributions, where Theorem 5.2.7 would not guarantee convergence to the correct

[2]For a rigorous explanation of this fact, consider the setting where $F_S = \frac{1}{|S|!}$ for all S. The bias term equals $(1 - \frac{1}{a_i(n-|S|-1)!}\prod_{j\notin S} a_j)(1 - \frac{1}{(n-|S|)!}\prod_{j\notin S} a_j)^{-1}$, which is less than 1 and decreasing over $S \subsetneq ([n] \setminus \{i\})$. Therefore, the bias is more significant for larger subsets S.

underlying parameters. We have followed the setup from Chu et al. (2011), and summarize our main findings below:

- By iterating our algorithm with a simple stopping criterion based on comparing $a^{(k)}$ to $a^{(k-1)}$, approximately-correct parameters can be reached using less than 10 iterations on average;
- The error due to noise in $\hat{p}_S$ can be mitigated by increasing the cardinality of parameter $\mathcal{S}_i$;
- The error due to the distribution being non-uniform is less than 2%, as long as the distribution is *unimodal*. We provide an explanation for this surprising accuracy in Section D.3.4.

5.3 Testing Model on Data from Online Retailer

In this section we test how the insights from our model translate to real-world price elasticities.

We consider sales data provided by a large Latin American online retailer, on their home and kitchen items, many of which are sold in discounted bundles. For each item and bundle, we are given its price and sales for 26 weeks starting June 1st, 2016. The last of these 26 weeks contains both Black Friday (November 25th, 2016) and the Cyber Monday following it. This is a special promotional period during which many items are heavily marked down.

We use an item's sales boost after a Black Friday markdown to measure its price elasticity, and test whether this agrees with the elasticity indicated by its bundle sales prior to Black Friday. As an example, consider Table 5.1. Having only seen the sales of the items at a fixed price, can we use their bundle sales to speculate on their sales boosts after a Black Friday markdown?

A naive observation of the data might suggest that neither item is particularly elastic, because the discounted bundle did not sell many copies (only 29) relative to the weekly sales of the items (222 and 30, respectively). However, the analysis in our chapter would suggest that the recipe book is *much less elastic* than the pressure cooker. This is because a large number of people (222) paid \$207 for the pressure cooker alone, declining to add the recipe book at the highly discounted price of \$5 by paying \$212 for the bundle instead.

Table 5.1: An example from the data. Can we speculate on the "?" entries using only the information in the table? (Answers below.)

	First 25 Weeks		Black Friday Week	
	Price in R$	Avg. Sales/wk	Price in R$	Sales
Pressure Cooker	$207	222	$190 ($17 markdown)	?
Recipe Book	$14	30	$3 ($11 markdown)	?
Bundle	$212 ($9 discount)	29	$212 (dominated by markdowns)	

By contract, much fewer (30) people purchased the recipe book alone, usually wanting it as part of a bundle.

In the end, indeed, the recipe book sold only 176 copies during Black Friday week, representing a dismal price elasticity for a massive 79% Black Friday markdown. Meanwhile, the pressure cooker sold 1153 copies during Black Friday week, achieving a similar sales increase from a much smaller 8% markdown and hence representing a much better price elasticity.

5.3.1 Relationship between Bundle Sales and Price Elasticity

We test whether the relationships from the example in Table 5.1 generalize to the other items in the data set. Most of the items have been offered in multiple bundles, so for each item, we consider aggregate statistics over the bundles containing it.

Definition 5.3.1. Define the following for each item i, based on the 25 weeks before Black Friday.

- $\mathsf{PctBund}(i)$: out of all copies of i sold, the fraction of which came from a bundle purchase.
- $\mathsf{PctBundPartners}(i)$: the average value of $\mathsf{PctBund}$ over the *partner* items of i, where an item j is considered a partner of i if there is some bundle containing both i and j.
- $\mathsf{AvgBundDisc}(i)$: the % discount provided by the bundles containing i (taking a weighted average).

For full details on how we processed these statistics from the data, we defer to Section D.5.

To illustrate these statistics, we calculate them for the items in Table 5.1. In the full data, the (pressure) cooker was only part of the bundle with the (recipe) book, while the

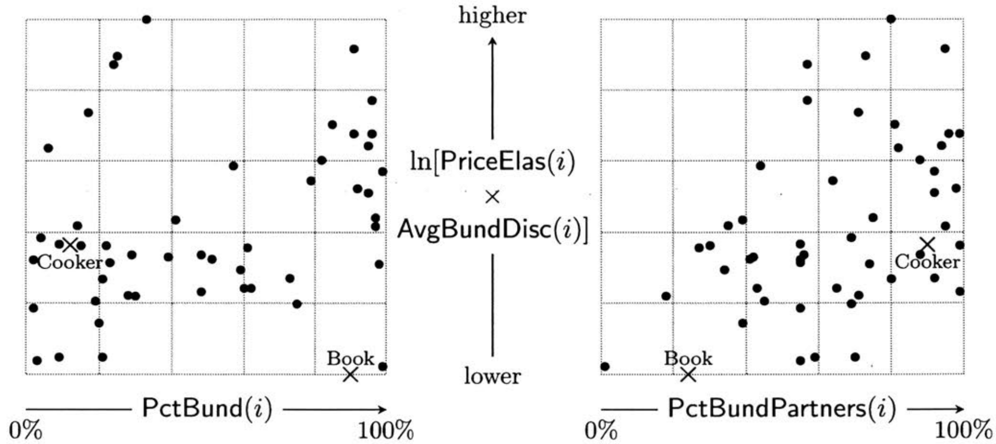


Figure 5-5: Relationship between bundle sales and price elasticity. The items from Table 5.1 are labeled "x".

book was part of many bundles with different kitchen appliances. An additional 235 copies of it were sold weekly in these other bundles. Therefore,

$$\mathsf{PctBund}(\text{Cooker}) = \frac{29}{29+222} \approx 12\% \quad \text{and} \quad \mathsf{PctBund}(\text{Book}) = \frac{29+235}{29+30+235} \approx 90\%.$$

The cooker's sole partner is the book, so $\mathsf{PctBundPartners}(\text{Cooker}) = \mathsf{PctBund}(\text{Book}) = 90\%$. Meanwhile, $\mathsf{PctBundPartners}(\text{Book}) = 24\%$ after averaging over all of its partners. Finally, AvgBundDisc considers the % discounts provided by the bundles, which for this one is $1 - \frac{212}{207+14} \approx 4\%$. Hence $\mathsf{AvgBundDisc}(\text{Cooker}) = 4\%$. $\mathsf{AvgBundDisc}(\text{Book})$ takes a weighted average over all of the bundles containing the book, ending up at 8%.

For each item i, we let $\mathsf{PriceElas}(i)$ denote its price elasticity, measured as its % increase in sales during Black Friday week, divided by its % decrease in individual price. We are interested in the relationship of $\mathsf{PriceElas}(i)$ with both $\frac{\mathsf{PctBund}(i)}{\mathsf{AvgBundDisc}(i)}$ and $\frac{\mathsf{PctBundPartners}(i)}{\mathsf{AvgBundDisc}(i)}$, where we have divided by the bundle discount, to penalize bundle sales percentages that came from highly discounted bundles.

For the 51 items which were marked down during Black Friday and had bundle sales prior to that, we plot $\ln[\mathsf{PriceElas}(i) \cdot \mathsf{AvgBundDisc}(i)]$ vs. both $\mathsf{PctBund}(i)$ and $\mathsf{PctBundPartners}(i)$, in Fig. 5-5. We will not attempt to express the y-axis on an absolute scale, and instead focus on relative magnitudes. Accordingly, we will use Kendall's τ coefficient to measure correlation with the x-axis.

In Fig. 5-5, $\mathsf{PctBund}(i)$ demonstrates a noticeable correlation with price elasticity, with

$\tau = .16$ (i.e. 58% of the pairs were concordant), which for 51 data points results in a two-sided p-value of .08. However, the correlation is weak with many outliers, as exemplified by the recipe book.

Meanwhile, $\mathsf{PctBundPartners}(i)$ demonstrates a highly significant correlation with price elasticity, with $\tau = .3$ and $p = .002$. This is exactly as explained by our model: if many people bought the *partners* of item i without adding i at a bundle discount, then they also wouldn't buy item i after an individual discount; on the other hand, if nobody bought the partners of item i individually, then a logical hypothesis is that everybody was willing to add i to the bundle.

It is also interesting to note that in Fig. 5-5, the uncertainty for a given value of $\mathsf{PctBundPartners}(i)$ appears to be one-sided. That is, a high $\mathsf{PctBundPartners}(i)$ could still imply a low price elasticity, likely caused by bundle sales that were driven by highly complementary items. However, a low $\mathsf{PctBundPartners}(i)$ never implied a high price elasticity, likely because substitutes being bundled together (which would cause this effect) rarely occurs in practice.

5.4 Conclusion

In this chapter we explore the phenomenon of bundle sales data containing richer information about customer valuations. We show how to fit a fundamental demand model from bundling theory to bundle sales data, thereby also providing the first method of estimating price elasticities under very limited data (no price changes, covariate information, etc.). We verify the existence of our phenomenon on a real-world data set, and specifically validate our theoretically-backed insight that the best indicator of an item's price elasticity is the sales of the *partner* items it is bundled with.

Our model emphasizes simplicity and identifiability from limited data, but does not capture aspects such as complements and substitutes. Nonetheless, on real data where its assumptions may be violated, we saw from Fig. 5-5 that it still provides a logical method of comparing price elasticities between items, which is correct significantly more often than not. We also saw from Fig. 5-5 that our model can be improved by identifying complementary items (as opposed to substitutes). In situations where this is possible because additional data is available, we believe that vast future research can be done, in analyzing higher-parameter models which incorporate our insights.

Chapter 6

Reaping the Benefits of Bundling under High Production Costs

It is well-known that selling different goods in a single bundle can significantly increase revenue, even when the valuations for the goods are independent. However, bundling is no longer profitable if the goods have high production costs. To overcome this challenge, we introduce a new mechanism, Pure Bundling with Disposal for Cost (PBDC), where after buying the bundle, the customer is allowed to return any subset of goods for their production cost. We derive both distribution-dependent and distribution-free guarantees on its profitability, which improve previous techniques. Our distribution-dependent bound suggests that the firm should never price the bundle such that the profit margin is less than 1/3 of the expected welfare, while also showing that PBDC is optimal for a large number of independent goods. Our distribution-free bound suggests that on the distributions where PBDC performs worst, individual sales perform well. Finally, we conduct extensive simulations which confirm that PBDC outperforms other simple pricing schemes overall.

6.1 Introduction

We study the monopolist pricing problem of a firm selling n heterogeneous items. For each item, customers have a valuation, which is their maximum willingness-to-pay, drawn from a known distribution. A customer wants at most one of each item. The firm offers take-it-or-leave-it prices for every subset of items, and the customer chooses the subset maximizing her surplus (valuation for the subset minus price), with the no-purchase option always being available. We assume the customer's valuation for a subset is additive over the items in the set. The objective of the firm is to maximize expected per-customer revenue.

In the full generality of the problem, the firm has $2^n - 1$ prices to set. However, it is important to find profitable yet *simple* pricing schemes that are determined by a small number of prices. Two such schemes are *Pure Components* (PC), where items are priced separately (and the price of a subset is understood to be the sum of its constituent prices), and *Pure Bundling* (PB), the strategy of only offering all the items together. A third scheme that generalizes both PC and PB is *Mixed Bundling* (MB), which offers individual item prices as well as a bundle price for all the items. MB can be seen as a form of price discrimination, where customers who highly value an item can buy it for its individual price, while customers with lower valuations still have a chance of buying it as part of a discounted bundle.

The efficacy of simple pricing schemes is of immense importance in retail, and has been studied over the past few decades in the economics literature, the operations research/marketing interface literature, and more recently, the computer science literature. For a single item, the solution is immediate: choose the price p maximizing $p(1 - F(p))$, where F is the CDF of the valuation (see Myerson (1981)). However, for two items, even if their valuations are independent, bundling can be better than individual sales.

For example, suppose we have two products with IID valuations, each of which is 1 half the time, and 2 half the time. If we sell the items individually, we can always get a sale for 1, or get a sale half the time for 2. In either case, the combined expected revenue is 2. However, if we sell the items as a bundle for 3, then the bundle will be purchased $\frac{3}{4}$ of the time, yielding an expected revenue of $\frac{9}{4}$.

The key observation is that the valuation of the bundle is more concentrated around its mean than the valuation of the individual items, which causes less consumer heterogeneity, and we can choose a price that is the highest willingness-to-pay for a larger fraction of customers. This makes it easier to reduce deadweight loss, which is revenue lost from pricing a customer with positive valuation out of the market, and consumer surplus, which is revenue lost from offering a customer a better price than necessary.

The power of bundling is even greater when valuations are negatively correlated—consider two products with marginal valuations that are uniform on $[0, 1]$ but correlated in a way such that they always sum to 1. In this case, offering the bundle at the price of

1 will always get a sale, extracting the entire welfare, while selling the items individually yields at most $\frac{1}{2}$, half the available surplus. These effects have long been known in the economics literature, following the pioneering work of Stigler (1963), Adams and Yellen (1976), Schmalensee (1984), and McAfee et al. (1989).

Of course, bundling is not always superior to individual sales—this is especially true once we consider production costs. For example, suppose we have two goods with IID valuations that are uniform on $[0, 3]$, but each cost 2 to produce. Selling them individually at price $\frac{5}{2}$ will yield a profit of $\frac{1}{12}$ per item and is better than selling them as a bundle—these are low-profit-margin items that are only valuable to a small fraction of the population, and by bundling them we may force a customer into consuming a good for which her valuation is less than the production cost.

Over the decades, a lot of work has been done to compare the profit of Pure Bundling versus Pure Components. Adams and Yellen (1976) write, "The chief defect of Pure Bundling is its difficulty in complying with Exclusion," where Exclusion refers to the principle that a transfer is better off not occuring when the consumer's valuation is below the producer's cost. It is observed in Schmalensee (1984) for the case of bivariate normal valuations that Pure Bundling is better when mean valuations are high compared to costs. Bakos and Brynjolfsson (1999) prove that bundling a large number of goods can extract almost all of the available surplus, but this is crucially dependent on the items being "information goods", i.e. goods with no production costs. Fang and Norman (2006) characterize conditions under which Pure Bundling outperforms Pure Components for a fixed number of items, and all of their conditions imply low costs. Li et al. (2013) define a measure of consumer heterogeneity that increases with costs, and present computational results showing Pure Bundling performs poorly relative to Pure Components as their measure of consumer heterogeneity increases.

The indisputable conclusion from all this work is that high costs are the greatest impediment to the magic of bundling. However, we argue that there is a simple way to enjoy the effects of bundling while allowing for the flexibility of individual sales—sell all of the items as a bundle, but allow the customer to return any subset of items for a refund equal to their total production cost. We call this scheme *Pure Bundling with Disposal for Cost* (PBDC). It is a strict improvement of Pure Bundling where the customer's valuations that were below the cost have been *truncated* to equal the cost. Meanwhile, the firm is indifferent between producing an item for its cost or returning its cost to the customer, but PBDC makes it easier to sell the bundle because customers with low valuations for specific items won't be priced out of the market.

Importantly, there is great flexibility in how to present PBDC to the customer in a transparent and attractive way. In fact, we show that PBDC has a few equivalent formulations which can already be seen in the market. One formulation is a *tariff* to enter the market, after which all products are sold at cost. Alternatively, PBDC can be introduced with an individual price for each item and a *per-item discount* for each item purchased beyond the first. From a marketing point of view, the tariff strategy is more attractive when the number of items is large, while the discount strategy is more attractive when the number of items is small.

Our scheme can be compared to that of Hitt and Chen (2005), who recognized the need for a middle ground between Pure Bundling and Pure Components. They introduced the scheme *Customized Bundling*, which prices each bundle based only on its size, and not which items are included. Chu et al. (2008)[1] perform extensive numerical experiments for the same scheme, calling it *Bundle-Size Pricing* (BSP), showing that it can extract 99% of the optimal profit in their simulations.

PBDC can be seen as orthogonal to BSP—while BSP imposes symmetric pricing across items but allows non-linear pricing based on quantity, PBDC allows asymmetric pricing across items based on cost but imposes additive pricing once the customer pays the tariff to enter the market. When all item costs are identical, PBDC is a simplified version of BSP, because instead of having n prices to decide, there is only one price to decide, be it thought of as the bundle price, the tariff, or the discount. However, since we are able to relate PBDC to Pure Bundling, it is much easier to analyze. Our work provides the first theoretical explanation for some of the empirical successes in Chu et al. (2008)—indeed, in their simulations, costs are either equal, or insignificant (equal to half of the product's mean valuation).

We present two types of theoretical bounds in this chapter. Both require that items have independent valuations, which is a standard and often necessary assumption in the bundling literature. Both also rely on a transformation from costs to negative valuations which as far as we know is new.

[1]See Chu et al. (2011) for the journal version.

First, we prove that PBDC, with an appropriately chosen bundle price, extracts the entire welfare as the number of items approaches infinity, so long as the valuations have uniformly bounded variances. This type of result is based on the law of large numbers, which says that the sum of cost-truncated valuations, which we denote with the random variable X, lies within $(1 \pm \varepsilon)\mathbb{E}[X]$ with high probability. Therefore, the bundle price of $(1-\varepsilon)\mathbb{E}[X]$ will be accepted by a $(1-\varepsilon)$-fraction of the customers, profiting almost the expected welfare, $\mathbb{E}[X] - C$ (C is the total cost of producing all the items), which is an upper bound on profit.

Bakos and Brynjolfsson (1999) have already proven this result for the case without costs, and Armstrong (1999) has proven this result for a cost-based two-part tariff which we show is equivalent to PBDC, so this result in itself is not new. However, our analysis introduces the use of Cantelli's stronger, one-sided concentration inequality in bundling, recovering previous bounds asymptotically and achieving a better bound when the coefficient of variation of X is large. In the latter case, both of the previous works recommend a bundle price of $C + \varepsilon(\mathbb{E}[X] - C)$, earning negligible profit, whereas our analysis never recommends a bundle price below $C + \frac{1}{3}(\mathbb{E}[X] - C)$ and guarantees a non-zero profit.

Furthermore, we recommend PBDC even when the number of items is small—the second type of theoretical bound we present is problem-independent, unaffected by the number of items or their variances. We prove that the expected profit of the best PBDC pricing is at least $\frac{1}{5.2}$ of the expected profit of the optimal mechanism, except in detectable pathological cases, where the best PC pricing achieves this guarantee. The benchmark in this theorem is the maximum expected profit that could be achieved from explicitly pricing all subsets of items[2]. This is a tighter benchmark than expected welfare, which could be infinite without distributional assumptions.

We use tools from the computer science literature to bound this benchmark, most notably from Babaioff et al. (2014), who prove in the costless case that the better of PB and PC earns at least $\frac{1}{6}$ of the optimal revenue. We improve their bound from $\frac{1}{6}$ to $\frac{1}{5.2}$ by using Cantelli's inequality, and enhancing their *core-tail decomposition* technique in analyzing the core and tail together. We also construct an example improving the upper bound from $\frac{12}{13} \approx \frac{1}{1.08}$ to $\frac{3+\ln 2}{3+2\ln 2} \approx \frac{1}{1.19}$, where the previous best-known example is from Hart and

[2]Technically, the optimal mechanism is also allowed to offer lotteries of items, which have been shown to be necessary for achieving the optimum, in Hart and Reny (2012).

Nisan (2012). Finally, we generalize the result of Babaioff et al. to the case with costs, where PBDC is needed instead of PB. We should point out that when the benchmark is the optimal mechanism, one cannot simply truncate all valuations from below by cost, because the optimal mechanism could exploit low valuations to reduce the cannibalization of higher-profit options. In general, profit is *non-monotone*, i.e. increasing customer valuations can decrease the optimal profit, as reported in Hart and Nisan (2012).

In addition to the theoretical considerations, we provide a continuation of the numerical experiments from Chu et al. (2008), extensively testing the efficacy of PBDC on a finite number of items. We use the same independent demand distributions with the same parameters as Chu et al. (2008). In their setting where costs are low and identical across items, PBDC is a special case of BSP. However, it still attains between 97.5% and 100% of the (nearly optimal) BSP profit. If we allow costs to vary and be more significant, then PBDC drastically outperforms other simple mechanisms (PC, PB, BSP), demonstrating its robustness under different scenarios. In fact, the worst case for PBDC is the aforementioned setting where it attains 97.5% of the profit of the best simple mechanism; contrast this with 79.9%, 16.8%, 59.5% for PC, PB, BSP, respectively, in their wost-case settings. In addition to being profit-maximizing, PBDC also achieves excellent global surplus in our simulations. Finally, we show that PBDC attains between 96.6% and 99.4% of the optimal profit (which prices all subsets) when $n = 3$, and scales well as n increases.

The general goal of our work is to dispel the myth that high costs should drive a firm away from bundling and toward individual sales. PBDC allows the firm to reap the benefits of bundling while preventing items from being consumed for utility below cost. We should point out that there do exist costless examples with independent valuations on which PBDC performs poorly relative to the optimal mechanism (which is why it is necessary to include PC in the statement of the second theoretical result). Here is a list, along with why PBDC (equivalent to PB) is ill-advised for each instance:

- Example 15 from Hart and Nisan (2012): there are various different valuations, each of which realizes to an exorbitant value with a small probability; bundling is ineffective because the probability that more than one valuation is non-zero is infinitessimal

- Examples 1 and 2 from Rubinstein (2016): there is a need to *partition* the items, i.e. split them into groups, and offer each group as a different bundle

- Example 6.4.2 from Section 6.4: there is a need to price-discriminate, i.e. offer high individual prices and a discounted bundle price

However, our numerical experiments demonstrate that over "average" instances, PBDC performs far better than these pathological constructions and the worst-case bound of $\frac{1}{5.2} \approx$ 19.2% suggest. Indeed, once PBDC has eliminated the effect of costs, selling everything under one bundle leaves very little to be desired.

6.1.1 Literature Review

Bundling has been the focus of many papers in three different disciplines: economics, computer science, and operations research/marketing science. In general, the literature can be classified into three categories: papers that provide insights, papers that suggest approximate algorithms with attractive worst-case bounds, and papers that develop computationally efficient algorithms. In this subsection we attempt to highlight the most important contributions to the bundling literature, independent of discipline.

Two Items. In the economics literature, the earliest recognition of bundling being able to increase the revenue from selling two items is usually attributed to Stigler (1963); other early research for two products includes Adams and Yellen (1976); Schmalensee (1984); McAfee et al. (1989). Since then, Venkatesh and Kamakura (2003); McCardle et al. (2007) have established conditions for bundling being optimal for two potentially correlated goods.

Simple Mechanisms. For more than two items, there is a great practical interest in finding simple pricing schemes that are both profitable and easy to explain to the customer; for surveys on how bundling has affected marketing practice see Stremersch and Tellis (2002); Venkatesh and Mahajan (2009). However, the only concrete, general pricing scheme we have found in this literature, other than the classical PC and PB, is the BSP proposed by Hitt and Chen (2005) and Chu et al. (2008). Our scheme, PBDC, attempts to add to this literature by providing a transparent, easy-to-compute heuristic.

Most of the attempts to prove that simple pricing schemes are indeed capturing most of the optimal profit have been restricted to special cases (see Manelli and Vincent (2006, 2007)), or empirical evidence, as in the case of BSP, where its great experimental success has been unexplained. That's where we turn to the computer science literature. There has been more general work on auctions with multiple buyers, or valuation functions where the valuation for a subset may not be additive over the items in the set, for which we refer to

Rubinstein and Weinberg (2015); Yao (2015) and the references therein. We focus on the case of a single buyer with additive valuations, which is the bundling problem.

In this special case, Hart and Nisan (2012) introduce performance guarantees for simple mechanisms, which are further studied in Hart and Reny (2012); Hart and Nisan (2013). One line of work (Li and Yuan (2013); Babaioff et al. (2014)) culminated in a proof that either PB or PC must be within $\frac{1}{6}$ of optimal, for arbitrary independent valuations. By relating PBDC to PB, and improving upon their techniques, we are able to prove that either PBDC or PC must be within $\frac{1}{5.2}$ of optimal for the independent case with costs. When all costs are equal, PBDC is a special case of BSP, so our work provides the first theoretical explanation for some of the successes in Chu et al. (2008).

Recently, mechanisms where items are partitioned before bundling have also been advocated as simple in Cai and Huang (2013); Rubinstein (2016). Our bound improves the theoretical guarantee for the partitioning scheme in Rubinstein (2016) from $\frac{1}{6}$ to $\frac{1}{5.2}$. The same core-tail decomposition of Babaioff et al. (2014) has also been recently used in Bateni et al. (2015); Yao (2015), so our new bound improves guarantees from those works as well.

Computational Solutions. Others have tried to tackle the problem with more items by giving up on simplicity and computing an explicit optimal or near-optimal solution using optimization techniques. A mixed integer programming formulation was first seen in Hanson and Martin (1990), and recently in the mechanism design literature, explicit polynomial-time solutions were provided via linear programming in Cai et al. (2012); Cai and Huang (2013).

As far as computing the optimal prices for simple mechanisms, Wu et al. (2008) use non-linear mixed integer programming to solve for the optimal BSP prices, while Rubinstein (2016) gives a PTAS for the optimal partitioning. Computation is another benefit of PBDC—like PB, it only requires calculating one price, which can be done via convolution.

Large Number of Items. Yet another line of work addresses the complexity of many items by claiming that PB is guaranteed to be optimal as the number of items approaches infinity, assuming independence and uniformly bounded variances. Traditionally, this line of work has dealt with information goods which have no marginal costs (see Bakos and Brynjolfsson (1999, 2000)), or showed that costs have a substantial effect on the efficacy of PB (see Fang and Norman (2006); Ibragimov and Walden (2010)). Armstrong (1999) advocates that the same result can be achieved with costs by using a cost-based two-part tariff, which we prove is equivalent to PBDC.

Our research strengthens this line of work by using Cantelli's one-sided concentration inequality to get a tighter problem-dependent bound. Furthermore, we advocate for PBDC even on a small number of items, both with our problem-independent bound, and our numerical experiments.

Closed-form Solutions. There is also interest in finding analytical closed-form solutions for the optimal pricing under simple cases of the problem. In the case of two independent valuations, one of which is uniform on $[0, b_1]$ and the other which is uniform on $[0, b_2]$, Eckalbar (2010) derives elementary equations for the optimal Mixed Bundling prices. Bhargava (2013) shows that the equations involve roots of high-degree polynomials once costs are introduced, and uses a linear approximation to record solutions. Our transformation in Section 6.2 shows that the problem with costs is equivalent to the problem for distributions uniform on $[a_1, b_1]$ and $[a_2, b_2]$, where a_1 and a_2 could be negative. The difficulty of analytical solutions in general is discussed in Wilson (1993); Armstrong (1996); Prasad et al. (2010).

6.1.2 Summary of Contributions and Outline of Chapter

- We introduce the idea of PBDC, eliminating the problem costs pose to bundling (Section 6.2):
  - We show that PBDC has equivalent formulations in the *cost-based two-part tariff* that exists in the theoretical literature, as well as a new *per-item discount* scheme
  - The idea of PBDC motivates a transformation from costs to negative valuations, enabling the analysis in subsequent sections

- We improve "large-number-of-items" bounds for the performance of PBDC, using Cantelli's inequality (Section 6.3):
  - We recover existing bounds asymptotically and achieve a better bound when the coefficient of variation is large
  - Our bound suggests that the firm should not price the bundle such that the profit margin is less than 1/3 of the expected welfare

- We provide finite-item, distribution-free bounds for the performance of PBDC (Section 6.4):
  - We generalize existing bounds to the case with costs, where PBDC is needed instead of PB
  - We improve existing bounds in both directions (upper and lower bound)
  - We compare this type of performance guarantee to that in the previous section
- We provide a continuation of the numerical experiments from Chu et al. (2008), demonstrating the efficacy of PBDC for a finite number of items (Section 6.5)

6.2 Set-up and Equivalence Propositions

A firm has n different items for sale. For each i, the cost incurred by the firm for selling item i is c_i, a non-negative real number. c_i can be thought of as an instantaneous production cost, the opportunity cost of saving the inventory for someone else, or the value of the item to the seller.

Each of the firm's customers has a valuation vector $x \in \mathbb{R}^n$ for the items. A customer wants at most one of each item, and her utility for a subset of items S is $\sum_{i \in S} x_i$. x can be thought of as a random vector drawn from D, the distribution of valuation vectors across the population. The firm is risk-neutral and its objective is to maximize the expected per-customer profit.

In the full generality of the problem, the firm's mechanism for selling the items is a *menu* $\mathcal{M}$ of *entries* (q, s), where $q \in [0,1]^n$ is the *allocation* indicating the fraction of each item transferred to the customer, and s is the *payment* that must be made for this allocation. If q has fractional entries, then the allocation can be thought of as a lottery where the customer only gets some items with a certain probability. The customer is also risk-neutral and chooses the entry maximizing her expected surplus, $q^T x - s$. We will assume that for every entry, the payment covers the expected cost for the firm to produce that allocation, i.e. $s \geq q^T c$, where $c = (c_1, \ldots, c_n)$. Simultaneously removing all entries in the menu with $s < q^T c$ cannot decrease the profit, since this can only force a customer who previously selected an entry with negative profit to select an entry with non-negative profit.

Let $\mathcal{X}$ denote the support of D. Given a menu $\mathcal{M}$, for all $x \in \mathcal{X}$, let $q_{\mathcal{M}}(x)$ denote the allocation chosen by a customer with valuation vector x, and $s_{\mathcal{M}}(x)$ denote the corresponding payment. We will omit the subscript $\mathcal{M}$ when the context is clear. $(q_{\mathcal{M}}(x), s_{\mathcal{M}}(x))$ must maximize the surplus of the customer among all entries of $\mathcal{M}$[3] (the mechanism is *incentive-compatible*), and one of these entries must be the no-purchase option with $q = 0, s = 0$ (the mechanism is *individually rational*).

The firm's profit maximization problem can be written as $\max_{\mathcal{M}} \mathbb{E}_{x \sim D}[s_{\mathcal{M}}(x) - q_{\mathcal{M}}(x)^T c]$. However, the optimization over general menus is intractable, and the resulting menu may not be practical.

Definition 6.2.1. A *pricing scheme* is a restricted class of menus, implied by a compact way of communicating the menu to the customer. It is assumed that the optimization problem over the restricted class of menus can be solved efficiently.

The following pricing schemes are frequently referenced throughout this chapter:

1. Pure Components (PC): items are offered individually, at respective non-negative prices $P_1^{\mathsf{PC}}, \ldots, P_n^{\mathsf{PC}}$.

2. Pure Components with Uniform Discount (PCUD): items are offered individually, at respective prices $P_1^{\mathsf{PCUD}}, \ldots, P_n^{\mathsf{PCUD}}$. There is an absolute discount of P_0^{PCUD} which is applied to each item bought beyond the first. For all i, $P_i^{\mathsf{PCUD}} \geq P_0^{\mathsf{PCUD}} \geq 0$ is imposed.

3. Pure Bundling (PB): all of the items are offered in a single bundle at non-negative price P_0^{PB}, and there are no separate sales.

4. Pure Bundling with Disposal (PBD): all of the items are offered in a single bundle at price P_0^{PBD}, with the agreement that if the customer buys the bundle, she can return any subset of items S for a refund equal to $\sum_{i \in S} P_i^{\mathsf{PBD}}$. For all i, $P_i^{\mathsf{PBD}} \geq 0$ is imposed. Furthermore, $P_0^{\mathsf{PBD}} \geq \sum_{i=1}^{n} P_i^{\mathsf{PBD}}$ is imposed.

5. Tariff Pricing (TP): there is a membership fee of P_0^{TP} to enter the market. If the customer enters the market, she can buy up to one unit of each item i for the price of P_i^{TP}. $P_0^{\mathsf{TP}}, P_1^{\mathsf{TP}}, \ldots, P_n^{\mathsf{TP}}$ are all imposed to be non-negative.

[3]In the case there are multiple maxima, the firm can choose between them; this can always be achieved by small perturbations.

6. Bundle-Size Pricing (BSP): the price of a subset S is $P^{\mathsf{BSP}}_{|S|}$, which is dependent only on the number of items in S, and not which items are in S. $0 = P^{\mathsf{BSP}}_0 \leq P^{\mathsf{BSP}}_1 \leq \ldots \leq P^{\mathsf{BSP}}_n$ is imposed.

PC and PB were introduced by Adams and Yellen (1976). PCUD and PBD are variations of PC and PB, respectively, and to the best of our knowledge, PCUD and PBD are new to this chapter. BSP was introduced by Hitt and Chen (2005); Chu et al. (2008), while the idea of TP could be seen in Armstrong (1999). Note that PC corresponds to the degenerate case of PBD where $P^{\mathsf{PBD}}_0 = \sum_{i=1}^n P^{\mathsf{PBD}}_i$.

Our first observation is the following:

Proposition 6.2.2. *PCUD, PBD, and TP represent the same class of menus. Specifically, given a PBD representation with prices*

$$(P^{\mathsf{PBD}}_0, P^{\mathsf{PBD}}_1, \ldots, P^{\mathsf{PBD}}_n) \tag{6.1}$$

the corresponding PCUD representation is

$$(P^{\mathsf{PCUD}}_0 = P^{\mathsf{PBD}}_0 - \sum_{i=1}^{n} P^{\mathsf{PBD}}_i, P^{\mathsf{PCUD}}_1 = P^{\mathsf{PBD}}_1 + P^{\mathsf{PCUD}}_0, \ldots, P^{\mathsf{PCUD}}_n = P^{\mathsf{PBD}}_n + P^{\mathsf{PCUD}}_0) \tag{6.2}$$

and the corresponding TP representation is

$$(P^{\mathsf{TP}}_0 = P^{\mathsf{PBD}}_0 - \sum_{i=1}^{n} P^{\mathsf{PBD}}_i, P^{\mathsf{TP}}_1 = P^{\mathsf{PBD}}_1, \ldots, P^{\mathsf{TP}}_n = P^{\mathsf{PBD}}_n). \tag{6.3}$$

The proofs of propositions are deferred to the appendix. Hereinafter, we will always refer to PBD instead of PCUD and TP for the analysis; however, the existence of PCUD and TP gives the firm additional flexibility in how to describe these menus to the customer. Specifically, when the number of items is small, PCUD should be used instead of TP, since it does not sound so enticing for one to pay a surcharge in order to buy a small number of items. On the other hand, when the number of items is large, P^{PCUD}_0 tends to be large, causing the individual items to be marked at exorbitant prices should PCUD be used. In this case, paying a membership fee to have access to all the items does not sound so bad.

We are especially interested in the following subclass of PBD, where $P^{\mathsf{PBD}}_i = c_i$ for all i. Similar subclasses can also be defined for PCUD and TP, following Proposition 6.2.2.

Pure Bundling with Disposal for Cost (PBDC): the bundle with all of the items is offered at P_0^{PBD}. If the customer buys the bundle, she can return any subset S of items for a refund of $\sum_{i \in S} c_i$.

Setting the refund for each item equal to its cost is a logical restriction to put on PBD. To see why, consider the following definitions:

Definition 6.2.3. The *welfare* generated by a customer with valuations $(x_1, \ldots, x_n)$ is $\sum_{i=1}^{n} \max\{x_i - c_i, 0\}$, which is realized when every item valued above cost is transferred and no other items are transferred. Welfare can be split up as follows:

- The *total surplus* is the welfare realized from transfers that occurred, equal to $\sum_{i=1}^{n} q_i(x)(x_i - c_i)$. Total surplus can be further split up depending on the price charged:
  - The *producer surplus* is another term for the profit earned by the firm, equal to $s(x) - \sum_{i=1}^{n} q_i(x)c_i$.
  - The *consumer surplus* is the utility gained by the customer, equal to $\sum_{i=1}^{n} q_i(x)x_i - s(x)$.
- The *deadweight loss* is the welfare lost because an item valued above cost was not transferred, equal to $\sum_{i:x_i > c_i}(1 - q_i(x))(x_i - c_i)$.
- The *overinclusion loss*[4] is the welfare lost because items were consumed for utility below cost, equal to $\sum_{i:x_i < c_i} q_i(x)(c_i - x_i)$.

It is clear from the equations that the sum of producer surplus, consumer surplus, deadweight loss, and overinclusion loss is $\sum_{i:x_i > c_i}(x_i - c_i)$, equal to welfare. Also, the fact that the consumer surplus is non-negative (since the customer can always choose the no-purchase option) implies that the profit is no greater than the total surplus, which in turn is no greater than the welfare.

PBDC (and thus PBD) is strictly better than PB in the following sense:

Proposition 6.2.4. *Given a PB menu with price P^{PB} which is at least $c_1 + \ldots + c_n$, consider instead the PBD menu with prices ($P_0^{\mathsf{PBD}} = P^{\mathsf{PB}}, P_1^{\mathsf{PBD}} = c_1, \ldots, P_n^{\mathsf{PBD}} = c_n$). For all x:*

[4] We use this terminology because Adams and Yellen (1976) refer to the act of not incurring this loss as *exclusion*.

- *The producer surplus is no less than before.*
- *The consumer surplus is no less than before.*
- *The deadweight loss is no more than before.*
- *The overinclusion loss is no more than before.*

Note that the preceding statements are not only in expectation; for *every* valuation vector x both the firm and the customer are better off. There is no reason to use PB if PBDC can be used instead, because PBDC is effectively PB where all valuations x_i have been replaced by $\max\{x_i, c_i\}$. This observation leads us to the following lemma:

Proposition 6.2.5. *The firm's problem of maximizing expected profit with distribution D and costs c is equivalent to the transformed problem of maximizing expected revenue with distribution D', where D' is the distribution D shifted downward by c_i in every dimension i. Furthermore, for any menu in the original problem, the corresponding menu in the transformed problem has the payment for each allocation reduced by the cost of producing that allocation.*

Proposition 6.2.5 is stated in precise mathematical language and proven in the appendix. If the original optimization problem was over a restricted class of menus, then the class restriction in the transformed setting can be found via the second statement in Proposition 6.2.5.

For the remainder of this chapter, we focus on bounding the revenue of PBDC in the transformed setting, which is more amenable to analysis. PBDC becomes the class of menus that offer the same price P for any non-empty subset of items (see Remark E.1.1 in the appendix for a technical proof of this). The customer makes a purchase if and only if her non-negative valuations (corresponding to valuations no less than cost) sum to at least P, in which case the firm earns P.

It may be tempting to truncate all negative customer valuations to 0 and claim that after this further transformation, PBDC is identical to PB. However, in Section 6.4, we bound the performance of the best PBDC menu relative to the *optimal* menu (with no restriction to a pricing scheme), which can be designed to exploit negative valuations to reduce the cannibalization of higher-priced menu entries. In general, revenue is *non-monotone*, i.e. increasing customer valuations can decrease the optimal revenue—see Hart and Nisan (2012).

6.3 Asymptotic Performance Bounds

In this section we analyze the performance of PBDC with a large number of items, whose costs have been transformed into negative valuations according to the previous section. We assume that the valuations for different items are *independent* random variables. Also making some assumptions on the means and variances of the individual distributions, PBDC is asymptotically optimal as the number of items becomes large.

Armstrong (1999) has already proven this result for *Cost-based Tariff Pricing* (TP with the additional restriction that $P_i^{\mathsf{TP}} = c_i$ for all i), which is equivalent to PBDC via Proposition 6.2.2. However, our analysis works under weaker assumptions, by employing Cantelli's inequality, along with other tools. To our knowledge, we are the first to use Cantelli's one-sided concentration inequality to get an improved performance bound for bundling; previous works by Bakos and Brynjolfsson (1999); Armstrong (1999); Fang and Norman (2006) all use the weaker Chebyshev's inequality. The analysis also motivates our finite-item, distribution-free bounds in Section 6.4, where we again make improvements using Cantelli's inequality.

Lemma 6.3.1. *(Cantelli's Inequality) Let X be a random variable with (finite) mean μ and variance σ^2. Let t be an arbitrary non-negative real number. Then*

$$\mathbb{P}[X - \mu \leq -t] \leq \frac{\sigma^2}{\sigma^2 + t^2}$$

We refer the reader to Lugosi (2009) for a proof, as well as more background. Our main result in this section is the following:

Theorem 6.3.2. *Suppose a firm is selling items to a customer with valuation vector x drawn from distribution D. Let $\mathrm{VAL}^+(D)$ denote the mean of the welfare, equal to $\mathbb{E}_{x\sim D}[\sum_i \max\{x_i, 0\}]$, and assume that $0 < \mathrm{VAL}^+(D) < \infty$. Furthermore, let $\mathrm{CV}^+(D)$ denote the coefficient of variation of the welfare, equal to $\frac{\sqrt{\mathrm{Var}_{x\sim D}[\sum_i \max\{x_i,0\}]}}{\mathbb{E}_{x\sim D}[\sum_i \max\{x_i,0\}]}$, and assume that $\mathrm{CV}^+(D) < \infty$. Then for all $\varepsilon \in [0, 1]$, the expected revenue of the PBDC menu with* *price* $(1-\varepsilon)\textsc{Val}^+(D)$ *is at least* $\frac{\varepsilon^2-\varepsilon^3}{\varepsilon^2+(\textsc{Cv}^+(D))^2}\cdot\textsc{Val}^+(D)$. *In particular, if*

$$\varepsilon = \frac{2(\textsc{Cv}^+(D))^{\frac{2}{3}}}{3(\textsc{Cv}^+(D))^{\frac{2}{3}}+2}, \tag{6.4}$$

then the expected revenue is at least

$$\frac{4}{4+24(\textsc{Cv}^+(D))^{\frac{2}{3}}+45(\textsc{Cv}^+(D))^{\frac{4}{3}}+27(\textsc{Cv}^+(D))^2}\cdot\textsc{Val}^+(D) \tag{6.5}$$

which in turn is at least

$$(1-6(\textsc{Cv}^+(D))^{\frac{2}{3}})\cdot\textsc{Val}^+(D). \tag{6.6}$$

(6.6) shows that when the coefficient of variation is close to 0, ε scales with $(\textsc{Cv}^+(D))^{\frac{2}{3}}$ and earns a $\left(1-\Theta((\textsc{Cv}^+(D))^{\frac{2}{3}})\right)$–fraction of the expected welfare, recovering the result from Bakos and Brynjolfsson (1999) and Armstrong (1999). However, for larger $\textsc{Cv}^+(D)$, we still get a non-zero revenue guarantee in (6.5), and interestingly our analysis never recommends offering the bundle below the price of $(1-\frac{2}{3})\textsc{Val}^+(D)=\frac{\textsc{Val}^+(D)}{3}$. Contrast this to the previous analyses, which recommend $\varepsilon=1$ when $\textsc{Cv}^+(D)\geq 1$, earning zero revenue. The value of ε in (6.4), recommended by our analysis, is useful even when the firm has the resources to compute the optimal value of ε from D—both as a managerial reference point, as well as in situations where the firm knows the mean and variance in demand but is uncertain about the exact distribution.

Theorem 6.3.2 treats the welfare as an abstract random variable, but the revenue guarantee is weak if the coefficient of variation is large. Independence is important in allowing the "law of large numbers" to control $\textsc{Cv}^+(D)$ when the number of items n is large.

Corollary 6.3.3. *Suppose a firm is selling n items to a customer with independent valuations $x_1,\ldots,x_n$ forming product distribution D. Let $\mu_{\min}$ be a lower bound on* $\mathbb{E}[\max\{x_i,0\}]$, *and let $\sigma^2_{\max}$ be an upper bound on* $\mathrm{Var}[\max\{x_i,0\}]$, *over $i=1,\ldots,n$. Suppose $\mu_{\min}>0$, $\sigma_{\max}<\infty$, and $n>(\frac{\sigma_{\max}}{\mu_{\min}})^2$. Then the expected revenue of an optimal menu within PBDC is at least*

$$(1-6(\frac{\sigma_{\max}}{\mu_{\min}})^{\frac{2}{3}}\frac{1}{\sqrt[3]{n}})\cdot\textsc{Val}^+(D).$$

Taking $n\to\infty$, we get the result that PBDC extracts the entire welfare. Note that truncating the random variables x_i from below by 0 can only increase the mean and decrease the variance, so any lower bound on $\mathbb{E}[x_i]$ and upper bound on $\mathrm{Var}[x_i]$ would also satisfy the conditions in Corollary 6.3.3.

Proof. Proof of Theorem 6.3.2. Let $X = \sum_{i=1}^{n} \max\{x_i, 0\}$ be a single random variable representing the welfare of a valuation vector drawn from D. As additional shorthand, let $\mu = \textsc{Val}^+(D)$ denote the mean of X, $\sigma = \textsc{Val}^+(D) \cdot \textsc{Cv}^+(D)$ denote the standard deviation of X, and $C = \textsc{Cv}^+(D)$ denote the coefficient of variation of X.

We would like to bound the probability that $X < (1-\varepsilon)\mu$ from above. Applying Cantelli's inequality with $t = \varepsilon\mu$, this probability is at most $\frac{\sigma^2}{\sigma^2+\varepsilon^2\mu^2}$. Therefore, our expected revenue is at least

$$(1-\varepsilon)\mu \cdot \left(1 - \frac{\sigma^2}{\sigma^2+\varepsilon^2\mu^2}\right) = \mu \cdot \frac{(1-\varepsilon)\varepsilon^2\mu^2}{\sigma^2+\varepsilon^2\mu^2}.$$

The fraction of expected welfare earned is

$$\begin{aligned} \frac{\varepsilon^2-\varepsilon^3}{\varepsilon^2+C^2} &\geq \frac{\varepsilon^2-\varepsilon^3}{\frac{2}{3}y^3C^{-\frac{2}{3}} + \frac{1}{3}C^{\frac{4}{3}} + C^2} && (6.7) \\ &\geq \frac{\varepsilon^2 - (1+\frac{2}{3}C^{-\frac{2}{3}})\varepsilon^3}{\frac{1}{3}C^{\frac{4}{3}} + C^2}. && (6.8) \end{aligned}$$

The first inequality uses the *weighted arithmetic mean–geometric mean inequality* (see Zhao (2008) for a reference), which yields $\frac{2y^3+C^2}{3} \geq (y^6C^2)^{\frac{1}{3}} = y^2C^{\frac{2}{3}}$. The second inequality is because for a fraction $\frac{a}{b}$ with $0 < a \leq b$, subtracting the same amount less than b from both the numerator and the denominator can only decrease the fraction.

Now, if we choose $\varepsilon = \dfrac{2C^{\frac{2}{3}}}{3C^{\frac{2}{3}}+2}$ (this is motivated by setting the derivative of (6.8) to zero), then the LHS of (6.7) becomes

$$\begin{aligned} \frac{4C^{\frac{4}{3}}(1-\frac{2}{3})}{(3C^{\frac{2}{3}}+2)^2(\frac{1}{3}C^{\frac{4}{3}}+C^2)} &= \frac{\frac{4}{3}}{(2+3C^{\frac{2}{3}})^2(\frac{1}{3}+C^{\frac{2}{3}})} \\ &= \frac{4}{4+24C^{\frac{2}{3}}+45C^{\frac{4}{3}}+27C^2} \\ &= 1 - 6C^{\frac{2}{3}}\left(\frac{4+\frac{15}{2}C^{\frac{2}{3}}+\frac{9}{2}C^{\frac{4}{3}}}{4+24C^{\frac{2}{3}}+45C^{\frac{4}{3}}+27C^2}\right) \\ &\geq 1 - 6C^{\frac{2}{3}} \end{aligned}$$

where the inequality holds because the expression in parentheses is less than 1. This establishes both (6.5) and (6.6), completing the proof of Theorem 6.3.2. □

Proof. Proof of Corollary 6.3.3. By independence, $\mathrm{Var}[\sum_{i=1}^{n} \max\{x_i, 0\}] = \sum_{i=1}^{n} \mathrm{Var}[\max\{x_i, 0\}]$ which is at most $n\sigma_{\max}^2$. Furthermore, $\mathbb{E}[\sum_{i=1}^{n} \max\{x_i, 0\}] \geq n\mu_{\min}$. Therefore, $\mathrm{Cv}^+(D)$ is upper bounded by $\frac{\sigma_{\max}}{\mu_{\min}\sqrt{n}}$, and it is easy to see from the proof of Theorem 6.3.2 that all of its statements continue to hold when $\mathrm{Cv}^+(D)$ is replaced by an upper bound on $\mathrm{Cv}^+(D)$. The condition $n > (\frac{\sigma_{\max}}{\mu_{\min}})^2$ ensures that $\mathrm{Cv}^+(D) < 1$, and the result follows immediately from substituting $\mathrm{Cv}^+(D) \leq \frac{\sigma_{\max}}{\mu_{\min}\sqrt{n}}$ into (6.6). □

6.4 Finite-item, Distribution-free Performance Bounds

In this section we analyze the performance of PBDC with only the independence assumption on the items, whose costs have been transformed into negative valuations according to Section 6.2. All proofs are deferred to the appendix, but we sketch the techniques needed to handle arbitrary distributions.

Theorem 6.4.1. *Suppose a firm is selling items to a customer with independent (and potentially negative) valuations forming product distribution D. Let $\mathrm{REV}(D)$ denote the expected revenue of an optimal menu (along with tie-breaking rules) for distribution D. Then the expected revenue of either the optimal menu within PBDC or the optimal menu within PC is at least*

$$\frac{1}{5.2} \cdot \mathrm{REV}(D).$$

In the previous section, we showed that with assumptions on the number of items and their variances, PBDC can earn almost all of the expected welfare, $\mathrm{VAL}^+(D)$. However, this is clearly false without distributional assumptions—$\mathrm{VAL}^+(D)$ can be infinite. To recover some guarantee on performance, we need to use the *core-tail decomposition*, a technique developed through Hart and Nisan (2012); Li and Yuan (2013); Babaioff et al. (2014).

The idea of the core-tail decomposition is to split off from each independent distribution all the valuations above a large positive cutoff (the "tail"). The remaining valuations (the "core") are bounded, and it can be shown using a concentration inequality that PB (in our case PBDC) performs well relative to the welfare of the core. Meanwhile, PC can be shown to perform well relative to the *optimal mechanism* in the tail. Finally, the core bound (relative to the expected welfare of the core) and the tail bound (relative to the optimal expected revenue for the tail) can be combined to get a performance guarantee relative to

Table 6.1: Comparison of Guarantees

	Corollary 6.3.3 (Section 6.3)	Theorem 6.4.1 (Section 6.4)
Dependence on n	bound only relevant for large n	**none**
Assumptions on Distributions	uniformly bounded variance	**none**
Benchmark	**expected welfare**	expected revenue of optimal menu
% of Benchmark Guaranteed	**100% as $n \to \infty$**	$\frac{1}{5.2}$, with the help of PC
Related Literature	Bakos and Brynjolfsson (1999) Armstrong (1999)	Babaioff et al. (2014)

The advantages of each bound are bolded.

the optimal expected revenue on D.

Theorem 6.4.1 improves upon the main result of Babaioff et al. (2014) by increasing the guarantee from $\frac{1}{6}$ to $\frac{1}{5.2}$, and allowing for negative valuations. The differences in our analysis can be summarized as follows:

- We analyze the core and tail together, and show that the worst case for PBDC in the core and worst case for PC in the tail cannot simultaneously occur
- We use Cantelli's inequality instead of Chebyshev's inequality in the core bound
- We show that the core bound and the tail bound can still be combined to upper-bound the optimal revenue on D when the optimal mechanism can exploit negative valuations

Table 6.1 compares Theorem 6.4.1 to the type of bound in the previous section, in particular Corollary 6.3.3. Essentially, to accommodate arbitrary distributions, we have to settle for a constant fraction of the optimum, compare against an optimum that is convoluted, and also allow ourselves to use PC in pathological cases.

One additional point worth mentioning is that it is unclear from Theorem 6.4.1 what the optimal prices for PBDC or PC are. It is assumed that the firm, knowing distribution D, can compute the optimal prices for both PBDC and PC and use the scheme with higher expected revenue, with the knowledge that it will be within $\frac{1}{5.2}$ of optimal. Meanwhile, Theorem 6.3.2, with its simpler analysis, has an explicit benchmark price of $\frac{(\mathrm{Cv}^+(D))^{2/3}+2}{3(\mathrm{Cv}^+(D))^{2/3}+2} \cdot \mathrm{Val}^+(D)$ for the bundle in PBDC.

Finally, we address the tightness of Theorem 6.4.1. First we present a theoretical upper bound.

Example 6.4.2. Consider an instance with 2 costless items, which have IID valuations distributed as follows. There is a point mass of size $1-\rho$ at 0, a point mass of size $\frac{\rho}{2}$ at 2, and the remaining $\frac{\rho}{2}$ mass distributed in an *equal-revenue* fashion on $[1,2)$, i.e. selling individually at any price in $[1,2)$ results in the same revenue. Formally, if Y is a random variable with this distribution, then

$$\mathbb{P}[Y \geq y] = \begin{cases} 1 & y = 0 \\ \rho & 0 < y \leq 1 \\ \frac{\rho}{y} & 1 \leq y \leq 2 \end{cases}$$

where the value of ρ is optimized to be $\frac{3}{3+\ln 2} \approx 0.81$.

Theorem 6.4.3. *Consider the instance in Example 6.4.2. The best possible PC revenue is* 2ρ*, attained by selling individual items at any price in* $[1,2]$*. The best possible PB revenue is also* 2ρ*, attained by selling the bundle at the price of* 2 *or* 3*. The optimal revenue is at least* $2\rho(2-\rho)$*; this value can be achieved by selling individual items at the price of* 2*, and the bundle at the discounted price of* 3*.*

Therefore, neither PC nor PB can obtain more than $\frac{3+\ln 2}{3+2\ln 2} \cdot \textsc{Rev}(D)$ *which is approximately*

$$\frac{1}{1.19} \cdot \textsc{Rev}(D).$$

In Example 6.4.2, both PC and PB perform poorly because there is a need to *price-discriminate*, i.e. allow customers who highly value an item to buy it for its individual price, but still give customers with lower valuations a chance of buying it as part of a discounted bundle. Very recently, Rubinstein (2016) constructed an example where both PC and PB perform poorly because there is a need to *partition* the items, i.e. split them into groups, and offer each group as a different bundle. In his example, the better of PC and PB can only obtain $\frac{1}{2}+\varepsilon$ of the revenue via partitioning, which is smaller than our bound. However, our example exhibits the worst-known loss from not price-discriminating, where even partitioning performs poorly relative to the optimal mechanism. Our example also only requires two IID items, following the examples of Hart and Nisan (2012); Hart and Reny (2012); the example in Rubinstein (2016) requires a large number of distinct items.

Nonetheless, there is a large gap between the best-known lower bound from Theorem 6.4.1 and the best-known upper bounds, and furthermore, being guaranteed only $\frac{1}{5.2} \approx 19.2\%$ of the optimal profit does not sound so enticing. However, this bound arises from a worst-case analysis that needs to address pathological instances, on which PBDC does not obtain $\frac{1}{5.2}$ of the optimum, but PC does. In the next section, we test the performance of PBDC over "average" instances.

6.5 Numerical Experiments

In this section we conduct a continuation of the numerical experiments from Chu et al. (2008) where PBDC is included as an additional pricing scheme. As a disclaimer, we should quote Chu et al. (2008) on the limitations inherent to this kind of numerical analysis:

> "Although we attempt to cover a large space of parameter values, the results clearly depend on the specific parameters we choose (i.e., the choice of grid). Further, there is no way for us to know whether we are under- or oversampling the relevant (i.e., empirically plausible) combinations of parameters. So, for example, when we describe average outcomes, these should certainly not be interpreted as outcomes that would be expected in an empirical sense—they should be interpreted narrowly as the average of the experiments we performed."

6.5.1 Procedure

For consistency, we follow the setup from Chu et al. (2008) as closely as possible. We use the same five families of valuation distributions commonly used to model demand—Exponential, Logit, Lognormal, Normal, and Uniform. We also use the same ranges of parameters for these families, as outlined in Table 6.2. The parameters were calibrated so that valuations across different families have similar means on average, and the highest means are around 10 times the lowest means. We allow for free disposal, just like Chu et al. (2008)—all negative valuations are converted to 0. We assume that valuations are independent across items.

As far as costs, we consider three scenarios:

1. *Heterogeneous Items*: valuation distributions fluctuate in accordance with Table 6.2, while costs are low. The cost of each item is set to 0.2, except in the case of Uniform distributions, where it is set to half the item's mean valuation. These are the same

Table 6.2: Ranges of Parameters, replicated from Chu et al. (2008)

Exponential	Marginal distributions are Exponential, with means chosen uniformly from $[0.2, 2]$. Thus the rates λ are in $[0.5, 5]$.
Logit	Marginal distributions are Gumbel, with fixed scale $\sigma = 0.25$ and means chosen uniformly from $[0, 2.5]$. Thus the locations μ are in $[-0.25\gamma, 2.5 - 0.25\gamma] \approx [-0.14, 2.36]$.
Lognormal	Marginal distributions are Lognormal. Logarithms of valuations are Normally distributed with means chosen uniformly from $[-1.5, 1]$ and fixed variance $\sigma^2 = 0.25$. Thus the original valuations have means in $[e^{-1.5+0.125}, e^{1+0.125}] \approx [0.25, 3.08]$.
Normal	Marginal distributions are Normal with means chosen uniformly from $[-1, 2.5]$ and variances chosen uniformly from $[0.25, 1.75]$.
Uniform	Marginal distributions are Uniform on $[0, b]$, where b is chosen uniformly from $[0.4, 4]$. Thus the means are in $[0.2, 2]$.

Note that Chu et al. (2008) have two separate families of Normal distributions, one with varying mean and one with varying variance. For convenience, we allow both to vary at the same time.

numbers used in Chu et al. (2008), so this scenario is a duplicate of some of their experiments.

2. *Heterogeneous Costs*: valuation distributions are identical, while costs fluctuate. The costs are chosen uniformly from $[0, 2.5]$, approximately the same range as the means. In the case of the bounded Uniform distribution, the costs are chosen uniformly from 0 to 0.75 times the maximum valuation, so that there always are some customers who value the item above cost. The fixed valuation distributions are disclosed in Table 6.3—we choose a mean that is on the high end of the range to avoid degenerate instances, where the welfare in the system is near 0 when costs are high.

3. *Heterogeneous Items and Costs*: both valuation distributions and costs are allowed to fluctuate (independently) according to the preceding scenarios.

The parameters and costs are summarized in Table 6.3.

We compare the four simple pricing schemes—PC, PB, BSP, and PBDC. Unlike Chu et al. (2008), we do not compute the optimal deterministic profit with $2^n - 1$ prices, since it is hard to compute, difficult to implement in practice, and could be far off from the optimal profit of a randomized mechanism anyway. Skipping this expensive computation allows us to consider n from 2 up to 6.

Table 6.3: Summary of Parameters and Costs

Taste Distribution	Range for Means	Fixed Mean	Range for Costs	Fixed Cost
Exponential	[0.20,2.00]	1.25	[0,2.5]	0.2
Logit	[0.00,2.50]	1.5	[0,2.5]	0.2
Lognormal	[0.25,3.08]	$e^{0.5+0.125} \approx 1.87$	[0,2.5]	0.2
Normal	[0.00,2.50]	1.5 (and variance 1)	[0,2.5]	0.2
Uniform	[0.20,2.00]	irrelevant	$[0, 1.5]\times$mean	$0.5\times$mean

For each combination of the 3 cost scenarios, 5 demand distributions, and 5 options for n, we randomly generate 200 instances, resulting in 15000 total instances. Chu et al. (2008) were able to discretize the parameter space for each combination and generate 220 instances in a grid. While generating instances in a grid is more reliable than generating instances randomly, we simply have too many combinations, because we allow costs to vary independently, allow for larger n, and in the case of Normal distributions, also allow variances to vary independently. Our randomized approach has the advantage of being scalable, and not depending on the exact grid chosen. Furthermore, we have verified that 200 instances per combination is enough, in that repeating the experiments does not cause the reported observations to change by any significance.

6.5.2 Observations

First, we report the performance of the simple pricing schemes separated by scenario. For each instance (out of the 15000), we compute which of PC, PB, BSP, PBDC earns the most profit on that instance, and record the performance of every pricing scheme as a *fraction* of this optimum. For each scenario (out of the 3), we report the median performance as well as 10'th percentile performance of every pricing scheme across the 1000 instances of each distribution family (200 for each of $n = 2, \ldots, 6$), in Table 6.4. We also count the number of instances on which each pricing scheme was best, in Table 6.5.

We know from Chu et al. (2008) that BSP is within 1% of the deterministic optimum in most of their settings, so there is minimal room for improvement under scenario 1. In fact, PBDC is a special case of BSP when all costs are identical, and very similar to PB when costs are low. However, as one can see in Table 6.4, PBDC still extracts close to 100% of the BSP profit under this scenario, hence it also extracts close to 100% of the deterministic optimum. For Uniform valuations, PBDC is no longer a special case of BSP, since costs

Table 6.4: Median and 10'th Percentile Performances of Pricing Schemes

		Heterogeneous Items				Heterogeneous Costs				Both Heterogeneous			
		PC	PB	BSP	PBDC	PC	PB	BSP	PBDC	PC	PB	BSP	PBDC
Exponential	0.1 %ile	.766	.940	**1**	.994	.850	.269	.807	**.995**	.884	.137	.759	**.978**
	0.5 %ile	.835	.972	**1**	.999	.931	.489	.907	**1**	.964	.403	.926	**1**
Logit	0.1 %ile	.826	.937	**1**	.988	.815	.063	.245	**.996**	.852	.001	.385	**.987**
	0.5 %ile	.873	.992	**1**	.998	.891	.481	*.595*	**1**	.938	*.168*	.894	**1**
Lognormal	0.1 %ile	.734	.982	**1**	.998	.775	.513	.760	**1**	.852	.015	.327	**.931**
	0.5 %ile	*.799*	.996	**1**	1	.861	.730	.880	**1**	.970	.245	.887	**1**
Normal	0.1 %ile	.825	.745	**1**	.957	.858	.297	.779	**.982**	.904	.010	.699	**.974**
	0.5 %ile	.890	.880	**1**	*.975*	.926	.547	.912	**1**	.978	.198	.933	**1**
Uniform	0.1 %ile	.904	.834	.940	**.949**	.872	.348	.875	**.948**	.914	.380	.605	**.937**
	0.5 %ile	.959	.867	.975	**.998**	.933	.578	.974	**1**	.982	.638	.875	**1**

For each scenario, the best performance in each row is **bolded**. The overall worst median performance of each pricing scheme is *italicized*.

vary proportionally with means. PBDC actually outperforms BSP in this setting—indeed, this is by far the worst setting for BSP listed in (Chu et al., 2008, tbl. 5), where it only extracts 91% of the deterministic optimum.

Scenario 2, where valuation distributions are identical but costs are allowed to fluctuate, really exhibits the power of PBDC, which allows customers to consume only the items they value above cost via self-selection. PC loses out on not bundling similar items that differ only in cost, while BSP is forced to compromise between charging cheap prices where high-cost items may be consumed for utility below cost, or charging expensive prices that result in a lot of deadweight loss in the low-cost items. In Section E.4, we show an instance that exemplifies why BSP performs so poorly when the costs in the setup from Chu et al. (2008) are increased.

When both valuation distributions and costs are allowed to vary under scenario 3, PBDC is still the best strategy by a significant margin. However, the benefits of bundling have decreased when items can be drastically different, so PC has gained ground. It seems intuitive to hypothesize that the performance of PC is inflated by the small values of n we are using. In the next subsection, we organize our reports separated by n, under scenario 3 (where both valuation distributions and costs are allowed to fluctuate).

6.5.3 Separation by n and Effects on Welfare

In this subsection, we allow both valuation distributions and costs to vary, and report averages across demand distributions, separated by n (instead of medians over the different choices for n, separated by demand distribution). Since the distribution families we're amalgamating were calibrated to have similar means over their ranges of parameters, it makes sense in this subsection to report average absolute profits, instead of median fractions. We also report the figures defined in Definition 6.2.3, in the same way as Chu et al. (2008).

In Table 6.6, we report the expected values of these figures across the 1000 instances for each n. The main conclusions are best summarized in Figures 6-1-6-2.

The first graph (Figure 6-1) shows that although PBDC optimizes from the perspective of a selfish monopolist interested only in Producer Surplus, it has a similar advantage in Total Surplus. There is no Overinclusion Loss, and the monopolist is encouraged to choose a low tariff price so that most customers can enter the market. PC also incurs no Overinclusion Loss, but incurs more Deadweight Loss because it does not bundle. PB

Table 6.5: Number of Instances on which each Pricing Scheme was Best

	Heterogeneous Items				Heterogeneous Costs				Both Heterogeneous			
	PC	PB	BSP	PBDC	PC	PB	BSP	PBDC	PC	PB	BSP	PBDC
Exponential	5	-	**995**	-	19	-	114	**867**	206	-	113	**681**
Logit	0	-	**1000**	-	121	-	45	**834**	179	-	133	**688**
Lognormal	0	-	**1000**	-	8	-	70	**922**	245	-	68	**687**
Normal	12	-	**988**	-	20	-	180	**800**	216	-	201	**583**
Uniform	228	-	293	**479**	145	-	348	**507**	370	-	129	**501**

Table 6.6: Report of Economics Figures, separated by n

Number of Items	Statistic	PC	PB	BSP	PBDC
2	Producer Surplus	0.427	0.301	0.412	**0.432**
	Consumer Surplus	0.287	0.194	0.250	0.292
	Total Surplus	0.714	0.495	0.662	0.724
	Deadweight Loss	0.192	0.351	0.224	0.183
	Overinclusion Loss	-	0.061	0.021	-
3	Producer Surplus	0.655	0.395	0.630	**0.683**
	Consumer Surplus	0.437	0.254	0.382	0.436
	Total Surplus	1.092	0.649	1.011	1.119
	Deadweight Loss	0.291	0.604	0.352	0.264
	Overinclusion Loss	-	0.130	0.020	-
4	Producer Surplus	0.870	0.457	0.827	**0.929**
	Consumer Surplus	0.587	0.293	0.497	0.582
	Total Surplus	1.456	0.749	1.324	1.511
	Deadweight Loss	0.396	0.905	0.498	0.342
	Overinclusion Loss	-	0.198	0.031	-
5	Producer Surplus	1.070	0.504	1.030	**1.167**
	Consumer Surplus	0.705	0.297	0.595	0.703
	Total Surplus	1.775	0.802	1.625	1.870
	Deadweight Loss	0.488	1.158	0.600	0.394
	Overinclusion Loss	-	0.304	0.039	-
6	Producer Surplus	1.265	0.553	1.206	**1.409**
	Consumer Surplus	0.844	0.346	0.697	0.828
	Total Surplus	2.108	0.899	1.902	2.237
	Deadweight Loss	0.587	1.440	0.736	0.459
	Overinclusion Loss	-	0.356	0.057	-

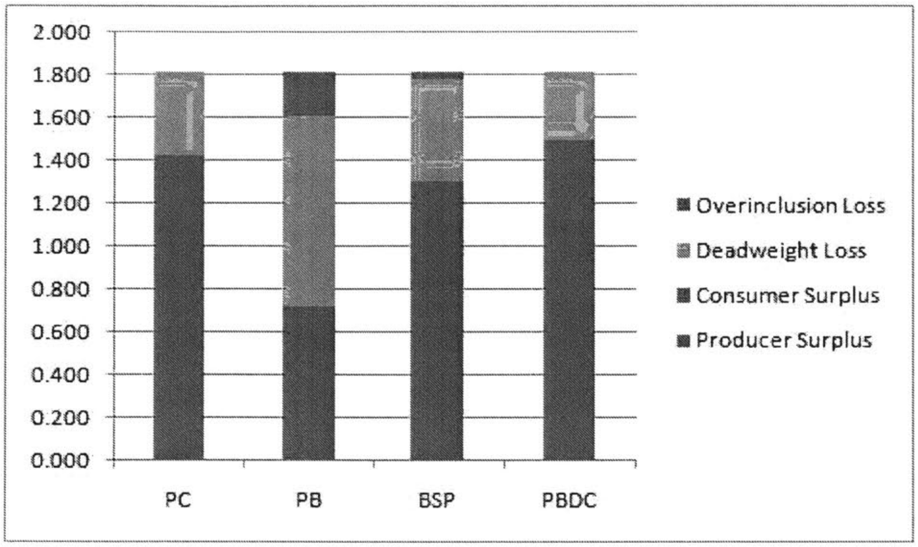


Figure 6-1: Breakdown of Welfare for each Pricing Scheme, averaged over n

incurs significantly more Overinclusion Loss than any other strategy, forcing the customer into buying every item at once. All in all, PBDC is equally attractive from the standpoint of an altruistic policymaker interested in maximizing Total Surplus.

The second graph (Figure 6-2) shows the profits of each pricing scheme as n increases. The PC profits increase linearly with n, since items are sold separately. Both the PB and the BSP profits are concave in n—that is, the marginal gain from having one more item to sell is decreasing. Indeed, PB is burdened with adding to its grand bundle another item that could be valued below cost, while BSP is burdened with an additional distinct item to consider in its item-symmetric cost structure. PBDC is the only pricing scheme where the profit is convex in n, as each item creates additional incentive for the customer to enter the market, and makes their total utility from entering the market more concentrated. This confirms the hypothesis that while Table 6.4 reports a small gap between PC and PBDC under scenario 3, this gap quickly widens as n increases.

6.5.4 Grid Instances and Comparing with the Deterministic Optimum for $n = 3$

In this subsection, we generate instances in a grid where both valuation distributions and costs are allowed to vary, for the $n = 3$ case. There are 3 possibilities for distribution mean and 3 possibilities for cost for each of 3 different items, resulting in a total of $3^6 = 729$

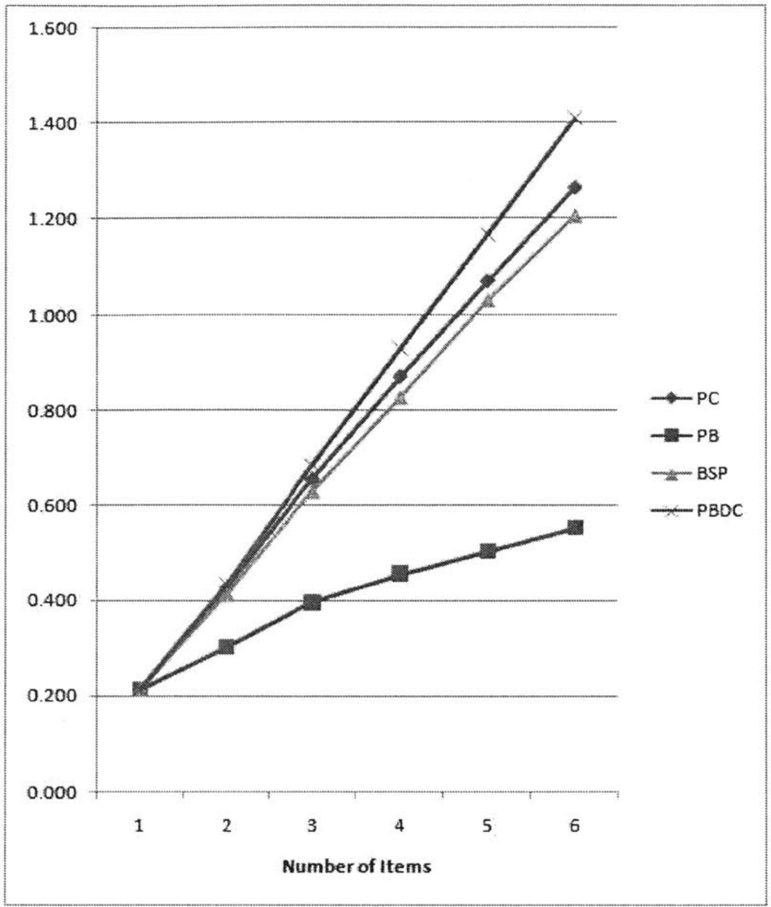


Figure 6-2: Average Profit of each Pricing Scheme, as a function of n

instances. This is repeated over the 5 different demand distributions. The grid is outlined in Table 6.7; we centered the grid around the values from Table 6.3.

We report the performance of each simple pricing scheme over these 729 instances in the same manner as Table 6.4, except this time every number is recorded as a fraction of the optimal deterministic profit, which is at least the profit of any simple pricing scheme. The results are displayed in Table 6.8.

In the median case, PBDC obtains between 96.6% to 99.4% of the deterministic optimum across the different demand distributions. This confirms both that PBDC is performing well relative to the optimal deterministic profit and not just other simple mechanisms, and that our earlier numbers with random instances are consistent.

To summarize our numerical experiments, we considered both scenarios with low costs and scenarios with high costs, and reported median performances over $n = 2, \ldots, 6$ for

Table 6.7: Grid for Items Parameters and Costs

Taste Distribution	Grid for Means	Grid for Costs
Exponential	$\{0.5, 1.25, 2\}$	$\{0, 1.25, 2.5\}$
Logit	$\{0.5, 1.5, 2.5\}$	$\{0, 1.25, 2.5\}$
Lognormal	$\{e^{0.125}, e^{0.625}, e^{1.125}\} \approx \{1.13, 1.87, 3.08\}$	$\{0, 1.25, 2.5\}$
Normal	$\{0.5, 1.5, 2.5\}$ (and fixed variance 1)	$\{0, 1.25, 2.5\}$
Uniform	$\{0.4, 1, 1.6\}$	$\{0, 0.75, 1.5\}\times$mean

Table 6.8: Median and 10'th Percentile Performances over the Grid

Taste Distribution	Statistic	PC	PB	BSP	PBDC
Exponential	0.1 %ile	.863	.159	.600	**.889**
	0.5 %ile	.932	.474	.872	**.966**
Logit	0.1 %ile	.881	.000	.219	**.966**
	0.5 %ile	.941	.308	.918	**.994**
Lognormal	0.1 %ile	.834	.575	.735	**.946**
	0.5 %ile	.909	.898	.959	**.989**
Normal	0.1 %ile	.877	.095	.593	**.944**
	0.5 %ile	.925	.479	.880	**.968**
Uniform	0.1 %ile	.874	.340	.461	**.899**
	0.5 %ile	.922	.723	.888	**.972**

different demand distributions. When costs are low, PC can earn as little as 79.9% of the profit of the optimal simple mechanism. When costs are high, PB can earn as little as 16.8% of the profit of the optimal simple mechanism, BSP can earn as little as 59.5%, and PC also falls behind as n increases. PBDC has the highest percentages overall, and is by far the most robust over different cost scenarios, always obtaining at least 97.5% of the profit of the optimal simple mechanism. We should point out that throughout our simulations, PBDC was also computationally much faster than BSP, requiring an optimization over 1 price instead of n.

6.6 Conclusion and Open Questions

In this chapter, we propose a simple strategy for the multi-product pricing problem: Pure Bundling with Disposal for Cost, or PBDC. We prove that PBDC is asymptotically optimal. When there are only a small number of items, we still guarantee that either PBDC or PC earns at least $\frac{1}{5.2} \approx 19.2\%$ of the optimal profit, and our simulations suggest that this is closer to 96.6%-99.4% in the average case, and that PC is not needed. While this is worse than the 99% achieved by Chu et al. (2008) for BSP in their experiments with lower costs, the pricing problem becomes much harder when costs are significant, and the existing simple pricing schemes (including BSP) fall behind PBDC by a great deal. Yet, production costs exceeding mean valuations is a common occurrence in industry, where only a small fraction of a company's customers may have interest in any particular item.

One caveat with PBDC is that the prices do reveal production costs to the customer. If this is undesired, a potential remedy is optimizing prices over the larger class of Tariff Pricing (TP) strategies, which has $n+1$ degrees of freedom and is guaranteed to be at least as profitable as PBDC. We believe that using TP instead of PBDC is very reasonable in practice, so long as the firm can accept the significant increase in computation time and decrease in the manager's ability to interpret the pricing.

However, the true demand distribution is never known, and must be constructed from data. When the given demand is prone to error, we hypothesize that there is additional benefit in choosing strategies that optimize one price at a time (such as PC, PB, PBDC) over strategies that optimize $\Theta(n)$ prices together (such as BSP, TP, MB). Besides, the theoretical guarantee for PBDC is no worse than that for TP, and PBDC is optimal as the number of items approaches infinity. We find it particularly interesting that as n increases and there are *more* potential prices to optimize, the benefit of optimizing only *one* price is greater.

All in all, PBDC captures the concentration effects of bundling and the selection effects of individual sales in a single heuristic that is computationally minimal and highly marketable. We hope our work on PBDC will have an impact on both the theory and practice of bundling, and be viewed as an effort to tie together the streams of research from three different disciplines: economics, computer science, and operations research.

Chapter 7

Improvements and Generalizations of Stochastic Knapsack and Markovian Bandits Approximation Algorithms

We study the multi-armed bandit problem with arms which are Markov chains with rewards. In the finite-horizon setting, the celebrated Gittins indices do not apply, and the exact solution is intractable. We provide approximation algorithms for the general model of Markov decision processes with non-unit transition times. When preemption isn't allowed, we provide a $(\frac{1}{2}-\varepsilon)$-approximation, along with an example showing this is tight. When preemption is allowed, we provide a $\frac{1}{12}$-approximation, which improves to a $\frac{4}{27}$-approximation when transition times are unity. Our model captures the Markovian Bandits model of Gupta et al., the Stochastic Knapsack model of Dean et al., and the Budgeted Learning model of Guha and Munagala. Our algorithms improve existing results in all three areas. In our analysis, we encounter and overcome to our knowledge a new obstacle—an algorithm that provably exists via analytical arguments, but cannot be found in polynomial time.

7.1 Introduction

We are interested in a broad class of stochastic control problems: there are multiple evolving systems competing for the attention of a single operator, who has limited time to extract as much reward as possible. Classical examples include a medical researcher allocating his time between different clinical trials, or a graduate student shifting her efforts between different ongoing projects. Before we describe the model in detail, we introduce three problems in the literature which motivated this work.

7.1.1 Markovian Bandits

The Markovian multi-armed bandit problem is the following: there are some number of Markov chains (*arms*), each of which only evolve to the next node[1] and return some reward when you play (*pull*) that arm; the controller has to allocate a fixed number of pulls among the arms to maximize expected reward. The reward returned by the next pull of an arm depends on the current node that arm is on. When an arm is pulled, the controller observes the transition taken before having to choose the next arm to pull. Multi-armed bandit (MAB) problems capture the tradeoff between *exploring* arms that could potentially transition to high-reward nodes, versus *exploiting* arms that have the greatest immediate payoff.

The infinite-horizon version of this problem with discounted rewards can be solved by the celebrated index policy of Gittins; see the book by Gittins et al. (2011) for an in-depth treatment of Gittins indices. However, Gittins indices are dependent on the time horizon being infinite (Gittins et al., 2011, sect. 3.4.1). The Gittins index measures the asymptotic performance of an arm, and does not apply when there are a finite number of time steps remaining.

Also, when we refer to multi-armed bandit, it is not to be confused with the Stochastic Bandits setting, where each arm is an unknown reward distribution, playing that arm collects a random sample from its distribution, and the objective is to learn which arm has the highest mean in a way that minimizes regret. For a comprehensive summary of Stochastic Bandits and other bandit settings, we refer to the survey by Bubeck and Cesa-Bianchi (2012). The main difference between Markovian Bandits and Stochastic Bandits

[1]We use the word *node* instead of *state* to avoid confusion with the notion of a state in dynamic programming.

is that in the former, the parameters governing the uncertainty are given as input and the challenge is computational, while in the latter, there is ambiguity in the parameters and the challenge is to compete with an adversary who knows the parameters in advance.

The finite-horizon Markovian Bandits problem is intractable even in special cases (see Goel et al. (2006), and the introduction of Guha and Munagala (2013)), so we turn to approximation algorithms. The state of the art is an LP-relative $\frac{1}{48}$-approximation[2] by Gupta et al. (2011a) (see Gupta et al. (2011b) for the conference version). We improve this bound by providing an LP-relative $\frac{4}{27}$-approximation for a more general model.

Martingale Reward Bandits and Bayesian Bandits

While Markovian Bandits is a different problem from Stochastic Bandits, it is a generalization of the closely related Bayesian Bandits problem, where each arm is an unknown reward distribution, but we have prior beliefs about what these distributions may be, and we update our beliefs as we collect samples from the arms. The objective is to maximize expected reward under a fixed budget of plays.

For each arm, every potential posterior distribution can be represented by a node in a Markov chain, and the transitions between nodes correspond to the laws of Bayesian inference. However, the resulting Markov chain is forced to satisfy the *martingale condition*, ie. the expected reward at the next node must equal the expected reward at the current node, by Bayes' law. This condition is not satisfied by Stochastic Knapsack with correlated rewards, as well as certain natural applications of the bandit model. For instance, in the marketing problems studied by Bertsimas and Mersereau (2007), the arms represent customers who may require repeated pulls (marketing actions) before they transition to a reward-generating node.

Nonetheless, fruitful research has been done in the Bayesian Bandits setting—Guha and Munagala (2013) show that constant-factor approximations can be obtained, also under a variety of side constraints. For the Bayesian Bandits problem with no side constraints, Farias and Madan (2011) show that *irrevocable* policies — policies which cannot start an arm, stop pulling it at some point, and resume it later — extract a constant fraction of the optimal (non-irrevocable) reward. Motivated by this, Guha and Munagala (2013) obtain a

[2]All of the problems we discuss will be maximization problems, for which an α-approximation refers to an algorithm that attains at least α of the optimum.

$(\frac{1}{2} - \varepsilon)$-approximation for Bayesian Bandits that is in fact a irrevocable policy.

Irrevocable Bandits

The above can be contrasted with the work of Gupta et al., who construct a non-martingale instance where irrevocable policies (they refer to these policies as *non-preempting*) can only extract an arbitrarily small fraction of the optimal reward (Gupta et al., 2011a, appx. A.3). Therefore, without the martingale assumption, one can only hope to compare irrevocable policies against the irrevocable optimum. We provide a $(\frac{1}{2} - \varepsilon)$-approximation for this problem, which we refer to as *Irrevocable Bandits*.

7.1.2 Stochastic Knapsack

The Stochastic Knapsack (SK) problem was introduced by Dean et al. (2004) (see Dean et al. (2008) for the journal version). We are to schedule some jobs under a fixed time budget. Each job has a stochastic reward and processing time whose distribution is known beforehand. We sequentially choose which job to perform next, only discovering its length and reward in real-time as it is being processed. The objective is to maximize the expected reward before the time budget is spent. A major focus of this work is on the benefit of *adaptive* policies (which can make dynamic choices based on the instantiated lengths of jobs processed so far) over *non-adaptive* policies (which must fix an ordering of the jobs beforehand), but in our work all policies will be adaptive.

Throughout Dean et al. (2008), the authors assume uncorrelated rewards — that is, the reward of a job is independent of its length. The state of the art for this setting is a $(\frac{1}{2} - \varepsilon)$-approximation by Bhalgat (2011); a $(\frac{1}{2} - \varepsilon)$-approximation is also obtained for the variant where jobs can be canceled at any time by Li and Yuan (2013). Gupta et al. (2011a) provide a $\frac{1}{8}$-approximation for Stochastic Knapsack with potentially correlated rewards, and a $\frac{1}{16}$-approximation for the variant with cancellation. We improve these bounds by providing an LP-relative $(\frac{1}{2} - \varepsilon)$-approximation for a problem which generalizes both variants with correlated rewards. Furthermore, we construct an example where the true optimum is as small as $\frac{1}{2} + \varepsilon$ of the optimum of the LP relaxation. Therefore, our bound is tight in the sense that one cannot hope to improve the approximation ratio using the same LP relaxation.

However, it is important to mention that our results, as well as the results of Gupta et al. (2011a), require the job sizes and budget to be given in unary, since these algorithms use a time-indexed LP. It appears that this LP is necessary whenever correlation is allowed — the non-time-indexed LP can be off by an arbitrarily large factor (Gupta et al., 2011a, appx. A.2). Techniques for discretizing the time-indexed LP if the job sizes and budget are given in binary are provided in Gupta et al. (2011a), albeit losing some approximation factor. Nonetheless, we always think of processing times as discrete hops on a Markov chain, given in unary. Note that the Stochastic Knapsack problem with correlated rewards and sizes given in binary can be shown to be PSPACE-hard (Dean et al., 2004, thm. 6).

7.1.3 Futuristic Bandits and Budgeted Bandits

Guha and Munagala (2007a,b) have studied many variants of *budgeted learning* problems — including switching costs, concave utilities, and Lagrangian budget constraints. See Guha and Munagala (2008) for an updated article that also subsumes some of their other works. Their basic setting, which we refer to as *Futuristic Bandits*, is identical to Bayesian Bandits (ie. there are Markov chains satisfying the martingale condition), except no rewards are dispensed during the execution of the algorithm. Instead, once the budget[3] is spent, we choose a single arm we believe to be best, and only earn the (expected) reward for that arm. A $\frac{1}{4}$-approximation is provided in Guha and Munagala (2008), and this is improved to a $(\frac{1}{3} - \varepsilon)$-approximation in Guha and Munagala (2013). Our algorithm works without the martingale assumption, but the approximation guarantee is only $\frac{4}{27}$.

7.1.4 MAB Superprocess with Multi-period Actions

Motivated by these examples, we now introduce our generalized model, which we call *MAB superprocess with multi-period actions.* Consider the Markovian Bandits setting, except we allow for a more general family of inputs, in two ways.

First, we allow transitions on the Markov chains to consume more than one pull worth of budget. We can think of these transitions as having a non-unit *processing time.* The processing times can be stochastic, and correlated with the node transition that takes place. The rewards can be accrued upon pulling the node, or only accrued if the processing time completes before the time budget runs out. The applications of such a generalization to

[3]In some variants, there is a cost budget instead of a time budget, and exploring each arm incurs a different cost. We explain in Section 7.2 why our model also generalizes this setting, which they refer to as *Budgeted Bandits.*

US Air Force jet maintenance have recently been considered in Kessler (2013), where it is referred to as *multi-period actions.*

The second generalization is that we allow each arm to be a Markov decision process; such a problem is referred to as *MAB superprocess* in Gittins et al. (2011). Now, when the controller pulls an arm, they have a choice of actions, each of which results in a different joint distribution on reward, processing time, and transition taken.

The purpose of the first generalization is to allow MAB to model the jobs from Stochastic Knapsack which have rewards correlated with processing time and can't be canceled. The purpose of the second generalization is to allow MAB to model Futuristic Bandits, where exploiting an arm corresponds to a separate action. The details of our reductions, along with examples, will be presented throughout Section 7.2, once we have introduced formal notation.

We consider two problem variants for our general model: the case with preemption (ie. we can start playing an arm, not play it for some time steps, and resume playing it later), and the case without preemption. The variant without preemption is necessary to generalize Stochastic Knapsack and Irrevocable Bandits. The variant with preemption generalizes Markovian Bandits and Futuristic Bandits.

7.1.5 Outline

This chapter can be outlined as follows:

- Reductions from existing problems to *MAB superprocess with multi-period actions* [sect. 7.2]

- Polynomial-sized LP relaxations for both variants of *MAB superprocess with multi-period actions*, and analytical proofs that they are indeed relaxations [sect. 7.2.5]

- A $(\frac{1}{2}-\varepsilon)$-approximation for *MAB superprocess with multi-period actions—no preemption*, with runtime polynomial in the input and $\frac{1}{\varepsilon}$ [sect. 7.3]

- A matching upper bound where it is impossible to obtain more than $\frac{1}{2}+\varepsilon$ of the optimum of the LP relaxation [sect. 7.3.1]

- A $\frac{4}{27}$-approximation for *MAB superprocess* (with preemption) [sect. 7.4]

Table 7.1: Comparison of results for SK

Problem	Previous Result	Result as a Special Case of Our Problems
Binary SK	$\frac{1}{2}-\varepsilon$ (Bhalgat, 2011)	-
Binary SK w/ Cancellation	$\frac{1}{2}-\varepsilon$ (Li and Yuan, 2013)	-
Unary Correlated SK	$\frac{1}{8}$ (Gupta et al., 2011a)	$\frac{1}{2}-\varepsilon$ [thm. 7.2.7]
Unary Correlated SK w/ Cancellation	$\frac{1}{16}$ (Gupta et al., 2011a)	$\frac{1}{2}-\varepsilon$ [thm. 7.2.7]

Table 7.2: Comparison of results for MAB

Problem	Previous Result	Result as a Special Case of Our Problems	Result with Martingale Assumption
Markovian Bandits	$\frac{1}{48}$	$\frac{4}{27}$ [thm. 7.2.8]	$\frac{1}{2}-\varepsilon$ (Guha and Munagala, 2013)
Irrevocable Bandits	-	$\frac{1}{2}-\varepsilon$ [thm. 7.2.7]	$\frac{1}{2}-\varepsilon$ (Guha and Munagala, 2013)
Futuristic Bandits	-	$\frac{4}{27}$ [thm. 7.2.8]	$\frac{1}{3}-\varepsilon$ (Guha and Munagala, 2013)
Budgeted Bandits	-	$\frac{1}{12}$ [thm. 7.2.9]	$\frac{1}{4}-\varepsilon$ (Guha and Munagala, 2008)

- A $\frac{1}{12}$-approximation for *MAB superprocess with multi-period actions* (and preemption) [sect. 7.4.3]

The ways in which these approximation ratios improve previous results on SK and MAB is summarized in Tables 7.1 and 7.2.

7.1.6 Sketch of Techniques

The sketch we provide here is brief, but a more detailed high-level overview is provided at the beginning of each subsection.

In the variant without preemption, we prove that given any feasible solution to the LP relaxation, there exists a policy which plays every node with half the probability it is played in the LP solution. This would yield a $\frac{1}{2}$-approximation, but the policy cannot be specified in polynomial time, because the previous argument is purely existential. Instead, we show how to approximate the policy via sampling, in a way that doesn't cause error propagation.

In the variant with preemption, we derive an approximation algorithm which uses priority indices, based on an optimal solution to the LP relaxation, to accomplish the explore-exploit tradeoff. Our priority-based policy is based on the ideas behind the *convex decomposition* and *gap filling* operations from Gupta et al. (2011a). We perform a tighter analysis for our algorithm and show how it can be generalized to the model of Markov decision processes with non-unit transition times. Our analysis uses Samuels' conjecture (Samuels,

1966) for $n = 3$ (which is proven) to bound the upper tail.

7.1.7 Related Work

The results on bandits, Stochastic Knapsack, and budgeted learning that are most related to our results have already been introduced in the earlier subsections, but we mention some additional results here. One such result for Stochastic Knapsack is the bi-criteria $(1-\varepsilon)$-approximation of Bhalgat et al. (2011) that uses $1+\varepsilon$ as much space; such a result is also obtained via alternate methods by Li and Yuan (2013) and generalized to the setting with both correlated rewards and cancellation. Also, Gupta et al. (2014) introduce the new *stochastic orienteering* problem, which associates jobs in SK with locations in a metric space. The benefit of adaptive policies for this problem is also addressed by Bansal and Nagarajan (2014).

Another example of a stochastic optimization problem where adaptive policies are necessary is the *stochastic matching* problem of Bansal et al. (2012) — in fact we use one of their lemmas in our analysis. Recently, the setting of stochastic matching has been integrated into online matching problems by Mehta and Panigrahi (2012).

All of the problems described thus far focus on expected reward. In contrast, Ilhan et al. (2011) study the variant of SK where the objective is to maximize the probability of achieving a target reward; older work on this model includes Carraway et al. (1993). Approximation algorithms for minimizing the expected sum of weighted completion times when the processing times are stochastic are provided in Möhring et al. (1999) and Skutella and Uetz (2001). SK with chance constraints — maximizing the expected reward subject to the probability of running overtime being at most p — is studied in Goel and Indyk (1999) and Kleinberg et al. (2000).

Looking at more comprehensive synopses, we point the reader interested in infinite-horizon Markovian Bandits to the book by Gittins et al. (2011). Families of bandit problems other than Markovian, including Stochastic and Adversarial, are surveyed by Bubeck and Cesa-Bianchi (2012). For an encyclopedic treatment of using dynamic programming to solve stochastic control problems, we refer the reader to the book by Bertsekas (1995). For an encyclopedic treatment of stochastic scheduling, we refer the reader to the book by Pinedo (2012).

7.2 Problem Definition

We first define the fully general *MAB superprocess with multi-period actions* (and preemption) problem described in the introduction. We introduce the variant without preemption later.

Problem 7.2.1 (Original Problem). There are $n \in \mathbb{N}$ arms which are Markov decision processes. For each arm i, let $\mathcal{S}_i$ denote its finite set of nodes, with the starting node, or *root node*, being ρ_i. To *play* an arm i that is currently on node $u \in \mathcal{S}_i$, we select an action a from the finite, non-empty action set A_u, after which the arm will transition to a new node $v \in \mathcal{S}_i$ in t time steps, accruing reward over this duration. We will also refer to this process as *playing action a on node u*, since for each pair (u, a), we are given as input the joint distribution of the destination node, transition time, and reward. Specifically, for all $u \in \mathcal{S}_i$, $a \in A_u$, and possible pairs of destination and transition time (v, t), let $p^a_{u,v,t}$ denote the probability of transitioning to node v in exactly t time steps, when action a is played on node u. We will refer to this transition by the quadruple (u, a, v, t), and conditioned on it occurring, let $R^a_{u,v,t,t'}$ denote the reward accrued t' time steps from the present, for all $t' = 0, \ldots, t-1$. $R^a_{u,v,t,t'}$ is a random variable with known distribution, taking values in $[0, \infty)$.

Each Markov decision process i starts on its root node, ρ_i. There is a total budget of $B \in \mathbb{N}$ time steps over which we would like to maximize the reward, in expectation. At each time step, if no arm is in the middle of a transition, then we may choose a new arm to play, along with an action[4]. We observe the destination and sequence of rewards over the transition time, realized according to the probabilities defined above. After the transition is over, we may choose a new arm and action to play. After B total time steps pass, our final reward is the sum of rewards collected up to that point, and an arm in the middle of transition is cut off.

7.2.1 Problem Simplification

We now perform a sequence of transformations to simplify the problem and notation, as well as aid in the analysis throughout the rest of the chapter. Assuming that the budget

[4]For convenience, we allow ourselves to not play an arm at a time step, even though playing is always beneficial, in that all rewards are non-negative.

B and transition times t are given in unary, all of the transformations can be performed in polynomial time.

TRANSFORMATION 1. We use $A = \cup_{i=1}^{n} \cup_{u \in \mathcal{S}_i} A_u$, a common set of actions for all nodes across all arms, defining incompatible actions to result in no transition.

TRANSFORMATION 2. We change all transition times t greater than B to equal B, cutting off rewards with $t' \geq B$. Since there are only B time steps, the exact value of any $t \geq B$ does not matter, and only the rewards with $t' = 0, \ldots, B-1$ may be obtainable.

TRANSFORMATION 3. We replace each random reward $R^a_{u,v,t,t'}$ with its deterministic equivalent $r^a_{u,v,t,t'} = \mathbb{E}[R^a_{u,v,t,t'}]$. This does not affect the objective of maximizing expected reward.

TRANSFORMATION 4. We add self-loops with unit-time and zero reward so that for all nodes u and actions a, $\sum_{v \in \mathcal{S}_i} \sum_{t=1}^{B} p^a_{u,v,t} = 1$. That is, an arm always makes a transition, instead of stopping.

TRANSFORMATION 5. We expand all transitions with non-unit processing times. For any transition (u, a, v, t) with $t > 1$, we:

1. Add *bridge nodes* $w_1, \ldots, w_{t-1}$;
2. Set transition probability $p^a_{u,w_1,1} = p^a_{u,v,t}$, and change $p^a_{u,v,t}$ to be 0;
3. Set transition probabilities $p^b_{w_1,w_2,1} = \ldots = p^b_{w_{t-1},v,1} = 1$ for all $b \in A$;
4. Set all other transition probabilities involving $w_1, \ldots, w_{t-1}$ to be 0;
5. Set rewards $r^a_{u,w_1,1,0} = r^a_{u,v,t,0}$, $r^b_{w_1,w_2,1,0} = r^a_{u,v,t,1}, \ldots, r^b_{w_{t-1},v,1,0} = r^a_{u,v,t,t-1}$ for all $b \in A$.

As long as we enforce the bridge nodes must be played as soon as they are reached, the new problem is equivalent to the old problem. Notationally, we will assume that they are played with action α. Also, we will eliminate the subscripts t, t' and just write $p^a_{u,v}$, $r^a_{u,v}$ now that all transitions with $t > 1$ have been eliminated.

TRANSFORMATION 6. For all $i \in [n]$, $u \in \mathcal{S}_i$, and $a \in A$, define $r^a_u = \sum_{v \in \mathcal{S}_i} p^a_{u,v} \cdot r^a_{u,v}$, and consider the Markov decision process that earns deterministic reward r^a_u every time action a is played on node u, instead of a random reward $r^a_{u,v}$ that depends on the destination node v. Under the objective of maximizing expected reward, the two processes are again equivalent.

Transformation 7. We convert each rooted Markov decision process into a layered acyclic digraph, up to depth B. That is, we assume there exists a function depth mapping nodes to $0, \ldots, B$ such that $\mathsf{depth}(\rho_i) = 0$ for all $i \in [n]$, and all transitions (u, a, v) with $p^a_{u,v} > 0$ satisfy $\mathsf{depth}(v) = \mathsf{depth}(u) + 1$. This can be done by expanding each node in the original graph into a time-indexed copy of itself for $t = 1, \ldots, B$—we refer to (Gupta et al., 2011a, appx. E.1) for the standard reduction, which immediately generalizes to the case of Markov decision processes with bridge nodes. We can cut off at depth B since there are only B time steps in total.

We restate the problem after all the transformations, summarizing the notation.

Problem 7.2.2 (Transformed Problem). An instance of *MAB superprocess with multi-period actions* consists of the following:

- A: the global set of actions, indexed by a, containing
  - α: the default action used to play bridge nodes;
- n: the number of arms, indexed by i, each with
  - $\mathcal{S}_i$: a finite set of nodes;
  - $\mathcal{B}_i$: the set of *bridge nodes*, a potentially empty subset of $\mathcal{S}_i$;
  - ρ_i: a root node in $\mathcal{S}_i \setminus \mathcal{B}_i$;
  - $p^a_{u,v}$: the probability of transitioning to node v when action a is played on node u, for all $a \in A$ and $u, v \in \mathcal{S}_i$ (with $u = v$ possible);
  - r^a_u: the reward obtained when action a is played on node u, for all $a \in A$ and $u \in \mathcal{S}_i$;
- B: the number of time steps, indexed by t;
- depth: a function from $\cup_{i=1}^n \mathcal{S}_i$ to $\{0, \ldots, B\}$ satisfying
  - $\mathsf{depth}(\rho_i) = 0$ for all i;
  - $\mathsf{depth}(v) = \mathsf{depth}(u) + 1$ for all (u, a, v) with $p^a_{u,v} > 0$.

The objective is to choose an arm and an action during each time step to maximize expected reward. At a time step, if the arm last played is on a bridge node, then the same arm must be played again.

7.2.2 Dynamic Programming

Algorithms for this problem are described in the form of an adaptive *policy*, a specification of which arm and action to play for each state the system could potentially be in. A state in this case is determined by the following information: the node each arm is on, and the time step we are at[5]. The optimal policy can be defined by an exponential-sized dynamic program. We write the Bellman state-updating equations as constraints to get a linear program whose feasible region is precisely the set of admissible policies. After adding in the objective function of maximizing expected reward, solving this exponential-sized linear program would be equivalent to solving our problem to optimality.

Definition 7.2.3. Define the following notation and related terminology.

1. For any positive integer m, let $[m]$ denote the set $\{1, \ldots, m\}$.
2. Let $\mathcal{S} = \bigcup_{i=1}^{n} \mathcal{S}_i$, the union of all nodes across all arms.
3. For all $u \in \mathcal{S}$, let $\mathsf{Par}(u) = \{(v, a) \in \mathcal{S} \times A : p_{v,u}^{a} > 0\}$, the set of *parents* of u, i.e. the (node, action) combinations that have a positive probability of transitioning to u.
4. Let $\boldsymbol{\mathcal{S}} = \mathcal{S}_1 \times \ldots \times \mathcal{S}_n$, the set of *joint nodes*, which are ordered n-tuples indicating the node each arm is on.
5. For all $\pi \in \boldsymbol{\mathcal{S}}$ and $u \in \mathcal{S}_i$, let π^u be the joint node such that $\pi_i^u = u$, and $\pi_j^u = \pi_j$ for all $j \neq i$.

A state in the dynamic program can then be defined as a joint node π along with a time t. Let $y_{\pi,t}$ be the probability of having arms on nodes according to π at the beginning of time t. Let $z_{\pi,i,t}^{a}$ be the probability we play arm i at time t with action a, when the arms are on nodes according to π. Note that some (π, t) pairs are impossible states[6], but for notational convenience we still have variables for these states.

Our objective is

$$\max \sum_{\pi \in \boldsymbol{\mathcal{S}}} \sum_{i=1}^{n} \sum_{a \in A} r_{\pi_i}^{a} \sum_{t=1}^{B} z_{\pi,i,t}^{a} \tag{7.1}$$

[5]Even though we have converted all Markov decision processes into layered acyclic digraphs, we cannot deduce the time elapsed from the nodes each arm is on, since we allow ourselves to not play any arm at a time step. Therefore, the time step must be included separately in the state information.

[6]For example we could never be at a joint node with two or more arms on bridge nodes, and we could not get an arm on a node of depth 5 at the beginning of time 5.

with the following constraints on how we can play the arms:

$$\sum_{i=1}^{n}\sum_{a\in A} z^a_{\pi,i,t} \leq y_{\pi,t} \qquad \pi\in\boldsymbol{\mathcal{S}},\ t\in[B] \tag{7.2a}$$

$$z^{\alpha}_{\pi,i,t} = y_{\pi,t} \qquad \pi\in\boldsymbol{\mathcal{S}},\ i:\pi_i\in\mathcal{B},\ t\in[B] \tag{7.2b}$$

$$z^a_{\pi,i,t} \geq 0 \qquad \pi\in\boldsymbol{\mathcal{S}},\ i\in[n],\ a\in A,\ t\in[B] \tag{7.2c}$$

The novel constraint is (7.2b), which guarantees that we must play a bridge node upon arrival. The remaining constraints update the $y_{\pi,t}$'s correctly:

$$y_{(\rho_1,\dots,\rho_n),1} = \quad 1 \tag{7.3a}$$

$$y_{\pi,1} = \quad 0 \qquad \pi\in\boldsymbol{\mathcal{S}}\setminus\{(\rho_1,\dots,\rho_n)\} \tag{7.3b}$$

$$\begin{aligned} y_{\pi,t} = \quad & y_{\pi,t-1} - \sum_{i=1}^{n}\sum_{a\in A} z^a_{\pi,i,t-1} \\ & + \sum_{i=1}^{n}\sum_{(u,a)\in\mathsf{Par}(\pi_i)} z^a_{\pi^u,i,t-1}\cdot p^a_{u,\pi_i} \qquad t>1,\ \pi\in\boldsymbol{\mathcal{S}} \end{aligned} \tag{7.3c}$$

Essentially, the only decision variables are the z-variables; there are as many y-variables as equalities in (7.3a)-(7.3c). These constraints guarantee $\sum_{\pi\in\boldsymbol{\mathcal{S}}} y_{\pi,t}=1$ for all $t\in[B]$, and combined with (7.2a), we obtain

$$\sum_{\pi\in\boldsymbol{\mathcal{S}}}\sum_{i=1}^{n}\sum_{a\in A} z^a_{\pi,i,t} \leq 1 \qquad t\in[B] \tag{7.4}$$

Let (ExpLP) denote the linear program defined by objective (7.1) and constraints (7.2a)-(7.2c), (7.3a)-(7.3c) which imply (7.4). This is the dynamic program for *MAB superprocess with multi-period actions* (and preemption).

7.2.3 No Preemption Variant

Now we define *MAB superprocess with multi-period actions—no preemption.* We follow the same set-up from Problem 7.2.1, and assume that the same sequence of transformations from Section 7.2.1 have been performed to arrive at Problem 7.2.2. However, we add the further constraint that for each arm, the set of time steps during which we play it must be contiguous. We encorce this by adding a *terminal node* to each arm, from which it cannot be played; if an arm not on its root node is not played during a time step, then it transitions to its terminal node.

Definition 7.2.4. Define the following notation and related terminology.

1. For all $i \in [n]$, let ϕ_i denote the terminal node of arm i, and let $\mathcal{S}_i' = \mathcal{S}_i \cup \{\phi_i\}$.

2. Let $\boldsymbol{\mathcal{S}}' = \mathcal{S}_1' \times \ldots \times \mathcal{S}_n' \setminus \{\pi : \pi_i \notin \{\rho_i, \phi_i\}, \pi_j \notin \{\rho_j, \phi_j\}, i \neq j\}$, where we have excluded the joint nodes with two or more arms in the middle of being processed.

3. For all $\pi \in \boldsymbol{\mathcal{S}}'$, let $I(\pi) = \{i : \pi_i \neq \phi_i\}$, the indices of arms that could be played from π.

4. For all $i \in [n]$, let $\boldsymbol{\mathcal{A}}_i = \{\pi \in \boldsymbol{\mathcal{S}}' : \pi_i \notin \{\rho_i, \phi_i\}\}$, the joint nodes with arm i *active*, i.e. in the middle of being processed.

5. Let $\boldsymbol{\mathcal{A}} = \bigcup_{i=1}^{n} \boldsymbol{\mathcal{A}}_i$, the set of joint nodes with an active arm.

6. For all $\pi \in \boldsymbol{\mathcal{S}}'$, let $\boldsymbol{\mathcal{P}}(\pi)$ denote the subset of $\boldsymbol{\mathcal{S}}'$ that would transition to π with *no play* during a time step.

   (a) If $\pi \notin \boldsymbol{\mathcal{A}}$, then $\boldsymbol{\mathcal{P}}(\pi) = \{\pi\} \cup \left(\bigcup_{i \notin I(\pi)} \{\pi^u : u \in \mathcal{S}_i \setminus \{\rho_i\}\}\right)$. $\boldsymbol{\mathcal{P}}(\pi)$ contains π because $\pi \notin \boldsymbol{\mathcal{A}}$, hence π does not contain an active arm which would transition to its terminal node with no play, and hence the system would remain at the same joint node π. Furthermore, for any i such that $i \notin I(\pi)$ (i.e. $\pi_i = \phi_i$), arm i would transition from any $u \in \mathcal{S}_i \setminus \{\rho_i\}$ to ϕ_i with no play, hence for such u, $\boldsymbol{\mathcal{P}}(\pi)$ contains joint node π^u.

   (b) If $\pi \in \boldsymbol{\mathcal{A}}$, then $\boldsymbol{\mathcal{P}}(\pi) = \emptyset$. This is because if joint node π has an active arm i, then we can only arrive at π by playing i, i.e. we cannot arrive at π with no play.

Now we can write the dynamic program for the variant without preemption. The objective is

$$\max \sum_{\pi \in \boldsymbol{\mathcal{S}}'} \sum_{i \in I(\pi)} \sum_{a \in A} r_{\pi_i}^a \sum_{t=1}^{B} z_{\pi,i,t}^a \tag{7.5}$$

with very similar constraints on the z-variables:

$$\sum_{i\in I(\pi)}\sum_{a\in A} z^a_{\pi,i,t} \le y_{\pi,t} \qquad \pi\in\boldsymbol{\mathcal{S}}',\ t\in[B] \tag{7.6a}$$

$$z^\alpha_{\pi,i,t} = y_{\pi,t} \qquad \pi\in\boldsymbol{\mathcal{S}}',\ i:\pi_i\in\mathcal{B},\ t\in[B] \tag{7.6b}$$

$$z^a_{\pi,i,t} \ge 0 \qquad \pi\in\boldsymbol{\mathcal{S}}',\ i\in I(\pi),\ a\in A,\ t\in[B] \tag{7.6c}$$

The only difference from (7.2a)-(7.2c) is that arms on terminal nodes cannot be played. However, moving forward, the state-updating constraints become more complicated, because now an arm can make a transition even while it is not being played, namely, the transition to the terminal node. We update the y-variables as follows:

$$y_{(\rho_1,\ldots,\rho_n),1} = 1 \tag{7.7a}$$

$$y_{\pi,1} = 0 \qquad \pi\in\boldsymbol{\mathcal{S}}'\setminus\{(\rho_1,\ldots,\rho_n)\} \tag{7.7b}$$

$$y_{\pi,t} = \sum_{\pi'\in\boldsymbol{\mathcal{P}}(\pi)}\Big(y_{\pi',t-1} - \sum_{i\in I(\pi')}\sum_{a\in A} z^a_{\pi',i,t-1}\Big) \qquad t>1,\pi\in\boldsymbol{\mathcal{S}}'\setminus\boldsymbol{\mathcal{A}} \tag{7.7c}$$

$$y_{\pi,t} = \sum_{a:(\rho_i,a)\in\mathsf{Par}(\pi_i)}\Big(\sum_{\pi'\in\boldsymbol{\mathcal{P}}(\pi^{\rho_i})} z^a_{\pi',i,t-1}\Big)\cdot p^a_{\rho_i,\pi_i} \quad t>1, i\in[n],\pi\in\boldsymbol{\mathcal{A}}_i,\mathsf{depth}(\pi_i)=1 \tag{7.7d}$$

$$y_{\pi,t} = \sum_{(u,a)\in\mathsf{Par}(\pi_i)} z^a_{\pi^u,i,t-1}\cdot p^a_{u,\pi_i} \qquad t>1, i\in[n],\pi\in\boldsymbol{\mathcal{A}}_i,\mathsf{depth}(\pi_i)>1 \tag{7.7e}$$

(7.7c) updates $y_{\pi,t}$ for $\pi\notin\boldsymbol{\mathcal{A}}$, ie. joint nodes with no active arms. Such a joint node π can only be arrived upon by making no play from a joint node in $\boldsymbol{\mathcal{P}}(\pi)$.

(7.7d), (7.7e) update $y_{\pi,t}$ for $\pi\in\boldsymbol{\mathcal{A}}$. To get to joint node $\pi\in\boldsymbol{\mathcal{A}}_i$, we must have played arm i during the previous time step and transitioned to node π_i. However, the restrictions on the previous joint node depend on whether $\mathsf{depth}(\pi_i)=1$. If so, then arm i was on ρ_i at time step $t-1$, so it's possible to get to π from any joint node in $\boldsymbol{\mathcal{P}}(\pi^{\rho_i})$. That is, in the previous joint node, there could have been an active arm that is not i. This is reflected in (7.7d). On the other hand, if $\mathsf{depth}(\pi_i)>1$, then arm i must have been the active arm at time step $t-1$, as described in (7.7e).

Like before, these equations guarantee that at each time step, we are at exactly one joint node, ie. $\sum_{\pi \in \boldsymbol{S}'} y_{\pi,t} = 1$. Combined with (7.6a), we obtain

$$\sum_{\pi \in \boldsymbol{S}'} \sum_{i \in I(\pi)} \sum_{a \in A} z^a_{\pi,i,t} \leq 1 \qquad\qquad t \in [B] \tag{7.8}$$

Let (ExpLP$'$) denote the linear program defined by objective (7.5) and constraints (7.6a)-(7.6c), (7.7a)-(7.7e) which imply (7.8). This is the dynamic program for *MAB superprocess with multi-period actions—no preemption.*

7.2.4 Reductions from SK and MAB

Before we proceed, we explain why our model captures the problems described in the introduction.

Markovian Bandits can be captured immediately by defining the action set to consist of a single action, and not having any transition times greater than unity. Irrevocable Bandits, the non-preempting variant, can be captured analogously by using the no preemption variant of our problem defined in Section 7.2.3.

We show how to reduce the Stochastic Knapsack variants to Problem 7.2.1, which can then be considered under the non-preempting variant in Section 7.2.3. A job in an instance of correlated Stochastic Knapsack can be given as follows, when processing times are in unary (Gupta et al., 2011a, sect. 2.1). Let B be the total time budget. For each $t \in [B]$, let $\tilde{p}_t$ be the probability of the job finishing after exactly t time steps, and in such a case, let $\tilde{r}_t$ be the reward returned upon completion. There are two problem variants, one where jobs cannot be canceled once started, and another where jobs can be canceled at any time (e.g., after observing that it will not finish before a critical threshold).

Transformation 8 (Correlated SK without Cancellation to Problem 7.2.1). There is a single action and we will ignore the superscript a. The set of nodes is $\{\rho, \phi\}$, with the root node being ρ. For each $t \in [B]$, we set $p_{\rho,\phi,t} = \tilde{p}_t$, with $R_{\rho,\phi,t,t-1}$ taking the deterministic value of $\tilde{r}_t$. This transition represents the job finishing after exactly t time steps, returning a reward upon processing the final time step.

Transformation 9 (Correlated SK with Cancellation to Problem 7.2.1). There is a single action and we will ignore the superscript a. The set of nodes is $\{S_1, \dots, S_B, \phi\}$, with the root node being S_1. Node S_t represents the job processing its t'th time step, and node ϕ represents the job finishing. For each $t \in [B]$, we set $p_{S_t,\phi,1} = \frac{\tilde{p}_t}{\sum_{t' \geq t} \tilde{p}_{t'}}$, the probability of the job finishing upon processing time step t conditioned on it not finishing before then. We set $p_{S_t,S_{t+1},1} = 1 - p_{S_t,\phi,1}$. The rewards are on the transitions from S_t to ϕ, with $R_{S_t,\phi,1,0}$ taking the deterministic value of $\tilde{r}_t$.

We show some examples of these transformations in Section F.1. As previously stated, for both variants of SK, we use the non-preempting adaptation of our problem defined in Section 7.2.3. However, we should point out that for SK with Cancellation, allowing preemption on the jobs under our model results in a distinct problem. In Section F.2, we construct an example where a policy that both preempts and cancels earns more reward than the best policy possible with only cancellation, even when rewards are uncorrelated with processing times.

Finally, we show how to reduce Futuristic Bandits to Problem 7.2.1. In Futuristic Bandits, there are n Markov chains with rewards on nodes. There is a budget of T "exploration" time steps. During these time steps, arms can be played so that they transition to different nodes, but no reward is obtained. At the end of these time steps, each arm is on some final node. Then there is a single "exploit" time step where the greatest reward among the n final nodes is obtained. The objective is to maximize the expected amount exploited.

TRANSFORMATION 10 (FUTURISTIC BANDITS TO PROBLEM 7.2.1). We can use the Markov chains from Futuristic Bandits directly as the arms in Problem 7.2.1. However, we place no rewards on nodes. Instead, we add a separate "exploit" action, which when played on a node, returns the reward of that node. The exploit action has processing time $T + 1$ and we set our time budget B to be $2T + 1$.

This is equivalent to the Futuristic Bandits problem. First, note that it is impossible to collect exploitation rewards from more than one arm. Also, we can explore for at most T time steps if we are going to earn any reward at all. Note that in our problem it is possible to stop exploring before T time steps pass. However, it is never beneficial to do so when the rewards on nodes satisfy the martingale condition, as assumed in Futuristic Bandits. The Budgeted Bandits generalization can also be reduced to Problem 7.2.1 by having different processing times for exploring different arms.

7.2.5 Polynomial-sized LP Relaxations

We now write the polynomial-sized LP relaxations of our earlier problems. We keep track of the probabilities of being on the nodes of each arm individually without considering their joint distribution. Let $s_{u,t}$ be the probability arm i is on node u at the beginning of time t. Let $x^a_{u,t}$ be the probability we play action a on node u at time t.

For both variants of the problem, we have the objective

$$\max \sum_{u \in \mathcal{S}} \sum_{a \in A} r^a_u \sum_{t=1}^{B} x^a_{u,t} \tag{7.9}$$

and constraints on how we can play each individual arm:

$$\sum_{a \in A} x^a_{u,t} \le s_{u,t} \qquad u \in \mathcal{S},\ t \in [B] \tag{7.10a}$$

$$x^\alpha_{u,t} = s_{u,t} \qquad u \in \mathcal{B},\ t \in [B] \tag{7.10b}$$

$$x^a_{u,t} \ge 0 \qquad u \in \mathcal{S},\ a \in A,\ t \in [B] \tag{7.10c}$$

Furthermore, there is a single constraint

$$\sum_{u \in \mathcal{S}} \sum_{a \in A} x^a_{u,t} \le 1 \qquad t \in [B] \tag{7.11}$$

enforcing that the total probabilities of plays across all arms cannot exceed 1 at any time step.

The state-updating constraints differ for the two variants of the problem. If we allow preemption, then they are:

$$s_{\rho_i,1} = 1 \qquad i \in [n] \tag{7.12a}$$

$$s_{u,1} = 0 \qquad u \in \mathcal{S} \setminus \{\rho_1, \dots, \rho_n\} \tag{7.12b}$$

$$s_{u,t} = s_{u,t-1} - \sum_{a \in A} x^a_{u,t-1} + \sum_{(v,a) \in \mathsf{Par}(u)} x^a_{v,t-1} \cdot p^a_{v,u} \qquad t > 1,\ u \in \mathcal{S} \tag{7.12c}$$

If we disallow preemption, then an arm can only be on a non-root node if we played the same arm during the previous time step. This is reflected in (7.13c)-(7.13d):

$$s_{\rho_i,1} = 1 \qquad i \in [n] \tag{7.13a}$$

$$s_{u,1} = 0 \qquad u \in \mathcal{S} \setminus \{\rho_1, \dots, \rho_n\} \tag{7.13b}$$

$$s_{\rho_i,t} = s_{\rho_i,t-1} - \sum_{a \in A} x^a_{\rho_i,t-1} \qquad t > 1,\ i \in [n] \tag{7.13c}$$

$$s_{u,t} = \sum_{(v,a) \in \mathsf{Par}(u)} x^a_{v,t-1} \cdot p^a_{v,u} \qquad t > 1,\ u \in \mathcal{S} \setminus \{\rho_1, \dots, \rho_n\} \tag{7.13d}$$

Let (`PolyLP`) denote the linear program defined by objective (7.9) and constraints (7.10a)-(7.10c), (7.11), (7.12a)-(7.12c). Similarly, let (`PolyLP`$'$) denote the linear program defined by objective (7.9) and constraints (7.10a)-(7.10c), (7.11), (7.13a)-(7.13d). We still have to prove the polynomial-sized linear programs are indeed relaxations of the exponential-sized linear programs. For any linear program LP, let $\mathsf{OPT}_{\mathrm{LP}}$ denote its optimal objective value.

Lemma 7.2.5. *Given a feasible solution* $\{z^a_{\pi,i,t}\}, \{y_{\pi,t}\}$ *to* (ExpLP)*, we can construct a solution to* (`PolyLP`) *with the same objective value by setting* $x^a_{u,t} = \sum_{\pi \in \boldsymbol{\mathcal{S}}: \pi_i = u} z^a_{\pi,i,t}$, $s_{u,t} = \sum_{\pi \in \boldsymbol{\mathcal{S}}: \pi_i = u} y_{\pi,t}$ *for all* $i \in [n], u \in \mathcal{S}_i, a \in A, t \in [B]$*. Thus the feasible region of* (`PolyLP`) *is a projection of that of* (ExpLP) *onto a subspace and* $\mathsf{OPT}_{\mathrm{ExpLP}} \le \mathsf{OPT}_{\mathtt{PolyLP}}$.

Lemma 7.2.6. *Given a feasible solution* $\{z^a_{\pi,i,t}\}, \{y_{\pi,t}\}$ *to* (ExpLP$'$)*, we can construct a solution to* (`PolyLP`$'$) *with the same objective value by setting* $x^a_{u,t} = \sum_{\pi \in \boldsymbol{\mathcal{S}}': \pi_i = u} z^a_{\pi,i,t}$, $s_{u,t} = \sum_{\pi \in \boldsymbol{\mathcal{S}}': \pi_i = u} y_{\pi,t}$ *for all* $i \in [n], u \in \mathcal{S}_i, a \in A, t \in [B]$*. Thus the feasible region of* (`PolyLP`$'$) *is a projection of that of* (ExpLP$'$) *onto a subspace and* $\mathsf{OPT}_{\mathrm{ExpLP}'} \le \mathsf{OPT}_{\mathtt{PolyLP}'}$.

Essentially, Lemma 7.2.5 says that `PolyLP` reduces from `ExpLP`, and Lemma 7.2.6 says that `PolyLP`$'$ reduces from `ExpLP`$'$. Recall that the feasible regions of `ExpLP` and `ExpLP`$'$ correspond exactly to the admissible policies in the two variants. These lemmas say that the performance of any adaptive policy can be upper bounded by the polynomial-sized relaxations. Our lemmas are analogous to similar statements from earlier works (e.g. Gupta et al., 2011a, lem. 2.1), but put into the context of an exponential-sized linear program. Their proofs are mostly analytical and deferred to Section F.3.

7.2.6 Main Results

Now that we have established the preliminaries, we are ready to state our main results in the form of theorems.

Theorem 7.2.7. *Given a feasible solution* $\{x^a_{u,t}\}, \{s_{u,t}\}$ *to* (PolyLP′)*, there exists a solution to* (ExpLP′) *with* $\sum_{\pi:\pi_i=u} z^a_{\pi,i,t} = \frac{1}{2}x^a_{u,t}$, $\sum_{\pi:\pi_i=u} y_{\pi,t} = \frac{1}{2}s_{u,t}$ *for all* $i \in [n], u \in \mathcal{S}_i, a \in A, t \in [B]$*, obtaining reward* $\frac{1}{2}\mathsf{OPT}_{\mathtt{PolyLP'}}$*. We can use sampling to turn this into a* $(\frac{1}{2} - \varepsilon)$*-approximation algorithm for MAB superprocess with multi-period actions—no preemption, with runtime polynomial in the input and* $\frac{1}{\varepsilon}$*.*

We prove this theorem in Section 7.3, and also show that it is tight, constructing an instance under the special case of correlated SK where it is impossible to obtain reward greater than $(\frac{1}{2} + \varepsilon)\mathsf{OPT}_{(\mathtt{PolyLP'})}$.

Theorem 7.2.8. *There is a* (PolyLP)*-relative* $\frac{4}{27}$*-approximation algorithm for MAB superprocess (with preemption), when all processing times are* 1.

Theorem 7.2.9. *There is a* (PolyLP)*-relative* $\frac{1}{12}$*-approximation algorithm for MAB superprocess with multi-period actions (and preemption).*

We prove these theorems in Section 7.4.

7.3 Proof of Theorem 7.2.7

In this section we prove Theorem 7.2.7. To build intuition, we will first present the upper bound, showing a family of examples with $\frac{\mathsf{OPT}_{\mathtt{ExpLP'}}}{\mathsf{OPT}_{\mathtt{PolyLP'}}}$ approaching $\frac{1}{2}$.

7.3.1 Construction for Upper Bound

Let N be a large integer. We will describe our $n = 2$ arms as stochastic jobs. Job 1 takes $N+1$ time with probability $1 - \frac{1}{N}$, in which case it returns a reward of 1. It takes 1 time with probability $\frac{1}{N}$, in which case it returns no reward. Job 2 deterministically takes 1 time and returns a reward of 1. The budget is $B = N + 1$ time steps.

Any actual policy can never get more than 1 reward, since it cannot get a positive reward from both jobs.

After all the reductions from Section 7.2, the Markov Chains representing these jobs can be denoted as follows. Let $\mathcal{S}_1 = \{S_0, S_1, \ldots, S_N, \phi_1\}$, with $\rho_1 = S_0$. There is only one action, and we will omit the action superscripts. The only uncertainty is at S_0, with $p_{S_0,S_1} = 1 - \frac{1}{N}, p_{S_0,\phi_1} = \frac{1}{N}$. The remaining transitions are $p_{S_1,S_2} = \ldots = p_{S_{N-1},S_N} = p_{S_N,\phi_1} = 1$, and self loop on the terminal node $p_{\phi_1,\phi_1} = 1$. The only reward is a reward of 1 on node S_N. Meanwhile, $\mathcal{S}_2$ consists only of nodes $\{\rho_2, \phi_2\}$, with $p_{\rho_2,\phi_2} = p_{\phi_2,\phi_2} = 1$, $r_{\rho_2} = 1$.

Consider the solution for (PolyLP′) with $x_{S_0,1} = 1, x_{S_1,2} = \ldots = x_{S_N,N+1} = 1 - \frac{1}{N}, x = 0, x_{\rho_2,2} = \ldots = x_{\rho_2,N+1} = \frac{1}{N}$, all other x-variables equal to 0, and s-variables determined using (7.13a)-(7.13d). It can be checked that this solution is feasible, and that its objective value of $2 - \frac{1}{N}$ is optimal, since all of the potential reward is acquired. Hence as we take $N \to \infty$, we get $\frac{\mathsf{OPT}_{\mathsf{ExpLP'}}}{\mathsf{OPT}_{\mathsf{PolyLP'}}} = \frac{1}{2}$.

Note that we can put all of $\mathcal{S}_1 \setminus \{\rho_1, \phi_1\}$ in $\mathcal{B}$ if we want; it doesn't change the example whether job 1 can be canceled once started. It also doesn't matter whether we allow preemption—both $\frac{\mathsf{OPT}_{\mathsf{ExpLP}}}{\mathsf{OPT}_{\mathsf{PolyLP}}}$ and $\frac{\mathsf{OPT}_{\mathsf{ExpLP'}}}{\mathsf{OPT}_{\mathsf{PolyLP'}}}$ are $\frac{1}{2} + \varepsilon$ for this example.

Let's analyze what goes wrong when we attempt to replicate the optimal solution to the LP relaxation in an actual policy. We start job 1 at time 1 with probability $x_{S_0,1} = 1$. If it does not terminate after 1 time step, which occurs with probability $1 - \frac{1}{N}$, then we play job 1 through to the end, matching $x_{S_1,2} = \ldots = x_{S_N,N+1} = 1 - \frac{1}{N}$. If it does, then we start job 2 at time 2. This occurs with unconditional probability $x_{\rho_2,2} = \frac{1}{N}$, as planned. However, in this case, we cannot start job 2 again at time 3 (since it has already been processed at time 2), even though $x_{\rho_2,3} = \frac{1}{N}$ is telling us to do so. The LP relaxation fails to consider that event "job 1 takes time 1" is *directly correlated* with event "job 2 is started at time 2", so the positive values specified by $x_{\rho_2,3}, \ldots, x_{\rho_2,N+1}$ are illegal plays.

Motivated by this example, we observe that if we only try to play u at time t with probability $\frac{x_{u,t}}{2}$, then we can obtain a solution to (ExpLP′) (and hence a feasible policy) that is a scaled copy of the solution to (PolyLP′).

7.3.2 Specification of Solution to (ExpLP')

Fix a solution $\{x^a_{u,t}, s_{u,t}\}$ to ($\texttt{PolyLP}'$). Our objective in this subsection is to construct a solution $\{z^a_{\pi,i,t}, y_{\pi,t}\}$ to ($\texttt{ExpLP}'$) such that

$$\sum_{\pi\in\boldsymbol{\mathcal{S}}':\pi_i=u} z^a_{\pi,i,t} = \frac{x^a_{u,t}}{2} \qquad\qquad i\in[n],\ u\in\mathcal{S}_i,\ a\in A \tag{7.14}$$

obtaining half the objective value of ($\texttt{PolyLP}'$). We will prove feasibility in Section 7.3.3.

The intuition for the construction is as follows. For any $u \in \mathcal{S}_i$ and $t \in [B]$, in order to play node u at time t, we must have started playing arm i at time $t - \mathsf{depth}(u)$, since preemption is not allowed. Therefore, it is possible to partition the combinations of u, a, t where $x^a_{u,t} > 0$ according to the time at which we must play ρ_i. Having established this, we only have to make decisions on which new arm to start, when the values of $x^a_{u,t}$ prescribe that the current arm should be stopped. To satisfy the global constraint (7.14), the decision to start arm i at time t (in a specific state) depends on the total probability of *being able to* start arm i at time t (conditioned on the past policy and realizations).

For convenience, define $x_{u,t} = \sum_{a\in A} x^a_{u,t}$ and $z_{\pi,i,t} = \sum_{a\in A} z^a_{\pi,i,t}$. We will complete the specification of $\{z^a_{\pi,i,t}, y_{\pi,t}\}$ over B iterations $t = 1,\dots,B$. On iteration t:

1. Compute $y_{\pi,t}$ for all $\pi\in\boldsymbol{\mathcal{S}}'$, using (7.7a)–(7.7e).

2. Define $\tilde{y}_{\pi,t} = y_{\pi,t}$ if $\pi \notin \boldsymbol{\mathcal{A}}$, and $\tilde{y}_{\pi,t} = y_{\pi,t} - \sum_{a\in A} z^a_{\pi,i,t}$ if $\pi\in\boldsymbol{\mathcal{A}}_i$ for some $i\in[n]$ (if $\pi\in\boldsymbol{\mathcal{A}}_i$, then $\{z^a_{\pi,i,t} : a\in A\}$ has already been set in a previous iteration).

3. For all $i\in[n]$, define $f_{i,t} = \sum_{\pi\in\boldsymbol{\mathcal{S}}':\pi_i=\rho_i} \tilde{y}_{\pi,t}$.

4. For all $i\in[n]$, $\pi\in\boldsymbol{\mathcal{S}}'$ such that $\pi_i=\rho_i$, and $a\in A$, set $z^a_{\pi,i,t} = \tilde{y}_{\pi,t}\cdot\frac{1}{2}\cdot\frac{x^a_{\rho_i,t}}{f_{i,t}}$.

5. For all $i\in[n]$, and $\pi\in\boldsymbol{\mathcal{S}}'$ such that $\pi_i=\rho_i$ and $\pi_j\in\{\rho_j,\phi_j\}$ for $j\neq i$, define $g_{\pi,i,t} = \sum_{\pi'\in\boldsymbol{\mathcal{P}}(\pi)} z_{\pi',i,t}$.

6. For all $i\in[n]$, $u\in\mathcal{S}_i\setminus\{\rho_i\}$, $\pi\in\boldsymbol{\mathcal{S}}'$ such that $\pi_i=u$, and $a\in A$, set $z^a_{\pi,i,t+\mathsf{depth}(u)} = g_{\pi^{\rho_i},i,t}\cdot\frac{x^a_{u,t+\mathsf{depth}(u)}}{x_{\rho_i,t}}$.

In Step 2, $\tilde{y}_{\pi,t}$ represents the probability that we are at joint node π and looking to start a new arm at time t, abandoning the arm in progress if there is any. In Step 3, $f_{i,t}$ is the total probability of *being able to* start arm i at time t, which we define as arm i being available when we are looking to start a new arm at time t. The normalization in Step 4 ensures that each arm is started with the correct probability at time t. In Step 5, $g_{\pi,i,t}$ is the probability arm i is started at time t, and other arms are on nodes $\{\pi_j : j \neq i\}$ while arm i executes (another arm j could have made a transition to ϕ_j during the first time step t). Step 6 specifies how to continue playing arm i in subsequent time steps if it is started at time t. Note that $g_{\pi^{\rho_i},i,t}$ is guaranteed to be defined in this case, since $\pi_i \notin \{\rho_i, \phi_i\}$ and $\pi \in \boldsymbol{\mathcal{S}}'$ implies $\pi_j \in \{\rho_j, \phi_j\}$ for all $j \neq i$.

This completes the specification of the solution to (ExpLP$'$). Every $y_{\pi,t}$ is set in Step 1, and every $z^a_{\pi,i,t}$ is set in either Step 4 or Step 6.

Using the definition of $f_{i,t}$, Step 4 guarantees that for $i \in [n]$, $a \in A$,

$$\begin{aligned}
\sum_{\pi \in \boldsymbol{\mathcal{S}}' : \pi_i = \rho_i} z^a_{\pi,i,t} &= \sum_{\pi \in \boldsymbol{\mathcal{S}}' : \pi_i = \rho_i} \tilde{y}_{\pi,t} \cdot \frac{1}{2} \cdot \frac{x^a_{\rho_i,t}}{f_{i,t}} \\
&= \frac{x^a_{\rho_i,t}}{2}
\end{aligned}$$

Meanwhile, Step 6 guarantees that for $i \in [n]$, $u \in \mathcal{S}_i \setminus \{\rho_i\}$, $a \in A$,

$$\begin{aligned}
\sum_{\pi \in \boldsymbol{\mathcal{S}}' : \pi_i = u} z^a_{\pi,i,t+\mathsf{depth}(u)} &= \sum_{\pi \in \boldsymbol{\mathcal{S}}' : \pi_i = u} g_{\pi^{\rho_i},i,t} \cdot \frac{x^a_{u,t+\mathsf{depth}(u)}}{x_{\rho_i,t}} \\
&= \frac{x_{\rho_i,t}}{2} \cdot \frac{x^a_{u,t+\mathsf{depth}(u)}}{x_{\rho_i,t}} \\
&= \frac{x^a_{u,t+\mathsf{depth}(u)}}{2}
\end{aligned}$$

We explain the second equality. Since $u \neq \rho_i$ implies arm i is the active arm in all of $\{\pi \in \boldsymbol{\mathcal{S}}' : \pi_i = u\}$, this set is equal to $\{\rho_1, \phi_1\} \times \cdots \times u \times \cdots \times \{\rho_n, \phi_n\}$. Summing $g_{\pi^{\rho_i},i,t}$ over all the possibilities for $\{\pi_j : j \neq i\}$ yields the total probability arm i is started at time t. This is equal to $\sum_{\pi \in \boldsymbol{\mathcal{S}}' : \pi_i = \rho_i} \sum_{a \in A} z^a_{\pi,i,t}$, which by the first calculation is equal to $\frac{x_{\rho_i,t}}{2}$.

The proof of (7.14) is now complete.

7.3.3 Proof of Feasibility

At a high level, the main challenge in proving feasibility is verifying (7.6a), i.e. the total probability of plays scheduled for joint node π and time t does not exceed the probability of being at joint node π and time t. Given the way the solution was constructed, it is mostly an analytical exercise (Lemma 7.3.1) to reduce (7.6a) to showing that the values of $f_{i,t}$ are sufficiently large. $f_{i,t}$ is the probability of being able to start arm i at time t, which we are not able to do if the current arm has not been prescribed to stop, or if arm i has already been played. Lemma 7.3.2 shows that the probability of these events occurring is just small enough when the probabilities of the LP relaxation, $x^a_{u,t}$, are scaled down by a factor of 2.

We will inductively prove feasibility over iterations $t = 1, \ldots, B$. Suppose all of the variables $\{z^a_{\pi,i,t'}, y_{\pi,t'}\}$ with $t' < t$ have already been set in a way that satisfies constraints (7.6a)-(7.6c), (7.7a)-(7.7e). Some of the variables $z^a_{\pi,i,t'}$ with $t' \geq t$ may have also been set in Step 6 of earlier iterations; if so, suppose they have already been proven to satisfy (7.6c).

On iteration t, we first compute in Step 1 $y_{\pi,t}$ for all $\pi \in \boldsymbol{\mathcal{S}}'$; these are guaranteed to satisfy (7.7a)-(7.7e) by definition. To complete the induction, we need to show that (7.6a)-(7.6c) hold after setting the z-variables in Step 4, and furthermore, (7.6c) holds for any $z^a_{\pi,i,t'}$ (with $t' > t$) we set in Step 6.

We first prove the following lemma:

Lemma 7.3.1. *Suppose $\pi \in \boldsymbol{\mathcal{A}}_i$ for some $i \in [n]$. Let $u = \pi_i$ (which is neither ρ_i nor ϕ_i). Then $\sum_{a \in A} z^a_{\pi,i,t} \leq y_{\pi,t}$, and furthermore if $u \in \mathcal{B}$, then $z^{\alpha}_{\pi,i,t} = y_{\pi,t}$.*

Proof. Proof. First suppose $\mathsf{depth}(u) = 1$. (7.7d) says $y_{\pi,t} = \sum_{a:(\rho_i,a)\in\mathsf{Par}(u)}(\sum_{\pi'\in\boldsymbol{\mathcal{P}}(\pi^{\rho_i})} z^a_{\pi',i,t-1}) \cdot p^a_{\rho_i,u}$. Every π' in the sum has $\pi'_i = \rho_i$, so $z^a_{\pi',i,t-1}$ was set in Step 4 of iteration $t-1$ to $\tilde{y}_{\pi',t-1} \cdot \frac{1}{2} \cdot \frac{x^a_{\rho_i,t-1}}{f_{i,t-1}}$. Substituting into (7.7d), we get

$$\begin{aligned} y_{\pi,t} &= \sum_{a:(\rho_i,a)\in\mathsf{Par}(u)} \Big(\sum_{\pi'\in\boldsymbol{\mathcal{P}}(\pi^{\rho_i})} \tilde{y}_{\pi',t-1} \cdot \frac{1}{2} \cdot \frac{x^a_{\rho_i,t-1}}{f_{i,t-1}} \Big) \cdot p^a_{\rho_i,u} \\ &= \Big(\sum_{\pi'\in\boldsymbol{\mathcal{P}}(\pi^{\rho_i})} \tilde{y}_{\pi',t-1} \Big) \cdot \frac{1}{2f_{i,t-1}} \cdot \sum_{a:(\rho_i,a)\in\mathsf{Par}(u)} x^a_{\rho_i,t-1} \cdot p^a_{\rho_i,u} \end{aligned}$$

Meanwhile, for all $a \in A$, $z^a_{\pi,i,t}$ was set in Step 6 of iteration $t-1$ to $g_{\pi^{\rho_i},i,t-1} \cdot \frac{x^a_{u,t}}{x_{\rho_i,t-1}}$. Hence

$$\begin{aligned}
z^a_{\pi,i,t} &= g_{\pi^{\rho_i},i,t-1} \cdot \frac{x^a_{u,t}}{x_{\rho_i,t-1}} \\
&= \sum_{\pi' \in \boldsymbol{\mathcal{P}}(\pi^{\rho_i})} z_{\pi',i,t-1} \cdot \frac{x^a_{u,t}}{x_{\rho_i,t-1}} \\
&= \sum_{\pi' \in \boldsymbol{\mathcal{P}}(\pi^{\rho_i})} \Big(\sum_{b \in A} \tilde{y}_{\pi',t-1} \cdot \frac{1}{2} \cdot \frac{x^b_{\rho_i,t-1}}{f_{i,t-1}} \Big) \cdot \frac{x^a_{u,t}}{x_{\rho_i,t-1}} \\
&= \Big(\sum_{\pi' \in \boldsymbol{\mathcal{P}}(\pi^{\rho_i})} \tilde{y}_{\pi',t-1} \Big) \cdot \frac{1}{2f_{i,t-1}} \cdot x^a_{u,t}
\end{aligned}$$

where the second equality is by the definition of $g_{\pi^{\rho_i},i,t-1}$, and the third equality uses the fact that $z^b_{\pi',i,t-1}$ was set in Step 4 of iteration $t-1$. To prove $\sum_{a \in A} z^a_{\pi,i,t} \le y_{\pi,t}$, it suffices to show $\sum_{a \in A} x^a_{u,t} \le \sum_{a:(\rho_i,a) \in \mathsf{Par}(u)} x^a_{\rho_i,t-1} \cdot p^a_{\rho_i,u}$. This follows immediately from combining constraints (7.10a) and (7.13d) of ($\texttt{PolyLP}'$). Furthermore, if $u \in \mathcal{B}$, then we can use (7.10b) to get $z^{\alpha}_{\pi,i,t} = y_{\pi,t}$.

Now suppose $\mathsf{depth}(u) > 1$. (7.7e) says $y_{\pi,t} = \sum_{(v,a) \in \mathsf{Par}(u)} z^a_{\pi^v,i,t-1} \cdot p^a_{v,u}$. Since $v \ne \rho_i$, $z^a_{\pi^v,i,t-1}$ was set in Step 6 of iteration $t' := t - \mathsf{depth}(u)$ to $g_{\pi^{\rho_i},i,t'} \cdot \frac{x^a_{v,t-1}}{x_{\rho_i,t'}}$. Substituting into (7.7e), we get

$$\begin{aligned}
y_{\pi,t} &= \sum_{(v,a) \in \mathsf{Par}(u)} g_{\pi^{\rho_i},i,t'} \cdot \frac{x^a_{v,t-1}}{x_{\rho_i,t'}} \cdot p^a_{v,u} \\
&= \frac{g_{\pi^{\rho_i},i,t'}}{x_{\rho_i,t'}} \sum_{(v,a) \in \mathsf{Par}(u)} x^a_{v,t-1} \cdot p^a_{v,u}
\end{aligned}$$

Meanwhile, for all $a \in A$, $z^a_{\pi,i,t}$ was set in Step 6 of iteration t' to $g_{\pi^{\rho_i},i,t'} \cdot \frac{x^a_{u,t}}{x_{\rho_i,t'}}$. To prove $\sum_{a \in A} z^a_{\pi,i,t} \le y_{\pi,t}$, it suffices to show $\sum_{a \in A} x^a_{u,t} \le \sum_{(v,a) \in \mathsf{Par}(u)} x^a_{v,t-1} \cdot p^a_{v,u}$. This is again obtained from (7.10a) and (7.13d), and if $u \in \mathcal{B}$, then we can use (7.10b) to get $z^{\alpha}_{\pi,i,t} = y_{\pi,t}$. □

By the lemma, $\tilde{y}_{\pi,t} \ge 0$ if $\pi \in \boldsymbol{\mathcal{A}}_i$ for some $i \in [n]$. On the other hand, $\tilde{y}_{\pi,t} \ge 0$ is immediate from definition if $\pi \notin \boldsymbol{\mathcal{A}}$. Therefore, $\tilde{y}_{\pi,t} \ge 0$ for all $\pi \in \boldsymbol{\mathcal{S}}'$, and (7.6c) is satisfied by all the z-variables set in Step 4 or Step 6. Furthermore, the lemma guarantees (7.6b) for the $z^a_{\pi,i,t}$ with $\pi_i \in \mathcal{B}$ set in previous iterations.

It remains to prove (7.6a). If $\pi \in \boldsymbol{\mathcal{A}}_i$, then the LHS of (7.6$a$) is

$$\begin{aligned}\sum_{j \in I(\pi)} z_{\pi,j,t} &= z_{\pi,i,t} + \sum_{j \in I(\pi)\setminus\{i\}} \tilde{y}_{\pi,t} \cdot \frac{1}{2} \cdot \frac{x_{\rho_j,t}}{f_{j,t}} \\ &= z_{\pi,i,t} + (y_{\pi,t} - z_{\pi,i,t}) \cdot \sum_{j \in I(\pi)\setminus\{i\}} \frac{1}{2} \cdot \frac{x_{\rho_j,t}}{f_{j,t}}\end{aligned}$$

For the first equality, note that $z_{\pi,j,t}$ for $j \neq i$ is set in Step 4 of the current iteration, but $z_{\pi,i,t}$ has already been set in an earlier iteration. The second equality is immediate from the definition of $\tilde{y}_{\pi,t}$. Note that $y_{\pi,t} - z_{\pi,i,t} \geq 0$, by Lemma 7.3.1. If we knew $\sum_{j \in I(\pi)\setminus\{i\}} \frac{1}{2} \cdot \frac{x_{\rho_j,t}}{f_{j,t}} \leq 1$, then we would have $\sum_{j \in I(\pi)} z_{\pi,j,t} \leq z_{\pi,i,t} + (y_{\pi,t} - z_{\pi,i,t})(1) = y_{\pi,t}$, which is (7.6a).

On the other hand, if $\pi \notin \boldsymbol{\mathcal{A}}$, then the LHS of (7.6$a$) is $y_{\pi,t} \cdot \sum_{j \in I(\pi)} \frac{1}{2} \cdot \frac{x_{\rho_j,t}}{f_{j,t}}$, where in this case all of the $z_{\pi,j,t}$ are set in Step 4 of the current iteration. Similarly, if we knew $\sum_{j \in I(\pi)} \frac{1}{2} \cdot \frac{x_{\rho_j,t}}{f_{j,t}} \leq 1$, then we would have (7.6a).

To complete the proof of feasibility, it suffices to show $\sum_{j=1}^{n} \frac{1}{2} \cdot \frac{x_{\rho_j,t}}{f_{j,t}} \leq 1$ (note that $f_{j,t}$ is always non-negative, by its definition and Lemma 7.3.1). This is implied by the following lemma, which proves a simpler statement:

Lemma 7.3.2. *$f_{i,t} \geq \sum_{j=1}^{n} \frac{x_{\rho_j,t}}{2}$ for all $i \in [n]$.*

Proof. Proof. Fix some $i \in [n]$. By the definitions in Step 2 and Step 3,

$$f_{i,t} = \sum_{\pi \in \boldsymbol{S}':\pi_i=\rho_i} y_{\pi,t} - \sum_{j \neq i} \sum_{\pi \in \boldsymbol{\mathcal{A}}_j:\pi_i=\rho_i} z_{\pi,j,t}$$

Let's start by bounding $\sum_{\pi \in \boldsymbol{S}':\pi_i=\rho_i} y_{\pi,t}$, the total probability arm i is still on ρ_i at the start of time t. This is equal to $1 - \sum_{t'<t} \sum_{\pi \in \boldsymbol{S}':\pi_i=\rho_i} z_{\pi,i,t'}$, where we subtract from 1 the total probability arm i was initiated before time t. By (7.14), $\sum_{\pi \in \boldsymbol{S}':\pi_i=\rho_i} z_{\pi,i,t'} = \frac{x_{\rho_i,t'}}{2}$ for all $t' < t$. Furthermore,

$$\sum_{t'<t} x_{\rho_i,t'} \leq 1 \tag{7.15}$$

from iteratively applying (7.13c) to (7.13a), and combining with (7.10a)[7]. Therefore, $\sum_{\pi \in \boldsymbol{S}':\pi_i=\rho_i} y_{\pi,t} \geq \frac{1}{2}$.

[7]Intuitively, we're arguing that solution to the LP relaxation still satisfies the total probability arm i being played from its root node not exceeding unity.

Now we bound the remaining term in the equation for $f_{i,t}$:

$$\begin{aligned}
\sum_{j\neq i}\sum_{\pi\in\mathcal{A}_j:\pi_i=\rho_i} z_{\pi,j,t} &= \sum_{j\neq i}\sum_{v\in\mathcal{S}_j\setminus\{\rho_j\}}\sum_{\pi:\pi_i=\rho_i,\pi_j=v} z_{\pi,j,t} \\
&\leq \sum_{j\neq i}\sum_{v\in\mathcal{S}_j\setminus\{\rho_j\}}\sum_{\pi:\pi_j=v} z_{\pi,j,t} \\
&= \frac{1}{2}\sum_{j\neq i}\sum_{v\in\mathcal{S}_j\setminus\{\rho_j\}} x_{v,t} \\
&\leq \frac{1}{2}\sum_{j=1}^{n}\sum_{v\in\mathcal{S}_j\setminus\{\rho_j\}} x_{v,t} \\
&\leq \frac{1}{2}\left(1-\sum_{j=1}^{n} x_{\rho_j,t}\right)
\end{aligned}$$

The first inequality uses the non-negativity of $z_{\pi,j,t}$ in the inductively proven (7.6c), the second equality uses (7.14), the second inequality uses the non-negativity of $x_{v,t}$ in (7.10c), and the third inequality uses (7.11).

Combining the two terms, we get $f_{i,t} \geq \sum_{j=1}^{n}\frac{x_{\rho_j,t}}{2}$, as desired. □

7.3.4 Approximation Algorithm via Sampling

We would like to infer a polynomial-time policy from this exponential-sized solution $\{z^a_{\pi,i,t}, y_{\pi,t}\}$ of (ExpLP$'$). It is not possible to merely compute the values of $z^a_{\pi,i,t}$ on the realized sample path, because $z^a_{\pi,i,t}$ depends on $f_{i,t}$, the total probability of being able to start arm i at time t over exponentially many states. To overcome this challenge, at each time step t, we *sample* (run the algorithm up to time t a large number of times, realizing new transitions each time) to estimate the values of $f_{i,t}$, before making a decision. We record the sampling results and the decisions prescribed by such, since future sampling depends on past algorithm decisions.

Hereinafter we will assume that the $\{x^a_{u,t}, s_{u,t}\}$ we are imitating is an *optimal* solution of (PolyLP$'$). Consider the following algorithm, which takes in as parameters a terminal time step $t\in[B]$, and probabilities $\lambda_{i,t'}$ for each $i\in[n], t'\leq t$ (which for now should be considered to be $f_{i,t'}$ to aid in the comprehension of the algorithm):

Policy(t, $\{\lambda_{i,t'} : i\in[n], t'\leq t\}$)

- Initialize $t' = 1$, $\mathsf{current} = 0$.
- While $t' \leq t$:

  1. If $\mathsf{current} = 0$, then
     (a) For each arm i that is on ρ_i, set $\mathsf{current} = i$ with probability $\frac{1}{2} \cdot \frac{x_{\rho_i,t'}}{\lambda_{i,t'}}$; if the sum of these probabilities exceeds 1 (ie. this step is inadmissible), then terminate with no reward.
     (b) If $\mathsf{current}$ was set in this way, leave t' unchanged and enter the next if block. Otherwise, leave $\mathsf{current}$ at 0 but increment t' by 1.
  2. If $\mathsf{current} \neq 0$, then
     (a) Let u denote the node arm $\mathsf{current}$ is on. For each $a \in A$, play action a on arm $\mathsf{current}$ with probability $\frac{x^a_{u,t'}}{x_{u,t'}}$.
     (b) Suppose we transition onto node v as a result of this play. With probability $\frac{x_{v,t'+1}}{s_{v,t'+1}}$, leave $\mathsf{current}$ unchanged. Otherwise, set $\mathsf{current} = 0$.
     (c) Increment t' by 1.

Define the following events and probabilities, which depend on the input passed into Policy:

- For all $i \in [n], t' \leq (t+1)$, let $\mathcal{A}_{i,t'}$ be the event that at the beginning of time t', $\mathsf{current} = 0$ and arm i is on ρ_i. Let $\mathsf{Free}(i,t') = \Pr[\mathcal{A}_{i,t'}]$.
- For all $i \in [n], t' \leq t$, let $\mathsf{Started}(i,t')$ be the probability that we play arm i from ρ_i at time t'.
- For all $u \in \mathcal{S}, a \in A, t' \leq t$, let $\mathsf{Played}(u,a,t')$ be the probability that we play action a on node u at time t'.

It is easy to see that Policy is an algorithmic specification of feasible solution $\{z^a_{\pi,i,t}, y_{\pi,t}\}$ if we run it on input $(B, \{f_{i,t} : i \in [n], t \in [B]\})$. Indeed, we would iteratively have for $t = 1, \ldots, B$:

- $\mathsf{Free}(i,t) = f_{i,t}$ for all $i \in [n]$
- $\mathsf{Started}(i,t) = \mathsf{Free}(i,t) \cdot \frac{1}{2} \cdot \frac{x_{\rho_i,t}}{f_{i,t}} = \frac{x_{\rho_i,t}}{2}$ for all $i \in [n]$

- $\mathsf{Played}(u,a,t) = \mathsf{Started}(i, t-\mathsf{depth}(u)) \cdot \frac{x^a_{u,t}}{x_{\rho_i, t-\mathsf{depth}(u)}} = \frac{x^a_{u,t}}{2}$ for all $u \in \mathcal{S}, a \in A$

The final statement can be seen inductively:

$$\begin{aligned}
\mathsf{Played}(u,a,t) &= \sum_{(v,b)\in\mathsf{Par}(u)} \mathsf{Played}(v,b,t-1)\cdot p^b_{v,u}\cdot\frac{x^a_{u,t}}{s_{u,t}} \\
&= \sum_{(v,b)\in\mathsf{Par}(u)} \Big(\mathsf{Started}(i,t-1-\mathsf{depth}(v))\cdot\frac{x^b_{v,t-1}}{x_{\rho_i,t-1-\mathsf{depth}(v)}}\Big)\cdot p^b_{v,u}\cdot\frac{x^a_{u,t}}{s_{u,t}} \\
&= \mathsf{Started}(i,t-\mathsf{depth}(u))\cdot\frac{x^a_{u,t}}{x_{\rho_i,t-\mathsf{depth}(u)}} \qquad (7.16)
\end{aligned}$$

where the first equality is by Steps 2a-b, the second equality is by the induction hypothesis, and the final equality is by (7.13d).

Therefore, we would have a $\frac{1}{2}$-approximation if we knew $\{f_{i,t} : i \in [n], t \in [B]\}$, but unfortunately computing $f_{i,t}$ requires summing exponentially many terms. We can try to approximate it by sampling, but we can't even generate a sample from the binary distribution with probability $f_{i,t}$ since that requires knowing the exact values of $f_{i,t'}$ for $t' < t$. So we give up trying to approximate $f_{i,t}$, and instead iteratively approximate the values of $\mathsf{Free}(i,t)$ when Policy is ran on previously approximated $\mathsf{Free}(i,t)$ values.

Fix some small $\varepsilon, \delta > 0$ that will be determined later. Let $\mu_{\varepsilon,\delta} = \frac{3\ln(2\delta^{-1})}{\varepsilon^2}$. Change Policy so that the probabilities in Step 1a are multiplied by $(1-\varepsilon)^2$ (and change the definitions of $\mathcal{A}_{i,t'}, \mathsf{Free}, \mathsf{Started}, \mathsf{Played}$ accordingly).

Sampling Algorithm

- Initialize $\mathsf{Free}^{\mathsf{emp}}(i,1) = 1$ for all $i \in [n]$.
- For $t = 2, \ldots, B$:
  1. Run $\mathsf{Policy}(t-1, \{\mathsf{Free}^{\mathsf{emp}}(i,t') : i \in [n], t' < t\})$ a total of $M = \frac{8|\mathcal{S}|B}{\varepsilon}\cdot\mu_{\varepsilon,\delta}$ times. For all $i \in [n]$, let $C_{i,t}$ count the number of times event $\mathcal{A}_{i,t}$ occurred.
  2. For each $i \in [n]$, if $C_{i,t} > \mu_{\varepsilon,\delta}$, set $\mathsf{Free}^{\mathsf{emp}}(i,t) = \frac{C_{i,t}}{M}$; otherwise set $\mathsf{Free}^{\mathsf{emp}}(i,t) = \sum_{j=1}^n \frac{x_{\rho_j,t}}{2}$.

Consider iteration t of Sampling Algorithm. $\{\mathsf{Free}^{\mathsf{emp}}(i,t') : i \in [n], t' < t\}$ have already been finalized, and we are sampling event $\mathcal{A}_{i,t}$ when (the ε-modified) Policy is ran on those finalized approximations to record values for $\{\mathsf{Free}^{\mathsf{emp}}(i,t) : i \in [n]\}$. For all $i \in [n]$, if $C_{i,t} > \mu_{\varepsilon,\delta}$, then the probability of $\frac{C_{i,t}}{M}$ lying in $\big((1-\varepsilon)\cdot\mathsf{Free}(i,t), (1+\varepsilon)\cdot\mathsf{Free}(i,t)\big)$ is at least[8] $1-\delta$. As far as when we have $C_{i,t} > \mu_{\varepsilon,\delta}$, note that if $\mathsf{Free}(i,t) > \frac{\varepsilon}{4|\mathcal{S}|B}$, then $\mathbb{E}[C_{i,t}] > 2\mu_{\varepsilon,\delta}$, so the Chernoff bound says $\Pr[C_{i,t} \leq \mu_{\varepsilon,\delta}] = O(\delta^{\frac{1}{\varepsilon^2}}) = O(\delta)$. We have discussed two $O(\delta)$ probability events in this paragraph of sampling/Chernoff yielding an unlikely and undesired result; call these events *failures*.

By the union bound, the probability of having any failure over iterations $t = 2, \ldots, B$ is at most $2(B-1)n(\delta + O(\delta)) = O(Bn\delta)$. Assuming no failures, we will inductively prove

$$\frac{(1-\varepsilon)^2}{1+\varepsilon}\cdot\frac{x_{\rho_i,t}}{2} \leq \mathsf{Started}(i,t) \leq \max\big\{(1-\varepsilon)\cdot\frac{x_{\rho_i,t}}{2}, \frac{\varepsilon}{4|\mathcal{S}|B}\big\} \tag{7.17}$$

for all $i \in [n]$. This is clear when $t = 1$ since $\mathsf{Started}(i,1) = \frac{x_{\rho_i,1}}{2}$ exactly for all $i \in [n]$.

Now suppose $t \geq 2$. We will first prove a lemma on the true probabilities $\mathsf{Free}(i,t)$, which is the "approximate" version of Lemma 7.3.2:

Lemma 7.3.3. *Suppose* Policy *is ran on input* $(t-1, \{\mathsf{Free}^{\mathsf{emp}}(i,t') : i \in [n], t' < t\})$ *and there were no failures while obtaining the sample average approximations* $\mathsf{Free}^{\mathsf{emp}}(i,t')$. *Then for all* $i \in [n]$, $\mathsf{Free}(i,t) \geq \frac{1}{2}\sum_{j=1}^{n} x_{\rho_j,t}$.

Proof. Proof. We know that event $\mathcal{A}_{i,t}$ will occur if at time t, arm i has not yet been started, and no other arm is active. By the union bound, $1 - \mathsf{Free}(i,t) \leq \sum_{t'<t}\mathsf{Started}(i,t') + \sum_{j=1}^{n}\sum_{u\in\mathcal{S}_j\setminus\{\rho_j\}}\sum_{a\in A}\mathsf{Played}(u,a,t)$. Assuming (7.17) holds, we can bound

$$\begin{aligned}
\sum_{t'<t}\mathsf{Started}(i,t') &\leq \sum_{t'<t}\big((1-\varepsilon)\cdot\frac{x_{\rho_i,t'}}{2} + \frac{\varepsilon}{4|\mathcal{S}|B}\big) \\
&\leq \frac{1-\varepsilon}{2}\sum_{t'<t}x_{\rho_i,t'} + \frac{\varepsilon}{4|\mathcal{S}|} \\
&\leq \frac{1-\varepsilon}{2} + \frac{\varepsilon}{4}
\end{aligned}$$

[8]This is because $\mathsf{Free}^{\mathsf{emp}}(i,t)$ is an average over $M \geq \frac{8}{\varepsilon}\cdot\mu_{\varepsilon,\delta}$ runs, which is enough samples to guarantee this probability; see Motwani and Raghavan (2010).

where the final inequality uses (7.15). Similarly, assuming (7.17) holds, we can bound

$$\begin{aligned}
\sum_{j=1}^{n}\sum_{u\in\mathcal{S}_j\setminus\{\rho_j\}}\sum_{a\in A}\mathsf{Played}(u,a,t) &= \sum_{j=1}^{n}\sum_{u\in\mathcal{S}_j\setminus\{\rho_j\}}\mathsf{Started}(i,t-\mathsf{depth}(u))\cdot\frac{x_{u,t}}{x_{\rho_i,t-\mathsf{depth}(u)}} \\
&\leq \sum_{j=1}^{n}\sum_{u\in\mathcal{S}_j\setminus\{\rho_j\}}\Big((1-\varepsilon)\cdot\frac{x_{\rho_i,t-\mathsf{depth}(u)}}{2}+\frac{\varepsilon}{4|\mathcal{S}|B}\Big)\cdot\frac{x_{u,t}}{x_{\rho_i,t-\mathsf{depth}(u)}} \\
&\leq \frac{1-\varepsilon}{2}\sum_{j=1}^{n}\sum_{u\in\mathcal{S}_j\setminus\{\rho_j\}}x_{u,t}+\frac{\varepsilon}{4B} \\
&\leq \frac{1}{2}\Big(1-\sum_{j=1}^{n}x_{\rho_j,t}\Big)+\frac{\varepsilon}{4}
\end{aligned}$$

where the equality uses (7.16), the second inequality uses the fact that $x_{u,t} \leq x_{\rho_i,t-\mathsf{depth}(u)}$, and the final inequality uses (7.11). Combining these bounds completes the proof of the lemma. □

By the description in Step 1a of Policy, for all $i \in [n]$, we have

$$\mathsf{Started}(i,t) = \mathsf{Free}(i,t)\cdot\frac{1}{2}\cdot\frac{x_{\rho_i,t}}{\mathsf{Free}^{\mathsf{emp}}(i,t)}\cdot(1-\varepsilon)^2 \tag{7.18}$$

If $C_{i,t} > \mu_{\varepsilon,\delta}$, then $\mathsf{Free}^{\mathsf{emp}}(i,t)$ will be set to $\frac{C_{i,t}}{M}$, and furthermore no failures implies $(1-\varepsilon)\cdot\mathsf{Free}(i,t) \leq \frac{C_{i,t}}{M} \leq (1+\varepsilon)\cdot\mathsf{Free}(i,t)$. Substituting into (7.18), we get $\frac{(1-\varepsilon)^2}{1+\varepsilon}\cdot\frac{x_{\rho_i,t}}{2} \leq \mathsf{Started}(i,t) \leq (1-\varepsilon)\cdot\frac{x_{\rho_i,t}}{2}$ which implies (7.17). On the other hand, if $C_{i,t} \leq \mu_{\varepsilon,\delta}$, then $\mathsf{Free}^{\mathsf{emp}}(i,t)$ will be set to $\sum_{j=1}^{n}\frac{x_{\rho_j,t}}{2}$, and assuming no failures it must have been the case that $\mathsf{Free}(i,t) \leq \frac{\varepsilon}{4|\mathcal{S}|B}$. Substituting into (7.18), we get $\mathsf{Started}(i,t) \leq \frac{\varepsilon}{4|\mathcal{S}|B}\cdot\frac{1}{2}\cdot\frac{x_{\rho_i,t}}{\sum_{j=1}^{n}x_{\rho_j,t}}\cdot(1-\varepsilon)^2 \leq \frac{\varepsilon}{4|\mathcal{S}|B}$ which implies the upper bound in (7.17). For the lower bound, Lemma 7.3.3 says $\mathsf{Free}(i,t) \geq \mathsf{Free}^{\mathsf{emp}}(i,t)$, so $\mathsf{Started}(i,t) \geq (1-\varepsilon)^2\cdot\frac{x_{\rho_i,t}}{2} \geq \frac{(1-\varepsilon)^2}{1+\varepsilon}\cdot\frac{x_{\rho_i,t}}{2}$.

This completes the induction for (7.17). The final thing to check is that with these new parameters $\{\mathsf{Free}^{\mathsf{emp}}(i,t) : i \in [n]\}$, the sum of the probabilities in Step 1a of Policy does not exceed 1. $\mathsf{Free}^{\mathsf{emp}}(i,t)$ will either get set to $\sum_{j=1}^{n}\frac{x_{\rho_j,t}}{2}$, or be at least $(1-\varepsilon)\cdot\mathsf{Free}(i,t)$, which is at least $(1-\varepsilon)\cdot\sum_{j=1}^{n}\frac{x_{\rho_j,t}}{2}$ by Lemma 7.3.3. In either case, $\mathsf{Free}^{\mathsf{emp}}(i,t) \geq (1-\varepsilon)\cdot\sum_{j=1}^{n}\frac{x_{\rho_j,t}}{2}$ for all $i \in [n]$, so the desired sum in Step 1a is at most $\frac{1}{1-\varepsilon}\cdot(1-\varepsilon)^2 \leq 1$.

We have an algorithm that fails with probability $O(Bn\delta)$, and when it doesn't fail, $\mathsf{Started}(i,t) \geq \frac{(1-\varepsilon)^2}{1+\varepsilon}\cdot\frac{x_{\rho_i,t}}{2}$ for all $i \in [n], t \in [B]$, which in conjunction with (7.16) shows that we obtain expected reward at least $\frac{(1-\varepsilon)^2}{1+\varepsilon} \cdot \frac{1}{2} \cdot \mathsf{OPT}_{\mathtt{PolyLP'}}$. Recall from Lemma 7.2.6 that $\mathsf{OPT}_{\mathtt{PolyLP'}} \geq \mathsf{OPT}_{\mathtt{ExpLP'}}$. Treating a failed run as a run with 0 reward, we can set $\delta = \Theta(\frac{\varepsilon}{Bn})$ to get a $(\frac{1}{2} - \varepsilon)$-approximation. Finally, note that the runtime of this approximation algorithm is polynomial in the input, $\frac{1}{\varepsilon}$, and $\ln(\frac{1}{\delta})$, completing the proof of Theorem 7.2.7.

7.4 Proof of Theorem 7.2.8

In this section we prove Theorem 7.2.8, and also show how to modify the proof to prove Theorem 7.2.9.

7.4.1 Description of Algorithm

The high-level description of the algorithm is as follows. A priority index, which is a time step t in $[B]$, is maintained for each arm. To start, the algorithm plays the arm, say i, with the lowest priority index[9]. Arm i will make a transition and its index will evolve. Arm i is played until it reaches a point where its index is much greater than the depth of the node it is on. Once this occurs, the algorithm switches to the arm that now has the smallest index, and repeats this process. Recall that switching back and forth between arms, i.e. preemption, is necessary for an algorithm to be within a constant factor of optimality. On the other hand, the constraint based on depth ensures that the algorithm does not switch away after an arm has been played a large number of times to reach a high-reward node. Our priority-based policy is motivated by the ideas from (Gupta et al., 2011a, sect. 4–5).

Fix an optimal solution $\{x^a_{u,t}, s_{u,t}\}$ to (PolyLP). The priority indices are maintained based on this solution. For an arm on node u, it will always have some *status* (u, a, t), which says that the next play of the arm should be with action a, and that this play has priority t. We allow $t = \infty$ to indicate that the algorithm will never try to play the arm again; in this case we omit the action argument.

We initialize each arm i to status (ρ_i, a, t) with probability $\frac{x^a_{\rho_i,t}}{C}$, for all $a \in A$ and $t \in [B]$, where $C > 0$ is some constant to be optimized later. With probability $1 - \sum_{a \in A} \sum_{t=1}^{B} \frac{x^a_{\rho_i,t}}{C}$, the arm is initialized to status (ρ_i, ∞) and never touched; note that this probability is at least $1 - \frac{1}{C}$.

If we play an arm and it transitions to node u, we need to decide what status (u, a, t)

[9]Please note that this is not to be confused with the colloquialism of "higher priorities going first".

to put that arm in. For all $i \in [n]$, $u \in \mathcal{S}_i \setminus \{\rho_i\}$, $a \in A$, $t \in [B]$, $(v,b) \in \mathsf{Par}(u)$, and $t' < t$, we prescribe a probability $q_{v,b,t',u,a,t}$ with which we will put arm i into status (u,a,t), conditioned on arriving at node u after playing action b on node v at priority t'. The evolution of statuses is independent of other arms. The following lemma shows that it is possible to solve for values of $q_{v,b,t',u,a,t}$ which are feasible, and respect the values of $x^a_{u,t}$:

Lemma 7.4.1. *Suppose we are given the x's of a feasible solution to* (`PolyLP`). *Then we can find* $\{q_{v,b,t',u,a,t} : u \in \mathcal{S}\setminus\{\rho_1,\dots,\rho_n\}, a \in A, t \in [B], (v,b) \in \mathsf{Par}(u), t' < t\}$ *in polynomial time such that*

$$\sum_{a\in A}\sum_{t:t>t'} q_{v,b,t',u,a,t} \le 1 \qquad u \in \mathcal{S}\setminus\{\rho_1,\dots,\rho_n\},\ (v,b)\in\mathsf{Par}(u),\ t'\in[B-1] \tag{7.19a}$$

$$\sum_{(v,b)\in\mathsf{Par}(u)}\sum_{t':t'<t} x^b_{v,t'}\cdot p^b_{v,u}\cdot q_{v,b,t',u,a,t} = x^a_{u,t} \quad u \in \mathcal{S}\setminus\{\rho_1,\dots,\rho_n\},\ a\in A,\ t\in\{2,\dots,B\} \tag{7.19b}$$

Furthermore, if $u \in \mathcal{B}$, then we can strengthen (7.19a) to $q_{v,b,t',u,\alpha,t'+1} = 1$.

(7.19a) ensures that the probabilities telling us what to do, when we arrive at node u after playing action b on node v at time t', are well-defined; the case where u is a bridge node will be needed to prove Theorem 7.2.9. For all $i \in [n]$, $u \in \mathcal{S}_i \setminus \{\rho_i\}$, $(v,b) \in \mathsf{Par}(u)$, and $t' \in [B-1]$, define $q_{v,b,t',u,\infty} = 1 - \sum_{a\in A}\sum_{t>t'} q_{v,b,t',u,a,t}$, the probability we abandon arm i after making the transition to u.

At a high level, Lemma 7.4.1 is a flow decomposition result and our analogue to the *convex decomposition* from Gupta et al. (2011a). It says that for each arm i, $\{x^a_{u,t} : u \in \mathcal{S}_i, t \in [B], a \in A\}$ is a flow satisfying precedence constraints on both nodes u and times t, and can be decomposed into "local instructions on each transition" (i.e. what to do upon arriving at u after playing b on v at priority t') to reconstruct the original flow. Its proof is deferred to Section F.4.

Having defined the values of $q_{v,b,t',u,a,t}$, the overall algorithm can now be described in two steps:

1. While there exists an arm with priority not ∞, play an arm with the smallest index (breaking ties arbitrarily) until it arrives at a status (u,a,t) such that $t \ge 2\cdot\mathsf{depth}(u)$

($t = \infty$ would suffice).

2. Repeat until all arms have priority ∞.

To clarify Step 1, once we start playing a lowest-index arm, its index will evolve and increase. However, we do not pause playing it once its index is no longer the lowest; instead, we pause playing it once it reaches a combination of index t and node u such that $t \geq 2 \cdot \mathsf{depth}(u)$.

We are also constrained by a budget of B time steps, but it will simplify the analysis to assume our algorithm finishes all the arms and collects reward only for plays up to time B. Under this assumption, the statuses an arm goes through is independent of the outcomes on all other arms; the inter-dependence only affects the order in which arms are played (and thus which nodes obtain reward).

Also, note that this is only a valid algorithm because Theorem 7.2.8 assumes all processing times are 1, so there are no bridge nodes. If there were bridge nodes, then we may not be allowed to switch to an arm with lowest index, being forced to play the arm on a bridge node.

7.4.2 Analysis of Algorithm

For all $i \in [n]$, $u \in \mathcal{S}_i$, $a \in A$, $t \in [B]$, let $\mathsf{time}(u, a, t)$ be the random variable for the time step at which our algorithm plays arm i from status (u, a, t), with $\mathsf{time}(u, a, t) = \infty$ if arm i never gets in status (u, a, t). Then $\Pr[\mathsf{time}(\rho_i, a, t) < \infty] = \frac{x^a_{\rho_i,t}}{C}$ for all $i \in [n]$, $a \in A$, $t \in [B]$. If u is a non-root node, then we can induct on $\mathsf{depth}(u)$ to prove for all $a \in A$, $t \in [B]$ that

$$\begin{aligned}
\Pr[\mathsf{time}(u,a,t) < \infty] &= \sum_{(v,b)\in\mathsf{Par}(u)} \sum_{t'<t} \Pr[\mathsf{time}(v,b,t') < \infty] \cdot p^b_{v,u} \cdot q_{v,b,t',u,a,t} \\
&= \sum_{(v,b)\in\mathsf{Par}(u)} \sum_{t'<t} \frac{x^b_{v,t'}}{C} \cdot p^b_{v,u} \cdot q_{v,b,t',u,a,t} \\
&= \frac{x^a_{u,t}}{C} \qquad (7.20)
\end{aligned}$$

where the final equality follows from Lemma 7.4.1.

For an event $\mathcal{A}$, let $\mathbb{1}_{\mathcal{A}}$ be the indicator random variable for $\mathcal{A}$. The expected reward obtained by our algorithm is

$$
\begin{aligned}
&\mathbb{E}\Big[\sum_{u\in\mathcal{S}}\sum_{a\in A} r_u^a \sum_{t=1}^{B} \mathbb{1}_{\{\mathsf{time}(u,a,t)\le B\}}\Big] \\
=\ & \sum_{u\in\mathcal{S}}\sum_{a\in A} r_u^a \sum_{t=1}^{B} \mathbb{E}[\mathbb{1}_{\{\mathsf{time}(u,a,t)\le B\}} \mid \mathsf{time}(u,a,t)<\infty]\cdot \Pr[\mathsf{time}(u,a,t)<\infty] \\
=\ & \sum_{u\in\mathcal{S}}\sum_{a\in A} r_u^a \sum_{t=1}^{B} \Pr[\mathsf{time}(u,a,t)\le B \mid \mathsf{time}(u,a,t)<\infty]\cdot \frac{x_{u,t}^a}{C}
\end{aligned}
$$

For the remainder of this subsection, we will set $C=3$ and prove for an arbitrary $i\in[n]$, $u\in\mathcal{S}_i$, $a\in A$, $t\in[B]$ that $\Pr[\mathsf{time}(u,a,t)\le B \mid \mathsf{time}(u,a,t)<\infty]\ge\frac{4}{9}$. It suffices to prove that $\Pr[\mathsf{time}(u,a,t)\le t \mid \mathsf{time}(u,a,t)<\infty]\ge\frac{4}{9}$, since $t\le B$.

Case 1. Suppose $t\ge 2\cdot\mathsf{depth}(u)$. We prove that conditioned on the event $\{\mathsf{time}(u,a,t)<\infty\}$, $\{\mathsf{time}(u,a,t)>t\}$ occurs with probability at most $\frac{5}{9}$.

Note that every node v can have at most one b,t' such that $\mathsf{time}(v,b,t')<\infty$; let $\mathsf{time}(v)$ denote this quantity (and be ∞ if $\mathsf{time}(v,b,t')=\infty$ for all $b\in A$, $t'\in[B]$). The nodes v that are played before u are those with $\mathsf{time}(v)<\mathsf{time}(u,a,t)$. Since our algorithm plays a node at every time step, $\mathsf{time}(u,a,t)>t$ if and only if there are t or more nodes $v\neq u$ such that $\mathsf{time}(v)<\mathsf{time}(u,a,t)$. But this is equivalent to there being exactly t nodes $v\neq u$ such that $\mathsf{time}(v)<\mathsf{time}(u,a,t)$ and $\mathsf{time}(v)\le t$. The $\mathsf{depth}(u)$ ancestors of u are guaranteed to satisfy this.

Hence the event $\{\mathsf{time}(u,a,t) > t\}$ is equivalent to $\{\mathsf{depth}(u)+\sum_{v\in\mathcal{S}\setminus\mathcal{S}_i}\mathbb{1}_{\{\mathsf{time}(v)<\mathsf{time}(u,a,t)\}}\cdot\mathbb{1}_{\{\mathsf{time}(v)\le t\}}=t\}$. But $t\ge 2\cdot\mathsf{depth}(u)$, so this implies $\{\sum_{v\in\mathcal{S}\setminus\mathcal{S}_i}\mathbb{1}_{\{\mathsf{time}(v)<\mathsf{time}(u,a,t)\}}\cdot\mathbb{1}_{\{\mathsf{time}(v)\le t\}}\ge\frac{t}{2}\} \implies \{\sum_{v\in\mathcal{S}\setminus\mathcal{S}_i}\mathbb{1}_{\{\mathsf{time}(v)<\mathsf{time}(u,a,t)\}}\ge\frac{t}{2}\}$. Now, whether the sum is at least $\frac{t}{2}$ is unchanged if we exclude all v such that $\mathsf{depth}(v)\ge\frac{t}{2}$. Indeed, if any such v satisfies $\mathsf{time}(v)<\mathsf{time}(u,a,t)$, then all of its ancestors also do, and its first $\lceil\frac{t}{2}\rceil$ ancestors ensure that the sum, without any nodes of depth at least $\frac{t}{2}$, is at least $\frac{t}{2}$. Thus, the last event is equivalent to

$$\Big\{\sum_{v\in\mathcal{S}\setminus\mathcal{S}_i:\mathsf{depth}(v)<\frac{t}{2}}\mathbb{1}_{\{\mathsf{time}(v)<\mathsf{time}(u,a,t)\}}\ge\frac{t}{2}\Big\} \tag{7.21}$$

Suppose $\mathsf{time}(v)=\mathsf{time}(v,b,t')$ for some $b\in A$ and $t'\in[B]$. We would like to argue that in order for both $\mathsf{time}(v)<\mathsf{time}(u,a,t)$ and $\mathsf{depth}(v)<\frac{t}{2}$ to hold, it must be the case that $t' \leq t$. Suppose to the contrary that $t' > t$. If $t' \geq 2 \cdot \mathsf{depth}(v)$, then the algorithm can only play (v, b, t') once t' becomes the lowest index, which must happen after (u, a, t) becomes the lowest index, hence $\mathsf{time}(v, b, t') < \mathsf{time}(u, a, t)$ is impossible. Otherwise, if $t' < 2 \cdot \mathsf{depth}(v)$, then $\mathsf{depth}(v) > \frac{t'}{2} > \frac{t}{2}$, violating $\mathsf{depth}(v) < \frac{t}{2}$. Thus indeed $t' \leq t$ and

$$
\begin{aligned}
(7.21) \quad & \iff \quad \Big\{ \sum_{v \in \mathcal{S} \setminus \mathcal{S}_i : \mathsf{depth}(v) < \frac{t}{2}} \sum_{b \in A} \sum_{t'=1}^{t} \mathbb{1}_{\{\mathsf{time}(v,b,t') < \mathsf{time}(u,a,t)\}} \geq \frac{t}{2} \Big\} \\
& \implies \quad \Big\{ \sum_{v \in \mathcal{S} \setminus \mathcal{S}_i} \sum_{b \in A} \sum_{t'=1}^{t} \mathbb{1}_{\{\mathsf{time}(v,b,t') < \infty\}} \geq \frac{t}{2} \Big\}
\end{aligned}
$$

We establish that the probability of interest $\Pr[\mathsf{time}(u, a, t) > t \mid \mathsf{time}(u, a, t) < \infty]$ is at most

$$
\begin{aligned}
& \Pr\Big[\sum_{v \in \mathcal{S} \setminus \mathcal{S}_i} \sum_{b \in A} \sum_{t'=1}^{t} \mathbb{1}_{\{\mathsf{time}(v,b,t') < \infty\}} \geq \frac{t}{2} \mid \mathsf{time}(u, a, t) < \infty \Big] \\
= \; & \Pr\Big[\sum_{j \neq i} \sum_{v \in \mathcal{S}_j} \sum_{b \in A} \sum_{t'=1}^{t} \mathbb{1}_{\{\mathsf{time}(v,b,t') < \infty\}} \geq \frac{t}{2} \Big]
\end{aligned}
$$

where we remove the conditioning due to independence between arms. Now, let

$$
Y_j = \min\Big\{ \sum_{v \in \mathcal{S}_j} \sum_{b \in A} \sum_{t'=1}^{t} \mathbb{1}_{\{\mathsf{time}(v,b,t') < \infty\}}, \frac{t}{2} \Big\}
$$

for all $j \neq i$. The previous probability is equal to $\Pr[\sum_{j \neq i} Y_j \geq \frac{t}{2}]$. Note that

$$
\begin{aligned}
\mathbb{E}\Big[\sum_{j \neq i} Y_j\Big] \;&\leq\; \sum_{j \neq i} \sum_{v \in \mathcal{S}_j} \sum_{b \in A} \sum_{t'=1}^{t} \Pr[\mathsf{time}(v, b, t') < \infty] \\
&\leq\; \sum_{t'=1}^{t} \sum_{v \in \mathcal{S}} \sum_{b \in A} \frac{x^b_{v,t'}}{3} \\
&\leq\; \frac{t}{3}
\end{aligned}
$$

where the second inequality uses (7.20), and the final inequality uses (7.11). We can do better than the Markov bound on $\Pr[\sum_{j \neq i} Y_j \geq \frac{t}{2}]$ because the random variables $\{Y_j\}_{j \neq i}$ are independent. Furthermore, each Y_j is non-zero with probability at most $\frac{1}{3}$ (arm j is never touched with probability at least $\frac{2}{3}$), so since is at most $\frac{t}{2}$ when it is non-zero, $\mathbb{E}[Y_j] \leq \frac{t}{6}$

for all $j \neq i$. We now invoke the following lemma:

Lemma 7.4.2. *Let $t > 0$ be arbitrary and $Y_1, \ldots, Y_m$ be independent non-negative random variables with individual expectations at most $\frac{t}{6}$ and sum of expectations at most $\frac{t}{3}$. Then $\Pr[\sum_{j=1}^{m} Y_j \geq \frac{t}{2}]$ is maximized when only two random variables are non-zero, each taking value $\frac{t}{2}$ with probability $\frac{1}{3}$ (and value 0 otherwise). Therefore, $\Pr[\sum_{j=1}^{m} Y_j \geq \frac{t}{2}] \leq 1 - (1 - \frac{1}{3})^2 = \frac{5}{9}$.*

This lemma would complete the proof that $\Pr[\mathsf{time}(u,a,t) > t \mid \mathsf{time}(u,a,t) < \infty] \leq \frac{5}{9}$ under Case 1, where $t \geq 2 \cdot \mathsf{depth}(u)$. We defer the proof of Lemma 7.4.2 to Section F.4. It uses the conjecture from Samuels (1966) for $n = 3$; the conjecture has been proven for $n \leq 4$ in Samuels (1968). The proof also uses a technical lemma from Bansal et al. (2012).

Case 2. Suppose $t < 2 \cdot \mathsf{depth}(u)$. Then $\mathsf{depth}(u)$ must be at least 1, so conditioned on $\mathsf{time}(u,a,t) < \infty$, there must be some $(v,b) \in \mathsf{Par}(u)$ and $t' < t$ such that $\mathsf{time}(v,b,t') < \infty$. Furthermore, the algorithm will play status (u,a,t) at time step $\mathsf{time}(v,b,t') + 1$, so $\mathsf{time}(u,a,t) \leq t$ will hold so long as $\mathsf{time}(v,b,t') \leq t'$, since $t' < t$. Thus $\Pr[\mathsf{time}(u,a,t) \leq t \mid \mathsf{time}(u,a,t) < \infty] \geq \Pr[\mathsf{time}(v,b,t') \leq t \mid \mathsf{time}(v,b,t') < \infty]$. We can iterate this argument until the the problem reduces to Case 1. This completes the proof that $\Pr[\mathsf{time}(u,a,t) \leq t \mid \mathsf{time}(u,a,t) < \infty] \geq \frac{4}{9}$ under Case 2.

Therefore, the expected reward obtained by our algorithm is at least $\sum_{u \in \mathcal{S}} \sum_{a \in A} r_u^a \sum_{t=1}^{B} (1 - \frac{5}{9}) \frac{x_{u,t}^a}{3}$, which is the same as $\frac{4}{27}\mathsf{OPT}_{\mathtt{PolyLP}}$, completing the proof of Theorem 7.2.8.

7.4.3 Proof of Theorem 7.2.9

In this subsection we show how to modify the algorithm and analysis when there are multi-period actions, to prove Theorem 7.2.9. As mentioned in Section 7.4.1, we must modify Step 1 of the algorithm when there are bridge nodes. If we arrive at a status (u,a,t) such that $t \geq 2 \cdot \mathsf{depth}(u)$ but $u \in \mathcal{B}$ (and $a = \alpha$), we are forced to immediately play the same arm again, instead of switching to another arm with a lower index.

The overall framework of the analysis still holds, except now the bound is optimized when we set $C = 6$. Our goal is to prove for an arbitrary $i \in [n]$, $u \in \mathcal{S}_i$, $a \in A$, $t \in [B]$ that $\Pr[\mathsf{time}(u,a,t) > t \mid \mathsf{time}(u,a,t) < \infty] \leq \frac{1}{2}$. We still have that event $\{\mathsf{time}(u,a,t) > t\}$

implies (7.21). Suppose for an arbitrary $v \in \mathcal{S} \setminus \mathcal{S}_i$ that $\mathsf{depth}(v) < \frac{t}{2}$ and $\mathsf{time}(v) < \mathsf{time}(u, a, t)$, where $\mathsf{time}(v) = \mathsf{time}(v, b, t')$. We can no longer argue that $t' \leq t$, but we would like to argue that $t' \leq \frac{3t}{2}$. Suppose to the contrary that $t' > \frac{3t}{2}$.

Then $t' > 3 \cdot \mathsf{depth}(v)$, so $t' \geq 2 \cdot \mathsf{depth}(v)$ ie. we would check priorities before playing (v, b, t'). However, if $v \in \mathcal{B}$, then it could be the case that $\mathsf{time}(v, b, t') < \mathsf{time}(u, a, t)$ even though $t' > t$. If so, consider w, the youngest (largest depth) ancestor of v that isn't a bridge node. Suppose $\mathsf{time}(w) = \mathsf{time}(w, b', t'')$; it must be the case that $t'' \leq t$. By the final statement of Lemma 7.4.1, the $\mathsf{depth}(v) - \mathsf{depth}(w)$ immediate descendents of w, which are bridge nodes, must have priority indices $t'' + 1, \ldots, t'' + \mathsf{depth}(v) - \mathsf{depth}(w)$, respectively. The youngest of these descendents is v, hence $t' = t'' + \mathsf{depth}(v) - \mathsf{depth}(w)$. But $t'' \leq t$ and $\mathsf{depth}(v) < \frac{t}{2}$, so $t' < \frac{3t}{2}$, causing a contradiction.

Therefore $t' \leq \frac{3t}{2}$. The bound on $\mathbb{E}[\sum_{j \neq i} Y_j]$ changes to

$$\begin{aligned}
\mathbb{E}\Big[\sum_{j \neq i} Y_j\Big] &\leq \sum_{j \neq i} \sum_{v \in \mathcal{S}_j} \sum_{b \in A} \sum_{t'=1}^{\frac{3t}{2}} \Pr[\mathsf{time}(v, b, t') < \infty] \\
&\leq \sum_{t'=1}^{\frac{3t}{2}} \sum_{v \in \mathcal{S}} \sum_{b \in A} \frac{x^b_{v,t'}}{6} \\
&\leq \frac{t}{4}
\end{aligned}$$

Thus $\Pr[\mathsf{time}(u, a, t) > t \mid \mathsf{time}(u, a, t) < \infty] \leq \Pr[\sum_{j \neq i} Y_j \geq \frac{t}{2}] \leq \frac{1}{2}$ where the final inequality is Markov's inequality. Note that we cannot use the stronger Samuels' conjecture here because we would need it for $n = 5$, which is unproven; if we could, then we could get a better approximation factor (and we would re-optimize C).

The rest of the analysis, including Case 2, is the same as before. Therefore, the expected reward obtained by our algorithm is at least $\sum_{u \in \mathcal{S}} \sum_{a \in A} r^a_u \sum_{t=1}^{B} (1 - \frac{1}{2}) \frac{x^a_{u,t}}{6}$, which is the same as $\frac{1}{12} \mathsf{OPT}_{\mathsf{PolyLP}}$, completing the proof of Theorem 7.2.9.

7.5 Conclusion and Open Questions

In this chapter, we presented a $(\frac{1}{2} - \varepsilon)$-approximation for the fully general *MAB superprocess with multi-period actions—no preemption* problem, by following a scaled copy of an optimal solution to the LP relaxation, and this is tight. However, when preemption is allowed, we were only able to obtain a $\frac{1}{12}$-approximation, using the solution to the LP relaxation mainly for generating priorities, and resorting to weak Markov-type bounds in the analysis. It seems difficult to follow a scaled copy of a solution to the LP relaxation when preemption is allowed, because arms can be paused and restarted. We do conjecture that our separation of $(2-\varepsilon)$ between the LP and the optimal algorithm is correct in this case, and that it is possible to obtain a $(\frac{1}{2}-\varepsilon)$-approximation, but this remains an open problem. Also, we have not explored how our techniques apply to certain extensions of the multi-armed bandit problem (switching costs, simultaneous plays, delayed feedback, contextual information, etc.).

Chapter 8

Conclusion and Future Directions

In this thesis, we have studied a broad range of topics in Revenue Management which can be classified along two dimensions: demand learning vs. revenue maximization, and single-period vs. multi-period (where learning and/or shared resource constraints link the time periods together).

<table>
<tr><th></th><th>Demand Learning</th><th>Revenue Maximization</th></tr>
<tr><td>Single-period</td><td>Estimation using Bundles
(Chapter 5)</td><td>Bundle Pricing
(Chapter 6)</td></tr>
<tr><td rowspan="2">Multi-period</td><td></td><td>Online Resource Allocation (Chapter 2)
Recommendation at Checkout (Chapter 3)
Single-leg RM (Chapter 4)</td></tr>
<tr><td colspan="2">Markovian Multi-Armed Bandits
(Chapter 7)</td></tr>
</table>

All of the chapters in this thesis, with the exception of Chapter 4, involve multiple items, and a lot of the focus is on the avoiding the "curse of dimensionality" by using simple heuristics. These heuristics are based on independent bid prices and protection levels in Chapters 2–3, simple bundling schemes in Chapters 5–6, and rounding a polynomial-sized LP-relaxation in Chapter 7.

We should reiterate that our work allows direct implementation on data instances, but does not directly address higher-level business strategy. We believe it is interesting future work to rigorously analyze the interplay between lower-order algorithmic optimization and higher-order business questions.

Another recurring theme in our work, more on the theoretical side, is the necessity of sampling to make algorithms run in polynomial-time. The chronologically-earliest appearance of this was in our work on Markovian multi-armed bandits and stochastic knapsack in Chapter 7. It is extremely interesting to us that the same analysis framework was useful in the chronologically-later-to-appear Chapters 3 and 4, on seemingly unrelated problems.

Finally, we discuss a direction where we have joint work that is not part of this thesis. Although the algorithms in Chapters 2–4 do not assume any stochastic model for the number of customers or their characteristics, they assume that a customer's choice model, as a function of her characteristics, has already been learned (e.g., from collaborative filtering on extensive historical transactions). Meanwhile, our work in Chapter 7 and the area of multi-armed bandits in general study the tradeoff between *exploring* the popularity of new products with limited historical data, vs. repeatedly *exploiting* products already known to be popular. We believe it is an exciting future direction to consider algorithms which incorporate the *learning* of choice patterns that are consistent across products and customers, while still *hedging* against strict uncertainty in the number of future customers and their characteristics.

Bibliography

T. Abdallah. On the benefit (or cost) of large-scale bundling. *Available at SSRN*, 2016.

W. J. Adams and J. L. Yellen. Commodity bundling and the burden of monopoly. *The quarterly journal of economics*, pages 475–498, 1976.

G. Aggarwal, G. Goel, C. Karande, and A. Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1253–1264. Society for Industrial and Applied Mathematics, 2011.

S. Agrawal, Z. Wang, and Y. Ye. A dynamic near-optimal algorithm for online linear programming. *Operations Research*, 62(4):876–890, 2014.

V. F. Araman and R. Caldentey. Revenue management with incomplete demand information. *Wiley Encyclopedia of Operations Research and Management Science*, 2011.

M. Armstrong. Multiproduct nonlinear pricing. *Econometrica: Journal of the Econometric Society*, pages 51–75, 1996.

M. Armstrong. Price discrimination by a many-product firm. *The Review of Economic Studies*, 66(1):151–168, 1999.

M. Babaioff, N. Immorlica, B. Lucier, and S. M. Weinberg. A simple and approximately optimal mechanism for an additive buyer. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 21–30. IEEE, 2014.

Y. Bakos and E. Brynjolfsson. Bundling information goods: Pricing, profits, and efficiency. *Management science*, 45(12):1613–1630, 1999.

Y. Bakos and E. Brynjolfsson. Bundling and competition on the internet. *Marketing science*, 19(1):63–82, 2000.

M. O. Ball and M. Queyranne. Toward robust revenue management: Competitive analysis of online booking. *Operations Research*, 57(4):950–963, 2009.

N. Bansal and V. Nagarajan. On the adaptivity gap of stochastic orienteering. In *Integer Programming and Combinatorial Optimization*, pages 114–125. Springer, 2014.

N. Bansal, A. Gupta, J. Li, J. Mestre, V. Nagarajan, and A. Rudra. When lp is the cure for your matching woes: Improved bounds for stochastic matchings. *Algorithmica*, 63(4): 733–762, 2012.

M. Bateni, S. Dehghani, M. Hajiaghayi, and S. Seddighin. Revenue maximization for selling multiple correlated items. In *Algorithms-ESA 2015*, pages 95–105. Springer, 2015.

D. Bergemann and K. H. Schlag. Pricing without priors. *Journal of the European Economic Association*, 6(2-3):560–569, 2008.

F. Bernstein, A. G. Kök, and L. Xie. Dynamic assortment customization with limited inventories. *Manufacturing & Service Operations Management*, 17(4):538–553, 2015.

D. P. Bertsekas. *Dynamic programming and optimal control.* Athena Scientific Belmont, MA, 1995.

D. Bertsimas and A. J. Mersereau. A learning approach for interactive marketing to a customer segment. *Operations Research*, 55(6):1120–1135, 2007.

A. Bhalgat. A (2+eps)-approximation algorithm for the stochastic knapsack problem. *Unpublished Manuscript*, 2011.

A. Bhalgat, A. Goel, and S. Khanna. Improved approximation results for stochastic knapsack problems. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1647–1665. SIAM, 2011.

H. K. Bhargava. Mixed bundling of two independently valued goods. *Management Science*, 59(9):2170–2185, 2013.

T. Bodea, M. Ferguson, and L. Garrow. Data set—choice-based revenue management: Data from a major hotel chain. *Manufacturing & Service Operations Management*, 11 (2):356–361, 2009.

A. Borodin and R. El-Yaniv. *Online computation and competitive analysis.* cambridge university press, 2005.

S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *arXiv preprint arXiv:1204.5721*, 2012.

N. Buchbinder and J. S. Naor. The design of competitive online algorithms via a primal–dual approach. *Foundations and Trends® in Theoretical Computer Science*, 3(2–3):93–263, 2009.

N. Buchbinder, K. Jain, and J. S. Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *European Symposium on Algorithms*, pages 253–264. Springer, 2007.

N. Buchbinder, S. Chen, J. Naor, and O. Shamir. Unified algorithms for online learning and competitive analysis. *Mathematics of Operations Research*, 41(2):612–625, 2016.

Y. Cai and Z. Huang. Simple and nearly optimal multi-item auctions. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 564–577. SIAM, 2013.

Y. Cai, C. Daskalakis, and S. M. Weinberg. An algorithmic characterization of multi-dimensional mechanisms. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 459–478. ACM, 2012.

R. L. Carraway, R. L. Schmidt, and L. R. Weatherford. An algorithm for maximizing target achievement in the stochastic knapsack problem with normal returns. *Naval Research Logistics (NRL)*, 40(2):161–173, 1993.

C. W. Chan and V. F. Farias. Stochastic depletion problems: Effective myopic policies for a class of dynamic optimization problems. *Mathematics of Operations Research*, 34(2): 333–350, 2009.

H. Chen, M. Hu, and G. Perakis. Distribution-free pricing. *available on SSRN*, 2017.

Y. Chen and V. F. Farias. Simple policies for dynamic pricing with imperfect forecasts. *Operations Research*, 61(3):612–624, 2013.

W. C. Cheung and D. Simchi-Levi. Efficiency and performance guarantees for choice-based network revenue management problems with flexible products. *available on SSRN*, 2016.

C. S. Chu, P. Leslie, and A. Sorensen. Nearly optimal pricing for multiproduct firms. Technical report, National Bureau of Economic Research, 2008.

C. S. Chu, P. Leslie, and A. Sorensen. Bundle-size pricing as an approximation to mixed bundling. *The American Economic Review*, pages 263–303, 2011.

J. Chung and V. R. Rao. A general choice model for bundles with multiple-category products: Application to market segmentation and optimal pricing for bundles. *Journal of Marketing Research*, 40(2):115–130, 2003.

D. F. Ciocan and V. Farias. Model predictive control for dynamic resource allocation. *Mathematics of Operations Research*, 37(3):501–525, 2012.

R. G. Cross. *Revenue management: Hard-core tactics for market domination.* Crown Business, 2011.

B. C. Dean, M. X. Goemans, and J. Vondrdk. Approximating the stochastic knapsack problem: The benefit of adaptivity. In *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, pages 208–217. IEEE, 2004.

B. C. Dean, M. X. Goemans, and J. Vondrák. Approximating the stochastic knapsack problem: The benefit of adaptivity. *Mathematics of Operations Research*, 33(4):945–964, 2008.

N. R. Devanur, K. Jain, and R. D. Kleinberg. Randomized primal-dual analysis of ranking for online bipartite matching. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 101–107. SIAM, 2013.

J. C. Eckalbar. Closed-form solutions to bundling problems. *Journal of Economics & Management Strategy*, 19(2):513–544, 2010.

R. El-Yaniv, A. Fiat, R. M. Karp, and G. Turpin. Optimal search and one-way trading online algorithms. *Algorithmica*, 30(1):101–139, 2001.

S. S. Eren and C. Maglaras. Monopoly pricing with limited demand information. *Journal of revenue and pricing management*, 9(1-2):23–48, 2010.

H. Fang and P. Norman. To bundle or not to bundle. *RAND Journal of Economics*, pages 946–963, 2006.

V. F. Farias and R. Madan. The irrevocable multiarmed bandit problem. *Operations Research*, 59(2):383–399, 2011.

V. F. Farias, S. Jagabathula, and D. Shah. A nonparametric approach to modeling choice with limited data. *Management science*, 59(2):305–322, 2013.

J. Feldman, N. Korula, V. Mirrokni, S. Muthukrishnan, and M. Pál. Online ad assignment with free disposal. In *International Workshop on Internet and Network Economics*, pages 374–385. Springer, 2009.

J. Feldman, M. Henzinger, N. Korula, V. Mirrokni, and C. Stein. Online stochastic packing applied to display ad allocation. *Algorithms-ESA 2010*, pages 182–194, 2010.

K. J. Ferreira, D. Simchi-Levi, and H. Wang. Online network revenue management using thompson sampling. *manuscript on SSRN*, 2016.

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

G. Gallego, A. Li, V.-A. Truong, and X. Wang. Online resource allocation with customer choice. *arXiv preprint arXiv:1511.01837*, 2015.

G. Gallego, A. Li, V.-A. Truong, and X. Wang. Online personalized resource allocation with customer choice. Technical report, Columbia University, 2016.

J. Gittins, K. Glazebrook, and R. Weber. *Multi-armed bandit allocation indices.* John Wiley & Sons, 2011.

A. Goel and P. Indyk. Stochastic load balancing and related problems. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 579–586. IEEE, 1999.

A. Goel, S. Guha, and K. Munagala. Asking the right questions: Model-driven optimization using probes. In *Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 203–212. ACM, 2006.

N. Golrezaei, H. Nazerzadeh, and P. Rusmevichientong. Real-time optimization of personalized assortments. *Management Science*, 60(6):1532–1551, 2014.

S. Guha and K. Munagala. Approximation algorithms for budgeted learning problems. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 104–113. ACM, 2007a.

S. Guha and K. Munagala. Model-driven optimization using adaptive probes. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 308–317. Society for Industrial and Applied Mathematics, 2007b.

S. Guha and K. Munagala. Sequential design of experiments via linear programming. *arXiv preprint arXiv:0805.2630*, 2008.

S. Guha and K. Munagala. Approximation algorithms for bayesian multi-armed bandit problems. *arXiv preprint arXiv:1306.3525*, 2013.

A. Gupta, R. Krishnaswamy, M. Molinaro, and R. Ravi. Approximation algorithms for correlated knapsacks and non-martingale bandits. *arXiv preprint arXiv:1102.3749*, 2011a.

A. Gupta, R. Krishnaswamy, M. Molinaro, and R. Ravi. Approximation algorithms for correlated knapsacks and non-martingale bandits. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 827–836. IEEE, 2011b.

A. Gupta, R. Krishnaswamy, V. Nagarajan, and R. Ravi. Running errands in time: Approximation algorithms for stochastic orienteering. *Mathematics of Operations Research*, 40(1):56–79, 2014.

J. N. Hagstrom. Computational complexity of pert problems. *Networks*, 18(2):139–147, 1988.

W. Hanson and R. K. Martin. Optimal bundle pricing. *Management Science*, 36(2):155–174, 1990.

S. Hart and N. Nisan. Approximate revenue maximization with multiple items. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 656–656. ACM, 2012.

S. Hart and N. Nisan. The menu-size complexity of auctions. In *Proceedings of the fourteenth ACM conference on Electronic commerce*, pages 565–566. ACM, 2013.

S. Hart and P. J. Reny. *Maximal revenue with multiple goods: Nonmonotonicity and other observations.* Center for the Study of Rationality, 2012.

L. M. Hitt and P.-y. Chen. Bundling with customer self-selection: A simple approach to bundling low-marginal-cost goods. *Management Science*, 51(10):1481–1493, 2005.

R. Ibragimov and J. Walden. Optimal bundling strategies under heavy-tailed valuations. *Management Science*, 56(11):1963–1976, 2010.

T. Ilhan, S. M. Iravani, and M. S. Daskin. Technical note-the adaptive knapsack problem with stochastic rewards. *Operations research*, 59(1):242–248, 2011.

S. Jasin and S. Kumar. A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. *Mathematics of Operations Research*, 37(2):313–345, 2012.

K. Jedidi, S. Jagpal, and P. Manchanda. Measuring heterogeneous reservation prices for product bundles. *Marketing Science*, 22(1):107–130, 2003.

B. Kalyanasundaram and K. R. Pruhs. An optimal deterministic algorithm for online b-matching. *Theoretical Computer Science*, 233(1):319–325, 2000.

R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 352–358. ACM, 1990.

T. Kesselheim, K. Radke, A. Tönnis, and B. Vöcking. An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions. In *European Symposium on Algorithms*, pages 589–600. Springer, 2013.

J. M. Kessler. *United States air force fighter jet maintenance models: effectiveness of index policies.* PhD thesis, Massachusetts Institute of Technology, 2013.

J. Kleinberg, Y. Rabani, and É. Tardos. Allocating bandwidth for bursty connections. *SIAM Journal on Computing*, 30(1):191–217, 2000.

Y. Lan, H. Gao, M. O. Ball, and I. Karaesmen. Revenue management with limited demand information. *Management Science*, 54(9):1594–1609, 2008.

B. Letham, W. Sun, and A. Sheopuri. Latent variable copula inference for bundle pricing from retail transaction data. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 217–225, 2014.

J. Li and W. Yuan. Stochastic combinatorial optimization via poisson approximation. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 971–980. ACM, 2013.

M. Li, H. Feng, F. Chen, and J. Kou. Numerical investigation on mixed bundling and pricing of information products. *International Journal of Production Economics*, 144(2): 560–571, 2013.

G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.

Q. Liu and G. Van Ryzin. On the choice-based linear programming model for network revenue management. *Manufacturing & Service Operations Management*, 10(2):288–310, 2008.

G. Lugosi. *Concentration-of-measure inequalities.* http://www.econ.upf.edu/ lugosi/anu.pdf, 2009.

C. Maglaras and J. Meissner. Dynamic pricing strategies for multiproduct revenue management problems. *Manufacturing & Service Operations Management*, 8(2):136–148, 2006.

S. Mahajan and G. van Ryzin. Stocking retail assortments under dynamic consumer substitution. *Operations Research*, 49(3):334–351, 2001.

M. Mahdian, H. Nazerzadeh, and A. Saberi. Online optimization with uncertain information. *ACM Transactions on Algorithms (TALG)*, 8(1):2, 2012.

A. M. Manelli and D. R. Vincent. Bundling as an optimal selling mechanism for a multiple-good monopolist. *Journal of Economic Theory*, 127(1):1–35, 2006.

A. M. Manelli and D. R. Vincent. Multidimensional mechanism design: Revenue maximization and the multiple-good monopoly. *Journal of Economic Theory*, 137(1):153–185, 2007.

V. H. Manshadi, S. O. Gharan, and A. Saberi. Online stochastic matching: Online actions based on offline statistics. *Mathematics of Operations Research*, 37(4):559–573, 2012.

R. P. McAfee, J. McMillan, and M. D. Whinston. Multiproduct monopoly, commodity bundling, and correlation of values. *The Quarterly Journal of Economics*, pages 371–383, 1989.

K. F. McCardle, K. Rajaram, and C. S. Tang. Bundling retail products: Models and analysis. *European Journal of Operational Research*, 177(2):1197–1217, 2007.

A. Mehta. Online matching and ad allocation. *Foundations and Trends® in Theoretical Computer Science*, 8(4):265–368, 2013.

A. Mehta and D. Panigrahi. Online matching with stochastic rewards. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 728–737. IEEE, 2012.

A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized online matching. *Journal of the ACM (JACM)*, 54(5):22, 2007.

A. Mehta, B. Waggoner, and M. Zadimoghaddam. Online stochastic matching with unequal probabilities. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1388–1404. SIAM, 2014.

M. Mitzenmacher and E. Upfal. *Probability and computing: Randomized algorithms and probabilistic analysis.* Cambridge University Press, 2005.

R. H. Möhring, A. S. Schulz, and M. Uetz. Approximation in stochastic scheduling: the power of lp-based priority policies. *Journal of the ACM (JACM)*, 46(6):924–942, 1999.

R. Motwani and P. Raghavan. *Randomized algorithms.* Chapman & Hall/CRC, 2010.

R. B. Myerson. Optimal auction design. *Mathematics of operations research*, 6(1):58–73, 1981.

J. P. Newman, M. E. Ferguson, L. A. Garrow, and T. L. Jacobs. Estimation of choice-based models using sales data from a single firm. *Manufacturing & Service Operations Management*, 16(2):184–197, 2014.

R. L. Phillips. *Pricing and revenue optimization.* Stanford University Press, 2005.

M. L. Pinedo. *Scheduling: theory, algorithms, and systems.* Springer Science & Business Media, 2012.

A. Prasad, R. Venkatesh, and V. Mahajan. Optimal bundling of technological products with network externality. *Management Science*, 56(12):2224–2236, 2010.

R. Ravi and B. Sun. *Customer-centric marketing: A pragmatic framework.* Mit Press, 2016.

M. I. Reiman and Q. Wang. An asymptotically optimal policy for a quantity-based network revenue management problem. *Mathematics of Operations Research*, 33(2):257–282, 2008.

A. Rubinstein. On the computational complexity of optimal simple mechanisms. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 21–28. ACM, 2016.

A. Rubinstein and S. M. Weinberg. Simple mechanisms for a subadditivebuyer and applications to revenue monotonicity. In *Proceedings of the sixteenth ACM conference on Electronic commerce*, pages 377–394. ACM, 2015.

P. Rusmevichientong, Z.-J. Shen, and D. Shmoys. Dynamic assortment optimization with a multinomial logic choice model and capacity constraint. *Operations Research*, 58(6): 1666–1680, 2010.

S. M. Samuels. On a Chebyshev-type inequality for sums of independent random variables. *Ann. Math. Stat.*, 37(1):248–259, 1966.

S. M. Samuels. *More on a Chebyshev-type inequality for sums of independent random variables.* Defense Technical Information Center, 1968.

R. Schmalensee. Gaussian demand and commodity bundling. *Journal of business*, pages S211–S230, 1984.

M. Skutella and M. Uetz. Scheduling precedence-constrained jobs with stochastic processing times on parallel machines. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 589–590. Society for Industrial and Applied Mathematics, 2001.

G. J. Stigler. United states v. loew's inc.: A note on block-booking. *Sup. Ct. Rev.*, page 152, 1963.

S. Stremersch and G. J. Tellis. Strategic bundling of products and prices: A new synthesis for marketing. *Journal of Marketing*, 66(1):55–72, 2002.

K. Talluri and G. Van Ryzin. Revenue management under a general discrete choice model of consumer behavior. *Management Science*, 50(1):15–33, 2004.

K. T. Talluri and G. J. Van Ryzin. *The theory and practice of revenue management*, volume 68. Springer Science & Business Media, 2006.

O. Toubia, D. I. Simester, J. R. Hauser, and E. Dahan. Fast polyhedral adaptive conjoint estimation. *Marketing Science*, 22(3):273–303, 2003.

G. van Ryzin and S. Mahajan. On the relationship between inventory costs and variety benefits in retail assortments. *Management Science*, 45(11):1496–1509, 1999.

G. Van Ryzin and J. McGill. Revenue management without forecasting or optimization: An adaptive algorithm for determining airline seat protection levels. *Management Science*, 46(6):760–775, 2000.

G. van Ryzin and G. Vulcano. A market discovery algorithm to estimate a general class of nonparametric choice models. *Management Science*, 61(2):281–300, 2014.

R. Venkatesh and W. Kamakura. Optimal bundling and pricing under a monopoly: Contrasting complements and substitutes from independently valued products*. *The Journal of business*, 76(2):211–231, 2003.

R. Venkatesh and V. Mahajan. 11 the design and pricing of bundles: a review of normative guidelines and practical approaches. *Handbook of pricing research in marketing*, page 232, 2009.

X. Wang, V. Truong, and D. Bank. Online advance admission scheduling for services, with customer preferences. In *Working paper*, 2015.

R. B. Wilson. *Nonlinear pricing.* Oxford University Press, 1993.

M. Wood. A new kind of e-commerce adds a personal touch. New York Times, 2014.

S.-y. Wu, L. M. Hitt, P.-y. Chen, and G. Anandalingam. Customized bundle pricing for information goods: A nonlinear mixed-integer programming approach. *Management Science*, 54(3):608–622, 2008.

A. C.-C. Yao. Probabilistic computations: Toward a unified measure of complexity. In *Foundations of Computer Science, 1977., 18th Annual Symposium on*, pages 222–227. IEEE, 1977.

A. C.-C. Yao. An n-to-1 bidder reduction for multi-item auctions and its applications. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 92–109. SIAM, 2015.

M. Yuan, Y. Pavlidis, M. Jain, and K. Caster. Walmart online grocery personalization: Behavioral insights and basket recommendations. In S. Link and J. C. Trujillo, editors, *Advances in Conceptual Modeling*, pages 49–64, 2016.

D. Zhang and W. L. Cooper. Revenue management for parallel flights with customer-choice behavior. *Operations Research*, 53(3):415–431, 2005.

H. Zhang, C. Shi, C. Qin, and C. Hua. Stochastic regret minimization for revenue management problems with nonstationary demands. *Naval Research Logistics (NRL)*, 63(6): 433–448, 2016.

W. Zhao and Y.-S. Zheng. Optimal dynamic pricing for perishable assets with nonhomogeneous demand. *Management science*, 46(3):375–388, 2000.

Y. Zhao. *Inequalities.* http://yufeizhao.com/wc08/ineq.pdf, 2008.

Appendix A

Appendix to Chapter 2

A.1 Deferred Proofs from Section 2.2

Proof. Proof of Lemma 2.2.1. Fix any adaptive algorithm (which knows the arrival information, but not the realizations of the customers' purchase decisions, at the start) and consider its execution on instance $\mathcal{I}$. Let $X_{t,i}^{(j)}$ be the indicator random variable (0 or 1) for the algorithm offering item i at price j to customer t, and $P_{t,i}^{(j)}$ be the indicator random variable for customer t accepting when item i is offered to her at price j. On a given run, the constraints $\sum_{t=1}^{T}\sum_{j=1}^{m_i} P_{t,i}^{(j)} X_{t,i}^{(j)} \leq k_i$ and $\sum_{i=1}^{n}\sum_{j=1}^{m_i} X_{t,i}^{(j)} \leq 1$ are satisfied. Therefore, they are still satisfied after taking an expectation over all runs, and furthermore we can use independence to show that $\mathbb{E}[P_{t,i}^{(j)} X_{t,i}^{(j)}] = \mathbb{E}[P_{t,i}^{(j)}] \cdot \mathbb{E}[X_{t,i}^{(j)}] = p_{t,i}^{(j)} x_{t,i}^{(j)}$. Therefore, the algorithm must satisfy constraints (2.3b) and (2.3c) of the LP. Since its revenue on a given run is $\sum_{t=1}^{T}\sum_{i=1}^{n}\sum_{j=1}^{m_i} P_{t,i}^{(j)} r^{(j)} X_{t,i}^{(j)}$, taking an expectation over it yields (2.3a), completing the proof. □

Proof. Proof of Proposition 2.2.2. The statement for $\sigma^{(j)}, \ldots, \sigma^{(j)}$ is immediate from the fact that the explicit value of $\sigma^{(j)}$ is $(1 - \frac{r^{(j-1)}}{r^{(j)}})(1 + \sum_{j'=2}^{m}(1 - \frac{r^{(j'-1)}}{r^{(j')}}))^{-1}$, for all $j \in [m]$. To prove the statement for $\alpha^{(1)}, \ldots, \alpha^{(m)}$, we show that the solution to the system of n equations formed by (2.5) and $\alpha^{(1)} + \ldots \alpha^{(m)} = 1$ is unique and strictly positive.

Let $\gamma^{(j)} = e^{-\alpha^{(j)}}$ for all j. Then the constraint $\alpha^{(1)} + \ldots \alpha^{(m)} = 1$ can be rewritten as $\prod_{j=1}^{m} \gamma^{(j)} = \frac{1}{e}$. Furthermore, we derive from (2.5) that for all $j > 1$, $\gamma^{(j)} = (1 - \frac{r^{(j-1)}}{r^{(j)}})\gamma^{(1)} +$

$\frac{r^{(j-1)}}{r^{(j)}}$. Therefore,

$$\gamma^{(1)} \cdot \prod_{j=2}^{m} \left((1 - \frac{r^{(j-1)}}{r^{(j)}})\gamma^{(1)} + \frac{r^{(j-1)}}{r^{(j)}} \right) = \frac{1}{e}. \tag{A.1}$$

Consider the LHS of (A.1) as a function of $\gamma^{(1)}$ on $[\frac{1}{e}, 1]$. This is a continuous, strictly increasing function which is at most $\frac{1}{e}$ when $\gamma^{(1)} = \frac{1}{e}$ and 1 when $\gamma^{(1)} = 1$. Therefore, there is a unique solution with $\gamma^{(1)} \in [\frac{1}{e}, 1)$, and the resulting value of $\alpha^{(1)}$ is positive. For $j > 1$, since $\gamma^{(j)}$ can also be written as $\gamma^{(1)} + \frac{r^{(j-1)}}{r^{(j)}}(1-\gamma^{(1)})$, it can be seen that $\gamma^{(j)} \in [\frac{1}{e}, 1)$, hence the unique value for $\alpha^{(j)}$ is positive as well.

Finally, to see that $\alpha^{(1)} \geq \frac{1}{m}$, observe that when $\gamma^{(1)} = e^{-1/m}$, the LHS of (A.1) is at least $(e^{-1/m})^m = \frac{1}{e}$. Therefore, $\gamma^{(1)}$ is at most $e^{-1/m}$ which implies that $\alpha^{(1)} \geq \frac{1}{m}$. □

Proof. Proof of Proposition 2.2.9. For the first inequality in (2.10), observe that $f(x) = \frac{x}{1-e^{-x}}$ is a strictly increasing function on $[0, 1]$. Since $\sigma^{(1)} \in (0, 1)$, $\frac{\sigma^{(1)}}{1-e^{-\sigma^{(1)}}} < \frac{1}{1-\frac{1}{e}}$, which is the desired result.

For the second inequality in (2.10), we show $\alpha^{(1)} > \sigma^{(1)}$, by showing that for all $j = 2, \ldots, m$, $\alpha^{(j)}$ is a smaller multiple of $\alpha^{(1)}$ than $\sigma^{(j)}$ is of $\sigma^{(1)}$. This suffices because both the fractions $\alpha^{(1)}, \ldots, \alpha^{(m)}$ and $\sigma^{(1)}, \ldots, \sigma^{(m)}$ must sum to 1. For a given j, we must establish that $\frac{\alpha^{(j)}}{\alpha^{(1)}} < \frac{\sigma^{(j)}}{\sigma^{(1)}}$. By definition, $\frac{\sigma^{(j)}}{\sigma^{(1)}} = 1 - \frac{r^{(j-1)}}{r^{(j)}} = \frac{1-e^{-\alpha^{(j)}}}{1-e^{-\alpha^{(1)}}}$. Therefore, is suffices to show that $\frac{\alpha^{(j)}}{\alpha^{(1)}} < \frac{1-e^{-\alpha^{(j)}}}{1-e^{-\alpha^{(1)}}}$, or $\frac{\alpha^{(j)}}{1-e^{-\alpha^{(j)}}} < \frac{\alpha^{(1)}}{1-e^{-\alpha^{(1)}}}$. This follows from the fact that the function $f(x) = \frac{x}{1-e^{-x}}$ is strictly increasing.

To prove (2.11), note that $\sigma^{(1)} = (1 + \sum_{j=2}^{m}(1 - \frac{r^{(j-1)}}{r^{(j)}}))^{-1}$, while $1 + \ln \frac{r^{(m)}}{r^{(1)}} = 1 + \sum_{j=2}^{m} \ln \frac{r^{(j)}}{r^{(j-1)}}$. Therefore, it suffices to show that for any $j = 2, \ldots, m$, $\ln \frac{r^{(j)}}{r^{(j-1)}} > 1 - \frac{r^{(j-1)}}{r^{(j)}}$. Letting $x = \ln \frac{r^{(j-1)}}{r^{(j)}} < 0$, the desired inequality becomes $-x > 1 - e^{x}$, which is immediate.

For (2.12), we would like to prove that $\alpha < \alpha^{(1)}$. Note that $\alpha^{(1)}$ is the unique solution to

$$\alpha^{(1)} + \sum_{j=2}^{m} \left[- \ln \left(1 - (1 - e^{-\alpha^{(1)}})(1 - \frac{r^{(j-1)}}{r^{(j)}})\right) \right] = 1, \tag{A.2}$$

while α is the unique solution to

$$\alpha + \sum_{j=2}^{m} (1 - e^{-\alpha}) \ln \frac{r^{(j)}}{r^{(j-1)}} = 1. \tag{A.3}$$

The LHS of (A.2), as a function of $\alpha^{(1)}$, is increasing over $(0, 1)$; the same can be said about the LHS of (A.3) as a function of α. Therefore, it suffices to show that if $\alpha^{(1)} = \alpha = x$, then the LHS of (A.2) is strictly less than the LHS of (A.3), for all $x \in (0,1)$.

Let $F = 1 - e^{-x}$ and consider any $j > 1$. Let $s = \frac{r^{(j-1)}}{r^{(j)}} \in (0,1)$. It suffices to show that $-\ln(1 - F(1-s)) < F \cdot \ln \frac{1}{s}$, which can be rearranged as $\frac{1-s^F}{1-s} > F$. For the final inequality, note that $f(s) = s^F$ is a strictly concave function on $(0,1)$, since $F \in (0,1)$. Therefore, $\frac{1-s^F}{1-s} > F$, because the LHS is the slope of the secant line through (s, s^F) and $(1,1)$, while the RHS is the slope of the tangent line through $(1,1)$. $\square$

A.2 Supplement to Section 2.3

The first subsection contains the deferred proofs from Section 2.3. In the second subsection, we explain how to optimize the randomized procedure for generating a single value function. In the third subsection, we put together the proof of Theorem 2.2.4.

The following inequality will be useful throughout the paper. For all $j = 2, \ldots, m$, (2.5) says that $1 - e^{-\alpha^{(j)}} \leq 1 - \frac{r^{(j-1)}}{r^{(j)}}$, where we have used the fact that $1 - e^{-\alpha^{(1)}} \leq 1$. Therefore, for all $j = 2, \ldots, m$, we can derive that

$$\frac{r^{(j-1)}}{r^{(j)}} \leq e^{-\alpha^{(j)}}. \tag{A.4}$$

A.2.1 Deferred Proofs

Proof. Proof of Theorem 2.3.1. Define $N_{t,i}$ to be the algorithm's value for N_i at the *end* of time t ($N_{0,i}$ is understood to be 0), for all $t \in [T]$ and $i \in [n]$. For all $t \in [T]$, define $R_t = r_{i_t^*}^{(j_t^*)}$ and $Z_t = \tilde{\Phi}_{i_t^*}(\tilde{L}_{i_t^*}^{(j_t^*)}) - \tilde{\Phi}_{i_t^*}(N_{i_t^*}/k_{i_t^*})$ if a sale was made during time t; define $R_t = Z_t = 0$ otherwise.

Consider the solution to the dual LP (2.18) formed by setting $y_i = \mathbb{E}[\tilde{\Phi}_i(\frac{N_{T,i}}{k_i})]$ for all $i \in [n]$, and $z_t = \mathbb{E}[Z_t]$ for all $t \in [T]$. We claim that this solution is feasible. The non-negativity constraint (2.18c) can be verified directly from the definitions.

Now, consider constraint (2.18b) for a fixed $t \in [T], i \in [n], j \in [m_i]$. Given the initializations of $\tilde{L}_i^{(1)}, \ldots, \tilde{L}_i^{(m_i)}, \tilde{\Phi}_i$ and the value of $N_{t-1,i}$, the algorithm will always make a decision during time t which earns pseudorevenue whose conditional expectation is at least $p_{t,i}^{(j)}(\tilde{\Phi}_i(\tilde{L}_i^{(j)}) - \tilde{\Phi}_i(\frac{N_{t-1,i}}{k_i}))$, by definition (2.15). Formally,

$$\mathbb{E}[Z_t | \tilde{L}_i^{(1)}, \ldots, \tilde{L}_i^{(m_i)}, \tilde{\Phi}_i, N_{t-1,i}] \geq p_{t,i}^{(j)}(\tilde{\Phi}_i(\tilde{L}_i^{(j)}) - \tilde{\Phi}_i(\frac{N_{t-1,i}}{k_i})),$$

for all values of $\tilde{L}_i^{(1)}, \ldots, \tilde{L}_i^{(m_i)}, \tilde{\Phi}_i, N_{t-1,i}$. By the tower property of conditional expectation, $z_t = \mathbb{E}[Z_t] \geq \mathbb{E}[p_{t,i}^{(j)}(\tilde{\Phi}_i(\tilde{L}_i^{(j)}) - \tilde{\Phi}_i(\frac{N_{t-1,i}}{k_i}))]$. Meanwhile, y_i has been set to $\mathbb{E}[\tilde{\Phi}_i(\frac{N_{T,i}}{k_i})]$. Since $N_{T,i} \geq N_{t-1,i}$ and $\tilde{\Phi}_i$ is increasing, $y_i \geq \mathbb{E}[\tilde{\Phi}_i(\frac{N_{t-1,i}}{k_i})]$. Therefore, the LHS of (2.18b), $p_{t,i}^{(j)} y_i + z_t$, is at least $\mathbb{E}[p_{t,i}^{(j)}(\tilde{\Phi}_i(\tilde{L}_i^{(j)}))]$. By (2.17), this is at least $r_i^{(j)}$, completing the proof of feasibility.

Applying weak duality, we obtain

$$\begin{aligned}
\mathsf{OPT}(\mathcal{I}) &\leq \sum_{i=1}^{n} k_i \mathbb{E}[\tilde{\Phi}_i(\frac{N_{T,i}}{k_i})] + \sum_{t=1}^{T} \mathbb{E}[Z_t] \\
&= \sum_{i=1}^{n} k_i \mathbb{E}\Big[\sum_{t=1}^{T} (\tilde{\Phi}_i(\frac{N_{t,i}}{k_i}) - \tilde{\Phi}_i(\frac{N_{t-1,i}}{k_i}))\Big] + \sum_{t=1}^{T} \mathbb{E}[Z_t] \\
&= \sum_{t=1}^{T} \mathbb{E}\Big[\sum_{i=1}^{n} k_i (\tilde{\Phi}_i(\frac{N_{t,i}}{k_i}) - \tilde{\Phi}_i(\frac{N_{t-1,i}}{k_i})) + Z_t\Big]. \qquad \text{(A.5)}
\end{aligned}$$

We now analyze the term inside the expectation,

$$\sum_{i=1}^{n} k_i (\tilde{\Phi}_i(\frac{N_{t,i}}{k_i}) - \tilde{\Phi}_i(\frac{N_{t-1,i}}{k_i})) + Z_t, \qquad \text{(A.6)}$$

for every $t \in [T]$. We would like to argue that it is at most $\frac{R_t}{F}$, on every sample path.

There are two cases. If an item $i = i_t^*$ was sold at price $j = j_t^*$ during time t, then (A.6) equals

$$k_i (\tilde{\Phi}_i(\frac{N_{t-1,i}+1}{k_i}) - \tilde{\Phi}_i(\frac{N_{t-1,i}}{k_i})) + \tilde{\Phi}_i(\tilde{L}_i^{(j)}) - \tilde{\Phi}_i(\frac{N_{t-1,i}}{k_i}). \qquad \text{(A.7)}$$

Indeed, $N_{t,i} = N_{t-1,i} + 1$, $N_{t,i} = N_{t-1,i}$ for all $i \neq i$, and $Z_t = \tilde{\Phi}_i(\tilde{L}_i^{(j)}) - \tilde{\Phi}_i(\frac{N_{t-1,i}}{k_i})$ by definition. Furthermore, since Z_t is positive, $N_{t-1,i}$ must by less than $\tilde{L}_i^{(j)} k$. Therefore, we can invoke (2.16) to get that (A.7) is at most $r_i^{(j)}/F$, which is equal to $\frac{R_t}{F}$ by definition. In the other case, if no item was sold during time t, then (A.7) is 0, while $R_t = 0$ too, so (A.7) is still at most $\frac{R_t}{F}$.

Substituting back into (A.5), we conclude that $\mathsf{OPT}(\mathcal{I}) \leq \sum_{t=1}^{T} \mathbb{E}[\frac{R_t}{F}]$, which is equal to $\frac{1}{F}\mathbb{E}[\mathsf{ALG}(\mathcal{I})]$ by definition. This completes the proof of Algorithm 1 having a competitive ratio at least F. $\square$

Proof. Proof of Theorem 2.3.3.

First we prove the following two properties implied by the comonotonic randomized rounding procedure for $\tilde{L}^{(0)}, \ldots, \tilde{L}^{(m)}$ from Definition 2.3.2:

$$\mathbb{E}[\tilde{L}^{(j)}] = L^{(j)}, \qquad j = 0, \ldots, m; \tag{A.8}$$

$$|(\tilde{L}^{(j)} - \tilde{L}^{(j')}) - (L^{(j)} - L^{(j')})| \leq \frac{1}{k}, \qquad 1 \leq j' < j \leq m. \tag{A.9}$$

For (A.8), note that $\mathbb{E}[\tilde{L}^{(j)}] = \frac{\lfloor L^{(j)}k \rfloor + 1}{k}(L^{(j)}k - \lfloor L^{(j)}k \rfloor) + \frac{\lfloor L^{(j)}k \rfloor}{k}(1 - (L^{(j)}k - \lfloor L^{(j)}k \rfloor)) = \frac{1}{k}(L^{(j)}k - \lfloor L^{(j)}k \rfloor) + \frac{\lfloor L^{(j)}k \rfloor}{k} = L^{(j)}$.

For (A.9), note that $|(\tilde{L}^{(j)} - \tilde{L}^{(j')}) - (L^{(j)} - L^{(j')})| = |(\tilde{L}^{(j)} - L^{(j)}) - (\tilde{L}^{(j')} - L^{(j')})|$. We will prove that $(\tilde{L}^{(j)} - L^{(j)}) - (\tilde{L}^{(j')} - L^{(j')}) \leq \frac{1}{k}$; the inequality that $(\tilde{L}^{(j)} - L^{(j)}) - (\tilde{L}^{(j')} - L^{(j')}) \geq -\frac{1}{k}$ follows by symmetry. The maximum value of $k\tilde{L}^{(j)}$ is $\lfloor kL^{(j)} \rfloor + 1$ while the minimum value of $k\tilde{L}^{(j')}$ is $\lfloor kL^{(j')} \rfloor$, hence the result is immediate unless $(\lfloor kL^{(j)} \rfloor + 1) - kL^{(j)} + kL^{(j')} - \lfloor kL^{(j')} \rfloor > 1$, i.e. $kL^{(j')} - \lfloor kL^{(j')} \rfloor > kL^{(j)} - \lfloor kL^{(j)} \rfloor$. However, in this case, if $k\tilde{L}^{(j)} = \lfloor kL^{(j)} \rfloor + 1$, then $W < kL^{(j)} - \lfloor kL^{(j)} \rfloor < kL^{(j')} - \lfloor kL^{(j')} \rfloor$ and hence $\tilde{L}^{(j')}$ is rounded up as well. Similarly, if $\tilde{L}^{(j')}$ is rounded down, then $\tilde{L}^{(j)}$ must be rounded down as well. If $\tilde{L}^{(j)}$ and $\tilde{L}^{(j')}$ are rounded in the same direction, then (iii) holds.

Having established (A.8) and (A.9), we now show that (2.16)–(2.17) are satisfied.

First we prove (2.17), the claim that $\mathbb{E}[\tilde{\Phi}(\tilde{L}^{(j)})] \geq r^{(j)}$, inductively. Clearly $\mathbb{E}[\Phi(\tilde{L}^{(0)})] \geq r^{(0)} = 0$. Now consider $j \in [m]$ and suppose we have established (2.17) for the $j-1$ case. We can compare expression (2.19) with $q = \tilde{L}^{(j)}$ and $q = \tilde{L}^{(j-1)}$ to obtain $\tilde{\Phi}(\tilde{L}^{(j)}) = \tilde{\Phi}(\tilde{L}^{(j)}) + (r^{(j)} - r^{(j-1)})\frac{\exp(\tilde{L}^{(j)} - \tilde{L}^{(j-1)}) - 1}{\exp(\alpha^{(j)}) - 1}$. Therefore,

$$\begin{aligned}
\mathbb{E}[\tilde{\Phi}(\tilde{L}^{(j)})] &\geq r^{(j-1)} + (r^{(j)} - r^{(j-1)})\frac{\mathbb{E}[\exp(\tilde{L}^{(j)} - \tilde{L}^{(j-1)})] - 1}{\exp(\alpha^{(j)}) - 1} \\
&\geq r^{(j-1)} + (r^{(j)} - r^{(j-1)})\frac{\exp(\mathbb{E}[\tilde{L}^{(j)} - \tilde{L}^{(j-1)}]) - 1}{\exp(\alpha^{(j)}) - 1} \\
&= r^{(j-1)} + (r^{(j)} - r^{(j-1)})\frac{\exp(\alpha^{(j)}) - 1}{\exp(\alpha^{(j)}) - 1}
\end{aligned}$$

where the first inequality uses the induction hypothesis, and the second inequality uses Jensen's inequality (the exponential function exp is convex). The equality follows from (A.8) and the definition that $\alpha^{(j)} = L^{(j)} - L^{(j-1)}$, completing the induction.

Now we prove (2.16) for an arbitrary $j \in [m]$ and $N \in \{0, \ldots, \tilde{L}^{(j)}k - 1\}$. Let $q = \frac{N}{k}$ and $\ell = \tilde{\ell}(q)$. Note that $1 \leq \ell \leq j$, and $\tilde{L}^{(\ell-1)} \leq q < \tilde{L}^{(\ell)}$. Substituting $q = \frac{N}{k}$ into the LHS of (2.16), we get $k\big(\tilde{\Phi}(q + \frac{1}{k}) - \tilde{\Phi}(q)\big) + \tilde{\Phi}(\tilde{L}^{(j)}) - \tilde{\Phi}(q)$. Adding and subtracting $\tilde{\Phi}(\tilde{L}^{(\ell)})$ and rearranging, we get

$$k\big(\tilde{\Phi}(q+\frac{1}{k})-\tilde{\Phi}(q)\big)+\tilde{\Phi}(\tilde{L}^{(\ell)})-\tilde{\Phi}(q)+\tilde{\Phi}(\tilde{L}^{(j)})-\tilde{\Phi}(\tilde{L}^{(\ell)}). \tag{A.10}$$

The following upper bound can be derived for expression (A.10):

$$\begin{aligned}
&k\big(\tilde{\Phi}(q+\frac{1}{k})-\tilde{\Phi}(q)\big)+\tilde{\Phi}(\tilde{L}^{(\ell)})-\tilde{\Phi}(q)+\tilde{\Phi}(\tilde{L}^{(j)})-\tilde{\Phi}(\tilde{L}^{(\ell)})\\
=&(r^{(\ell)}-r^{(\ell-1)})\frac{e^{q+1/k-\tilde{L}^{(\ell-1)}}(k-(k+1)e^{-1/k})+e^{\tilde{L}^{(\ell)}-\tilde{L}^{(\ell-1)}}}{e^{\alpha^{(\ell)}}-1}+\sum_{\ell'=\ell+1}^{j}(r^{(\ell')}-r^{(\ell'-1)})\frac{e^{\tilde{L}^{(\ell')}-\tilde{L}^{(\ell'-1)}}-1}{e^{\alpha^{(\ell')}}-1}\\
\leq&(r^{(\ell)}-r^{(\ell-1)})\frac{e^{\tilde{L}^{(\ell)}-\tilde{L}^{(\ell-1)}}(k-(k+1)e^{-1/k})+e^{\tilde{L}^{(\ell)}-\tilde{L}^{(\ell-1)}}}{e^{\alpha^{(\ell)}}-1}+\sum_{\ell'=\ell+1}^{j}(r^{(\ell')}-r^{(\ell'-1)})\frac{e^{\tilde{L}^{(\ell')}-\tilde{L}^{(\ell'-1)}}-1}{e^{\alpha^{(\ell')}}-1}\\
=&(r^{(\ell)}-r^{(\ell-1)})\frac{e^{\tilde{L}^{(\ell)}-\tilde{L}^{(\ell-1)}-\alpha^{(\ell)}}(1+k)(1-e^{-1/k})}{1-e^{-\alpha^{(\ell)}}}+\sum_{\ell'=\ell+1}^{j}(r^{(\ell')}-r^{(\ell'-1)})\frac{e^{\tilde{L}^{(\ell')}-\tilde{L}^{(\ell'-1)}-\alpha^{(\ell')}}-e^{-\alpha^{(\ell')}}}{1-e^{-\alpha^{(\ell')}}}.
\end{aligned} \tag{A.11}$$

The inequality holds because $k-(1+k)e^{-1/k}>0$ for all $k\in\mathbb{N}$, and q is at most $\tilde{L}^{(\ell)}-1/k$.

It suffices to show that expression (A.11) is bounded from above by

$$r^{(j)}\frac{(1+k)(e^{1/k}-1)}{1-e^{-\alpha^{(1)}}}. \tag{A.12}$$

To assist in this task, we would like to establish the following for all $\ell'=\ell+1,\ldots,j$ and $\ell''\in\{\ell,\ldots,\ell'-1\}$:

$$\begin{aligned}
&(r^{(\ell'-1)}-r^{(\ell'-2)})\frac{e^{\tilde{L}^{(\ell'-1)}-\tilde{L}^{(\ell''-1)}-L^{(\ell'-1)}+L^{(\ell''-1)}}(1+k)(1-e^{-1/k})}{1-e^{-\alpha^{(\ell'-1)}}}\\
&+(r^{(\ell')}-r^{(\ell'-1)})\frac{e^{\tilde{L}^{(\ell')}-\tilde{L}^{(\ell'-1)}-\alpha^{(\ell')}}-e^{-\alpha^{(\ell')}}}{1-e^{-\alpha^{(\ell')}}}\\
\leq&(r^{(\ell')}-r^{(\ell'-1)})\frac{e^{\max\{\tilde{L}^{(\ell')}-\tilde{L}^{(\ell''-1)}-L^{(\ell')}+L^{(\ell''-1)},\tilde{L}^{(\ell')}-\tilde{L}^{(\ell'-1)}-L^{(\ell')}+L^{(\ell'-1)}\}}(1+k)(1-e^{-1/k})}{1-e^{-\alpha^{(\ell')}}}.
\end{aligned} \tag{A.13}$$

But $\frac{r^{(\ell'-1)}-r^{(\ell'-2)}}{1-e^{-\alpha^{(\ell'-1)}}}=\frac{r^{(\ell')}-r^{(\ell'-1)}}{1-e^{-\alpha^{(\ell')}}}\cdot\frac{r^{(\ell'-1)}}{r^{(\ell')}}$ due to the definition of α in (2.5), and $\frac{r^{(\ell'-1)}}{r^{(\ell')}}\leq e^{-\alpha^{(\ell')}}$ due to (A.4). Substituting back into inequality (A.13), it suffices to prove

$$\begin{aligned}
&e^{\tilde{L}^{(\ell'-1)}-\tilde{L}^{(\ell''-1)}-L^{(\ell')}+L^{(\ell''-1)}}(1+k)(1-e^{-1/k})+e^{\tilde{L}^{(\ell')}-\tilde{L}^{(\ell'-1)}-\alpha^{(\ell')}}-e^{-\alpha^{(\ell')}}\\
\leq&e^{\max\{\tilde{L}^{(\ell')}-\tilde{L}^{(\ell''-1)}-L^{(\ell')}+L^{(\ell''-1)},\tilde{L}^{(\ell')}-\tilde{L}^{(\ell'-1)}-L^{(\ell')}+L^{(\ell'-1)}\}}(1+k)(1-e^{-1/k})
\end{aligned}$$

where we have used Definition 2.2.3 to rewrite the first exponent. Now,

$$e^{\tilde{L}^{(\ell'-1)}-\tilde{L}^{(\ell''-1)}-L^{(\ell')}+L^{(\ell''-1)}}(k-(1+k)e^{-1/k})\leq e^{\tilde{L}^{(\ell')}-\tilde{L}^{(\ell''-1)}-L^{(\ell')}+L^{(\ell''-1)}}(k-(1+k)1-e^{-1/k}),$$

since $k-(1+k)e^{-1/k}>0$ and $\tilde{L}^{(\ell'-1)} \leq \tilde{L}^{(\ell')}$. Thus it remains to prove that

$$e^{\tilde{L}^{(\ell'-1)}-\tilde{L}^{(\ell''-1)}-L^{(\ell')}+L^{(\ell''-1)}} + e^{\tilde{L}^{(\ell')}-\tilde{L}^{(\ell'-1)}-\alpha^{(\ell')}} - e^{-\alpha^{(\ell')}}$$
$$\leq e^{\max\{\tilde{L}^{(\ell')}-\tilde{L}^{(\ell''-1)}-L^{(\ell')}+L^{(\ell''-1)},\tilde{L}^{(\ell')}-\tilde{L}^{(\ell'-1)}-L^{(\ell')}+L^{(\ell'-1)}\}}. \tag{A.14}$$

We consider two cases. First suppose $\tilde{L}^{(\ell')}-\tilde{L}^{(\ell''-1)}-L^{(\ell')}+L^{(\ell''-1)} \leq \tilde{L}^{(\ell')}-\tilde{L}^{(\ell'-1)}-L^{(\ell')}+L^{(\ell'-1)}$, i.e. $\tilde{L}^{(\ell'-1)}-\tilde{L}^{(\ell''-1)} \leq L^{(\ell'-1)}-L^{(\ell''-1)}$. Then the LHS of (A.14) equals $e^{-\alpha^{(\ell')}} + e^{\tilde{L}^{(\ell')}-\tilde{L}^{(\ell'-1)}-\alpha^{(\ell')}} - e^{-\alpha^{(\ell')}} = e^{\tilde{L}^{(\ell')}-\tilde{L}^{(\ell'-1)}-\alpha^{(\ell')}}$, which equals the RHS of (A.14) by the assumption that $\tilde{L}^{(\ell'-1)}-\tilde{L}^{(\ell''-1)} \leq L^{(\ell'-1)}-L^{(\ell''-1)}$. In the second case, suppose $\tilde{L}^{(\ell')}-\tilde{L}^{(\ell''-1)}-L^{(\ell')}+L^{(\ell''-1)} > \tilde{L}^{(\ell')}-\tilde{L}^{(\ell'-1)}-L^{(\ell')}+L^{(\ell'-1)}$, i.e. $\tilde{L}^{(\ell'-1)}-\tilde{L}^{(\ell''-1)} > L^{(\ell'-1)}-L^{(\ell''-1)}$. Then inequality (A.14) can be rearranged as

$$e^{-\alpha^{(\ell')}}(e^{\tilde{L}^{(\ell'-1)}-\tilde{L}^{(\ell''-1)}-L^{(\ell'-1)}+L^{(\ell''-1)}}-1)(e^{\tilde{L}^{(\ell')}-\tilde{L}^{(\ell'-1)}}-1) \geq 0.$$

The first bracket is positive by the assumption that $\tilde{L}^{(\ell'-1)}-\tilde{L}^{(\ell''-1)} > L^{(\ell'-1)}-L^{(\ell''-1)}$ and the second bracket is non-negative since $\tilde{L}^{(\ell'-1)} \leq \tilde{L}^{(\ell')}$. This finishes the proof of (A.14), and hence (A.13).

Equipped with (A.13), we return the task of proving that expression (A.11) is at most expression (A.12). If we inductively apply inequality (A.13) to expression (A.11) for $\ell' = \ell+1,\ldots,j$ (when $\ell'=\ell+1$, $\ell''=\ell$; when $\ell'=\ell+2$, $\ell''=\ell$ if we arrived at case two during iteration $\ell+1$ and $\ell''=\ell+1$ otherwise,...), we conclude that expression (A.11) is bounded from above by

$$(r^{(j)}-r^{(j-1)})\frac{e^{\tilde{L}^{(j)}-\tilde{L}^{(\ell''-1)}-L^{(j)}+L^{(\ell''-1)}}(1+k)(1-e^{-1/k})}{1-e^{-\alpha^{(j)}}}$$

for some $\ell'' \in \{\ell,\ldots,j\}$. The fact that $1-e^{-\alpha^{(1)}} = \frac{r^{(j)}}{r^{(j)}-r^{(j-1)}}(1-e^{-\alpha^{(j)}})$, due to (2.5), and the fact that $(\tilde{L}^{(j)}-\tilde{L}^{(\ell''-1)})-(L^{(j)}-L^{(\ell''-1)}) \leq 1/k$, due to (A.9), complete the proof of expression (A.11) being at most expression (A.12), and thus the proof of Theorem 2.3.3 for general m.

Finally, when $m=1$, $\alpha^{(1)}=1$. In the above proof, since j and ℓ are always 1, (A.11) can be replaced by $r^{(1)}\cdot\frac{(1)(1+k)(1-e^{-1/k})}{1-e^{-1}}$, where we have used the fact that $L^{(1)}=k$ always. This is immediately at most $\frac{r^{(j)}}{F}$, for the improved value of $F=\frac{1-e^{-\alpha^{(1)}}}{(1+k)(1-e^{-1/k})}$, completing the proof of Theorem 2.3.3 in its entirety. □

A.2.2 Optimizing the Randomized Procedure

We can explicitly formulate the optimization problem over randomized procedures for a single item with starting inventory k and m prices $r^{(1)}, \ldots, r^{(m)}$. Using the "balls in bins" counting argument, the number of configurations satisfying (2.13) is $D := \binom{k+m-1}{m-1}$.

We refer to these configurations in an arbitrary order using the index $d \in [D]$, where we let ρ_d denote the probability of choosing configuration d, $f_d(\cdot)$ denote the value function for d, and $L_d^{(j)}$ denote the value of $\tilde{L}^{(j)}$ under configuration d for all $j = 0, \ldots, m$. The optimization problem of satisfying (2.16)–(2.17) with a maximal value of F can be formulated as follows:

$$\widetilde{\mathsf{CR}} := \sup F \tag{A.15a}$$

$$k(f_d(\frac{N+1}{k}) - f_d(\frac{N}{k})) + f_d(L_d^{(j)}) - f_d(\frac{N}{k}) \leq \frac{r^{(j)}}{F} \quad d \in [D], j \in [m], 0 \leq N \leq kL_d^{(j)} - 1 \tag{A.15b}$$

$$f_d(1) \geq \ldots \geq f_d(\frac{1}{k}) \geq f_d(0) = 0 \qquad d \in [D] \tag{A.15c}$$

$$\sum_{d=1}^{D} \rho_d f_d(L_d^{(j)}) \geq r^{(j)} \qquad j \in [m] \tag{A.15d}$$

$$\sum_{d=1}^{D} \rho_d = 1 \tag{A.15e}$$

$$f_d(0), f_d(\frac{1}{k}), \ldots, f_d(1) \in \mathbb{R}; \rho_d \geq 0 \qquad d \in [D] \tag{A.15f}$$

Constraint (A.15b) corresponds to (2.16), constraint (A.15d) corresponds to (2.17), while constraint (A.15c) enforces the definition of a value function in (2.14). We let $\widetilde{\mathsf{CR}}$ denote the optimal objective value of (A.15). Unfortunately, it is difficult to solve (A.15) exactly, since the number of configurations D is exponential in the number of prices m, and constraint (A.15d) is non-linear.

Nonetheless, (A.15) is useful at determining the best competitive ratio which could be established *using our analysis*. We know that the randomized procedure from Definition 2.3.2 (based on Φ) is an optimal solution to (A.15) as $k \to \infty$, since it achieves the optimal competitive ratio possible.

We can also solve (A.15) exactly when $k = 1$, in which case $D = m$, where we will let $d \in [D]$ denote the configuration with $\tilde{L}^{(0)} = \ldots = \tilde{L}^{(d-1)} = 0$ and $\tilde{L}^{(d)} = \ldots = \tilde{L}^{(m)} = 1$. (A.15b) reduces to $2f_d(1) \leq \frac{r^{(j)}}{F}$, and needs to hold for $d \in [D]$, $j \geq d$ (for $j < d$, $kL_d^{(j)} - 1 = -1$). However, clearly only the constraint with $j = d$ is binding. As a result, (A.15b) corresponds to m constraints. (A.15d) corresponds to m constraints of the form $\sum_{d=1}^{j} \rho_d f_d(1) \geq r^{(j)}$, for $j \in [m]$.

Not counting $f_d(0)$, which must be set to 0, there are $2m+1$ variables: $\{f_d(1), \rho_d : d \in [D]\}$ and F. Consider the system of equations obtained in these $2m+1$ variables by setting (A.15b), (A.15d), and (A.15e) to equality. It can be checked that the unique solution is

$$f_d(1) = \frac{r^{(d)}}{\sigma^{(1)}}, \forall d \in [D]; \rho_d = \sigma^{(d)}, \forall d \in [D]; F = \frac{\sigma^{(1)}}{2} \tag{A.16}$$

with $\sigma^{(1)}, \ldots, \sigma^{(m)}$ defined from $r^{(1)}, \ldots, r^{(m)}$ according to (2.6). Furthermore, this solution is both feasible, satisfying the non-negativity constraints in (A.15c) and (A.15f), and optimal. Therefore, the value of $\widetilde{\mathsf{CR}}$ is $\frac{\sigma^{(1)}}{2}$.

A.2.3 Proof of Theorem 2.2.4

Now we put together the proof of Theorem 2.2.4. For all items $i \in [n]$, $\widetilde{\mathsf{CR}}_i$ is defined to be the optimal objective value of (A.15), with $k = k_i$, $m = m_i$, and $r^{(1)} = r_i^{(1)}, \ldots, r^{(m)} = r_i^{(m_i)}$. Consider Algorithm 1, where for all i, the randomized procedure used to initialize $\tilde{\Phi}_i$ is an optimal solution to (A.15) achieving the objective value of $\widetilde{\mathsf{CR}}_i$. For all i, (2.16)–(2.17) is satisfied as long as $F \leq \widetilde{\mathsf{CR}}_i$. Therefore, the maximum value of F satisfying the conditions of Theorem 2.3.1 is $\min_i \widetilde{\mathsf{CR}}_i$. By Theorem 2.3.1, this algorithm achieves a competitive ratio of $\min_i \widetilde{\mathsf{CR}}_i$.

To establish bounds (i)–(iii) from Theorem 2.2.4, for all i, we need to find a feasible randomized procedure with an objective value in (A.15) equal to the bound. For bounds (i) and (iii), this is established directly by the randomized procedure from Definition 2.3.2 and Theorem 2.3.3. For bound (ii), we need to split the k_i units of item i into k_i disparate items. For each single-unit item, its value function in Algorithm 1 is initialized according to the randomized procedure described by (A.16). This yields a value of $\frac{\sigma_i^{(1)}}{2}$, completing the proof of Theorem 2.2.4.

A.3 Deferred Proofs from Section 2.4

Proof. Proof of Lemma 2.4.1. Since the algorithm was willing to sell item i at price j, it must be the case that $W_i < L_i^{(j)}$. Let ℓ denote $\ell_i(W_i)$, which is at most j. Since we can ignore measure-zero events, we assume that $W_i \neq L^{(\ell-1)}$. We can rearrange Z_t as

$$\begin{aligned} & r_i^{(j)} - r_i^{(\ell)} + r_i^{(\ell)} - \left(r_i^{(\ell-1)} + (r_i^{(\ell)} - r_i^{(\ell-1)}) \frac{\exp(W_i - L_i^{(\ell-1)}) - 1}{\exp(\alpha_i^{(\ell)}) - 1} \right) \\ &= r_i^{(j)} - r_i^{(\ell)} + (r_i^{(\ell)} - r_i^{(\ell-1)}) \frac{\exp(\alpha_i^{(\ell)}) - \exp(W_i - L_i^{(\ell-1)})}{\exp(\alpha_i^{(\ell)}) - 1}. \end{aligned}$$

Adding $Y_i = \Phi_i'(W_i) = (r_i^{(\ell)} - r_i^{(\ell-1)}) \frac{\exp(W_i - L_i^{(\ell-1)})}{\exp(\alpha_i^{(\ell)}) - 1}$ to this expression, we get $r_i^{(j)} - r_i^{(\ell)} + \frac{r_i^{(\ell)} - r_i^{(\ell-1)}}{1 - \exp(-\alpha_i^{(\ell)})}$, which can be re-written as $r_i^{(j)} - r_i^{(\ell)} + \frac{r_i^{(\ell)}}{1 - \exp(-\alpha_i^{(1)})}$ due to (2.5). The result follows immediately. $\square$

Proof. Proof of Lemma 2.4.2. It suffices to show that constraint (2.18b) holds for all $t \in [T]$ and $i \in [n]$. Since $p_{t,i}^{(j)} \in \{0, 1\}$ and the constraint clearly holds when $p_{t,i}^{(j)} = 0$, it suffices to show that $\mathbb{E}[Y_i + Z_t] \geq r_i^{(j_{t,i})}$, where $j_{t,i} \neq 0$. We will let $j = j_{t,i}$ for brevity.

Fix the realization of $W_{i'}$ for all $i' \neq i$, and consider the run of the algorithm on a modified instance with item i removed. Having fixed the values of $W_{i'}$, such a run is deterministic. Let Z^{crit} denote the pseudorevenue earned on this run during time t, possibly 0. Φ_i maps $[0, L_i^{(j)}]$ to $[0, r_i^{(j)}]$ bijectively, so we can set W^{crit} to be the value in $[0, L_i^{(j)}]$ for which $\Phi_i(W^{\mathsf{crit}}) = \max\{r_i^{(j)} - Z^{\mathsf{crit}}, 0\}$.

We now consider the run of the algorithm on the full instance with item i, which is dependent on the realization of W_i. The following two claims from Devanur et al. (2013) generalize to our multi-price setting.

1. Dominance: if $W_i \in [0, W^{\mathsf{crit}})$, then in the run with item i, item i gets matched.

<u>Proof</u>: Since $W^{\mathsf{crit}} > W_i$ and $W_i \geq 0$, $W^{\mathsf{crit}} > 0$. Therefore, $\Phi_i(W^{\mathsf{crit}}) > 0$. Thus $\Phi_i(W^{\mathsf{crit}}) = r_i^{(j)} - Z^{\mathsf{crit}}$ (as opposed to $\Phi_i(W^{\mathsf{crit}}) = 0$), and moreover since $W_i < W^{\mathsf{crit}}$ and Φ_i is strictly increasing, $\Phi_i(W_i) < r_i^{(j)} - Z^{\mathsf{crit}}$. This implies $r_i^{(j)} - \Phi_i(W_i) > \max\{Z^{\mathsf{crit}}, 0\}$, since $Z^{\mathsf{crit}} \geq 0$. Thus on the run with item i, either i is already matched before time t, or it is matched to customer t.

2. Monotonicity: $Z_t \geq Z^{\mathsf{crit}}$ (regardless of the realization of W_i).

Proof: fix the realization of W_i. We compare two deterministic runs of the algorithm: one with item i, and one without. We can inductively establish over $t = 0, \ldots, T$ that at the end of time t, the set of unmatched items in the run with i is a superset of that in the run without i. Therefore, in the run with i, since the algorithm is maximizing pseudorevenue over a superset of items, its pseudorevenue Z_t can be no less than Z^{crit}.

Now, conditioned on the realizations of $W_{i'}$ for $i' \neq i$, which determines the values of Z^{crit} and W^{crit}, we have $Z_t \geq Z^{\text{crit}}$ (by Monotonicity) and in turn $Z^{\text{crit}} \geq r_i^{(j)} - \Phi_i(W^{\text{crit}})$ (by the definition of W^{crit}). Meanwhile, as long as i gets matched, Y_i gets set to $\Phi_i'(W_i)$, so by Dominance, $\mathbb{E}[Y_i|\{W_{i'} : i' \neq i\}] \geq \int_0^{W^{\text{crit}}} \Phi_i'(w)dw = \Phi_i(W^{\text{crit}}) - \Phi_i(0) = \Phi_i(W^{\text{crit}})$. Therefore, $\mathbb{E}[Y_i + Z_t|\{W_{i'} : i' \neq i\}] \geq r_i^{(j)}$. The proof follows from the tower property of conditional expectation. $\square$

A.4 Deferred Proofs from Section 2.5

Proof. Proof of Proposition 2.5.2. The unique solution to the system (2.21) is obtained inductively over $j = 2, \ldots, m$ by setting $B_j = \frac{r^{(j-1)}e^{-\alpha^{(j-1)}}}{r^{(j)}e^{-\alpha^{(j)}}} B_{j-1}$. By (A.4), $\frac{r^{(j-1)}}{r^{(j)}} \leq e^{-\alpha^{(j)}}$, hence $B_j \leq e^{-\alpha^{(j-1)}} B_{j-1}$. But $\alpha^{(j-1)} > 0$ by Proposition 2.2.2, completing the proof that $B_j < B_{j-1}$ for $j = 2, \ldots, m$. The fact that $0 < B_m$ is immediate. $\square$

Proof. Proof of Lemma 2.5.3. Consider the execution of an online algorithm on this randomized instance. For all $i \in [n]$ and group of customers $t \in [n]$, let $Q_{t,i}$ denote the number of group-t customers to which item π_i is sold, which is a random variable with respect to the random permutation π as well as any randomness in the algorithm. Let $q_{t,i} = \mathbb{E}[Q_{t,i}]$.

Clearly if $i < t$, then $Q_{t,i} = 0$, because group-t customers have no interest in item π_i. Otherwise, for any $i, i' \geq t$, we argue that $q_{t,i} = q_{t,i'}$. This is because while group t is arriving, the online algorithm cannot distinguish between items π_i and $\pi_{i'}$, hence any items it allocates are equally likely to be item π_i and item $\pi_{i'}$. Therefore, we let q_t denote the value of $q_{t,i}$ for $i \geq t$.

Now, consider item π_n. Since it only has k units of inventory, we know that $\sum_{t=1}^n Q_{t,n} \leq k$ on every sample path. Using the linearity of expectation, we get that

$$\sum_{t=1}^{n} q_t \leq k. \tag{A.17}$$

Furthermore, for a $t \in [n]$, on every sample path, $\sum_{i=t}^{n} Q_{t,i} \leq k$, since there are only k customers in group t. Therefore, $(n+1-t)q_t \leq k$, or

$$q_t \leq \frac{k}{n+1-t}. \tag{A.18}$$

For this proof, let $M_j = \sum_{j'=1}^{j} \beta_{j'}$, for all $j = 0, \ldots, m$. For all $j \in [m]$, let $\lambda_j = \frac{1}{k}\sum_{t=M_{j-1}n+1}^{M_j n} q_t$. Substituting into (A.17), we get the constraint that $\sum_{j=1}^{m} \lambda_j \leq 1$. For any $j \in [m-1]$, summing inequality (A.18) for $t = M_{j-1}n+1, \ldots, M_j n$ yields $\lambda_j \leq \ln \frac{B_j}{B_{j+1}}$, since $n \to \infty$, and $B_j = 1 - M_{j-1}$, $B_{j+1} = 1 - M_j$ by definition. It is also clear from definition that $\lambda_j \geq 0$ for all $j \in [m]$.

Finally, the total expected revenue is

$$\sum_{j=1}^{m} r^{(j)} \sum_{t=M_{j-1}n+1}^{M_j n} q_t(n+1-t), \tag{A.19}$$

since for each group t there are $n+1-t$ items for each of which q_t copies are sold in expectation. Consider any $j \in [m]$. Since $\sum_{t=M_{j-1}n+1}^{M_j n} q_t = \lambda_j k$ by definition, $\sum_{t=M_{j-1}n+1}^{M_j n} q_t(n+1-t)$ is maximized by setting q_t to its upper bound in (A.18) for $t = M_{j-1}n+1, M_{j-1}n+2, \ldots$ until the capacity of $\lambda_j k$ is reached. Since $n \to \infty$, we can simply compute the value of t for which

$$\frac{k}{n - M_{j-1}n} + \ldots + \frac{k}{n-t} = \lambda_j k, \tag{A.20}$$

with $t \in [M_{j-1}n, M_j n]$. Letting $t = (M_{j-1} + y\beta_j)n$ with $y \in [0,1]$, and using the definition of B_j, (A.20) becomes $\ln \frac{B_j}{B_j - y\beta_j} = \lambda_j$, or $y\beta_j = B_j(1 - e^{-\lambda_j})$. Therefore,

$$\begin{aligned}\sum_{t=M_{j-1}n+1}^{M_j n} q_t(n+1-t) &\leq \sum_{t=M_{j-1}n+1}^{(M_{j-1}+B_j(1-e^{-\lambda_j}))n} \frac{k}{n+1-t} \cdot (n+1-t) \\ &= B_j(1-e^{-\lambda_j})nk\end{aligned}$$

Substituting into (A.19), we get that the expected revenue of the online algorithm is at most (2.23), where $\sum_{j=1}^{m} \lambda_j \leq 1$, $\lambda_j \leq \ln \frac{B_j}{B_{j+1}}$ for $j \in [m-1]$, and $\lambda_j \geq 0$ for $j \in [m]$, completing the proof. $\square$

Proof. Proof of Lemma 2.5.4. We use backward induction over $j = m, \ldots, 1$. When $j = m$, (2.25) becomes $nkr^{(m)}B_m(1 - \exp(-\tau))$, since $A_m = \alpha^{(m)}$ by definition. Meanwhile, (2.24) is maximized by setting $\lambda_m = \tau$, resulting in the same expression and establishing the base case.

Now suppose $j < m$ and that we have already established the lemma in the $j + 1$ case. If we set $\lambda_j = \lambda$, for some $\lambda \in [0, \tau]$, then the maximum value of (2.24) subject to $\lambda_{j+1}, \ldots, \lambda_m \geq 0$ and $\lambda_{j+1} + \ldots + \lambda_m \leq \tau - \lambda$ is, by the inductive hypothesis,

$$r^{(j)}B_j(1 - \exp(-\lambda))nk + nk \sum_{\ell=j+1}^{m} r^{(\ell)}B_\ell\Big(1 - \exp\big(-\alpha^{(\ell)} + \frac{A_{j+1} - (\tau - \lambda)}{m - (j+1) + 1}\big)\Big). \quad (A.21)$$

Consider this expression as a function of λ. The derivative is

$$r^{(j)}B_j \exp(-\lambda)nk + nk \sum_{\ell=j+1}^{m} r^{(\ell)}B_\ell \cdot \frac{-1}{m - j} \cdot \exp\big(-\alpha^{(\ell)} + \frac{A_{j+1} - (\tau - \lambda)}{m - j}\big) \quad (A.22)$$

and the second derivative is clearly negative, so the function is concave. Therefore, it is maximized by setting the derivative to 0. By definition (2.21), $r^{(\ell)}B_\ell e^{-\alpha^{(\ell)}}$ is identical for all $\ell = j + 1, \ldots, m$, and equal to $r^{(j)}B_j e^{-\alpha^{(j)}}$. Thus setting (A.22) to 0 implies:

$$\begin{aligned}\exp(\alpha^{(j)} - \lambda) &= \frac{1}{m - j} \sum_{\ell=j+1}^{m} \exp\big(\frac{A_{j+1} - (\tau - \lambda)}{m - j}\big)\\ \alpha^{(j)} - \lambda &= \frac{A_{j+1} - (\tau - \lambda)}{m - j}.\end{aligned}$$

Rearranging and using the definition that $A_{j+1} = A_j - \alpha^{(j)}$, we get $\lambda = \alpha^{(j)} - \frac{A_j - \tau}{m-j+1}$. Substituting this value of λ into (A.21), the expression $\frac{A_{j+1}-(\tau-\lambda)}{m-(j+1)+1}$ is equal to $\frac{A_j-\tau}{m-j+1}$, hence (A.21) is equal to (2.25), completing the induction and the proof of the lemma. □

A.5 Deriving the Multi-price Value Function Φ_i

Throughout this chapter, we have proven results critically dependent on the exact definitions of $\alpha_i^{(1)}, \ldots, \alpha_i^{(m_i)}$ in (2.5), and Φ_i in (2.8). In this section we explain how to derive the system of equations in (2.5), and the functional form in (2.8). In Subsection A.5.1, we use the same method to derive the optimal value function when the price of an item i can take any value in the continuum $[r^{\min}, r^{\max}]$. We omit the subscript i throughout this section.

Consider constraints (2.16)–(2.17) in Theorem 2.3.3 for a single item with $k \to \infty$. Let $w = \frac{N}{k}$, and we deterministically set $\tilde{\Phi}$ to some Φ. The goal is to solve for the Φ which maximizes the value of F.

Observe that

$$\lim_{k\to\infty} k(\Phi(\frac{N+1}{k}) - \Phi(\frac{N}{k})) = \lim_{k\to\infty} \frac{\Phi(w+1/k) - \Phi(w)}{1/k},$$

which is equal to the derivative of Φ as w, by definition (Φ will end up not being differentiable on a discrete set of measure 0, which can be ignored). Therefore, (2.16) is equivalent to

$$\Phi'(w) - \Phi(w) \leq r^{(j)}(\frac{1}{F} - 1), \tag{A.23}$$

and needs to hold for all $j \in [m], w \in [0, L^{(j)}]$. For a fixed $w \in (L^{(j-1)}, L^{(j)})$, (A.23) needs to hold for all $j' = j, \dots, m$, but is clearly binding when $j' = j$. Therefore, it suffices to fix a $j \in [m]$ and consider (A.23) when $w \in (L^{(j-1)}, L^{(j)})$.

We should point out that this simplification via the "binding" argument is not possible for a finite k and random $\tilde{\Phi}$, because then (A.23) becomes $\tilde{\Phi}'(w) - \tilde{\Phi}(w) \leq \frac{r^{(j)}}{F} - \tilde{\Phi}(L^{(j)})$, and the RHS in fact may not be increasing in j.

If we set (A.23) to equality for some $j \in [m]$ and all $w \in (L^{(j-1)}, L^{(j)})$, and solve the differential equation, we get that $\Phi(w)$ must be of the form $Ce^w - r^{(j)}(\frac{1}{F} - 1)$ on $(L^{(j-1)}, L^{(j)})$. Setting $\Phi(L^{(j-1)}) = r^{(j-1)}$ and $\Phi(L^{(j)}) = r^{(j)}$, we obtain

$$\begin{aligned} C &= \frac{r^{(j)} - r^{(j-1)}}{e^{L^{(j)}} - e^{L^{(j-1)}}}; \\ F &= \frac{1}{1 - \frac{r^{(j-1)}}{r^{(j)}}} \cdot (1 - e^{-\alpha^{(j)}}). \end{aligned} \tag{A.24}$$

The RHS of (A.24) is the largest value of F which allows (A.23) to hold on segment j. It is dependent on $\alpha^{(j)}$, which is equal to $L^{(j)} - L^{(j-1)}$, the length of segment j. For (A.23) to hold on all segments $j \in [m]$, F must be set to $\min_j \frac{1}{1-r^{(j-1)}/r^{(j)}} \cdot (1 - e^{-\alpha^{(j)}})$.

Therefore, we would like to choose segment lengths $\alpha^{(1)}, \dots, \alpha^{(m)}$ summing to 1 to maximize the minimum $\frac{1}{1-r^{(j-1)}/r^{(j)}} \cdot (1 - e^{-\alpha^{(j)}})$, which is accomplished by setting $\frac{1}{1-r^{(j-1)}/r^{(j)}} \cdot (1 - e^{-\alpha^{(j)}})$ equal for all $j \in [m]$. This yields the system of equations (2.5), and Proposition 2.2.2. The resulting value of F is equal to $1 - e^{-\alpha^{(1)}}$, since $r^{(0)} = 0$. The resulting value of C, when substituted into the equation for $\Phi(w)$ on each segment $(L^{(j-1)}, L^{(j)})$, yields (2.8).

The derivation of Φ we just completed, starting from condition (A.23), comes from our analysis of MULTI-PRICE BALANCE. We note that the exact same inequality (A.23) can also be derived from our analysis of MULTI-PRICE RANKING.

A.5.1 Continuum of Feasible Prices

Let the feasible price set for the item be $[r^{\min}, r^{\max}]$, where $0 < r^{\min} < r^{\max}$. Using the same "binding" argument, it suffices to maximize the value of F for which the following can hold:

$$\Phi'(w) - \Phi(w) \leq r^{\min}(\frac{1}{F} - 1), \qquad w \in (0, \alpha); \qquad \text{(A.25)}$$

$$\Phi'(w) - \frac{\Phi(w)}{F} \leq 0, \qquad w \in (\alpha, 1). \qquad \text{(A.26)}$$

Φ must also satisfy $\Phi(0) = 0, \Phi(\alpha) = r^{\min}, \Phi(1) = r^{\max}$, while $\alpha \in (0,1)$ is an arbitrary "booking limit" for the lowest price of $r^{\min}$.

We know from before that under the optimal solution to (A.25), the value of F can be at most $1 - e^{-\alpha}$. Solving the differential equation where (A.26) is set to equality, $\Phi(w)$ must take the form $Ce^{w/F}$ on $(\alpha, 1)$. Substituting $\Phi(\alpha) = r^{\min}$ and $\Phi(1) = r^{\max}$ yields

$$C = (r^{\min})^{\frac{1}{1-\alpha}}(r^{\max})^{-\frac{\alpha}{1-\alpha}};$$
$$F = \frac{1-\alpha}{\ln \frac{r^{\max}}{r^{\min}}}.$$

Therefore, the value of F is also bounded from above by $\frac{1-\alpha}{\ln(r^{\max}/r^{\min})}$. F is maximized by setting $\frac{1-\alpha}{\ln(r^{\max}/r^{\min})}$ equal to the other upper bound of $1 - e^{-\alpha}$; the value at which equality is achieved is then the competitive ratio.

Letting $R = \ln(r^{\max}/r^{\min})$, the solution to $\frac{1-\alpha}{R} = 1 - e^{-\alpha}$ can be written as $W(Re^{R-1}) - R + 1$, where W is the Lambert-W function, the inverse function to $f(x) = xe^x$ for $x \in \mathbb{R}_{\geq 0}$.

Indeed, when $\alpha = W(Re^{R-1})$, the following can be derived:

$$\begin{aligned}
\frac{1-\alpha}{R} &= 1 - e^{-\alpha} \\
Re^{-\alpha} &= \alpha + R - 1 \\
Re^{R-1} &= (\alpha + R - 1)e^{\alpha+R-1} \\
W(Re^{R-1}) &= \alpha + R - 1
\end{aligned}$$

Substituting $\alpha = W(\ln(r^{\max}/r^{\min})e^{\ln(r^{\max}/r^{\min})-1}) - \ln(r^{\max}/r^{\min}) + 1$ into the formula for C, and using the fact that $\Phi(w) = Ce^{w/F}$, we get

$$\Phi(w) = (r^{\min})^{\frac{1-w}{1-\alpha}}(r^{\max})^{\frac{w-\alpha}{1-\alpha}}, \qquad w \in [\alpha, 1].$$

Meanwhile, the derivation preceding Subsection A.5.1 implies that

$$\Phi(w) = r^{\min} \cdot \frac{e^w - 1}{e^\alpha - 1}, \qquad w \in [0, \alpha].$$

It can be checked that indeed $\Phi(0) = 0$, $\Phi(\alpha) = r^{\min}$ (Φ is continuous at $w = \alpha$), and $\Phi(1) = r^{\max}$. Furthermore, unlike the case of discrete prices, it can be checked that Φ is also differentiable at $w = \alpha$ (on $[\alpha, 1]$, use the form that $\Phi(w) = Ce^{w/F}$, hence $\Phi'(\alpha) = \frac{\Phi(\alpha)}{F}$).

A.6 Supplement to Numerical Experiments

We provide additional details about our choice estimation. We define 8 customer types, one for each combination of the 3 following binary features.

1. Group: whether the customer indicated a party size greater than 1.

2. CRO: whether the customer booked using the Central Reservation Office, as opposed to the hotel's website or a Global Distribution System (for details on these terms, see Bodea et al. (2009)).

3. VIP: whether the customer had any kind of VIP status.

We did not use features such as: whether the booking date is a weekend, whether the check-in date is a weekend, the length of stay, or the number of days in advance booked. Such features did not result in a more predictive model.

We estimate the mean MNL utilities for each of the 8 products separately for each customer type. The results are displayed in Table A.1. The total share of each customer type (out of all the transactions) is also displayed. We should point out that it is possible for a customer to choose the higher fare for a room, even if the lower fare was also offered. This is because the higher fares are often packaged with additional offers, such as airline services, city attractions, in-room services, etc.

We have shifted the mean utilities so that for each customer type, the weights of both the no-purchase option, and the most-preferred purchase option, is equal to 0. The large weights on the no-purchase options ensure that the revenue-maximizing assortments tend to include both the low and high fares.

In the setting with greater fare differentiation (Subsection 2.7.5), the high prices of the King, Queen, Suite, and Two-double rooms are adjusted to \$614, \$608, \$768, \$612, respectively (twice the lower fares). The mean utility of the no-purchase option is increased by 2 for every customer type, to ensure that the revenue-maximizing assortments still include both the low and high fares.

A.6.1 Details on the Forecasting Bid-price Algorithms

To forecast the remaining number of customers, we assume that we know the average number of customers interested in each occupancy date (1340), as well as the overall trend for how far in advance customers book, which is plotted in Figure A-1. As an example of how to use these numbers, consider the occupancy date March 31st. At the start, we forecast there to be 1340 arrivals. However, suppose by March 6th, 500 customers have arrived. Since we know from Figure A-1 that roughly 50% of the total population interested in March 31st will have already booked by March 6th (25 days in advance), we expect there to only be 500 customers remaining.

To forecast the breakdown of remaining customers by type, we assume that we know the aggregate distribution of customer type over all occupancy dates. For example, from Section A.6, we know that 28% of all customers are of Type 3. Then we would estimate $28\% \times 500 = 140$ of the 500 remaining customers to be of Type 3. Alternatively, one can try to learn the specific distribution of customers interested in March 31st. Suppose that only 100, or 20%, of the 500 bookings made before March 6th came from customers of Type 3. Then we would instead estimate $20\% \times 500 = 100$ of the 500 remaining customers to be of

Table A.1: MNL choice models for the 8 customer types. The suffix "L" on a room type means lower fare, while the suffix "H" on a room type means higher fare.

Customer Type				MNL Mean Utilities								
Group?	CRO?	VIP?	Share	KingL	QueenL	SuiteL	2DoubleL	KingH	QueenH	SuiteH	2DoubleH	NoBuy
			0.16	-0.36	-1.22	-2.56	-1.04	0	-0.23	-2.25	-1.8	0
		✓	0.03	-0.82	-1.98	-2.16	-2.09	0	-1.02	-1.45	-1.82	0
	✓		0.28	-1.67	$-\infty$	-3.78	-2.71	0	-1.33	-1.8	-1.58	0
	✓	✓	0.09	-2.13	$-\infty$	-3.38	-3.76	0	-2.12	-1	-1.59	0
✓			0.19	-0.54	-0.97	-2.26	0	-0.91	-1.47	-2.78	-1.41	0
✓		✓	0.04	-0.09	-0.82	-0.95	-0.14	0	-1.35	-1.07	-0.51	0
✓	✓		0.18	-0.93	$-\infty$	-2.56	-0.76	0	-1.66	-1.41	-0.27	0
✓	✓	✓	0.03	-1.39	$-\infty$	-2.16	-1.8	0	-2.45	-0.61	-0.28	0

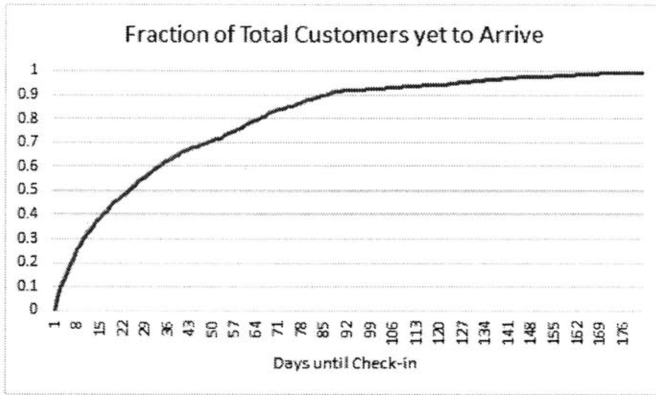


Figure A-1: Distribution of arrivals over the days before check-in, formed by aggregating all transactions.

Type 3.

To use the forecasted information, algorithms incorporate it into the LP (2.27), and set the *bid price* of each item i equal to the shadow price of constraint i in (2.27b). These algorithms then offer each customer t the assortment S (from the available items) maximizing $\sum_{(i,j)\in S} p_{t,i}^{(j)}(S)\left(r_i^{(j)} - \lambda_i\right)$.

We clarify the exact way in which the forecasted information is incorporated into the LP. Let there be A customer types, indexed by $a = 1, \dots, A$. We use $p_{a,i}^{(j)}(S)$ to denote the probability of a customer of type a choosing product (i, j) from assortment S. Suppose that when we want to re-solve the LP (2.27), the forecasted number of remaining customers of type a is N_a, for all $a \in [A]$, and the remaining inventory of item i is K_i, for all $i \in [n]$. We can formulate the following LP, which is a modification of (2.27):

$$\begin{aligned}
\max \sum_{a=1}^{A} \sum_{S} x_a(S) \sum_{(i,j)\in S} r_i^{(j)} p_{a,i}^{(j)}(S) & & \\
\sum_{a=1}^{A} \sum_{S} x_a(S) \sum_{j:(i,j)\in S} p_{a,i}^{(j)}(S) \le k_i & & i \in [n] \\
\sum_{S} x_a(S) = N_a & & a \in [A] \\
x_a(S) \ge 0 & \qquad a \in [A], S \subseteq \{(i,j) : i \in [n], j \in [m_i]\} &
\end{aligned}$$

We have set $T = \sum_{a=1}^{A} N_a$ and $|\{t : \text{type of customer } t \text{ is } a\}| = N_a$; note that the ordering of remaining customers is inconsequential for the LP.

Although this LP has an exponential number of variables, it can easily be solved using column generation (e.g., see Liu and Van Ryzin (2008)). Fix an optimal primal solution $(x_a^*(S) : a \in [A], S \subseteq \{(i,j) : i \in [n], j \in [m_i]\})$ and an optimal dual solution $(y_i^* : i \in [n]), (z_a^* : a \in [A])$. The bid-price algorithm sets the bid price of each item i equal to y_i^*.

We should point out that for every bid-price algorithm based on dual variables, there is a corresponding *random assignment* algorithm based on primal variables. Such an algorithm would, for each customer type a, offer each assortment S with probability proportional to $x_a^*(S)$. We have confirmed that these algorithms perform similarly in the simulations. We compare with the bid-price algorithms instead of the random assignment algorithms because they follow a form more similar to our MULTI-PRICE BALANCE algorithm.

Appendix B

Appendix to Chapter 3

B.1 Exponential Inventory Balancing on Our Problem

First we describe an example with large starting inventories before discussing the why using an exponential penalty function does not seem to improve the competitive ratio for our problem, even as the starting inventories become large.

B.1.1 Example

Consider a "scaled-up" version of Example 3.2.10 with initial inventories and selling season multiplied by k, a large integer. Now the randomized instance is defined as follows:

- $n = 3$;
- $r_1 = r_1^{\mathsf{disc}} = 0$, $b_1 = kN$;
- $r_2 = r_2^{\mathsf{disc}} = \frac{1}{N}$, $b_2 = kN$;
- $r_3 = r_3^{\mathsf{disc}} = M$, $b_3 = k$;
- $\mathcal{S}_1 = \{\{1,2\}\}$; $\phi_1(\{1,2\},\{1,2\}) = 1$;
- $\mathcal{S}_2 = \{\{2,3\}\}$; $\phi_2(\{2,3\},\{2,3\}) = \frac{1}{N}$;
- The arrivals are:
  - $T = kN, i_1 = \ldots = i_{kN} = 1$, w.p. $1 - \frac{C}{M}$;
  - $T = 2kN, i_1 = \ldots = i_{kN} = 1, i_{kN+1} = \ldots = i_{2kN} = 2$, w.p. $\frac{C}{M}$.

The expected value of $\mathsf{OPT}(i_1, \ldots, i_T)$ has been multiplied by k, so it is now $k(1+C)$. Intuitively, the offline fractional solution will always extract the full revenue of $\frac{1}{N} \cdot kN = k$ from item 2, and extract revenue of $M \cdot k$ from item 3 w.p. $\frac{C}{M}$. Details on the feasible LP solutions can be found in the proof of Theorem 3.2.11.

Now, consider any online algorithm. It faces the same conundrum as before: it must determine what fraction of the first kN customers to offer item 2 to without knowing whether the subsequent kN customers will come. Let $\alpha \in [0,1]$ be the fraction of the first kN customers served. If the subsequent kN customers come, $(1-\alpha)kN$ units of item 2 would be remaining for the purpose of selling item 3. The number of units of item 3 sold is then $\min\{X, k\}$, where X is a Binomial random variable with parameters $((1-\alpha)kN, \frac{1}{N})$. Clearly

$$\mathbb{E}[\min\{X, k\}] < \mathbb{E}[X] = (1-\alpha)k. \tag{B.1}$$

In the proof of Theorem 3.2.11 we explicitly compute the difference between $\mathbb{E}[\min\{X, k\}]$ and $\mathbb{E}[X]$, although for any fixed $\alpha < 1$ the law of large numbers suggests that $\mathbb{E}[\min\{X, k\}] \to \mathbb{E}[X]$ as $k \to \infty$.

Nonetheless, for any k, we can conclude from (B.1) that the expected revenue of an online algorithm serving α of the first kN customers is less than $\alpha k + C(1-\alpha)k$, as $M, N \to \infty$ (for details, see the proof of Theorem 3.2.11). Choosing $C = 1$, $\alpha k + C(1-\alpha)k = k$ regardless of what α the algorithm chooses, and $\mathbb{E}[\mathsf{OPT}(i_1, \ldots, i_T)] = 2k$. Therefore, the competitive ratio must be less than $\frac{1}{2}$ even as $\min\{b_1, \ldots, b_n\} \to \infty$.

B.1.2 Comparison of Exponential Inventory Balancing and Protection Level in Expectation

For their problem, Golrezaei et al. (2014) recommend offering a revenue-maximizing assortment during every time period, where the revenue of each item is scaled by a function of its fraction of starting inventory remaining. This function is called a *penalty function*. The intuition is that if a large fraction of an item has already been sold, then lower priority should given to selling the remaining units. As the starting inventories become large, their *exponential* penalty function accomplishes the optimal trade-off between selling a high-revenue item versus selling a low-revenue item that may not be wanted later.

In our problem, there a further trade-off between selling a high-revenue item versus saving it for *even higher* revenue later. As a result, selling too much of an item as an add-on has the added downside that the inventory could be being disposed for an arbitrarily small fraction of its potential revenue, and every unit depleted as an add-on incurs the *same* risk in revenue loss. The example in the previous subsection illustrates this—every unit of item 2 sold to the first kN customers loses the same fraction ($\frac{1}{N}$) of item 3's large revenue (M), should the final kN customers come.

Therefore, placing higher value on the final remaining units of each item in the form of an exponential penalty function appears to yield no benefit. Instead, our algorithm hedges against the worst-case arrival sequence by withholding half of the inventory of each item (in expectation) from being sold as an add-on. In that sense, our algorithm also scales the revenue of each item by a penalty factor, but our penalty factor is a function of the item's *expected* fraction of starting inventory remaining, and that function over $[0,1]$ is the function which is 0 on $[0,\frac{1}{2}]$ and 1 on $(\frac{1}{2},1]$.

B.2 Example Demonstrating the Importance of Correlation and Protection Level in Expectation

- $n = 3$; $b_1 = 1, b_2 = 1, b_3 = 2$;
- all revenues and discounted revenues are 1;
- $\mathcal{S}_1 = \{\{1,2,3\}\}$:
  - $\phi_1(\{1,2\},\{1,2,3\}) = \frac{1}{2}$; $\phi_1(\{1,3\},\{1,2,3\}) = \frac{1}{2}$;
  - $\phi_1(\{1,2\},\{1,2\}) = \frac{1}{2}$;
  - $\phi_1(\{1,3\},\{1,3\}) = \frac{1}{2}$;
- $\mathcal{S}_2 = \{\{2,3\}\}$; $\phi_2(\{2,3\},\{2,3\}) = 1$;
- $T = 2, i_1 = 1, i_2 = 2$.

Our algorithm would execute as follows. It offers $\{1,2,3\}$ to the first customer. Half the time, item 2 is sold and the second customer cannot be served. The other half, item 3 is sold, and sold again when it is offered as an add-on for the second customer.

In the end, items 1 and 2 are both guaranteed to be sold (since a customer of that type arrived), while the units of item 3 sold is either 0 (if the first customer bought item 2) or 2 (if the first customer bought item 3). Note that using a deterministic protection level of $\frac{1}{2}b_3 = 1$ would not have sufficed to achieve these sale probabilities of item 3. The negative correlation between the sales of items 2 and 3 during $t = 1$ was very relevant here.

B.3 Omitted Proofs

Proof. Proof of Lemma 3.2.7. Consider any algorithm, which could know $i_1, \ldots, i_T$ at the start of the time horizon, and consider a sample path with that algorithm. For all $t \in [T]$ and $S \in \mathcal{S}_{i_t}$, let $Y^t(S)$ be the indicator random variable for assortment S being offered during time t. For all $t \in [T]$ and $j \in [n]$, let P_j^t be the indicator random variable for customer t buying item j.

Clearly, at most one assortment from $\mathcal{S}_{i_t}$ can be offered during each $t \in [T]$, so we have $\sum_{S \in \mathcal{S}_{i_t}} Y^t(S) \leq 1$. Taking the expectation (with respect to both the randomness in the algorithm, and the randomness in the customers' decisions) on both sides and letting $y^t(S) := \mathbb{E}[Y^t(S)]$, we have

$$\sum_{S \in \mathcal{S}_{i_t}} y^t(S) \leq 1.$$

During time t, inventory of item $j \in [n]$ gets depleted if and only if $P_j^t = 1$, which is only possible if some $S \in \mathcal{S}_{i_t}$ is offered. Therefore, the depletion of item j during time t is equal to $\sum_{S \in \mathcal{S}_{i_t}} P_j^t \cdot Y^t(S)$. Conditioned on $Y^t(S) = 1$, P_j^t is an independent binary random variable which is 1 with probability $p_{i_t j}(S)$. Since the total depletion of item j over all the time periods cannot exceed b_j, we have

$$\begin{aligned}
\sum_{t=1}^{T} \sum_{S \in \mathcal{S}_{i_t}} P_j^t \cdot Y^t(S) &\leq b_j, \\
\sum_{t=1}^{T} \sum_{S \in \mathcal{S}_{i_t}} \mathbb{E}[P_j^t | Y^t(S) = 1] \cdot \Pr[Y^t(S) = 1] &\leq b_j, \\
\sum_{t=1}^{T} \sum_{S \in \mathcal{S}_{i_t}} p_{i_t j}(S) y^t(S) &\leq b_j.
\end{aligned}$$

Using the same arguments, the revenue on a run of the algorithm is

$$\sum_{t=1}^{T} \sum_{S \in \mathcal{S}_{i_t}} (r_{i_t} P_{i_t}^t + \sum_{j \neq i_t} r_j^{\mathsf{disc}} P_j^t) Y^t(S),$$

whose expectation is

$$\sum_{t=1}^{T} \sum_{S \in \mathcal{S}_{i_t}} (r_{i_t} + \sum_{j \neq i_t} r_j^{\mathsf{disc}} p_{i_t j}(S)) y^t(S),$$

since $p_{i_t i_t}(S) = 1$ for all $S \in \mathcal{S}_{i_t}$.

We have shown that for every algorithm, its expected revenue is equal to the objective value of the LP on a feasible solution of the LP. Therefore, the expected revenue of the algorithm is no greater than $\mathsf{OPT}(i_1, \ldots, i_T)$, completing the proof of Lemma 3.2.7. □

Proof. Proof of Theorem 3.2.11. First we show that the expected value of $\mathsf{OPT}(i_1, \ldots, i_T)$ is at least $C + 1$. In the case customers $N + 1, \ldots, 2N$ don't arrive (which occurs with probability $1 - \frac{C}{M}$), a revenue of $N(\frac{1}{N}) = 1$ can be obtained by offering assortment $\{1, 2\}$ during each of $t = 1, \ldots, N$. In the case customers $N + 1, \ldots, 2N$ arrive (which occurs with probability $\frac{C}{M}$), it can be checked that $y^1(\{1\}) = \ldots = y^N(\{1\}) = 1$, $y^{N+1}(\{2, 3\}) = \ldots = y^{2N}(\{2, 3\}) = 1$ is a feasible solution to the LP. Therefore, in this case, $\mathsf{OPT}(i_1, \ldots, i_T) \geq N(\frac{1}{N} + \frac{M}{N}) = 1 + M$. The expected value of $\mathsf{OPT}(i_1, \ldots, i_T)$ is at least

$$(1 - \frac{C}{M})(1) + \frac{C}{M}(1 + M) = 1 + C,$$

as desired.

Now consider any online algorithm. It must determine what fraction of the first N customers to offer item 2 to (as an add-on), without knowing whether the subsequent N customers requesting item 2 will arrive. Let $\alpha \in [0, 1]$ denote the fraction of the first N customers the online algorithm offers item 2 to. After deciding α, should customers $N + 1, \ldots, 2N$ arrive, the best the algorithm can do is use every remaining unit of item 2 to try to sell item 3. That is, the algorithm should serve $N - \alpha N$ of the customers of type 2, and offer item 3 as an add-on while it is available.

The expected revenue of such an algorithm is

$$\alpha N(\frac{1}{N}) + \frac{C}{M}\Big((N - \alpha N)(\frac{1}{N}) + \big(1 - (1 - \frac{1}{N})^{N - \alpha N}\big) M\Big), \tag{B.2}$$

which can be explained as follows:

- αN of the first N customers are offered item 2 as an add-on, guaranteeing a revenue of $\frac{1}{N}$ per customer;
- only if the next N customers arrive (which occurs w.p. $\frac{C}{M}$), is there additional revenue:
  - the remaining $N - \alpha N$ units of item 2 will be sold at the price of $\frac{1}{N}$ each;
  - the probability that item 3 is *not* sold is equal to the probability that all $N-\alpha N$ customers are offered $\{2,3\}$ and reject item 3, which occurs with probability $(1-\frac{1}{N})^{N-\alpha N}$.

(B.2) can be rearranged as $\alpha + C\big(1-(1-\frac{1}{N})^{N-\alpha N}\big) + \frac{C(1-\alpha)}{M}$. Taking the limit as $M, N \to \infty$, the expression is equal to

$$\alpha + C(1-e^{\alpha-1}).$$

We would like to argue that this expression, which is a continuous and differentiable function of α over the domain $[0,1]$, is maximized at $\alpha = 1-\ln C$. Indeed, its derivative is $1-Ce^{\alpha-1}$, which is positive for $\alpha \in [0, 1-\ln C)$, 0 when $\alpha = 1-\ln C$, and negative for $\alpha \in (1-\ln C, 1]$.

Therefore, the expected revenue of any online algorithm is no greater than $1 - \ln C + C(1-\frac{1}{C}) = C - \ln C$. Thus the competitive ratio of any online algorithm cannot exceed

$$\frac{C-\ln C}{C+1}.$$

Substituting in the optimized value of $C = \frac{1}{w}$ (where $we^w = 1$) completes the proof of Theorem 3.2.11. $\square$

Proof. Proof of Proposition 3.3.6. By the linearity of expectation, the expected units of item j sold as an add-on after the iteration equals the expected units of item j sold as an add-on before the iteration plus the probability that item j is sold as an add-on during the iteration. That is, if Step 6 is executed, then

$$\begin{aligned} &d_j(t+1, i_1, \ldots, i_{t+1}, \mathcal{L}^1, \ldots, \mathcal{L}^{t-1}, \mathcal{L}, ()) \\ &= d_j(t, i_1, \ldots, i_t, \mathcal{L}^1, \ldots, \mathcal{L}^{t-1}, \mathcal{L}) + h_j(t, i_1, \ldots, i_t, \mathcal{L}^1, \ldots, \mathcal{L}^{t-1}, \mathcal{L}) \end{aligned}$$

(for any i_{t+1}) and the desired inequality follows from the triangle inequality. On the other hand, if Step 11 is executed, then

$$\begin{aligned}
&d_j(t, i_1, \ldots, i_t, \mathcal{L}^1, \ldots, \mathcal{L}^{t-1}, \mathcal{L} + (\ell, \rho)) \\
&= d_j(t, i_1, \ldots, i_t, \mathcal{L}^1, \ldots, \mathcal{L}^{t-1}, \mathcal{L}) + \rho \cdot h_j(t, i_1, \ldots, i_t, \mathcal{L}^1, \ldots, \mathcal{L}^{t-1}, \mathcal{L})
\end{aligned}$$

and the desired inequality follows from the triangle inequality combined with the fact that $\rho < 1$. $\square$

Proof. Proof of Proposition 3.3.7. If the d_j^{curr}'s are updated in Step 6, then $d_j^{\text{curr}} \leq \frac{b_j}{2} - \varepsilon$ by the guarantee in Step 5, while if they are updated in Step 11, then $\rho \cdot \hat{h}_j \leq (\frac{b_j}{2} - \varepsilon) - d_j^{\text{curr}}$ (since $\hat{h}_j \geq 0$) by the choice of ρ. $\square$

Proof. Proof of Proposition 3.3.8. The first statement follows from the definition of $\mathcal{F}_K^t$, which is the set of items that appear in $\mathcal{L}^1, \ldots, \mathcal{L}^{t-1}, \mathcal{L}$. The second statement follows from the fact that after sub-iteration K of iteration t, j will always appear in F in Pseudocode 3, and never appear in G in Pseudocode 4. $\square$

Proof. Proof of Theorem 3.3.10. Consider some iteration $t \in [T]$. In Step 7 of SellTo with $s = t$, an extra sub-iteration is possible only for each item that was added to $\mathcal{L}^t$ by Step 9 of ConstructList. However, items can be added to protection lists at most once, by Proposition 3.3.8. Therefore, the total number of sub-iterations over $t = 1, \ldots, T$ is at most $T + n$.

As a result, SampleAddon is called is at most $T + n$ times, and the total number of probabilities h_j estimated is at most $n(T+n)$. Applying Lemma 3.3.9 with $\varepsilon_2 = \frac{\varepsilon}{T+n}$, each of the estimates $\hat{h}_j$ is outside of $[h_j - \varepsilon_2, h_j + \varepsilon_2]$ with probability at most $2e^{-2\varepsilon_2^2 M} = \frac{\varepsilon}{n(T+n)}$. By the union bound, the probability that *any* estimate is not within ε_2 of its true value is at most ε.

Therefore, with probability at least $1 - \varepsilon$, there are no failures in the sampling and the requirements on $\hat{h}_j$ in Proposition 3.3.6 are always met. For all $j \in [n]$, there is no error in the algorithm's value for d_j^{curr}, i.e. $\varepsilon_1 = 0$, at the start. Inductively applying Proposition 3.3.6, the total error accumulated over at most $T + n$ sub-iterations is at most $(T + n)\varepsilon_2 = \varepsilon$. That is, $|\hat{d}_j - \bar{d}_j| \leq \varepsilon$. We can also inductively apply Proposition 3.3.7 to see that $\hat{d}_j \leq \frac{b_j}{2} - \varepsilon$.

Combining the two preceding inequalities establishes that with probability at least $1-\varepsilon$, $\hat{d}_j - \varepsilon \leq \bar{d}_j \leq \frac{b_j}{2}$ for all $j \in [n]$, as desired. Now we analyze the runtime.

The runtime of SellTo with time horizon t is $O((t+n)n)$ (in Step 11, $\mathcal{F}_k^s$ can easily be computed in $O(n)$ time by having an index from each item to its position in the protection lists). Therefore, the total runtime of $\mathsf{SampleAddon}$ over the up to $T+n$ times it is called is

$$O\big((T+n)M(T+n)n)\big) = O\Big(\frac{(T+n)^4 n}{\varepsilon^2} \cdot \ln \frac{(T+n)n}{\varepsilon}\Big), \tag{B.3}$$

which is polynomial in T, n, and $\frac{1}{\varepsilon}$. It is easy to see that this is the bottleneck operation in $\mathsf{OnlineSamplingPersonalizedAssortment}(\varepsilon)$, completing the proof of Theorem 3.3.10. □

Appendix C

Appendix to Chapter 4

C.1 Full Proof of Theorem 4.2.2

Definition C.1.1. Define the following:

- $S_{i,t}$: the indicator random variable for whether inventory unit i is sold by the end of time t, i.e. the value of $\texttt{sold}[i]$ at the end of time t, defined for all $i \in [k]$ and $t = 0, \ldots, T$;
- i_t: the inventory unit assigned to customer t, taking a value in $[k]$ for all $t \in [T]$;
- $\ell_{i,t}$: the value such that $\texttt{level}[i] = r^{(\ell_{i,t})}$ at the end of time t, taking a value in $\{0, 1, \ldots, m\}$ for all $i \in [k]$ and $t = 0, \ldots, T$;
- j_t: the value in $\{0, 1, \ldots, m\}$ such that $V_t = r^{(j_t)}$, defined for all $t \in [T]$.

Fix the deterministic sequence of valuations $V_1, \ldots, V_T$ chosen by the adversary. i_t, $\ell_{i,t}$, and j_t are not random variables; they are determined by $V_1, \ldots, V_T$.

We would like to write the random variables $S_{i,t}$ in terms of the other random variables. By definition, $S_{i,0} = 0$ for all $i \in [k]$. For $t > 0$, the following equations hold:

$$S_{i_t,t} = S_{i_t,t-1} + X_t; \tag{C.1}$$

$$S_{i,t} = S_{i,t-1}, \text{ for } i \neq i_t. \tag{C.2}$$

(C.1)–(C.2) are easy to see. In the algorithm, the only inventory unit that could potentially be sold during time t is i_t. This explains why (C.2) holds for all $i \neq i_t$. It also explains why $S_{i_t,t} = 1$ if and only if $S_{i_t,t-1} = 1$ or $X_t = 1$. Furthermore, $S_{i_t,t-1}$ and X_t cannot both be 1, since the algorithm does not try to sell inventory unit i_t again at time t if it has already been sold. This completes the explanation for (C.1).

We now analyze the state of the `sold` array during the execution of the algorithm.

Lemma C.1.2. *At the end of each time step t, the probability that any inventory unit i has been sold is $\frac{1}{q}\sum_{j=1}^{\ell_{i,t}} q^{(j)}$. Formally, for all $t = 0, \dots, T$,*

$$\mathbb{E}[S_{i,t}] = \frac{1}{q}\sum_{j=1}^{\ell_{i,t}} q^{(j)}, \ \textit{for } i \in [k]. \tag{C.3}$$

Proof. Proof. We proceed by induction on t. (C.3) is true at time $t = 0$, where $\mathbb{E}[S_{i,0}] = 0$ and $\ell_{i,0} = 0$ for all $i \in [k]$.

Now suppose we are at the end of some time $t > 0$ and (C.3) was true at the end of time $t-1$. We need to prove that (C.3) is still true at the end of time t. For $i \neq i_t$, $S_{i,t} = S_{i,t-1}$, by (C.2). The value of `level`$[i]$ is unchanged by the algorithm during time t, so $\ell_{i,t} = \ell_{i,t-1}$ as well. The inductive hypothesis from time $t-1$ then establishes that $\mathbb{E}[S_{i,t}] = \frac{1}{q}\sum_{j=1}^{\ell_{i,t}} q^{(j)}$.

It remains prove $\mathbb{E}[S_{i_t,t}] = \frac{1}{q}\sum_{j=1}^{\ell_{i_t,t}} q^{(j)}$. This is immediate if j_t is no greater than $\ell_{i_t,t-1}$ (the value of the ℓ variable during iteration t of the algorithm), since both $S_{i,t}$ and $\ell_{i,t}$ would be unchanged. If $j_t > \ell_{i_t,t-1}$, the following can be derived (let $\ell = \ell_{i_t,t-1}$ for brevity):

$$\begin{aligned}
\mathbb{E}[S_{i_t,t}] &= \mathbb{E}[S_{i_t,t-1}] + \mathbb{E}[X_t] \\
&= \mathbb{E}[S_{i_t,t-1}] + \mathbb{E}[X_t | S_{i_t,t-1} = 0] \cdot \Pr[S_{i_t,t-1} = 0] \\
&= \frac{1}{q}\sum_{j=1}^{\ell} q^{(j)} + \Pr[X_t = 1 | S_{i_t,t-1} = 0]\Big(1 - \frac{1}{q}\sum_{j=1}^{\ell} q^{(j)}\Big) \\
&= \frac{1}{q}\sum_{j=1}^{\ell} q^{(j)} + \Big(\sum_{j=\ell+1}^{j_t} \frac{q^{(j)}}{\sum_{j'=\ell+1}^{m} q^{(j')}}\Big)\Big(\frac{\sum_{j=\ell+1}^{m} q^{(j)}}{q}\Big) \\
&= \frac{1}{q}\sum_{j=1}^{j_t} q^{(j)}.
\end{aligned}$$

The first equality follows from (C.1) and the linearity of expectation. The second equality conditions on $S_{i_t,t-1}$ being 0, since the value of X_t is 0 if $S_{i_t,t-1} = 1$. The third equality uses the value of $\mathbb{E}[S_{i_t,t-1}]$ guaranteed by the inductive hypothesis. In the fourth equality, the probability of getting a sale, conditioned on Algorithm 7 reaching line 7, is equal to the probability of choosing a price at most $r^{(j_t)}$, the valuation of customer t. The final equality achieves the desired result because $j_t = \ell_{i_t,t}$, the new value for `level[i]` after line 13 of iteration t of the algorithm.

This completes the induction and the proof of the lemma. $\square$

Now we analyze the expected revenue of the algorithm, which is $\mathbb{E}[\mathsf{ALG}]$, or $\sum_{t=1}^{T}\mathbb{E}[P_t X_t]$. As argued earlier, there cannot be a sale in a time step t where $j_t \leq \ell_{i_t,t-1}$, so for these time steps $X_t = 0$ and $\mathbb{E}[P_t X_t] = 0$. The following lemma derives the value of $\mathbb{E}[P_t X_t]$ when $j_t > \ell_{i_t,t-1}$.

Lemma C.1.3. *Suppose $j_t > \ell_{i_t,t-1}$ in a time step $t \in [T]$. Then the expected revenue earned by the algorithm during time step t is $\frac{1}{q}(r^{(j_t)} - r^{(\ell_{i_t,t-1})})$.*

Proof. Proof. Let $t \in [T]$ be any time step for which $j_t > \ell_{i_t,t-1}$. For brevity, let ℓ denote $\ell_{i_t,t-1}$. The following can be derived:

$$
\begin{aligned}
\mathbb{E}[P_t X_t] &= \mathbb{E}[P_t X_t | S_{i_t,t-1} = 0] \Pr[S_{i_t,t-1} = 0] \\
&= \Big(\sum_{j=\ell+1}^{m} r^{(j)} \mathbb{E}[X_t | P_t = r^{(j)}] \Pr[P_t = r^{(j)} | S_{i_t,t-1} = 0]\Big)\Big(1 - \Pr[S_{i_t,t-1} = 1]\Big) \\
&= \Big(\sum_{j=\ell+1}^{m} r^{(j)} \mathbb{1}[j_t \geq j] \Pr[P_t = r^{(j)} | S_{i_t,t-1} = 0]\Big)\Big(1 - \frac{1}{q}\sum_{j'=1}^{\ell} q^{(j')}\Big) \\
&= \Big(\sum_{j=\ell+1}^{j_t} r^{(j)} \frac{q^{(j)}}{\sum_{j'=\ell+1}^{m} q^{(j')}}\Big)\Big(\frac{\sum_{j'=\ell+1}^{m} q^{(j')}}{q}\Big) \\
&= \frac{1}{q}\sum_{j=\ell+1}^{j_t} r^{(j)}\Big(1 - \frac{r^{(j-1)}}{r^{(j)}}\Big)
\end{aligned}
$$

The first equality conditions on $S_{i_t,t-1}$ being 0; note that $X_t = 0$ if $S_{i_t,t-1} = 1$. The second equality conditions on the value of P_t, where we drop the conditioning on $S_{i_t,t-1}$ in the term $\mathbb{E}[X_t | P_t = r^{(j)}]$ since $P_t \neq \infty$ already implies $S_{i_t,t-1} = 0$. This term becomes $\mathbb{1}[j_t \geq j]$ in the third equality, since it is deterministically 1 or 0 depending on whether $V_t \geq r^{(j)}$, or equivalently $j_t \geq j$. The third equality also uses Lemma C.1.2, for the value of $\Pr[S_{i_t,t-1} = 1]$. The fourth equality uses the offering probabilities from line 7 of Algorithm 7. The fifth equality uses the explicit definition of $q^{(j)}$ from Definition 4.2.1, and it is easy to see that the final expression is equal to $\frac{1}{q}(r^{(j_t)} - r^{(\ell)})$. $\square$

Lemma C.1.3 in turn implies the following lemma.

Lemma C.1.4. *The expected revenue earned by the algorithm up to time t, $\sum_{t'=1}^{t} \mathbb{E}[P_{t'}X_{t'}]$, is $\frac{1}{q}\sum_{i=1}^{k} r^{(\ell_{i,t})}$.*

Proof. Proof. The customers up to time t can be partitioned according to the inventory unit they were assigned, so

$$\sum_{t'=1}^{t} \mathbb{E}[P_{t'}X_{t'}] = \sum_{i=1}^{k} \sum_{t' \le t: i_{t'}=i} \mathbb{E}[P_{t'}X_{t'}]. \tag{C.4}$$

Consider any i. For each t' assigned to i, $\mathbb{E}[P_{t'}X_{t'}]$ is 0 if $j_{t'} \le \ell_{i,t'-1}$. Denote the remaining t' such that $j_{t'} > \ell_{i,t'-1}$ by $t'_1, \dots, t'_N$, where $N \ge 0$ and $t'_1 < \dots < t'_N$. Using Lemma C.1.3,

$$\sum_{t' \le t: i_{t'}=i} \mathbb{E}[P_{t'}X_{t'}] \quad = \quad \sum_{n=1}^{N} \frac{1}{q}(r^{(j_{t'_n})} - r^{(\ell_{i,t'_n-1})}).$$

Before time t'_n, `level`$[i]$ was last updated at time t'_{n-1}, so $\ell_{i,t'_n-1} = j_{t'_{n-1}}$. Therefore, the sum telescopes and the remaining term is $\frac{1}{q}r^{(j_{t'_N})}$ (note that $r^{(\ell_{i,t'_1-1})} = r^{(0)} = 0$). Now, $j_{t'_N} = \ell_{i,t'_N}$, and `level`$[i]$ is not updated again in time steps $t'_N+1, \dots, t$, so $\ell_{i,t'_N} = \ell_{i,t}$. Substituting $\sum_{t' \le t: i_{t'}=i} \mathbb{E}[P_{t'}X_{t'}] = \frac{1}{q}r^{(\ell_{i,t})}$ into (C.4) completes the proof. $\square$

Having established the revenue of our online algorithm, we compare it to the offline optimum. Knowing the sequence of valuations $V_1, \dots, V_T$ in advance, it is clear that the following algorithm is optimal:

1. Find the $\min\{k, T\}$ customers with the largest valuations;

2. Charge each of these customers t her maximum willingness-to-pay V_t;

3. Reject all other customers.

The revenue OPT would be the the sum of the $\min\{k, T\}$ largest valuations.

Definition C.1.5. For all $t \in [T]$, let $M^k(t)$ be a vector consisting of the k largest elements from $(V_1, \dots, V_t)$, in any order. If $t < k$, fill in the remaining entries of $M^k(t)$ with zeros.

Then $\mathsf{OPT} = \sum_{i=1}^{k} M_i^k(T)$, where $M_i^k(T)$ denotes the i'th entry of $M^k(T)$. It turns out that $M^k(t)$ is closely tracked by the `level` array from Algorithm 7, as t progresses from

1 to T. Both $\ell_{i,t}$ (the value of `level[i]` at the end of time t) and $M^k(t)$ are deterministic functions of $V_1, \dots, V_t$.

Lemma C.1.6. *For all $t = 0, \dots, T$, the entries of the vector $(r^{(\ell_{1,t})}, \dots, r^{(\ell_{k,t})})$ is a permutation of the entries of the vector $M^k(t)$.*

Proof. Proof. We proceed by induction on t. At time $t = 0$, both $M^k(0)$ and $(r^{(\ell_{1,0})}, \dots, r^{(\ell_{k,0})})$ is a vector of k zeros, so the statement is true.

Now consider $t > 0$, and suppose that $M^k(t-1)$ is a permutation of $(r^{(\ell_{1,t-1})}, \dots, r^{(\ell_{k,t-1})})$. Therefore, a minimum entry in $M^k(t-1)$ is equal to a minimum entry in $(r^{(\ell_{1,t-1})}, \dots, r^{(\ell_{k,t-1})})$, which in turn is equal to $r^{(\ell_{i_t,t-1})}$, by Definition C.1.1.

If $j_t > \ell_{i_t,t-1}$, or equivalently $V_t = r^{(j_t)} > r^{(\ell_{i_t,t-1})}$, then by the definition of $M^k(t)$, V_t must be added to $M^k(t-1)$ and replace any minimum entry equal to $r^{(\ell_{i_t,t-1})}$. Meanwhile, $\ell_{i_t,t} = j_t$, and $\ell_{i,t} = \ell_{i,t-1}$ for all $i \neq i_t$, thus the only change from $(r^{(\ell_{1,t-1})}, \dots, r^{(\ell_{k,t-1})})$ to $(r^{(\ell_{1,t})}, \dots, r^{(\ell_{k,t})})$ is that the entry at index i_t has been replaced by $r^{(j_t)}$. Since $M^k(t-1)$ and $(r^{(\ell_{1,t-1})}, \dots, r^{(\ell_{k,t-1})})$ go through the same change at time t, $M^k(t)$ is still a permutation of $(r^{(\ell_{1,t})}, \dots, r^{(\ell_{k,t})})$.

If instead $j_t \leq \ell_{i_t,t-1}$, then every entry of $M^k(t-1)$ is already at least $r^{(j_t)} = V_t$, so $M^k(t-1)$ incurs no change at time t. Similarly, $\ell_{i,t} = \ell_{i,t-1}$ for all $i \in [k]$, so $(r^{(\ell_{1,t-1})}, \dots, r^{(\ell_{k,t-1})})$ incurs no change as well. In both cases, we have established that $M^k(t)$ is a permutation of $(r^{(\ell_{1,t})}, \dots, r^{(\ell_{k,t})})$, completing the induction and the proof. □

With Lemma C.1.4 and Lemma C.1.6, it is easy to establish the competitiveness of Algorithm 7. Our online algorithm is designed so that during each time step t, it earns exactly $\frac{1}{q}$ of the amount that the offline optimum would increase by with the addition of V_t, and it does not need to observe V_t beforehand to accomplish this.

Proof. Proof of Theorem 4.2.2. Fix any sequence of valuations $(V_1, \dots, V_T)$. $\mathbb{E}[\mathsf{ALG}]$ is equal to $\sum_{t=1}^{T} \mathbb{E}[P_t X_t]$, which in turn is equal to $\frac{1}{q} \sum_{i=1}^{k} r^{(\ell_{i,T})}$, by Lemma C.1.4. Meanwhile, $\mathsf{OPT} = \sum_{i=1}^{k} M_i^k(T)$, and the entries of $M^k(T)$ is a permutation of the entries of $(r^{(\ell_{1,T})}, \dots, r^{(\ell_{k,T})})$, by Lemma C.1.6. Therefore, $\mathsf{OPT} = \sum_{i=1}^{k} r^{(\ell_{i,T})} = q \cdot \mathbb{E}[\mathsf{ALG}]$, completing the proof of Theorem 4.2.2. □

C.2 Proofs from Sections 4.2.3–4.2.4

Proof. Proof of Lemma 4.2.5. Let $i = i_t^*$, for brevity. Let $S_{i,t}$ be the indicator random variable for inventory unit i being sold by the end of time t. For all $k' \in \{0, \ldots, k\}$,

$$\begin{aligned}
&\Pr[S_{i,t-1} = 0 | I_{t-1} = k'] \\
&= \frac{\Pr[I_{t-1} = k' | S_{i,t-1} = 0] \Pr[S_{i,t-1} = 0]}{\Pr[I_{t-1} = k' | S_{i,t-1} = 0] \Pr[S_{i,t-1} = 0] + \Pr[I_{t-1} = k' | S_{i,t-1} = 1] \Pr[S_{i,t-1} = 1]}
\end{aligned} \tag{C.5}$$

by Bayes' law. For all $t \in [T]$, $i \in [k]$, and $k' \in \{0, \ldots, k\}$, we explain how to compute $\Pr[I_{t-1} = k' | S_{i,t-1} = 0]$ in polynomial time; $\Pr[I_{t-1} = k' | S_{i,t-1} = 1]$ can be computed analogously.

First we argue that the Bernoulli random variables $\{S_{i',t-1} : i' \in [k]\}$ are independent. To see this, note that the assignment procedure in Algorithm 7 is deterministic. Therefore, each $S_{i',t-1}$ is only dependent on the prices chosen for the customers assigned to i', and while these prices could be dependent on each other, they are independent from the prices chosen for customers not assigned to i'.

Furthermore, $I_{t-1} = k - \sum_{i'=1}^{k} S_{i',t-1}$. By independence, $\Pr[I_{t-1} = k' | S_{i,t-1} = 0] = \Pr[\sum_{i' \neq i} S_{i',t-1} = k - k']$. $\sum_{i' \neq i} S_{i',t-1}$ is simply the sum of $k-1$ independent Bernoulli random variables with known mean (from Lemma C.1.2), hence the probability that it equals a specific value can be computed using dynamic programming.

We elaborate on the dynamic programming. For notational convenience, without loss of generality assume $i = k$. We will inductively for $a = 0, \ldots, k-1$ maintain the value of $\Pr[\sum_{i'=1}^{a} S_{i',t-1} = b]$ for all $b \in \{0, \ldots, k\}$. It is easy to initialize this for $a = 0$. Given $\Pr[\sum_{i'=1}^{a} S_{i',t-1} = b]$ for all $b \in \{0, \ldots, k\}$, note that

$$\begin{aligned}
&\Pr[\sum_{i'=1}^{a+1} S_{i',t-1} = b] \\
&= \Pr[\sum_{i'=1}^{a} S_{i',t-1} = b-1] \Pr[S_{a+1,t-1} = 1] + \Pr[\sum_{i'=1}^{a} S_{i',t-1} = b] \Pr[S_{a+1,t-1} = 0]
\end{aligned}$$

for all $b \in \{0, \ldots, k\}$. Each iteration of a can be computed in time linear in k, and there are less than k iterations.

$\Pr[I_{t-1} = k' | S_{i,t-1} = 1]$ can be computed analogously. It is clear that both procedures can be done in time $O(k^2)$ (ignoring the $O(t)$ time it may take to compute the assignment procedure), completing the proof of Lemma 4.2.5. □

Proof. Proof of Lemma 4.2.7 and Theorem 4.2.8. We argue that Lemma 4.2.7 and Theorem 4.2.8 are the special cases of Lemma 4.3.2 and Theorem 4.3.3 from the stochastic-valuation model. It is easy to check that the statements are analogous, so it suffices to show that Exp (as defined in Section 4.3.2) executed on deterministic valuations is identical to Algorithm 7' (as defined in Section 4.2.3).

We show that the decision rule for a single time period t, and any amount of inventory remaining k', is the same. Let $i = i_t^*$ and $\ell = \ell_t$, for brevity. Consider the values of i and ℓ during iteration t of Algorithm 7 (with the deterministic valuations $V_1, \ldots, V_T$). First consider any $j = \ell + 1, \ldots, m$.

$$\begin{aligned}\Pr[P_t^{\mathsf{A1}'} = r^{(j)} | I_{t-1}^{\mathsf{A1}'} = k'] &= \Pr[S_{i,t-1} = 0 | I_{t-1}^{\mathsf{A1}} = k'] \cdot \frac{q^{(j)}}{\sum_{j'=\ell+1}^{m} q^{(j')}} \\ &= \Pr[S_{i,t-1} = 0 | I_{t-1}^{\mathsf{A1}} = k'] \Pr[P_t^{\mathsf{A1}} = r^{(j)} | S_{i,t-1} = 0, I_{t-1}^{\mathsf{A1}} = k'] \\ &= \Pr[P_t^{\mathsf{A1}} = r^{(j)} \cap S_{i,t-1} = 0 | I_{t-1}^{\mathsf{A1}} = k'] \\ &= \Pr[P_t^{\mathsf{A1}} = r^{(j)} | I_{t-1}^{\mathsf{A1}} = k']\end{aligned}$$

The first equality holds by the specification of algorithm Algorithm 7'. The second equality holds by the specification of Algorithm 7, where we can add the conditioning on $I_{t-1}^{\mathsf{A1}} = k'$ in the second probability due to independence. The final equality follows because $P_t^{\mathsf{A1}} = r^{(j)} \neq \infty$ implies $S_{i,t-1} = 0$.

If $j = m + 1$, then

$$\begin{aligned}\Pr[P_t^{\mathsf{A1}'} = \infty | I_{t-1}^{\mathsf{A1}'} = k'] &= \Pr[S_{i,t-1} = 1 | I_{t-1}^{\mathsf{A1}} = k'] \\ &= \Pr[P_t^{\mathsf{A1}} = \infty | I_{t-1}^{\mathsf{A1}} = k']\end{aligned}$$

since the event $S_{i,t-1} = 1$ occurs if and only if the event $P_t^{\mathsf{A1}} = \infty$ occurs.

Finally, clearly if $j \leq \ell$, then both $\Pr[P_t^{\mathsf{A1}'} = r^{(j)} | I_{t-1}^{\mathsf{A1}'} = k']$ and $\Pr[P_t^{\mathsf{A1}} = r^{(j)} | I_{t-1}^{\mathsf{A1}} = k']$ are 0.

We have shown that $\Pr[P_t^{\mathsf{A1}'} = r^{(j)} | I_{t-1}^{\mathsf{A1}'} = k'] = \Pr[P_t^{\mathsf{A1}} = r^{(j)} | I_{t-1}^{\mathsf{A1}} = k']$ for all $j \in \{1, \ldots, m, m+1\}$, so it is the same decision rule as Exp, completing the proof. □

Proof. Proof of Lemma 4.2.10. Fix a valuation sequence $V_1, \ldots, V_T$ and consider any sample path in the execution of $\mathcal{A}$; let the sample path be depicted by the sequence of random prices $P_1^{\mathcal{A}}, \ldots, P_T^{\mathcal{A}}$. The revenue $\mathsf{ALG}^{\mathcal{A}}$ on that sample path is given by $\sum_{t:V_t \geq P_t^{\mathcal{A}}} P_t^{\mathcal{A}}$; note that the cardinality of the set $\{t : V_t \geq P_t^{\mathcal{A}}\}$ is at most k.

On that same sample path, the modified algorithm $\mathcal{A}'$ would sell to the k customers with the smallest indices in $\{t : V_t \geq \min\{P_t^{\mathcal{A}}, r^{(m)}\}\}$ (or all the customers in that set if its cardinality is less than k). Let $\mathcal{S}$ denote the set of customers served by the modified algorithm. Let $b = |\{t \in \mathcal{S} : P_t^{\mathcal{A}} = \infty\}|$, the number of customers with valuation $r^{(j)}$ served by the modified algorithm that were rejected by the original algorithm.

It is easy to see that

$$\begin{aligned} \mathsf{ALG}^{\mathcal{A}'} - \mathsf{ALG}^{\mathcal{A}} &= \sum_{t \in \mathcal{S}} \min\{P_t^{\mathcal{A}}, r^{(m)}\} - \sum_{t:V_t \geq P_t^{\mathcal{A}}} P_t^{\mathcal{A}} \\ &\geq br^{(m)} - \sum_{t \in \mathcal{S}'} P_t^{\mathcal{A}} \end{aligned} \tag{C.6}$$

where $\mathcal{S}'$ is the set of customers that are no longer served by $\mathcal{A}'$ because it used up b extra units of inventory. Since $|\mathcal{S}'| \leq b$, and $P_t^{\mathcal{A}} \leq r^{(m)}$ for all t such that $P_t^{\mathcal{A}} \leq V_t$, it is immediate that (C.6) is non-negative. Since this holds on every sample path for $\mathcal{A}$, we have completed the proof that $\mathbb{E}[\mathsf{ALG}^{\mathcal{A}'}] \geq \mathbb{E}[\mathsf{ALG}^{\mathcal{A}}]$. □

Proof. Proof of Theorem 4.2.11 The inventory level I_{t-1} is equal to $k - \sum_{i=1}^{k}(1 - S_{i,t-1})$, where $S_{i,t-1}$ is the indicator random variable for inventory unit i being sold by the end of time $t-1$. We will hereafter omit the subscript $t-1$.

Each term $(1 - S_i)$ is independent and equal to 1 with probability $(\sum_{j=b_i+1}^{m})q^{(j)}/q$, which is the probability that inventory unit i has not been sold. We will denote it using p_i and let $Y_i = 1 - S_i$, for brevity. As long as b_i (the index in $0, \ldots, m$ of the highest valuation assigned to inventory unit i) is not 0 or m, $p_i \in (0, 1)$. We will without loss of generality assume that $p_i \in (0, 1)$ for all i, redefining k and re-indexing as necessary (if $p_i = 0$ or $p_i = 1$ then Y_i is deterministic and we can remove it from analysis of the random sum). By the assumptions in the statement of the theorem, this re-indexing does not cause i_t^* to fall outside of $1, \ldots, k$; in fact we can without loss of generality assume $i_t^* = 1$. Furthermore, k_1, k_2 are at least 0 at at most the re-defined k, since they correspond to inventory levels that are realized with non-zero probability.

After all of these transformations, the statement reduces to

$$\Pr[Y_1 = 1 | \sum_{i=1}^{k} Y_i = k_1] < \Pr[Y_1 = 1 | \sum_{i=1}^{k} Y_i = k_2] \tag{C.7}$$

where each Y_i is an independent Bernoulli random variable of probability $p_i \in (0, 1)$ and $0 \leq k_1 < k_2 \leq k$. Furthermore, we can without loss of generality assume that $k_2 = k_1 + 1$.

If $k_1 = 0$, then (C.7) is clearly true, since the LHS is 0 while the RHS is non-zero. So assume that $k_1 > 0$ and we can rewrite (C.7) as follows:

$$\frac{\Pr[Y_1 = 1 \cap \sum_{i=1}^{k} Y_i = k_1]}{\Pr[\sum_{i=1}^{k} Y_i = k_1]} < \frac{\Pr[Y_1 = 1 \cap \sum_{i=1}^{k} Y_i = k_1 + 1]}{\Pr[\sum_{i=1}^{k} Y_i = k_1 + 1]}$$

$$\frac{p_1 \Pr[\sum_{i=2}^{k} Y_i = k_1 - 1]}{p_1 \Pr[\sum_{i=2}^{k} Y_i = k_1 - 1] + (1 - p_1) \Pr[\sum_{i=2}^{k} Y_i = k_1]} < \frac{p_1 \Pr[\sum_{i=2}^{k} Y_i = k_1]}{p_1 \Pr[\sum_{i=2}^{k} Y_i = k_1] + (1 - p_1) \Pr[\sum_{i=2}^{k} Y_i = k_1 + 1]}$$

$$\left(1 + \frac{(1 - p_1)}{p_1} \cdot \frac{\Pr[\sum_{i=2}^{k} Y_i = k_1]}{\Pr[\sum_{i=2}^{k} Y_i = k_1 - 1]}\right)^{-1} < \left(1 + \frac{(1 - p_1)}{p_1} \cdot \frac{\Pr[\sum_{i=2}^{k} Y_i = k_1 + 1]}{\Pr[\sum_{i=2}^{k} Y_i = k_1]}\right)^{-1}$$

Therefore, it suffices to prove that:

$$\frac{\Pr[\sum_{i=2}^{k} Y_i = k_1]}{\Pr[\sum_{i=2}^{k} Y_i = k_1 - 1]} > \frac{\Pr[\sum_{i=2}^{k} Y_i = k_1 + 1]}{\Pr[\sum_{i=2}^{k} Y_i = k_1]}$$

$$\Pr[\sum_{i=2}^{k} Y_i = k_1]^2 > \Pr[\sum_{i=2}^{k} Y_i = k_1 + 1] \Pr[\sum_{i=2}^{k} Y_i = k_1 - 1]$$

$$\left(\sum_{S \subseteq \{2,\ldots,k\}: |S| = k_1} \prod_{i \in S} p_i \prod_{i \notin S} (1 - p_i)\right)^2 > \left(\sum_{S: |S| = k_1 + 1} \prod_{i \in S} p_i \prod_{i \notin S} (1 - p_i)\right) \left(\sum_{S: |S| = k_1 - 1} \prod_{i \in S} p_i \prod_{i \notin S} (1 - p_i)\right) \tag{C.8}$$

After expanding, both sides are a sum of terms of the form

$$\prod_{i=2}^{k} p_i^{a_i} (1 - p_i)^{2 - a_i} \tag{C.9}$$

where each a_i is 0, 1, or 2 and the sum $\sum_{i=2}^{k} a_i$ equals $2k_1$, the total number of times that a "positive" term p_i (as opposed to a "negative" term $(1 - p_i)$) appears in the product. Let b denote the total number of $i = 2, \ldots, k$ such that $a_i = 1$, which must be even.

Now, observe that the total number of times the term (C.9) appears in the LHS of the expansion of (C.8) is $\binom{b}{b/2}$ (because we choose $b/2$ of the b indices that are "positive" to come from the first bracket; the remaining $b/2$ must come from the second bracket) while the total number of times this term appears in the RHS is $\binom{b}{b/2+1}$ (because we choose $b/2 + 1$ of the b indices that are "positive" to come from the first bracket), with the latter being strictly less. Furthermore, none of these terms are 0, since all of the values of p_i lie strictly between 0 and 1. Therefore, the inequality is strict, completing the proof of the theorem. $\square$

C.3 Proofs from Section 4.3

Proof. Proof of Lemma 4.3.2. We proceed by induction on t. (4.7) is true for $t = 0$, since $\Pr[I_0^{\mathsf{Exp}} = k] = \Pr[I_0^{\mathsf{A1}} = k] = 1$.

Now suppose $t > 0$ and that (4.7) has been established for time $t-1$. Then for every k' such that $\Pr[I_{t-1}^{\mathsf{Exp}} = k'] > 0$, (4.6) holds by definition. Indeed, since $\Pr[I_{t-1}^{\mathsf{A1}} = k'] = \Pr[I_{t-1}^{\mathsf{Exp}} = k']$ by the inductive hypothesis, $\Pr[I_{t-1}^{\mathsf{A1}} = k'] > 0$ for such k'.

We now consider (4.7) for time t. Note that $I_t^{\mathsf{Exp}} = I_{t-1}^{\mathsf{Exp}} - \mathbb{1}(V_t \geq P_t^{\mathsf{Exp}})$. Therefore,

$$\Pr[I_t^{\mathsf{Exp}} = k'] = \Pr[I_{t-1}^{\mathsf{Exp}} = k' + 1 \cap V_t \geq P_t^{\mathsf{Exp}}] + \Pr[I_{t-1}^{\mathsf{Exp}} = k' \cap V_t < P_t^{\mathsf{Exp}}] \tag{C.10}$$

for $k' \in \{0, \dots, k-1\}$, while

$$\Pr[I_t^{\mathsf{Exp}} = k] = \Pr[I_{t-1}^{\mathsf{Exp}} = k \cap V_t < P_t^{\mathsf{Exp}}]. \tag{C.11}$$

Now, for any $k' \in \{0, \dots, k\}$, if $\Pr[I_{t-1}^{\mathsf{Exp}} = k'] > 0$, then the following can be derived:

$$\begin{aligned}
& \Pr[V_t \geq P_t^{\mathsf{Exp}} | I_{t-1}^{\mathsf{Exp}} = k'] \\
= \; & \sum_{j=1}^{m+1} \Pr[V_t \geq P_t^{\mathsf{Exp}} | P_t^{\mathsf{Exp}} = r^{(j)}, I_{t-1}^{\mathsf{Exp}} = k'] \Pr[P_t^{\mathsf{Exp}} = r^{(j)} | I_{t-1}^{\mathsf{Exp}} = k'] \\
= \; & \sum_{j=1}^{m+1} \Pr[V_t \geq r^{(j)}] \Pr[P_t^{\mathsf{Exp}} = r^{(j)} | I_{t-1}^{\mathsf{Exp}} = k'] \\
= \; & \sum_{j=1}^{m+1} \Pr[V_t \geq r^{(j)}] \Pr[P_t^{\mathsf{A1}} = r^{(j)} | I_{t-1}^{\mathsf{A1}} = k'] \\
= \; & \Pr[V_t \geq P_t^{\mathsf{A1}} | I_{t-1}^{\mathsf{A1}} = k'].
\end{aligned} \tag{C.12}$$

In the second equality, we remove the conditioning on $I_{t-1}^{\mathsf{Exp}} = k'$, since the valuation V_t is an independent random variable unaffected by any history. The third equality follows because we have already established (4.6) for time t. The final equality also requires independence.

By the inductive hypothesis that (4.7) holds for time $t-1$, $\Pr[I_{t-1}^{\mathsf{Exp}} = k'] = \Pr[I_{t-1}^{\mathsf{A1}} = k']$.

If $\Pr[I_{t-1}^{\mathsf{Exp}} = k']$ is also non-zero, then the following can be derived using (C.12):

$$\begin{aligned} \Pr[V_t \geq P_t^{\mathsf{Exp}} | I_{t-1}^{\mathsf{Exp}} = k'] \Pr[I_{t-1}^{\mathsf{Exp}} = k'] &= \Pr[V_t \geq P_t^{\mathsf{A1}} | I_{t-1}^{\mathsf{A1}} = k'] \Pr[I_{t-1}^{\mathsf{A1}} = k'] \\ \Pr[V_t \geq P_t^{\mathsf{Exp}} \cap I_{t-1}^{\mathsf{Exp}} = k'] &= \Pr[V_t \geq P_t^{\mathsf{A1}} \cap I_{t-1}^{\mathsf{A1}} = k']. \end{aligned} \tag{C.13}$$

If instead $\Pr[I_{t-1}^{\mathsf{Exp}} = k'] = \Pr[I_{t-1}^{\mathsf{A1}} = k'] = 0$, then $\Pr[V_t \geq P_t^{\mathsf{Exp}} \cap I_{t-1}^{\mathsf{Exp}} = k'] \leq \Pr[I_{t-1}^{\mathsf{Exp}} = k'] = 0$. Similarly, $\Pr[V_t \geq P_t^{\mathsf{A1}} \cap I_{t-1}^{\mathsf{A1}} = k'] = 0$, and therefore, (C.13) still holds.

We can analogously to (C.12) and (C.13) derive for all $k' \in \{0, \ldots, k\}$ that

$$\Pr[V_t < P_t^{\mathsf{Exp}} \cap I_{t-1}^{\mathsf{Exp}} = k'] = \Pr[V_t < P_t^{\mathsf{A1}} \cap I_{t-1}^{\mathsf{A1}} = k']. \tag{C.14}$$

We can substitute (C.13) and (C.14) into (C.10) to see that

$$\begin{aligned} \Pr[I_t^{\mathsf{Exp}} = k'] &= \Pr[I_{t-1}^{\mathsf{A1}} = k' + 1 \cap V_t \geq P_t^{\mathsf{A1}}] + \Pr[I_{t-1}^{\mathsf{A1}} = k' \cap V_t < P_t^{\mathsf{A1}}] \\ &= \Pr[\sum_{t'=1}^{t-1} X_{t'}^{\mathsf{A1}} = k - k' - 1 \cap X_t^{\mathsf{A1}} = 1] + \Pr[\sum_{t'=1}^{t-1} X_{t'}^{\mathsf{A1}} = k - k' \cap X_t^{\mathsf{A1}} = 0] \\ &= \Pr[I_t^{\mathsf{A1}} = k'] \end{aligned}$$

for all $k' \in \{0, \ldots, k-1\}$. We can similarly substitute (C.14) into (C.11) to see that $\Pr[I_t^{\mathsf{Exp}} = k] = \Pr[I_t^{\mathsf{A1}} = k]$. This completes the induction and the proof of Lemma 4.3.2. $\square$

Proof. Proof of Theorem 4.3.3. The following is straight-forward to derive:

$$\begin{aligned} \mathbb{E}[\mathsf{ALG}^{\mathsf{Exp}}] &= \sum_{t=1}^{T} \mathbb{E}[P_t^{\mathsf{Exp}} \cdot \mathbb{1}(V_t \geq P_t^{\mathsf{Exp}})] \\ &= \sum_{t=1}^{T} \sum_{j=1}^{m+1} r^{(j)} \Pr[V_t \geq r^{(j)}] \Pr[P_t^{\mathsf{Exp}} = r^{(j)}] \\ &= \sum_{t=1}^{T} \sum_{j=1}^{m+1} r^{(j)} \Pr[V_t \geq r^{(j)}] \sum_{k'=0}^{k} \Pr[P_t^{\mathsf{Exp}} = r^{(j)} | I_{t-1}^{\mathsf{Exp}} = k'] \Pr[I_{t-1}^{\mathsf{Exp}} = k'] \\ &= \sum_{t=1}^{T} \sum_{j=1}^{m+1} r^{(j)} \Pr[V_t \geq r^{(j)}] \sum_{k'=0}^{k} \Pr[P_T^{\mathsf{A1}} = r^{(j)} | I_{t-1}^{\mathsf{A1}} = k'] \Pr[I_{t-1}^{\mathsf{A1}} = k'] \\ &= \sum_{t=1}^{T} \sum_{j=1}^{m+1} r^{(j)} \Pr[V_t \geq r^{(j)}] \Pr[P_t^{\mathsf{A1}} = r^{(j)}] \\ &= \sum_{t=1}^{T} \mathbb{E}[P_t^{\mathsf{A1}} \cdot \mathbb{1}(V_t \geq P_t^{\mathsf{A1}})]. \end{aligned}$$

The second and sixth equalities use the independence of V_t, while the fourth equality uses both statements of Lemma 4.3.2. The final expression is equal to $\mathbb{E}[\mathsf{ALG}^{\mathsf{A1}}]$, completing the proof of Theorem 4.3.3. $\square$

Proof. Proof of Lemma 4.3.7. The first statement is easy to see. Since every sample path fails at time $T+1$ by definition, for any sample path $\mathbf{H}_T^{\mathsf{Samp}}$, it must have a unique *first* point of failure in $[T+1]$, say t'. $\mathbf{H}_T^{\mathsf{Samp}}$ then falls under exactly one of the events, namely the one with $t = t'$ and $\mathbf{h}_t = (0, P_1^{\mathsf{Samp}}, V_1^{\mathsf{Samp}}, \ldots, 0, P_{t-1}^{\mathsf{Samp}}, V_{t-1}^{\mathsf{Samp}}, 1)$. Therefore, the events are mutually exclusive and collectively exhaustive. The case for Exp is argued analogously.

The final statement is argued inductively. For all $t \in \{0, \ldots, T\}$, let $\mathbf{g}_t = (f_1, p_1, v_1, \ldots, f_t, p_t, v_t)$ be a vector of realizations to the end of time t, and let $\mathcal{G}_t$ denote the set of such vectors containing no failures. Let $\mathbf{G}_t^{\mathsf{Samp}} = (F_1^{\mathsf{Samp}}, P_1^{\mathsf{Samp}}, V_1^{\mathsf{Samp}}, \ldots, F_t^{\mathsf{Samp}}, P_t^{\mathsf{Samp}}, V_t^{\mathsf{Samp}})$, and
$\mathbf{G}_t^{\mathsf{Exp}} = (F_1^{\mathsf{Exp}}, P_1^{\mathsf{Exp}}, V_1^{\mathsf{Exp}}, \ldots, F_t^{\mathsf{Exp}}, P_t^{\mathsf{Exp}}, V_t^{\mathsf{Exp}})$.

We would like to inductively establish that $\Pr[\mathbf{G}_t^{\mathsf{Samp}} = \mathbf{g}_t] = \Pr[\mathbf{G}_t^{\mathsf{Exp}} = \mathbf{g}_t]$ for all $t \in \{0, \ldots, T\}$ and $\mathbf{g}_t \in \mathcal{G}_t$. This is clearly true for $t = 0$. For $t > 0$, take any $\mathbf{g}_t \in \mathcal{G}_t$, and we can write

$$\begin{aligned}
\Pr[\mathbf{G}_t^{\mathsf{Samp}} = \mathbf{g}_t] &= \Pr[\mathbf{G}_{t-1}^{\mathsf{Samp}} = \mathbf{g}_{t-1}] \cdot \Pr[F_t^{\mathsf{Samp}} = 0 | \mathbf{G}_{t-1}^{\mathsf{Samp}} = \mathbf{g}_{t=1}] \cdot \Pr[P_t^{\mathsf{Samp}} = p_t | \mathbf{G}_{t-1}^{\mathsf{Samp}} = \mathbf{g}_{t=1}, F_t^{\mathsf{Samp}} = 0] \\
&\quad \cdot \Pr[V_t^{\mathsf{Samp}} = v_t | \mathbf{G}_{t-1}^{\mathsf{Samp}} = \mathbf{g}_{t=1}, F_t^{\mathsf{Samp}} = 0, P_t^{\mathsf{Samp}} = p_t]; \\
\Pr[\mathbf{G}_t^{\mathsf{exp}} = \mathbf{g}_t] &= \Pr[\mathbf{G}_{t-1}^{\mathsf{exp}} = \mathbf{g}_{t-1}] \cdot \Pr[F_t^{\mathsf{Exp}} = 0 | \mathbf{G}_{t-1}^{\mathsf{exp}} = \mathbf{g}_{t=1}] \cdot \Pr[P_t^{\mathsf{Exp}} = p_t | \mathbf{G}_{t-1}^{\mathsf{exp}} = \mathbf{g}_{t=1}, F_t^{\mathsf{Exp}} = 0] \\
&\quad \cdot \Pr[V_t^{\mathsf{Exp}} = v_t | \mathbf{G}_{t-1}^{\mathsf{exp}} = \mathbf{g}_{t=1}, F_t^{\mathsf{Exp}} = 0, P_t^{\mathsf{Exp}} = p_t].
\end{aligned}$$

We will prove that $\Pr[\mathbf{G}_t^{\mathsf{Samp}} = \mathbf{g}_t] = \Pr[\mathbf{G}_t^{\mathsf{Exp}} = \mathbf{g}_t]$ by arguing that each term in the expression for $\Pr[\mathbf{G}_t^{\mathsf{Samp}} = \mathbf{g}_t]$ equals the corresponding term in the expression for $\Pr[\mathbf{G}_t^{\mathsf{Exp}} = \mathbf{g}_t]$. The first terms are equal because of the inductive hypothesis. The second terms are equal because both algorithms are sampling runs of Algorithm 7 and trying to hit a run with $I_{t-1}^{\mathsf{A1}} = k - \sum_{t'=1}^{t-1} \mathbb{1}(v_{t'} \geq p_{t'})$. The third terms are identical because because we have conditioned on $F_t^{\mathsf{Samp}} = 0$. The fourth terms are equal because V_t^{Samp} and V_t^{Exp} are IID and none of the conditioning has any effect.

Having established this, note that for every $t \in [T+1]$ and $\mathbf{h}_t \in \mathcal{F}_t$ there exists a unique $\mathbf{g}_{t-1} \in \mathcal{G}_{t-1}$ such that $\mathbf{g}_{t-1}$ is a prefix of $\mathbf{h}_t$. We know that for this $\mathbf{g}_{t-1}$, $\Pr[\mathbf{G}_{t-1}^{\mathsf{Samp}} =$

$\mathbf{g}_{t-1}] = \Pr[\mathbf{G}_{t-1}^{\mathsf{Exp}} = \mathbf{g}_{t-1}]$. Therefore, it suffices to prove that $\Pr[F_t^{\mathsf{Samp}} = 0 | \mathbf{G}_{t-1}^{\mathsf{Samp}} = \mathbf{g}_{t-1}] = \Pr[F_t^{\mathsf{Exp}} = 0 | \mathbf{G}_{t-1}^{\mathsf{Exp}} = \mathbf{g}_{t-1}]$. By the same argument as the previous paragraph, these two probabilities are equal. Therefore, $\Pr[\mathbf{H}^{\mathsf{Samp}} = \mathbf{h}_t] = \Pr[\mathbf{H}^{\mathsf{Exp}} = \mathbf{h}_t]$, completing the proof of Lemma 4.3.7. □

Proof. Proof of Theorem 4.3.10. Applying Lemma 4.3.9 to (4.15), we see that

$$\begin{aligned}
\mathbb{E}[\mathsf{ALG}^{\mathsf{Samp}}] &\geq \mathbb{E}[\mathsf{ALG}^{\mathsf{Exp}}] - \mathbb{E}[\mathsf{OPT}]\Big(\sum_{t=1}^{T} \frac{1}{eCt^2}\Big) \\
&\geq \mathbb{E}[\mathsf{ALG}^{\mathsf{Exp}}] - \mathbb{E}[\mathsf{OPT}]\frac{1}{e\lceil\frac{6}{e\pi^2\varepsilon}\rceil}\Big(\frac{\pi^2}{6}\Big) \\
&\geq \mathbb{E}[\mathsf{ALG}^{\mathsf{Exp}}] - \varepsilon \cdot \mathbb{E}[\mathsf{OPT}]
\end{aligned}$$

Furthermore, we know from Theorem 4.3.3 that $\mathbb{E}[\mathsf{ALG}^{\mathsf{Exp}}] = \mathbb{E}[\mathsf{ALG}^{\mathsf{A1}}] = \frac{1}{q}\mathbb{E}[\mathsf{OPT}]$. This establishes the competitiveness.

The statement about runtime also follows easily from the specification of Algorithm 8 since the number of sample runs during each time period t, $\lceil\frac{6}{e\pi^2\varepsilon}\rceil(k+1)t^2$, is polynomial in $\frac{1}{\varepsilon}$. □

C.4 A Continuum of Prices

In this section we show how to modify Algorithm 7 for the setting where valuations could take any value in $0 \cup [1, R]$. The competitive ratio obtained will be $\frac{1}{1+\ln R}$, recovering the competitive ratio from Ball and Queyranne (2009).

Consider Algorithm 9. Now `val`$[i]$ keeps track of the highest valuation assigned to inventory unit i thus far, starting at 0. It is easy to check that the price distributions specified in lines 8 and 10 are proper.

To analyze the competitiveness of Algorithm 9, we prove lemmas analogous to Lemmas C.1.2–C.1.3. We use the same notation as in Definition C.1.1, except instead of $\ell_{i,t}$ and j_t, we use $v_{i,t}$ to denote the value of `val`$[i]$ at the end of time t, taking a value in $0 \cup [1, R]$.

Lemma C.4.1. *At the end of each time step t, the probability that any inventory unit i has been sold is 0 if* $v_{i,t} = 0$*, and* $\frac{1+\ln v_{i,t}}{1+\ln R}$ *if* $v_{i,t} \geq 1$*. Formally, for all* $t = 0, \ldots, T$,

$$\mathbb{E}[S_{i,t}] = \mathbb{1}(v_{i,t} > 0) \cdot \frac{1 + \ln v_{i,t}}{1 + \ln R}, \textit{ for } i \in [k]. \tag{C.15}$$

Algorithm 9 Weakly Randomized Online Algorithm for Continuum of Prices

$\texttt{val}[i] = 0, \texttt{sold}[i] = \texttt{false}$ for $i = 1, \ldots, k$
$t = 1$
while customer t arrives **do**
    $v = \min_{i'}\{\texttt{val}[i']\}$
    $i = \min\{i' : \texttt{val}[i'] = v\}$
    if $\texttt{sold}[i] = \texttt{false}$ **then**
        if v=0 **then**
            offer price 1 w.p. $\frac{1}{1+\ln R}$, and price r w.p. $\frac{1}{r(1+\ln R)}$ for all $r \in (1, R]$
        else
            offer price r w.p. $\frac{1}{r(\ln R - \ln v)}$ for all $r \in (v, R]$
        end if
    else
        reject the customer by choosing price ∞
    end if
    observe valuation V_t and purchase decision X_t
    if $V_t > v$ **then**
        $\texttt{val}[i] = V_t$
        if $X_t = 1$ **then**
            $\texttt{sold}[i] = \texttt{true}$
        end if
    end if
    $t = t + 1$
end while

Input: Customers $t = 1, 2, \ldots$ arriving online, with each valuation V_t revealed after the price P_t is chosen.
Output: For each customer t, a (possibly random) price P_t for her.

Proof. Proof. We proceed by induction on t. (C.15) is true at time $t = 0$, where $\mathbb{E}[S_{i,0}] = 0$ and $v_{i,0} = 0$ for all $i \in [k]$.

Now suppose we are at the end of some time $t > 0$ and (C.15) was true at the end of time $t - 1$. It suffices to prove that $\mathbb{E}[S_{i_t,t}] = \mathbb{1}(v_{i_t,t} > 0) \cdot \frac{1+\ln v_{i_t,t}}{1+\ln R}$. This is immediate if $V_t \le v_{i_t,t-1}$, by the induction hypothesis. Otherwise, if $V_t > v_{i_t,t-1}$, we consider two cases. Let $v = v_{i_t,t-1}$ for brevity.

We know that $\mathbb{E}[S_{i_t,t}] = \mathbb{E}[S_{i_t,t-1}] + \mathbb{E}[X_t | S_{i_t,t-1} = 0] \cdot \Pr[S_{i_t,t-1} = 0]$.

If $v = 0$, then this equals

$$\begin{aligned} \Pr[X_t = 1 | S_{i_t,t-1} = 0] &= \frac{1}{1+\ln R}(1 + \int_1^{v_{i,t}} \frac{1}{r} dr) \\ &= \frac{1 + \ln v_{i,t}}{1 + \ln R} \end{aligned}$$

as desired. On the other hand, if $v > 0$, then

$$\begin{aligned}
\mathbb{E}[S_{i_t,t}] &= \frac{1+\ln v}{1+\ln R} + \frac{1}{\ln R - \ln v}\int_v^{v_{i,t}} \frac{1}{r}dr\Big(1 - \frac{1+\ln v}{1+\ln R}\Big) \\
&= \frac{1+\ln v}{1+\ln R} + \frac{\ln v_{i,t} - \ln v}{\ln R - \ln v}\Big(\frac{\ln R - \ln v}{1+\ln R}\Big) \\
&= \frac{1+\ln v_{i,t}}{1+\ln R}.
\end{aligned}$$

This completes the induction and the proof of the lemma. □

Lemma C.4.2. *Suppose $V_t = v_{i_t,t} > v_{i_t,t-1}$ in a time step $t \in [T]$. Then the expected revenue earned by the algorithm during time step t is $\frac{1}{1+\ln R}(v_{i_t,t} - v_{i_t,t-1})$.*

Proof. Proof. Let $t \in [T]$ be any time step for which $V_t > v_{i_t,t-1}$. Again, let v denote $\ell_{i_t,t-1}$, and we consider the two cases $v = 0$ and $v > 0$. If $v = 0$, then

$$\begin{aligned}
\mathbb{E}[P_t X_t] &= \frac{1}{1+\ln R}\Big(1 + \int_1^R r\mathbb{E}[X_t | P_t = r]\frac{1}{r}dr\Big) \\
&= \frac{1}{1+\ln R}\Big(1 + \int_1^R \mathbb{1}[v_{i_t,t} \geq r]dr\Big) \\
&= \frac{1}{1+\ln R}(1 + v_{i_t,t} - 1).
\end{aligned}$$

In the first equality, a sale is guaranteed if $P_t = 1$, earning revenue 1. The final term is the desired expression.

If $v > 0$, then

$$\begin{aligned}
\mathbb{E}[P_t X_t] &= \frac{1}{\ln R - \ln v}\Big(\int_v^R r\mathbb{E}[X_t | P_t = r]\frac{1}{r}dr\Big)\Big(\frac{\ln R - \ln v}{1+\ln R}\Big) \\
&= \frac{1}{1+\ln R}\Big(\int_v^R \mathbb{1}[v_{i_t,t} \geq r]dr\Big) \\
&= \frac{1}{1+\ln R}(v_{i_t,t} - v)
\end{aligned}$$

as desired. □

With these two lemmas, the rest of the proof follows Section C.1. Indeed, Lemma C.1.4 says that $\mathbb{E}[\mathsf{ALG}] = \frac{1}{1+\ln R}\sum_{i=1}^k v_{i,T}$. Meanwhile, Lemma C.1.6 says that $\mathsf{OPT} = \sum_{i=1}^k v_{i,T}$. Therefore, $\frac{\mathbb{E}[\mathsf{ALG}]}{\mathsf{OPT}} \geq \frac{1}{1+\ln R}$, and since $V_1, \ldots, V_T$ was arbitrary, Algorithm 9 is $\frac{1}{1+\ln R}$-competitive.

C.5 Upper Bounds on the Competitive Ratio relative to the DLP

First, it is well-known that the DLP overestimates the optimum by a factor of $1-\frac{1}{e}$, even when the feasible price set $\mathcal{P}$ consists of a singleton (i.e. the dynamic pricing problem is trivial because there is only one price to choose from). The example requires the starting inventory k to be 1. Without loss of generality assume $\mathcal{P}=\{1\}$. Consider T customers, each of whom have a valuation exceeding 1 with probability $\frac{1}{T}$, and a valuation of 0 otherwise. It is easy to check that $\mathsf{OPT}_{\mathsf{LP}}=1$ in this case, by setting $x_t^{(1)}=1$ for all $t\in[T]$. Meanwhile, any algorithm cannot have expected revenue exceeding $1-(1-\frac{1}{T})^T$, where we have subtracted from 1 the probability of all customers having valuation 0. As $T\to\infty$, $\frac{\mathbb{E}[\mathsf{ALG}]}{\mathsf{OPT}_{\mathsf{LP}}}$ approaches $1-\frac{1}{e}$.

However, the gap becomes even larger if $\mathcal{P}$ contains more than one price. We illustrate in the case of two feasible prices.

Lemma C.5.1. *Consider the stochastic-valuation model defined in Section 4.3, and let $\mathcal{P}=\{1,r\}$, $k=1$. For all $r\geq 1$, there exists a distribution over $\mathbf{v}_1,\dots,\mathbf{v}_T$ such that for any online algorithm,*

$$\frac{\mathbb{E}[\mathsf{ALG}(\mathbf{v}_1,\dots,\mathbf{v}_T)]}{\mathbb{E}[\mathsf{OPT}_{\mathsf{LP}}(\mathbf{v}_1,\dots,\mathbf{v}_T)]}\leq\min\{1-\frac{1}{e},\frac{r-r/e}{2r-1-r/e}\}. \tag{C.16}$$

If $r\leq\frac{1}{1-1/e}$, then the RHS of (C.16) is equal to $1-\frac{1}{e}\approx .632$. However, if $r>\frac{1}{1-1/e}$, then we show that the upper bound is $\frac{r-r/e}{2r-1-r/e}$, which decreases to $\frac{e-1}{2e-1}\approx .387$ as $r\to\infty$.

Proof. Proof. Suppose that $r>\frac{1}{1-1/e}$, and let $p=\frac{1}{r(1-1/e)}$, which is in $(0,1)$. Consider the following distribution over $\mathbf{v}_1,\dots,\mathbf{v}_T$:

- The first valuation distribution is deterministically $\mathbf{v}_1=(v_1^{(0)},v_1^{(1)},v_1^{(2)})=(0,1,0)$, i.e. the first customer deterministically has valuation 1.
- With probability p, valuation distributions $\mathbf{v}_2,\dots,\mathbf{v}_T$ are all equal to $(1-\frac{1}{T-1},0,\frac{1}{T-1})$. When this occurs, each of the $T-1$ customers $2,\dots,T$ are willing to pay r with probability $\frac{1}{T-1}$, and 0 otherwise.
- With probability $1-p$, valuation distributions $\mathbf{v}_2,\dots,\mathbf{v}_T$ are all equal to $(1,0,0)$. When this occurs, all customers $2,\dots,T$ will never make a purchase.

We first compute the expected value of $\mathsf{OPT}_{\mathsf{LP}}(\mathbf{v}_1, \ldots, \mathbf{v}_T)$. With probability $1-p$, $\mathsf{OPT}_{\mathsf{LP}}(\mathbf{v}_1, \ldots, \mathbf{v}_T) = 1$, setting $x_1^{(1)} = 1$. With probability p, $\mathsf{OPT}_{\mathsf{LP}}(\mathbf{v}_1, \ldots, \mathbf{v}_T) = r$, setting $x_1^{(1)} = x_1^{(2)} = 0$ and $x_2^{(2)} = \ldots = x_T^{(2)} = 1$. Therefore, $\mathbb{E}[\mathsf{OPT}_{\mathsf{LP}}(\mathbf{v}_1, \ldots, \mathbf{v}_T)] = 1 - p + pr$.

We now consider the optimal strategy for the online algorithm. It has to decide, at time 1, whether to sell the only unit of inventory at price 1, without knowing whether $\mathbf{v}_2, \ldots, \mathbf{v}_T$ are equal to $(1 - \frac{1}{T-1}, 0, \frac{1}{T-1})$ or $(1, 0, 0)$. Conditioned on it deciding to sell, $\mathsf{ALG}(\mathbf{v}_1, \ldots, \mathbf{v}_T)$ is deterministically 1. Conditioned on it deciding to wait, $\mathsf{ALG}(\mathbf{v}_1, \ldots, \mathbf{v}_T)$ is r with probability

$$\frac{1}{r(1-1/e)} \cdot (1 - (1 - \frac{1}{T-1})^{T-1}), \tag{C.17}$$

and 0 otherwise.

We explain (C.17). If the online algorithm decides to wait, then it will offer price r to all customers beyond the first. It gets a sale if $\mathbf{v}_2 = \ldots = \mathbf{v}_T = (1 - \frac{1}{T-1}, 0, \frac{1}{T-1})$, which occurs with probability $p = \frac{1}{r(1-1/e)}$, and further if at least 1 of the valuations $V_2, \ldots, V_T$ realizes to r, which yields the second term in (C.17).

Thus the expected revenue from deciding to wait is (C.17) multiplied by r, or

$$\frac{1}{(1-1/e)} \cdot (1 - (1 - \frac{1}{T-1})^{T-1}), \tag{C.18}$$

which is always greater than 1. Therefore, the online algorithm is better off waiting, in which case its expected revenue is (C.18). Taking $T \to \infty$, (C.18) approaches 1.

As $T \to \infty$, the distribution we constructed over $\mathbf{v}_1, \ldots, \mathbf{v}_T$ is such that for the best online algorithm,

$$\begin{aligned}
\frac{\mathbb{E}[\mathsf{ALG}(\mathbf{v}_1, \ldots, \mathbf{v}_T)]}{\mathbb{E}[\mathsf{OPT}_{\mathsf{LP}}(\mathbf{v}_1, \ldots, \mathbf{v}_T)]} &= \frac{1}{1-p+pr} \\
&= \frac{r(1-1/e)}{r(1-1/e) - 1 + r} \\
&= \frac{r - r/e}{2r - 1 - r/e},
\end{aligned}$$

as desired. □

Appendix D

Appendix to Chapter 5

D.1 Proof of Lemmas from Section 5.2

Proof. Proof of Lemma 5.2.3. Let f_i denote the deficit $\max\{P_i - x_i, 0\}$ of each item i. First suppose $\sum_{i=1}^{n} f_i \leq d$. It is clear that the customer should buy all items with $P_i \leq x_i$ regardless of whether the discount is relevant. For the items with $P_i > x_i$, the customer's deficit from buying all of them is $\sum_{i:P_i > x_i}(P_i - x_i) = \sum_{i=1}^{n} f_i$, but this is covered by the discount of d, hence the customer's utility is maximized when she buys the bundle. On the other hand, if $\sum_{i=1}^{n} f_i > d$, then the customer's utility is increased from not buying all items with $P_i > x_i$, since the reduction in deficit is $\sum_{i=1}^{n} f_i$ while the discount lost is only d. □

Proof. Proof of Lemma 5.2.4. Consider what has to happen for the customer to choose subset S. If the bundle deal didn't exist, then we would need $x_i \geq P_i$ for all $i \in S$ and $x_i < P_i$ for all $i \notin S$. However, the bundle cannibalizes some of the situations where the items in S would have been individually bought. By Lemma 5.2.3, these are situations where $\sum_{i=1}^{n} f_i \leq d$, or $\sum_{i \notin S}(P_i - x_i) \leq d$. By independence, we have

$$p_S^* = \Big(\prod_{i \in S} q_i^*\Big)\Big(\prod_{i \notin S}(1 - q_i^*)\Big)\Big(1 - \Pr\big[\sum_{i \notin S}(P_i - x_i) \leq d \big| x_i < P_i \; \forall i \notin S\big]\Big).$$

In order for $\sum_{i \notin S}(P_i - x_i) \leq d$, it must be the case that $x_i \geq P_i - d$ for all $i \notin S$. Therefore,

$$\Pr\big[\sum_{i \notin S}(P_i - x_i) \leq d \big| x_i < P_i \; \forall i \notin S\big] = \Pr\big[\sum_{i \notin S}(P_i - x_i) \leq d \big| P_i - d \leq x_i < P_i \; \forall i \notin S\big]\Big(\prod_{i \notin S} a_i^*\Big)$$

by the laws of conditional probability, completing the proof. ☐

Proof. Proof of Lemma 5.2.6. For all $i \in S$, the conditional distribution y_i of $P_i - x_i$ is uniform on $(0, d]$. Therefore, the probability that $\sum_{i \in S} y_i \leq d$ is the volume of the unit simplex in $|S|$ dimensions, or $\frac{1}{|S|!}$. ☐

D.2 Analysis of Algorithm

In this section we prove error bounds for our iterative algorithm, which imply convergence. We first provide a high-level overview of the proof and techniques.

Recall that we are trying to solve for variables q_i and a_i from a system of equations in high-degree polynomials. While this is generally intractable, we exploit the structure in customer bundle selection to derive an algorithm which is able to alternate between isolating the two sets of variables (the q_i's and a_i's), and iteratively use them to improve the accuracy in the other.

To analyze the error of such an algorithm, we first derive an expression for its initial error (Section D.2.3), based on both the error in the input (the error in $\hat{p}_S$ and F_S) and the error from our approximation used for the starting solution. We then carefully bound the error propagation over iterations, and show that the error bound decreases (Section D.2.4). A general error bound then follows by induction (Section D.2.5).

Overall our proof is quite elementary and straight-forward, despite requiring heavy notation and a long sequence of inequalities. This Section D.2 highlights the structure of the overall proof, while the proofs of individual statements are deferred to Section D.2.8.

D.2.1 No Division by Zero

First, it is imperative to verify that the algorithm never divides by zero. This is true as long as we don't initialize any of the input probabilities to be zero. Indeed, if $\hat{p}_S > 0$ for all $S \subseteq [n]$, then for all $i \in [n]$ and $k \geq 0$, the following can be verified inductively:

1. $q_i^{(0)} \in (0, 1)$ in (5.6), for all $i \in [n]$;

2. $a^{(k)} \in [0, 1)$ in (5.7), for all $k \geq 0$ and $i \in [n]$;

3. $q^{(k+1)} \in (0, 1)$ in (5.8), for all $k \geq 0$ and $i \in [n]$.

Since every $q_i^{(k)}$ is in $(0,1)$ and every $a_i^{(k)}$ is in $[0,1)$, there cannot be division by 0 in the algorithm.

To ensure that $\hat{p}_s \neq 0$, we use a standard trick, where we start counting the number of customers who selected each subset S at 1 (instead of 0). This method of smoothing the input causes a negligible error if the sales counts are high, and is desirable anyway if the sales counts are low.

D.2.2 Setup and Notation

Definition D.2.1. Define the following.

- For all $S \subseteq [n]$, let p_S^* be the probability that a customer with valuation vector x drawn from D purchases exactly the subset of items S.
- For all $S \neq \emptyset$, let $F_S^* = \Pr\left[\sum_{i \in S}(P_i - x_i) \leq d \middle| P_i - d \leq x_i < P_i \ \forall i \in S\right]$.

With this notation, we can state Lemma 5.2.4 simply as

$$p_S^* = \Big(\prod_{i \in S} q_i^*\Big)\Big(\prod_{i \notin S}(1 - q_i^*)\Big)\Big(1 - F_{[n]\setminus S}\prod_{i \notin S} a_i^*\Big), \qquad S \neq [n]. \tag{D.1}$$

Definition D.2.2. The algorithm we analyze will always set the parameters $\mathcal{S}_1 = \ldots = \mathcal{S}_n = \{\emptyset\}$.

Definition D.2.3. For all $i \in [n]$, define the following quantities:

- $p_i^* := \frac{p_{\{i\}}^*}{p_\emptyset^*}$ and $p_{-i}^* := p_{[n]\setminus\{i\}}^*$
- $\hat{p}_i := \frac{\hat{p}_{\{i\}}}{\hat{p}_\emptyset}$ and $\hat{p}_{-i} := \hat{p}_{[n]\setminus\{i\}}$
- $Q_i^* := \prod_{j \neq i} q_j^*$ and $A_i^* := \prod_{j \neq i} a_i^*$
- $Q_i^{(k)} := \prod_{j \neq i} q_j^{(k)}$ and $A_i^{(k)} := \prod_{j \neq i} a_i^{(k)}$, for all $k \geq 0$

Also define the following quantities:

- $A^* := \prod_{j=1}^n a_j^*$
- $A^{(k)} := \prod_{j=1}^n a_j^{(k)}$, for all $k \geq 0$
- $F_{n-1} := \frac{1}{(n-1)!}$ (recall that our estimate $F_{[n]\setminus\{i\}}$ of $F_{[n]\setminus\{i\}}^*$ is $\frac{1}{(n-1)!}$ for all $i \in [n]$)

- $F_n := \frac{1}{n!}$ (recall that our estimate $F_{[n]}$ of $F^*_{[n]}$ is $\frac{1}{n!}$)

Using this shorthand, we can derive the following:

$$
\begin{aligned}
p^*_i &= \frac{q^*_i}{1-q^*_i} \cdot \frac{1-F^*_{[n]\setminus\{i\}}A^*_i}{1-F^*_{[n]}A^*} && \text{(D.2)}\\
\hat{p}_i &= \frac{q^{(0)}_i}{1-q^{(0)}_i} && \text{(D.3)}\\
\hat{p}_i &= \frac{q^{(k+1)}_i}{1-q^{(k+1)}_i} \cdot \frac{1-F_{n-1}A^{(k)}_i}{1-F_nA^{(k)}} && \text{(D.4)}\\
p^*_{-i} &= Q^*_i(1-q^*_i)(1-a^*_i) && \text{(D.5)}\\
\hat{p}_{-i} &= Q^{(k)}_i(1-q^{(k)}_i)(1-a'^{(k)}_i) && \text{(D.6)}
\end{aligned}
$$

(D.2) follows from dividing (D.1) for $S = \{i\}$ by (D.1) for $S = \emptyset$, while (D.5) follows directly from (D.1) for $S = n \setminus \{i\}$. (D.3) follows from instruction (5.6) in the algorithm when $\mathcal{S}_i = \emptyset$. Similarly, (D.4) follows from instruction (5.8) when $\mathcal{S}_i = \emptyset$. Finally, (D.6) follows from instruction (5.7), where we have defined

$$a'^{(k)}_i := 1 - \frac{\hat{p}_{[n]\setminus\{i\}}}{(\prod_{j\neq i} q^{(k)}_j)(1-q^{(k)}_i)},$$

which is equal to $a^{(k)}_i$ when $a'^{(k)}_i$ is non-negative.

Assumption D.2.4. For all $i \in [n]$, we make the following mild assumptions on probabilities not being 0, which allows us to not worry about division by 0 in the analysis. Recall that we discussed how to set $\{\hat{p}_S : S \subseteq [n]\}$ so that none of them are 0 in Section D.2.1.

1. $p^*_i \in (0,\infty)$ (i.e. $p^*_{\{i\}} > 0$, $p^*_\emptyset > 0$) and $0 < q^*_i < 1$

2. $\hat{p}_i \in (0,\infty)$ (i.e. $\hat{p}_{\{i\}} > 0$, $\hat{p}_\emptyset > 0$) and $0 < q^{(k)}_i < 1$, for all $k \geq 0$

3. $p^*_{-i} > 0$ (i.e. $p^*_{[n]\setminus\{i\}} > 0$) and $a^*_i < 1$

4. $\hat{p}_{-i} > 0$ (i.e. $\hat{p}_{[n]\setminus\{i\}} > 0$) and $a'^{(k)}_i < 1$, for all $k \geq 0$

Note the correspondence between Assumptions 1,2,3,4 and equations (D.2),(D.3)–(D.4),(D.5),(D.6), respectively. Now we derive the following equations, which help us bound the errors in our estimates. (D.7) and (D.9) are obtained by taking (D.2) divided by (D.3) and (D.4) respectively, while (D.8) is obtained by taking (D.6) divided by (D.5).

$$\frac{q_i^*}{q_i^{(0)}} \cdot \frac{1-q_i^{(0)}}{1-q_i^*} = \frac{p_i^*}{\hat{p}_i} \cdot \frac{1-F_{[n]}^* A^*}{1-F_{[n]\setminus\{i\}}^* A_i^*} \tag{D.7}$$

$$\frac{1-a_i'^{(k)}}{1-a_i^*} = \frac{\hat{p}_{-i}}{p_{-i}^*} \cdot \frac{Q_i^*(1-q_i^*)}{Q_i^{(k)}(1-q_i^{(k)})}, \ \forall k \geq 0 \tag{D.8}$$

$$\frac{q_i^*}{q_i^{(k+1)}} \cdot \frac{1-q_i^{(k+1)}}{1-q_i^*} = \frac{p_i^*}{\hat{p}_i} \cdot \frac{1-F_{n-1}A_i^{(k)}}{1-F_{[n]\setminus\{i\}}^* A_i^*} \cdot \frac{1-F_{[n]}^* A^*}{1-F_n A^{(k)}}, \ \forall k \geq 0 \tag{D.9}$$

Definition D.2.5. The following constants are not known to the algorithm, but we derive error bounds which are parametrized by them:

1. Let $L \geq 0$ be a bound on the input error such that both $\frac{p_i^*}{\hat{p}_i}$ and $\frac{p_{-i}^*}{\hat{p}_{-i}}$ lie in $[\frac{1}{1+L}, 1+L]$, for all $i \in [n]$.
2. Let $C \geq 0$ be a lower bound on $\Pr[x_i < P_i - d | x_i < P_i]$, i.e. $a_i^* \leq 1 - C$ for all $i \in [n]$.
3. Let $F_{n-1}^{\max}$ (resp. $F_{n-1}^{\min}$) be a constant in $[\frac{1}{(n-1)!}, 1)$ (resp. $[0, \frac{1}{(n-1)!}]$) such that $F_{n-1}^{\min} \leq F_{[n]\setminus\{i\}}^* \leq F_{n-1}^{\max}$ for all $i \in [n]$.
4. Let $F_n^{\max}$ (resp. $F_n^{\min}$) be a constant in $[\frac{1}{n!}, 1)$ (resp. $[0, \frac{1}{n!}]$) such that $F_n^{\min} \leq F_{[n]}^* \leq F_n^{\max}$.

D.2.3 Initialization

Lemma D.2.6. *The following bounds on the initial estimates $q_1^{(0)}, \ldots, q_n^{(0)}$ of $q_1^*, \ldots, q_n^*$ can be derived from (D.7):*

$$\frac{1}{1+L} \leq \frac{q_i^*}{q_i^{(0)}} \cdot \frac{1-q_i^{(0)}}{1-q_i^*} \leq \frac{1+L}{1-F_{n-1}^{\max}(1-C)^{n-1}}. \tag{D.10}$$

We have bounded $\frac{q_i^*}{q_i^{(0)}} \cdot \frac{1-q_i^{(0)}}{1-q_i^*}$, but we would like to bound the individual terms $\frac{q_i^*}{q_i^{(0)}}$ and $\frac{1-q_i^{(0)}}{1-q_i^*}$. The following lemma accomplishes this via the observation that they are either both greater than 1, both less than 1, or both equal to 1.

Lemma D.2.7. *Let q^*, q be real numbers in $(0,1)$ and $R \geq 0$ be a bound such that $\frac{q^*}{q} \cdot \frac{1-q}{1-q^*} \leq 1+R$. Then both $\frac{q^*}{q} \leq 1+R$ and $\frac{1-q}{1-q^*} \leq 1+R$. Taking reciprocals, we also have that if $\frac{1}{1+R} \leq \frac{q^*}{q} \cdot \frac{1-q}{1-q^*}$, then $\frac{1}{1+R} \leq \frac{q^*}{q}$ and $\frac{1}{1+R} \leq \frac{1-q}{1-q^*}$.*

The following bounds on the initial estimates $a_1^{(0)}, \ldots, a_n^{(0)}$ of $a_1^*, \ldots, a_n^*$ are then immediate by applying Lemmas D.2.6–D.2.7 to (D.8):

$$\frac{1 - F_{n-1}^{\max}(1-C)^{n-1}}{(1+L)^{n+1}} \leq \frac{1-a_i^{(0)}}{1-a_i^*} \leq \frac{(1+L)^{n+1}}{(1 - F_{n-1}^{\max}(1-C)^{n-1})^{n-1}}. \tag{D.11}$$

Note that we do not need to distinguish between $a_i'^{(k)}$ and $a_i^{(k)} = \max\{a_i'^{(k)}, 0\}$ when using (D.8) to bound $\frac{1-a_i^{(k)}}{1-a_i^*}$, since if $a_i'^{(k)} < 0$ and $a_i^{(k)} = 0$, then $1 < \frac{1-a_i^{(k)}}{1-a_i^*} < \frac{1-a_i'^{(k)}}{1-a_i^*}$.

At this point, we focus on bounding the errors in our estimates of $1 - a_i^*$. Bounds on the errors in q_i^* and $1 - q_i^*$ will then follow in Section D.2.5.

For all $k \geq 0$, our goal is to inductively find a $B^{(k)} > 0$ such that

$$\frac{1}{1+B^{(k)}} \leq \frac{1-a_i^{(k)}}{1-a_i^*} \leq 1 + B^{(k)} \tag{D.12}$$

for all $i \in [n]$. We would like to prove that the sequence of error bounds $(B^{(0)}, B^{(1)}, B^{(2)}, \ldots)$ decreases toward some small error in a small number of iterations. The error cannot be 0 unless $L = 0$ and $F_{n-1}^{\min} = F_{n-1} = F_{n-1}^{\max}, F_n^{\min} = F_n = F_n^{\max}$. Our specific strategy is to show exponential decay, i.e. find α and γ such that $B^{(k+1)} = \alpha + \gamma \cdot B^{(k)}$ is a feasible value (i.e., satisfies (D.12)).

We can see from (D.11) that the following is a feasible value for $B^{(0)}$:

$$B^{(0)} := \frac{(1+L)^{n+1}}{(1 - F_{n-1}^{\max}(1-C)^{n-1})^{n-1}} - 1. \tag{D.13}$$

Finally, C was a lower bound on $1 - a_i^*$ across $i \in [n]$. $C = 0$ is a valid such bound, but our algorithm's performance can improve if C is larger. Nonetheless, for technical convenience, we do not want situations where it is greater than $1/(1 + B^{(0)})$ or $1/n$. In this case, we will simply redefine it to be 0.

Definition D.2.8. If C is large enough such that $\frac{B^{(0)}}{1+B^{(0)}} \leq 1 - C$ or $Cn \leq 1$ is violated, then redefine $C := 0$, which is guaranteed to fix the violations while still satisfying Definition D.2.5.

D.2.4 Induction Step

In this section, given an iteration $k \geq 0$ and $B^{(k)} \in (0, B^{(0)}]$ such that $\frac{1}{1+B^{(k)}} \leq \frac{1-a_i^{(k)}}{1-a_i^*} \leq 1 + B^{(k)}$ for all $i \in [n]$, we find a $B^{(k+1)}$ such that $\frac{1}{1+B^{(k+1)}} \leq \frac{1-a_i^{(k+1)}}{1-a_i^*} \leq 1 + B^{(k+1)}$ for all $i \in [n]$. We will show that both $\frac{1-a_i^{(k+1)}}{1-a_i^*}$ and $\frac{1-a_i^*}{1-a_i^{(k+1)}}$ are no greater than $1 + B^{(k+1)}$. Fix some parameter $z \in [0, B^{(k)}]$, which affects the value of $B^{(k+1)}$ (we explain how to choose z later).

The following theorem is the key technical ingredient in our analysis:

Theorem D.2.9. *Let $m \geq 2$. Let $a_1^*, \ldots, a_m^*$ be underlying values in $[0, 1)$, and let $a_1, \ldots, a_m$ be respective estimates for these values, also in $[0, 1)$. Let $B > 0$ be a bound on the error of these estimates, where for all $i \in [m]$, $\frac{1}{1+B} \leq \frac{1-a_i}{1-a_i^*} \leq 1 + B$.*

Let $F, F^*, F^{\min}, F^{\max}, C, z$ be constants satisfying the following:

1. *$0 \leq F^{\min} \leq F^{\max} < 1$, and both F, F^* lie in $[F^{\min}, F^{\max}]$*
2. *$a_i^* \leq 1 - C$ for all $i \in [m]$, $\frac{B}{1+B} \leq 1 - C$, $Cm \leq 1$, and $C \geq 0$*
3. *$0 \leq z \leq B$*

Then the following upper bounds hold for $E := \dfrac{1 - F \prod_{i=1}^m a_i}{1 - F^* \prod_{i=1}^m a_i^*}$ and its reciprocal:

$$E \leq \quad 1 + \max\left\{B \cdot \frac{F^{\max}(1-C)^{m-1}}{1 + z(1 - F^{\max}(1-C)^{m-1})}, \frac{(F^{\max} - F)(1-C)^m}{1 - F^{\max}(1-C)^m} + B \cdot \frac{mCF(1-C)^{m-1}}{1 - F^{\max}(1-C)^m}\right\} \tag{D.14}$$

$$E^{-1} \leq \quad 1 + \max\{B \cdot \frac{F(\frac{1}{1+z})(1 - \frac{C}{1+z})^{m-1}}{1 - F(\frac{B}{1+B})(1 - \frac{C}{1+B})^{m-1}}, \frac{(F - F^{\min})(1-C)^m}{1 - F(1 - \frac{C}{1+B})^m} + B \cdot \frac{mCF(\frac{1}{1+z})(1 - \frac{C}{1+z})^{m-1}}{1 - F(1 - \frac{C}{1+B})^m}\}. \tag{D.15}$$

Definition D.2.10. Based on (D.14) and (D.15), for $m = n - 1$ and $m = n$, define the following constants $\alpha_m, \overline{\alpha}_m$, and functions $\gamma_m, \overline{\gamma}_m$ of z:

$$\begin{aligned}
\alpha_m &= \frac{(F_m^{\max} - F_m)(1-C)^m}{1 - F_m^{\max}(1-C)^m} \\
\gamma_m(z) &= \max\left\{\frac{mCF_m(1-C)^{m-1}}{1 - F_m^{\max}(1-C)^m}, \frac{F_m^{\max}(1-C)^{m-1}}{1 + z(1 - F_m^{\max}(1-C)^{m-1})} - \frac{\alpha_m}{B^{(0)}}\right\} \\
\overline{\alpha}_m &= \frac{(F_m - F_m^{\min})(1-C)^m}{1 - F_m(1 - \frac{C}{1+B^{(0)}})^m} \\
\overline{\gamma}_m(z) &= \max\left\{\frac{mCF_m(\frac{1}{1+z})(1 - \frac{C}{1+z})^{m-1}}{1 - F_m(1 - \frac{C}{1+B^{(0)}})^m}, \frac{F_m(\frac{1}{1+z})(1 - \frac{C}{1+z})^{m-1}}{1 - F_m(\frac{B^{(0)}}{1+B^{(0)}})(1 - \frac{C}{1+B^{(0)}})^{m-1}} - \frac{\overline{\alpha}_m}{B^{(0)}}\right\}.
\end{aligned}$$

Note that $\alpha_m, \overline{\alpha}_m \geq 0$, and $\gamma_m(z), \overline{\gamma}_m(z) \geq 0$ when $z \geq 0$. Moreover, $\gamma_m(z)$ is weakly decreasing (non-increasing) in z. The following lemma shows that $\overline{\gamma}_m(z)$ is also weakly decreasing in z:

Lemma D.2.11. *Consider the expression $\frac{1}{1+z}(1-\frac{C}{1+z})^{m-1}$ as a function of z, where $m \in \{2,3,\ldots\}$ and $C \in [0, \frac{1}{m}]$. This is a weakly decreasing function over $[0,\infty)$. Therefore, the maximum of this function is $(1-C)^{m-1}$, attained at $z = 0$.*

Lemma D.2.12. *We can apply Theorem D.2.9 to obtain the following:*

$$\frac{1 - F_{n-1}A_i^{(k)}}{1 - F^*_{[n]\setminus\{i\}}A_i^*} \leq 1 + \alpha_{n-1} + \gamma_{n-1}(z) \cdot B^{(k)}, \ \forall i \in [n] \tag{D.16}$$

$$\frac{1 - F_n A^{(k)}}{1 - F^*_{[n]}A^*} \leq 1 + \alpha_n + \gamma_n(z) \cdot B^{(k)} \tag{D.17}$$

$$\frac{1 - F^*_{[n]\setminus\{i\}}A_i^*}{1 - F_{n-1}A_i^{(k)}} \leq 1 + \overline{\alpha}_{n-1} + \overline{\gamma}_{n-1}(z) \cdot B^{(k)}, \ \forall i \in [n] \tag{D.18}$$

$$\frac{1 - F^*_{[n]}A^*}{1 - F_n A^{(k)}} \leq 1 + \overline{\alpha}_n + \overline{\gamma}_n(z) \cdot B^{(k)}. \tag{D.19}$$

Definition D.2.13. Define the following. h and $\overline{h}$ are functions over $z, w \geq 0$, and note that they take values greater than 1 over this domain (the inequality is strict because $L > 0$). Similarly, α and $\overline{\alpha}$ are constants greater than 0. γ and $\overline{\gamma}$ are non-negative functions over $z \geq 0$. Lemma D.2.11 established that $h(z,w), \overline{h}(z,w), \gamma(z), \overline{\gamma}(z)$ are also weakly decreasing in z.

$$\begin{aligned}
h(z,w) &= (1+L)^{n+1} \cdot (1 + \alpha_{n-1} + w\gamma_{n-1}(z))^{n-1} \cdot (1 + \overline{\alpha}_n + w\overline{\gamma}_n(z))^{n-1} \\
&\quad \cdot (1 + \overline{\alpha}_{n-1} + w\overline{\gamma}_{n-1}(z)) \cdot (1 + \alpha_n + w\gamma_n(z)) \\
\alpha &= (n+1)L + (n-1)(\alpha_{n-1} + \overline{\alpha}_n) + (\overline{\alpha}_{n-1} + \alpha_n) \\
\gamma(z) &= (n-1)(\gamma_{n-1}(z) + \overline{\gamma}_n(z)) + (\overline{\gamma}_{n-1}(z) + \gamma_n(z)) \\
\overline{h}(z,w) &= (1+L)^{n+1} \cdot (1 + \alpha_{n-1} + w\gamma_{n-1}(z)) \cdot (1 + \overline{\alpha}_n + w\overline{\gamma}_n(z)) \\
&\quad \cdot (1 + \overline{\alpha}_{n-1} + w\overline{\gamma}_{n-1}(z))^{n-1} \cdot (1 + \alpha_n + w\gamma_n(z))^{n-1} \\
\overline{\alpha} &= (n+1)L + (\alpha_{n-1} + \overline{\alpha}_n) + (n-1)(\overline{\alpha}_{n-1} + \alpha_n) \\
\overline{\gamma}(z) &= (\gamma_{n-1}(z) + \overline{\gamma}_n(z)) + (n-1)(\overline{\gamma}_{n-1}(z) + \gamma_n(z))
\end{aligned}$$

The intuition for Definition D.2.13 is that when w is close to 0, $1+\alpha+w\gamma(z)$ is supposed to be a good approximation of $h(z,w)$ (and there is a respective approximation for $\overline{h}(z,w)$). The following lemma bounds the error in this approximation:

Lemma D.2.14. *Let $m \in \mathbb{N}$, and let $a_1,\ldots,a_m$ be non-negative real numbers. Then $\dfrac{\prod_{i=1}^m(1+a_i)-1}{\sum_{i=1}^m a_i}$ as a function of $(a_1,\ldots,a_m)$ takes values in $[1,\infty)$ (over the domain where $\sum_{i=1}^m a_i \neq 0$), and is weakly increasing in a_i for all $i \in [m]$.*

Therefore, if we define

$$H(z) := \frac{h(z,B^{(0)})-1}{\alpha+B^{(0)}\gamma(z)} \tag{D.20}$$

$$\overline{H}(z) := \frac{\overline{h}(z,B^{(0)})-1}{\overline{\alpha}+B^{(0)}\overline{\gamma}(z)}, \tag{D.21}$$

then Lemma D.2.14 says that for all $0 \le w \le B^{(0)}$, $\frac{h(z,w)-1}{\alpha+w\gamma(z)} \le H(z)$ and $\frac{\overline{h}(z,w)-1}{\overline{\alpha}+w\overline{\gamma}(z)} \le \bar{H}(z)$, since $w \le B^{(0)}$ implies that $h(z,w) \le h(z,B^{(0)})$. $B^{(0)}$ is a feasible choice of w, so we have

$$h(z,B^{(k)}) \le 1+H(z)(\alpha+B^{(k)}\gamma(z)) \tag{D.22}$$

$$\overline{h}(z,B^{(k)}) \le 1+\overline{H}(z)(\overline{\alpha}+B^{(k)}\overline{\gamma}(z)). \tag{D.23}$$

Note that to the contrary, $z' \le z$ implies that $h(z',w) \ge h(z,w)$, since $h(z,w), \overline{h}(z,w), \gamma(z), \overline{\gamma}(z)$ are weakly decreasing in z. Therefore, $H(z)$ and $\overline{H}(z)$ are also weakly decreasing in z.

We are finally ready to proceed with the induction step. Applying Lemma D.2.12 to (D.9), and using Lemma D.2.7, we obtain

$$\max\left\{\frac{q_j^*}{q_j^{(k+1)}}, \frac{1-q_j^{(k+1)}}{1-q_j^*}\right\} \le (1+L)(1+\alpha_{n-1}+\gamma_{n-1}(z)\cdot B^{(k)})(1+\overline{\alpha}_n+\overline{\gamma}_n(z)\cdot B^{(k)}) \tag{D.24}$$

$$\max\left\{\frac{1-q_j^*}{1-q_j^{(k+1)}}, \frac{q_j^{(k+1)}}{q_j^*}\right\} \le (1+L)(1+\overline{\alpha}_{n-1}+\overline{\gamma}_{n-1}(z)\cdot B^{(k)})(1+\alpha_n+\gamma_n(z)\cdot B^{(k)}). \tag{D.25}$$

Applying (D.24) $n-1$ times and (D.25) once to (D.8), we obtain

$$\frac{1-a_i^{(k+1)}}{1-a_i^*} \le h(z,B^{(k)}) \le 1+H(z)\cdot\alpha+H(z)\cdot\gamma(z)\cdot B^{(k)}$$

where the second inequality uses (D.22). Applying (D.24) once and (D.25) $n-1$ times to the reciprocal of (D.8), we obtain $\frac{1-a_i^*}{1-a_i^{(k+1)}} \leq \overline{h}(z, B^{(k)})$, which by (D.23) is at most $\leq 1 + \overline{H}(z) \cdot \overline{\alpha} + \overline{H}(z) \cdot \overline{\gamma}(z) \cdot B^{(k)}$. Therefore, if we define

$$\begin{aligned} \tilde{\alpha}(z) &:= \max\{H(z) \cdot \alpha, \overline{H}(z) \cdot \overline{\alpha}\} && \text{(D.26)} \\ \tilde{\gamma}(z) &:= \max\{H(z) \cdot \gamma(z), \overline{H}(z) \cdot \overline{\gamma}(z)\}, && \text{(D.27)} \end{aligned}$$

then both $\frac{1-a_i^{(k+1)}}{1-a_i^*}$ and $\frac{1-a_i^*}{1-a_i^{(k+1)}}$ are no greater than $1 + \tilde{\alpha}(z) + \tilde{\gamma}(z) \cdot B^{(k)}$. Thus it suffices to set $B^{(k+1)} := \tilde{\alpha}(z) + \tilde{\gamma}(z) \cdot B^{(k)}$, completing the induction. Of course, at this point, we have not verified that $B^{(k+1)} < B^{(k)}$. We describe when this is the case in Section D.2.5.

D.2.5 Final Error Bounds

In this section we consolidate the induction from Sections D.2.3–D.2.4, in two theorems, which bound the errors in the algorithm's estimates of the a_i^* and q_i^* parameters, respectively. We illustrate explicit values the bounds can take in Section D.2.6.

Theorem D.2.15. *Let $n, L, C, F_{n-1}^{\min}, F_{n-1}^{\max}, F_n^{\min}, F_n^{\max}$ be constants satisfying Definitions D.2.5-D.2.8. Let $B^{(0)}$ be defined as in (D.13), and let z be a non-negative parameter. Let $\tilde{\alpha}(z), \tilde{\gamma}(z)$ be as defined through the sequence of Definition D.2.10, Definition D.2.13, (D.20)-(D.21), (D.26)-(D.27), and suppose they satisfy the following conditions:*

$$\text{(i) } \tilde{\gamma}(z) < 1, \text{ (ii) } \frac{\tilde{\alpha}(z)}{1-\tilde{\gamma}(z)} < B^{(0)}, \text{ (iii) } \frac{\tilde{\alpha}(z)}{1-\tilde{\gamma}(z)} \geq z. \tag{D.28}$$

Furthermore, define

$$\begin{aligned} B^{(\infty)} &:= \frac{\tilde{\alpha}(z)}{1-\tilde{\gamma}(z)} \\ B^{(k)} &:= B^{(\infty)} + \tilde{\gamma}(z)^k (B^{(0)} - B^{(\infty)}) \ \forall k \geq 0. \end{aligned}$$

Then, for all $k \geq 0$ and $i \in [n]$,

$$\frac{1}{1+B^{(k)}} \leq \frac{1-a_i^{(k)}}{1-a_i^*} \leq 1 + B^{(k)}.$$

In Theorem D.2.15, z is a parameter which can be optimized to make the error bounds $B^{(k)}$ as small as possible. However, it must satisfy the conditions in (D.28) which guarantee that $B^{(k)}$ is decreasing in k. Note that if the errors in the input observations are too large, then it is possible that no z can satisfy (D.28). In these extremely noisy cases, there is no guarantee that iterating our algorithm will improve the estimates; in fact, it may overfit to the noisy input, as we will see in Section D.2.6 when we display explicit values taken by the bounds.

We propose the following convenient method for choosing z which compares just two possibilities; we will follow it in displaying most of the explicit values shown in Section D.2.6. Note that (iii) is guaranteed to be satisfied (contingent on (i) being satisfied) if we choose $z = 0$. However, since $\gamma(z), \overline{\gamma}(z), H(z), \overline{H}(z)$ are weakly decreasing in z, $\tilde{\alpha}(z)$ and $\tilde{\gamma}(z)$ are also weakly decreasing in z, which means that choosing $z = 0$ results in the worst bounds. Therefore, we do the following:

1. Consider $z = B^{(0)}$. This is the maximum possible value for z, since it is imposed that $z \leq B^{(k)}$ for all $k \geq 0$. If (i) and (ii) in (D.28) are not satisfied, then they cannot be satisfied by any z, since $\tilde{\alpha}(z)$ and $\tilde{\gamma}(z)$ are at their minimum when $z = B^{(0)}$.

2. If (i) and (ii) in (D.28) are satisfied, then define $\tilde{B} := \dfrac{\tilde{\alpha}(B^{(0)})}{1 - \tilde{\gamma}(B^{(0)})}$. (i) and (ii) imply $\tilde{B} < B^{(0)}$.

3. Now consider $z = \tilde{B}$. If (i) and (ii) are still satisfied when z is this smaller value, then we claim that (iii) is guaranteed to be satisfied, because (iii) would be equivalent to $\dfrac{\tilde{\alpha}(\tilde{B})}{1 - \tilde{\gamma}(\tilde{B})} \geq \dfrac{\tilde{\alpha}(B^{(0)})}{1 - \tilde{\gamma}(B^{(0)})}$, which is true since $\tilde{\alpha}(z)$ and $\tilde{\gamma}(z)$ are weakly decreasing.

Returning to Theorem D.2.15, we now translate Theorem D.2.15 to bounds on $\frac{q_i^*}{q_i^{(k)}}$ and $\frac{1-q_i^*}{1-q_i^{(k)}}$ in the following way.

Theorem D.2.16. *Define*

$$
\begin{aligned}
R^{(0)} &:= \frac{1+L}{1 - F_{n-1}^{\max}(1-C)^{n-1}} - 1 \\
R^{(k+1)} &:= \max\big\{(1+L)(1+\alpha_{n-1}+\gamma_{n-1}(z)\cdot B^{(k)})(1+\overline{\alpha}_n+\overline{\gamma}_n(z)\cdot B^{(k)}), \\
&\qquad (1+L)(1+\overline{\alpha}_{n-1}+\overline{\gamma}_{n-1}(z)\cdot B^{(k)})(1+\alpha_n+\gamma_n(z)\cdot B^{(k)})\big\}\ \forall k \geq 0 \quad \text{(D.29)}
\end{aligned}
$$

Then $\frac{1}{1+R^{(k)}} \leq \frac{q_j^*}{q_j^{(k)}} \cdot \frac{1-q_j^{(k)}}{1-q_j^*} \leq 1+R^{(k)}$ *for* $k = 0, 1$. *Furthermore, suppose conditions (i)-(iii) in Theorem D.2.15 are satisfied. Then* $\frac{1}{1+R^{(k)}} \leq \frac{q_j^*}{q_j^{(k)}} \cdot \frac{1-q_j^{(k)}}{1-q_j^*} \leq 1+R^{(k)}$ *also holds for* $k \geq 2$.

Theorem D.2.16 follows easily from Theorem D.2.15, and (D.7), (D.24), (D.25). It implies multiplicative bounds on $\frac{q_i^*}{q_i^{(k)}}$ and $\frac{1-q_i^*}{1-q_i^{(k)}}$ via Lemma D.2.7.

D.2.6 Explicit Values of Error Bounds

We now illustrate explicit values the bounds from Theorems D.2.15–D.2.16 can take. We consider values of n in $\{3,4,5\}$ and values of L in $\{0,0.02,0.05\}$. We consider values of C in $\{0,0.05\}$.

We consider two approaches for setting the $F^{\min}$'s and $F^{\max}$'s. In the first approach, we directly impose that both $\frac{F^*_{[n]\setminus\{i\}}}{F_{n-1}}$ and $\frac{F^*_{[n]}}{F_n}$ lie in $[\frac{1}{1+\varepsilon_F}, 1+\varepsilon_F]$ for all $i \in [n]$, for an error parameter ε_F taking values in $\{0,0.25,1\}$. This implies that $F_m^{\min} = \frac{1}{m!(1+\varepsilon_F)}$ and $F_m^{\max} = \frac{1+\varepsilon_F}{m!}$ for $m \in \{n-1,n\}$.

In the second approach, we assume some lower bound on the conditional density of each D_i on $[P_i - d, P_i)$. Specifically, for all $i \in [n]$ and $y \in [P_i - d, P_i)$, we impose that $f_{x_i|P_i-d\le x_i<P_i}(y) \ge \frac{1-\varepsilon_{\mathsf{Unif}}}{d}$, where $\varepsilon_{\mathsf{Unif}}$ is an error parameter taking values in $\{0,0.1,0.2\}$. Explicit values of $F_m^{\min}$ and $F_m^{\max}$ for $m \in \{n-1,n\}$ can then be derived. The derivation is deferred to the end of Section D.2.9.

In Tables D.1–D.3, we display explicit values taken by the formulas from Section D.2.5, for different combination of n, L, C, and ε_F or $\varepsilon_{\mathsf{Unif}}$ (and the values of $F^{\min}, F^{\max}$ implied). For each combination, if conditions (i)–(ii) are satisfied by $\tilde{B}$ and thus we have the exponentially decaying error bounds in Theorems D.2.15–D.2.16, then we display all of $B^{(0)}, B^{(\infty)}, R^{(0)}, R^{(\infty)}$, where $R^{(\infty)}$ is defined by substituting $B^{(\infty)}$ into (D.29). We display arrows going from the initial error bounds to the asymptotic error bounds. If conditions (i)–(ii) are not satisfied for a combination, then we only display $B^{(0)}$ (which is still a valid error bound), and $R^{(0)}$ followed by $R^{(1)}$ ($R^{(0)}$ and $R^{(1)}$ are both valid error bounds, and either one could be smaller).

Additional details about the numbers in Tables D.1–D.3 can be found in Section D.2.9. In general, the tables are arranged in order of smallest error bounds ($n = 5$) to largest error bounds ($n = 3$). In the top-left corners, we have error bounds decreasing to 0 and thus convergence to the true values a_i^* and q_i^*. Over the possibilities of ε_F or $\varepsilon_{\mathsf{Unif}}$ from left to right, the ranges $[F_m^{\min}, F_m^{\max}]$ (for $m = n-1, n$) are roughly increasing. From top to bottom, note that the error bounds are increasing as L gets bigger, and C gets smaller.

We would like to point out that these error bounds arise when the inaccuracies in

Table D.1: Error Bounds for $n = 5$, with different values of L, C, and ε_F or $\varepsilon_{\mathsf{Unif}}$.

$n = 5$		$\varepsilon_F = \varepsilon_{\mathsf{Unif}} = 0$	$\varepsilon_F = \frac{1}{4}$	$\varepsilon_{\mathsf{Unif}} = 0.1$	$\varepsilon_F = 1$	$\varepsilon_{\mathsf{Unif}} = 0.2$
	$F_4^{\min}$	0.0417	0.0333	0.0273	0.0208	0.0171
	$F_4^{\max}$		0.0521	0.1039	0.0833	0.1893
	$F_5^{\min}$	0.0083	0.0067	0.0049	0.0042	0.0027
	$F_5^{\max}$		0.0104	0.0352	0.0167	0.0863
$L = 0$	$C = .05$	$B^{(k)} : 0.15 \to 0$	$B^{(k)} : 0.19 \to 0.05$	$B^{(k)} : 0.42 \to 0.32$	$B^{(k)} : 0.32 \to 0.21$	$B^{(k)} : 0.96 \to 0.95$
		$R^{(k)} : 0.04 \to 0$	$R^{(k)} : 0.04 \to 0.01$	$R^{(k)} : 0.09 \to 0.06$	$R^{(k)} : 0.07 \to 0.04$	$R^{(k)} : 0.18 \to 0.16$
	$C = 0$	$B^{(k)} : 0.19 \to 0$	$B^{(k)} : 0.24 \to 0.06$	$B^{(k)} : 0.55 \to 0.39$	$B^{(k)} : 0.42 \to 0.25$	$B^{(k)} : 1.32 \to 1.22$
		$R^{(k)} : 0.04 \to 0$	$R^{(k)} : 0.05 \to 0.01$	$R^{(k)} : 0.12 \to 0.07$	$R^{(k)} : 0.09 \to 0.05$	$R^{(k)} : 0.23 \to 0.19$
$L = .02$	$C = .05$	$B^{(k)} : 0.29 \to 0.16$	$B^{(k)} : 0.34 \to 0.20$	$B^{(k)} : 0.60 \to 0.49$	$B^{(k)} : 0.49 \to 0.37$	$B^{(k)} : 1.22 \to 1.20$
		$R^{(k)} : 0.06 \to 0.03$	$R^{(k)} : 0.07 \to 0.03$	$R^{(k)} : 0.11 \to 0.08$	$R^{(k)} : 0.09 \to 0.06$	$R^{(k)} : 0.21 \to 0.18$
	$C = 0$	$B^{(k)} : 0.34 \to 0.17$	$B^{(k)} : 0.39 \to 0.22$	$B^{(k)} : 0.75 \to 0.59$	$B^{(k)} : 0.59 \to 0.41$	$B^{(k)} : 1.61 \to 1.52$
		$R^{(k)} : 0.06 \to 0.03$	$R^{(k)} : 0.08 \to 0.04$	$R^{(k)} : 0.14 \to 0.10$	$R^{(k)} : 0.11 \to 0.07$	$R^{(k)} : 0.26 \to 0.21$
$L = .05$	$C = .05$	$B^{(k)} : 0.54 \to 0.44$	$B^{(k)} : 0.59 \to 0.48$	$B^{(k)} : 0.91 \to 0.81$	$B^{(k)} : 0.78 \to 0.65$	$B^{(k)} : 1.70 \to 1.62$
		$R^{(k)} : 0.09 \to 0.06$	$R^{(k)} : 0.10 \to 0.07$	$R^{(k)} : 0.15 \to 0.12$	$R^{(k)} : 0.13 \to 0.10$	$R^{(k)} : 0.24 \to 0.22$
	$C = 0$	$B^{(k)} : 0.59 \to 0.46$	$B^{(k)} : 0.66 \to 0.51$	$B^{(k)} : 1.08 \to 0.91$	$B^{(k)} : 0.90 \to 0.69$	$B^{(k)} : 2.10 \to 2.02$
		$R^{(k)} : 0.10 \to 0.07$	$R^{(k)} : 0.11 \to 0.08$	$R^{(k)} : 0.17 \to 0.13$	$R^{(k)} : 0.15 \to 0.10$	$R^{(k)} : 0.30 \to 0.25$

Table D.2: Error Bounds for $n=4$, with different values of L, C, and ε_F or $\varepsilon_{\mathsf{Unif}}$.

$n=4$		$\varepsilon_F=\varepsilon_{\mathsf{Unif}}=0$	$\varepsilon_F=\frac{1}{4}$	$\varepsilon_{\mathsf{Unif}}=0.1$	$\varepsilon_F=1$	$\varepsilon_{\mathsf{Unif}}=0.2$
	$F_3^{\min}$	0.1667	0.1333	0.1215	0.0833	0.0853
	$F_3^{\max}$		0.2083	0.2710	0.3333	0.3813
	$F_4^{\min}$	0.0417	0.0333	0.0273	0.0208	0.0171
	$F_4^{\max}$		0.0521	0.1039	0.0833	0.1893
$L=0$	$C=.05$	$B^{(k)}: 0.59 \to 0$ $R^{(k)}: 0.17 \to 0$	$B^{(k)}: 0.80 \to 0.25$ $R^{(k)}: 0.22 \to 0.06$	$B^{(k)}: 1.21 \to 0.83$ $R^{(k)}: 0.30 \to 0.16$	$B^{(k)}: 1.74 \to 1.39$ $R^{(k)}: 0.40 \to 0.27$	$B^{(0)}=2.28$ $R^{(k)}=0.49, 0.39$
	$C=0$	$B^{(k)}: 0.73 \to 0$ $R^{(k)}: 0.20 \to 0$	$B^{(k)}: 1.02 \to 0.34$ $R^{(k)}: 0.26 \to 0.08$	$B^{(k)}: 1.58 \to 1.00$ $R^{(k)}: 0.37 \to 0.19$	$B^{(0)}=2.38$ $R^{(k)}=0.50, 0.35$	$B^{(0)}=3.22$ $R^{(k)}=0.62, 0.46$
$L=.02$	$C=.05$	$B^{(k)}: 0.75 \to 0.37$ $R^{(k)}: 0.19 \to 0.08$	$B^{(k)}: 0.99 \to 0.54$ $R^{(k)}: 0.24 \to 0.11$	$B^{(k)}: 1.44 \to 1.14$ $R^{(k)}: 0.33 \to 0.20$	$B^{(k)}: 2.03 \to 1.79$ $R^{(k)}: 0.43 \to 0.32$	$B^{(0)}=2.62$ $R^{(k)}=0.52, 0.43$
	$C=0$	$B^{(k)}: 0.91 \to 0.49$ $R^{(k)}: 0.22 \to 0.10$	$B^{(k)}: 1.23 \to 0.70$ $R^{(k)}: 0.29 \to 0.14$	$B^{(0)}=1.85$ $R^{(k)}=0.40, 0.27$	$B^{(0)}=2.73$ $R^{(k)}=0.53, 0.39$	$B^{(0)}=3.66$ $R^{(k)}=0.65, 0.51$
$L=.05$	$C=.05$	$B^{(k)}: 1.03 \to 0.83$ $R^{(k)}: 0.23 \to 0.15$	$B^{(k)}: 1.30 \to 1.07$ $R^{(k)}: 0.28 \to 0.19$	$B^{(k)}: 1.82 \to 1.76$ $R^{(k)}: 0.37 \to 0.28$	$B^{(k)}: 2.50 \to 2.49$ $R^{(k)}: 0.47 \to 0.38$	$B^{(0)}=3.19$ $R^{(k)}=0.56, 0.50$
	$C=0$	$B^{(0)}=1.21$ $R^{(k)}=0.26, 0.18$	$B^{(0)}=1.57$ $R^{(k)}=0.33, 0.23$	$B^{(0)}=2.29$ $R^{(k)}=0.44, 0.33$	$B^{(0)}=3.31$ $R^{(k)}=0.58, 0.46$	$B^{(0)}=4.39$ $R^{(k)}=0.70, 0.58$

Table D.3: Error Bounds for $n = 3$, with different values of L, C, and ε_F or $\varepsilon_{\mathsf{Unif}}$

$n = 3$		$\varepsilon_F = \varepsilon_{\mathsf{Unif}} = 0$	$\varepsilon_F = \frac{1}{4}$	$\varepsilon_{\mathsf{Unif}} = 0.1$	$\varepsilon_F = 1$	$\varepsilon_{\mathsf{Unif}} = 0.2$
	$F_2^{\min}$	0.5000	0.4000	0.4050	0.2500	0.3200
	$F_2^{\max}$		0.6250	0.5950	1.0000	0.6800
	$F_3^{\min}$	0.1667	0.1333	0.1215	0.0833	0.0853
	$F_3^{\max}$		0.2083	0.2710	0.3333	0.3813
$L = 0$	$C = .05$	$B^{(0)} = 2.32 \to 0$ $R^{(k)} = 0.82 \to 0$	$B^{(0)} = 4.26$ $R^{(k)} = 1.29, 1.22$	$B^{(0)} = 3.66$ $R^{(k)} = 1.16, 1.06$	$B^{(0)} = 104.19$ $R^{(k)} = 9.26, 66.53$	$B^{(0)} = 5.70$ $R^{(k)} = 1.59, 1.58$
	$C = 0$	$B^{(0)} = 3.00 \to 0$ $R^{(k)} = 1.00 \to 0$	$B^{(0)} = 6.11$ $R^{(k)} = 1.67, 1.52$	$B^{(0)} = 5.10$ $R^{(k)} = 1.47, 1.31$	$B^{(0)} = \infty$ $R^{(0)} = \infty$	$B^{(0)} = 8.77$ $R^{(k)} = 2.13, 2.02$
$L = .02$	$C = .05$	$B^{(0)} = 2.59$ $R^{(k)} = 0.86, 0.78$	$B^{(0)} = 4.70$ $R^{(k)} = 1.34, 1.31$	$B^{(0)} = 4.05$ $R^{(k)} = 1.20, 1.15$	$B^{(0)} = 112.87$ $R^{(k)} = 9.46, 73.05$	$B^{(0)} = 6.25$ $R^{(k)} = 1.64, 1.68$
	$C = 0$	$B^{(0)} = 3.33$ $R^{(k)} = 1.04, 0.90$	$B^{(0)} = 6.70$ $R^{(k)} = 1.72, 1.62$	$B^{(0)} = 5.60$ $R^{(k)} = 1.52, 1.40$	$B^{(0)} = \infty$ $R^{(0)} = \infty$	$B^{(0)} = 9.57$ $R^{(k)} = 2.19, 2.13$
$L = .05$	$C = .05$	$B^{(0)} = 3.04$ $R^{(k)} = 0.91, 0.88$	$B^{(0)} = 5.40$ $R^{(k)} = 1.41, 1.44$	$B^{(0)} = 4.67$ $R^{(k)} = 1.27, 1.27$	$B^{(0)} = 126.86$ $R^{(k)} = 9.77, 83.82$	$B^{(0)} = 7.15$ $R^{(k)} = 1.72, 1.83$
	$C = 0$	$B^{(0)} = 3.86$ $R^{(k)} = 1.10, 1.01$	$B^{(0)} = 7.64$ $R^{(k)} = 1.80, 1.75$	$B^{(0)} = 6.41$ $R^{(k)} = 1.59, 1.53$	$B^{(0)} = \infty$ $R^{(0)} = \infty$	$B^{(0)} = 10.87$ $R^{(k)} = 2.28, 2.28$

the observations $\hat{p}_S$ and approximations F_S result in worst-case error. The error in our algorithm's estimates could be significantly less, and even in combinations where the error bounds are not decreasing, it could be the case that iterating decreases the error.

Finally, note that if we compare the tables to each other, the numbers are generally increasing from Table D.1 ($n = 5$) to Table D.3 ($n = 3$)—indeed, it can be seen that the error bounds, roughly, scale with $F_{n-1} = \frac{1}{(n-1)!}$. However, when $n = 5$, it takes more samples to achieve the same bound L on the input error. We discuss this in Section D.2.7.

D.2.7 Iterations, Samples, and Bounds on Other Quantities

In this section we discuss (i) the number of iterations, (ii) the number of samples, and (iii) bounds on the errors in our estimates of the three original probabilities of interest.

For (i), the rate of convergence to the asymptotic error bound of $B^{(\infty)}$ is that of exponential decay. Every iteration, the error bound is multiplied by $\tilde{\gamma}(z) < 1$, but then a fixed error of $\tilde{\alpha}(z)$ is accumulated due to inaccuracies in the observations $\hat{p}_S$ and approximations F_S. The number of iterations required to guarantee a desired error bound can be calculated directly from Theorem D.2.15:

Corollary D.2.17. *Suppose the conditions in Theorem D.2.15 are satisfied. Let $B' \in (B^{(\infty)}, B^{(0)}]$ be our error threshold. Then for $k \geq K$, $\frac{1}{1+B'} \leq \frac{1-a_i^{(k)}}{1-a_i^*} \leq 1 + B'$ for all $i \in [n]$, where*

$$K := \left\lceil \left(\ln \tilde{\gamma}(z)\right)^{-1} \left(\ln(B' - B^{(\infty)}) - \ln(B^{(0)} - B^{(\infty)})\right) \right\rceil.$$

Now we compute (ii), the number of samples needed to guarantee a specific L with high probability. Recall that L was a bound such that both $\frac{p_i^*}{\hat{p}_i} = \frac{p^*_{\{i\}}}{\hat{p}_{\{i\}}} \cdot \frac{\hat{p}_\emptyset}{p^*_\emptyset}$ and $\frac{p^*_{-i}}{\hat{p}_{-i}} = \frac{p^*_{[n]\setminus\{i\}}}{\hat{p}_{[n]\setminus\{i\}}}$ lie in $[\frac{1}{1+L}, 1 + L]$, for all $i \in [n]$. Expressions of the form $\frac{\hat{p}_S}{p^*_S}$ can be controlled with the multiplicative Chernoff bound:

Proposition D.2.18. *Suppose we have T independent samples of an event that occurs with probability p^*. Let $\hat{p}$ be the sample average approximation of p^* over the T samples. Then for all $\varepsilon \in (0, 1)$,*

$$\Pr[1 - \varepsilon \leq \frac{\hat{p}}{p^*} \leq 1 + \varepsilon] \geq 1 - 2e^{-\frac{\varepsilon^2 T p^*}{3}}.$$

To use Proposition D.2.18, we need a lower bound on the probability p^*. We base such a lower bound on the following definition, which will also be a useful benchmark in the numerical experiments:

$$p_{\min} := \min_{i \in [n]} \big(\min\{\Pr[x_i < P_i - d], \Pr[P_i - d \leq x_i < P_i], \Pr[P_i \leq x_i]\} \big). \tag{D.30}$$

$\Pr[x_i < P_i - d]$, $\Pr[P_i - d \leq x_i < P_i]$, and $\Pr[P_i \leq x_i]$, where x_i is a valuation drawn from distribution D_i, were the three probabilities that we were originally interested in estimating, for each $i \in [n]$. If these probabilities are small, then more samples are required achieve a specific L:

Theorem D.2.19. *Given an instance with distributions $D_1, \ldots, D_n$ and values of $P_1, \ldots, P_n, d$, define $p_{\min}$ as in (D.30). Let $L > 0$ be the desired bound on multiplicative error. Then for any $\delta > 0$, $\frac{p_i^*}{\hat{p}_i}, \frac{p_{-i}^*}{\hat{p}_{-i}} \in [\frac{1}{1+L}, 1+L]$ $\forall i \in [n]$ with probability at least $1-\delta$, so long as the number of samples T is at least*

$$\frac{3}{2}\Big(\ln \frac{4n+2}{\delta}\Big)\Big(\frac{2}{L}+1\Big)^2\Big(\frac{1-p_{\min}}{p_{\min}^{n+1}}\Big).$$

The sample bound is exponential in n, which is explained by the following intuition. As n increases, the fraction of customers for which the discount is relevant sharply decreases, and as a result, many more samples are required to see the impact of the discount. In general, we think of the bundles analyzed in this chapter as being small, motivated by fashion retailers and budget airlines.

Finally, we discuss (iii), i.e. how to translate, on the k'th iteration, $q^{(k)}$ and $a^{(k)}$ into estimates of $\Pr[x_i < P_i - d]$, $\Pr[P_i - d \leq x_i < P_i]$, and $\Pr[P_i \leq x_i]$ and obtain bounds on the errors in these estimates. First we bound $\frac{a_i^{(k)}}{a_i^*}$. It can be derived from $\frac{1}{1+B^{(k)}} \leq \frac{1-a_i^{(k)}}{1-a_i^*} \leq 1 + B^{(k)}$ that

$$1 - B^{(k)}\Big(\frac{1}{a_i^*} - 1\Big) \leq \frac{a_i^{(k)}}{a_i^*} \leq 1 + B^{(k)}\Big(\frac{1}{a_i^*} - 1\Big).$$

Now, recall from Definition 5.2.2 that $\Pr[x_i < P_i - d] = (1-q_i^*)(1-a_i^*)$, $\Pr[P_i - d \leq x_i < P_i] = (1-q_i^*)a_i^*$, and $\Pr[P_i \leq x_i] = q_i^*$. Our estimates of these probabilities are thus bounded as follows:

$$\begin{aligned}
\frac{1}{(1+R^{(k)})(1+B^{(k)})} \leq & \quad \frac{(1-q_i^{(k)})(1-a_i^{(k)})}{(1-q_i^*)(1-a_i^*)} \leq & (1+R^{(k)})(1+B^{(k)}) \\
\frac{1-B^{(k)}(\frac{1}{a_i^*}-1)}{1+R^{(k)}} \leq & \quad \frac{(1-q_i^{(k)})a_i^{(k)}}{(1-q_i^*)a_i^*} \leq & (1+R^{(k)})(1+B^{(k)}(\frac{1}{a_i^*}-1)) \\
\frac{1}{1+R^{(k)}} \leq & \quad \frac{q_i^{(k)}}{q_i^*} \leq & 1+R^{(k)}.
\end{aligned}$$

For each specific i, if $\frac{1}{a_i^*}$ is large, then our estimate $(1-q_i^{(k)})a_i^{(k)}$ of $\Pr[P_i - d \leq x_i < P_i]$ could be poor even if $B^{(k)}$ and $R^{(k)}$ are small. Furthermore, the error in the lower bound on $\frac{a_i^{(k)}}{a_i^*}$ is additive instead of multiplicative, so the lower bound could be negative and thus meaningless. Intuitively, small $a_i^* = \Pr[P_i - d \leq x_i | x_i < P_i]$ corresponds to the case where the bundle discount is irrelevant, so the firm is essentially conducting individual sales, and we can accurately estimate q_i^* and $1-q_i^*$ but not a_i^*. Nonetheless, if we are in a regime where $B^{(k)} \xrightarrow{k\to\infty} 0$, so long as $a_i^* \neq 0$, $B^{(k)}$ will eventually be much smaller than $\frac{1}{a_i^*} - 1$ (and the convergence of $B^{(k)}$ is not adversely affected by small a_i^*).

D.2.8 Supplement to Section D.2

Proof. Proof of Theorem D.2.9

First we prove the following propositions, which will aid us in the proof of Theorem D.2.9.

Proposition D.2.20. *Let $F, G_1, \ldots, G_m, H_1, \ldots, H_m$ are arbitrary constants, and consider any expression of the form $\dfrac{1-\prod_{i=1}^m (G_i a_i + H_i)}{1 - F\prod_{i=1}^m a_i}$ as a function of $(a_1, \ldots, a_m)$ over a polytope where both the numerator and denominator and positive. For any $i \in [m]$, the sign of the i'th partial derivative does not depend on a_i. In other words, the extreme values of the function are attained at the vertices of the feasible region.*

Proof. Proof. The i'th partial derivative is

$$\frac{-G_i \prod_{j\neq i}(G_j a_j + H_j)(1 - F\prod_{i=1}^m a_i) - (1 - \prod_{i=1}^m (G_i a_i + H_i))(-F\prod_{j\neq i} a_j)}{(1-F\prod_{i=1}^m a_i)^2}.$$

The coefficient of a_i in the numerator is

$$G_i\Big(\prod_{j\neq i}(G_j a_j + H_j)\Big)F\Big(\prod_{j\neq i} a_j\Big) - G_i\Big(\prod_{j\neq i}(G_j a_j + H_j)\Big)F\Big(\prod_{j\neq i} a_j\Big) = 0$$

and the denominator is always positive, completing the proof. □

Proposition D.2.21. *Let $\delta \geq 0$ be such that $0 \leq y - \delta \leq y \leq 1$. Then $y^m - (y-\delta)^m \leq my^{m-1}\delta$.*

Proof. Proof. The statement follows immediately from the fact that $f(y) := y^m$ is convex over $y \in [0,1]$ and my^{m-1} (the derivative) is a subgradient of f at y, i.e. $f(y-\delta) - f(y) \geq my^{m-1}((y-\delta) - y)$. □

Now we finally proceed to the main proof of Theorem D.2.9.

First let's analyze the maximum value of E. This expression is maximized when the a_i's are minimized, i.e. $a_i = \max\{a_i^* - B(1-a_i^*), 0\}$ for all $i \in [m]$. Now consider the problem of maximizing

$$\frac{1 - F\prod_{i=1}^m \max\{a_i^* - B(1-a_i^*), 0\}}{1 - F^* \prod_{i=1}^m a_i^*} \tag{D.31}$$

over $a_1^*, \dots, a_m^* \in [0, 1-C]$. If any a_i^* is small enough such that $a_i^* - B(1-a_i^*) < 0$, ie. $a_i^* < \frac{B}{1+B}$, then a_i^* can be increased to $\frac{B}{1+B}$, decreasing the denominator while leaving the numerator unchanged. Therefore, (D.31) is at most

$$\frac{1 - F\prod_{i=1}^m (a_i^* - B(1-a_i^*))}{1 - F^* \prod_{i=1}^m a_i^*} \tag{D.32}$$

where $a_1^*, \dots, a_m^* \in [\frac{B}{1+B}, 1-C]$ (note that $\frac{B}{1+B} \leq 1-C$ by assumption).

By Proposition D.2.20, (D.32) is maximized when each a_i^* is $\frac{B}{1+B}$ or $1-C$. Observe that if any $a_i^* = \frac{B}{1+B}$, then $\prod_{j=1}^m (a_j^* - B(1-a_j^*)) = 0$, hence the expression is maximized when $a_j^* = 1-C$ for all $j \neq i$. Therefore, the maximum value of (D.32) is the greater of

$$\frac{1}{1 - F^{\max}(\frac{B}{1+B})(1-C)^{m-1}} = \frac{1+B}{1+B-BF^{\max}(1-C)^{m-1}} \tag{D.33}$$

and

$$\frac{1 - F(1-C-BC)^m}{1 - F^{\max}(1-C)^m}. \tag{D.34}$$

where we have also made F^* as large as possible.

Putting together (D.33) and (D.34), we have

$$\begin{aligned}
E &\leq \max\{1+\frac{BF^{\max}(1-C)^{m-1}}{1+B-BF^{\max}(1-C)^{m-1}}, \\
&\qquad 1+\frac{F^{\max}(1-C)^m - F(1-C)^m}{1-F^{\max}(1-C)^m} + \frac{F(1-C)^m - F(1-C-BC)^m}{1-F^{\max}(1-C)^m}\} \\
&= 1+\max\{\frac{BF^{\max}(1-C)^{m-1}}{1+B(1-F^{\max}(1-C)^{m-1})}, \\
&\qquad \frac{(F^{\max}-F)(1-C)^m}{1-F^{\max}(1-C)^m} + \frac{F((1-C)^m-(1-C-BC)^m)}{1-F^{\max}(1-C)^m}\} \\
&\leq 1+\max\{\frac{BF^{\max}(1-C)^{m-1}}{1+z(1-F^{\max}(1-C)^{m-1})}, \\
&\qquad \frac{(F^{\max}-F)(1-C)^m}{1-F^{\max}(1-C)^m} + \frac{F(m(1-C)^{m-1})(BC)}{1-F^{\max}(1-C)^m}\}
\end{aligned}$$

where the final inequality follows from $z \in [0, B]$ and Proposition D.2.21 (note that $1-C-BC \geq 0$ since $\frac{B}{1+B} \leq 1-C$). This completes the proof of (D.14).

Now let's analyze the minimum value of E. E is minimized when the a_i's are maximized, i.e. $a_i = a_i^* + \frac{B}{1+B}(1-a_i^*)$ for all $i \in [m]$. Therefore, we can consider the problem of minimizing

$$\frac{1-F\prod_{i=1}^m (a_i^* + \frac{B}{1+B}(1-a_i^*))}{1-F^*\prod_{i=1}^m a_i^*}$$

over $a_1^*, \ldots, a_m^* \in [0, 1-C]$.

Again employing Proposition D.2.20, this expression is minimized when each a_i^* is 0 or $1-C$. If any $a_i^* = 0$, then $\prod_{j=1}^m a_j^* = 0$, hence the expression is minimized when $a_j^* = 1-C$ for all $j \neq i$. Therefore, the minimum value of this expression is the lesser of

$$1-F(\frac{B}{1+B})(1-C+\frac{B}{1+B}C)^{m-1} \tag{D.35}$$

and

$$\frac{1-F(1-C+\frac{B}{1+B}C)^m}{1-F^{\min}(1-C)^m}. \tag{D.36}$$

where we have also made F^* as small as possible.

Putting together (D.35) and (D.36), we have

$$\begin{aligned}
E^{-1} &\leq \max\{1 + \frac{F(\frac{B}{1+B})(1-C+\frac{B}{1+B}C)^{m-1}}{1-F(\frac{B}{1+B})(1-C+\frac{B}{1+B}C)^{m-1}}, \\
&\quad 1 + \frac{F(1-C)^m - F^{\min}(1-C)^m}{1-F(1-C+\frac{B}{1+B}C)^m} + \frac{F(1-C+\frac{B}{1+B}C)^m - F(1-C)^m}{1-F(1-C+\frac{B}{1+B}C)^m}\} \\
&= 1 + \max\{\frac{BF\frac{1}{1+B}(1-\frac{C}{1+B})^{m-1}}{1-F(\frac{B}{1+B})(1-C+\frac{B}{1+B}C)^{m-1}}, \\
&\quad \frac{(F-F^{\min})(1-C)^m}{1-F(1-C+\frac{B}{1+B}C)^m} + \frac{F((1-C+\frac{B}{1+B}C)^m - (1-C)^m)}{1-F(1-C+\frac{B}{1+B}C)^m}\} \\
&\leq 1 + \max\{\frac{BF\frac{1}{1+B}(1-\frac{C}{1+B})^{m-1}}{1-F(\frac{B}{1+B})(1-C+\frac{B}{1+B}C)^{m-1}}, \\
&\quad \frac{(F-F^{\min})(1-C)^m}{1-F(1-C+\frac{B}{1+B}C)^m} + \frac{F(m(1-C+\frac{B}{1+B}C)^{m-1})(\frac{B}{1+B}C)}{1-F(1-C+\frac{B}{1+B}C)^m}\} \\
&\leq 1 + \max\{\frac{BF\frac{1}{1+z}(1-\frac{C}{1+z})^{m-1}}{1-F(\frac{B}{1+B})(1-\frac{C}{1+B})^{m-1}}, \\
&\quad \frac{(F-F^{\min})(1-C)^m}{1-F(1-\frac{C}{1+B})^m} + \frac{F(m(1-\frac{C}{1+z})^{m-1})(\frac{B}{1+z}C)}{1-F(1-\frac{C}{1+B})^m}\}
\end{aligned}$$

where the second inequality uses Proposition D.2.21, while the final inequality uses $z \in [0, B]$ and Lemma D.2.11. This completes the proof of (D.15) and the proof of Theorem D.2.9. □

Proof. Proof of Lemma D.2.6. Fix an arbitrary $i \in [n]$. First we establish the lower bound. Since $F^*_{[n]} = \Pr[\sum_{j=1}^n (P_j - x_j) \leq d | P_j - d \leq x_j < P_j \ \forall j \in [n]]$ while $F^*_{[n]\setminus\{i\}} = \Pr[\sum_{j\neq i}(P_j - x_j) \leq d | P_j - d \leq x_j < P_j \ \forall j \neq i]$, $F^*_{[n]} \leq F^*_{[n]\setminus\{i\}}$. Furthermore, $A^* \leq A^*_i$, since $a_i < 1$. As a result, $\frac{1-F^*_{[n]}A^*}{1-F^*_{[n]\setminus\{i\}}A^*_i} \geq 1$, and we can see from (D.7) that $\frac{q^*_i}{q^{(0)}_i} \cdot \frac{1-q^{(0)}_i}{1-q^*_i} \geq \frac{p^*_i}{\hat{p}_i} \geq \frac{1}{1+L}$.

For the upper bound, the numerator of $\frac{1-F^*_{[n]}A^*}{1-F^*_{[n]\setminus\{i\}}A^*_i}$ is at most 1, while the denominator is at least $1 - F^{\max}_{n-1}(1-C)^{n-1}$. $\frac{p^*_i}{\hat{p}_i}$ is also at most $1+L$, completing the proof. □

Proof. Proof of Lemma D.2.7. If $\frac{q^*}{q} \leq 1$, we are done immediately since $R \geq 0$. Otherwise, if $\frac{q^*}{q} > 1$, then $\frac{1-q^*}{1-q} < 1$, so we can obtain from $\frac{q^*}{q} \cdot \frac{1-q}{1-q^*} \leq 1+R$ that $\frac{q^*}{q} \leq (1+R)\frac{1-q^*}{1-q} < 1+R$. We can use the same argument to show that $\frac{1-q}{1-q^*} \leq 1+R$ in either case. □

Proof. Proof of Lemma D.2.11. Let $y = \frac{1}{1+z}$, so that $y \in (0, 1]$. The derivative of $y(1-$

$Cy)^{m-1}$ with respect to y is

$$\begin{aligned}(1-Cy)^{m-1}+y(m-1)(1-Cy)^{m-2}(-C) &= (1-Cy)^{m-2}(1-Cy-Cy(m-1))\\ &= (1-Cy)^{m-2}(1-Cym)\end{aligned}$$

which is non-negative since $Cm \leq 1$ and $y \leq 1$. Thus the derivative with respect to z is non-positive. As a result, the expression is maximized when $z = 0$, completing the proof. □

Proof. Proof of Lemma D.2.12. Fix some $i \in [n]$. First let's apply Theorem D.2.9 with $m = n-1$, $\{a_1, \ldots, a_m\} = \{a_j^{(k)} : j \neq i\}$. Note that choosing $B = B^{(k)}, F = F_{n-1}, F^* = F^*_{[n]\setminus\{i\}}, F^{\min} = F^{\min}_{n-1}, F^{\max} = F^{\max}_{n-1}$ satisfies the conditions imposed on those constants. Finally, using the same C and z satisfies the remaining conditions of Theorem D.2.9 (note that $\frac{B^{(k)}}{1+B^{(k)}} \leq \frac{B^{(0)}}{1+B^{(0)}} \leq 1-C$ and $C(n-1) \leq Cn \leq 1$). Therefore, we can use (D.14) to obtain

$$\begin{aligned}\frac{1-F_{n-1}A_i^{(k)}}{1-F^*_{[n]\setminus\{i\}}A_i^*} &\leq 1+\max\{B^{(k)}\cdot\frac{F_m^{\max}(1-C)^{m-1}}{1+z(1-F_m^{\max}(1-C)^{m-1})}, \alpha_m + B^{(k)}\cdot\frac{mCF_m(1-C)^{m-1}}{1-F_m^{\max}(1-C)^m}\}\\ &= 1+\alpha_m+B^{(k)}\cdot\max\{\frac{F_m^{\max}(1-C)^{m-1}}{1+z(1-F_m^{\max}(1-C)^{m-1})}-\frac{\alpha_m}{B^{(k)}}, \frac{mCF_m(1-C)^{m-1}}{1-F_m^{\max}(1-C)^m}\}\\ &\leq 1+\alpha_m+\gamma_m(z)\cdot B^{(k)}\end{aligned}$$

where $m = n-1$ and the final inequality follows because $B^{(k)} \leq B^{(0)}$, completing the proof of (D.16).

Similarly, we can use (D.15) to obtain

$$\begin{aligned}&\frac{1-F^*_{[n]\setminus\{i\}}A_i^*}{1-F_{n-1}A_i^{(k)}}\\ \leq\;& 1+\max\{B^{(k)}\cdot\frac{F_m(\frac{1}{1+z})(1-\frac{C}{1+z})^{m-1}}{1-F_m(\frac{B^{(k)}}{1+B^{(k)}})(1-\frac{C}{1+B^{(k)}})^{m-1}},\\ &\qquad \overline{\alpha}_m\cdot\frac{1-F_m(1-\frac{C}{1+B^{(0)}})^m}{1-F_m(1-\frac{C}{1+B^{(k)}})^m}+B^{(k)}\cdot\frac{mCF_m(\frac{1}{1+z})(1-\frac{C}{1+z})^{m-1}}{1-F_m(1-\frac{C}{1+B^{(k)}})^m}\}\\ \leq\;& 1+\max\{B^{(k)}\cdot\frac{F_m(\frac{1}{1+z})(1-\frac{C}{1+z})^{m-1}}{1-F_m(\frac{B^{(0)}}{1+B^{(0)}})(1-\frac{C}{1+B^{(0)}})^{m-1}}, \overline{\alpha}_m+B^{(k)}\cdot\frac{mCF_m(\frac{1}{1+z})(1-\frac{C}{1+z})^{m-1}}{1-F_m(1-\frac{C}{1+B^{(0)}})^m}\}\\ =\;& 1+\overline{\alpha}_m+B^{(k)}\cdot\max\{\frac{F_m(\frac{1}{1+z})(1-\frac{C}{1+z})^{m-1}}{1-F_m(\frac{B^{(0)}}{1+B^{(0)}})(1-\frac{C}{1+B^{(0)}})^{m-1}}-\frac{\overline{\alpha}_m}{B^{(k)}}, \frac{mCF_m(\frac{1}{1+z})(1-\frac{C}{1+z})^{m-1}}{1-F_m(1-\frac{C}{1+B^{(0)}})^m}\}\\ \leq\;& 1+\overline{\alpha}_m+\overline{\gamma}_m(z)\cdot B^{(k)}\end{aligned}$$

where $m = n-1$ and both the second and final inequalities follow because $B^{(k)} \leq B^{(0)}$, completing the proof of (D.18).

(D.17) and (D.19) follow by applying Theorem D.2.9 with the following changes: $m = n$, $a_i = a_i^{(k)}$ for all $i \in [n]$, and $F = F_n, F^* = F^*_{[n]}, F^{\min} = F_n^{\min}, F^{\max} = F_n^{\max}$. This yields $\frac{1 - F_n A^{(k)}}{1 - F^*_{[n]} A^*} \leq 1 + \alpha_m + \gamma_m(z) \cdot B^{(k)}$ and $\frac{1 - F^*_{[n]} A^*}{1 - F_n A^{(k)}} \leq 1 + \overline{\alpha}_m + \overline{\gamma}_m(z) \cdot B^{(k)}$ where $m = n$, completing the proof of Lemma D.2.12. $\square$

Proof. Proof of Lemma D.2.14. It is immediate that $\frac{\prod_{i=1}^m (1+a_i) - 1}{\sum_{i=1}^m a_i} \geq 1$, since $\prod_{i=1}^m (1 + a_i) \geq 1 + \sum_{i=1}^m a_i$ when $a_1, \ldots, a_m \geq 0$. To prove that $\frac{\prod_{i=1}^m (1+a_i) - 1}{\sum_{i=1}^m a_i}$ is weakly increasing, fix some $i \in [m]$ and the i'th partial derivative is $\frac{(\prod_{j \neq i}(1+a_j)) \sum_{j=1}^m a_j - (\prod_{j=1}^m (1+a_j) - 1)}{(\sum_{j=1}^m a_j)^2}$. The numerator can be rewritten as

$$\begin{aligned}
(\prod_{j \neq i}(1+a_j))(\sum_{j=1}^m a_j - (1+a_i)) + 1 &= 1 - (1 - \sum_{j \neq i} a_j) \prod_{j \neq i}(1+a_j) \\
&\geq 1 - \exp(-\sum_{j \neq i} a_j) \prod_{j \neq i}(1+a_j) \\
&\geq 1 - \exp(-\sum_{j \neq i} a_j) \prod_{j \neq i} \exp(a_j) \\
&= 0
\end{aligned}$$

where both inequalities use the fact that $1 + y \leq e^y$ for all $y \in \mathbb{R}$, the first inequality uses the fact that all $a_j \geq 0$, and the second inequality uses the fact that $\exp(-\sum_{j \neq i} a_j) \geq 0$. $\square$

Proof. Proof of Theorem D.2.15. We know that $\frac{1}{1+B^{(k)}} \leq \frac{1-a_i^{(k)}}{1-a_i^*} \leq 1 + B^{(k)}$ for all $i \in [n]$ when $k = 0$, from (D.11). Now proceed inductively and assume that we have established $\frac{1}{1+B^{(k)}} \leq \frac{1-a_i^{(k)}}{1-a_i^*} \leq 1 + B^{(k)}$ for all $i \in [n]$, for some $k \geq 0$. Under the conditions of Theorem D.2.15, Section D.2.4 shows that $\frac{1}{1+(\tilde{\alpha}(z)+\tilde{\gamma}(z)B^{(k)})} \leq \frac{1-a_i^{(k+1)}}{1-a_i^*} \leq 1 + (\tilde{\alpha}(z) + \tilde{\gamma}(z)B^{(k)})$ for all $i \in [n]$. But by the definitions of $B^{(k)}$ and $B^{(\infty)}$,

$$\begin{aligned}
\tilde{\alpha}(z) + \tilde{\gamma}(z)B^{(k)} &= \tilde{\alpha}(z) + \tilde{\gamma}(z)(B^{(\infty)} + \tilde{\gamma}(z)^k (B^{(0)} - B^{(\infty)})) \\
&= (\tilde{\alpha}(z) + \tilde{\gamma}(z)\frac{\tilde{\alpha}(z)}{1 - \tilde{\gamma}(z)}) + \tilde{\gamma}(z)^{k+1}(B^{(0)} - B^{(\infty)}) \\
&= (\frac{\tilde{\alpha}(z)}{1 - \tilde{\gamma}(z)}) + \tilde{\gamma}(z)^{k+1}(B^{(0)} - B^{(\infty)})
\end{aligned}$$

which is precisely the definition of $B^{(k+1)}$, completing the proof. □

Proof. Proof of Theorem D.2.19. Set $\varepsilon = \frac{L}{2+L}$, which is less than 1, so Proposition D.2.18 applies. Let $\mathcal{S} := \{S \subset [n] : |S| \in \{0, 1, n-1\}\}$. It is easy to see that if $1 - \varepsilon \leq \frac{\hat{p}_S}{p^*_S} \leq 1 + \varepsilon$ for all $S \in \mathcal{S}$, then $\frac{p^*_{-i}}{\hat{p}_{-i}} \leq 1 + L$ and $\frac{p^*_{-i}}{\hat{p}_{-i}} \geq \frac{1}{1+L}$. We can further calculate that $\frac{p^*_i}{\hat{p}_i} \leq \frac{1+\varepsilon}{1-\varepsilon} = L$ (by the definition of ε) and $\frac{p^*_i}{\hat{p}_i} \geq \frac{1-\varepsilon}{1+\varepsilon} > 1 - L > \frac{1}{1+L}$. Therefore, it suffices to find a δ such that with probability at least $1 - \delta$, $1 - \varepsilon \leq \frac{\hat{p}_S}{p^*_S} \leq 1 + \varepsilon$ for all $S \in \mathcal{S}$.

By Proposition D.2.18, the probability that any $\frac{\hat{p}_S}{p^*_S}$ is outside of $[1-\varepsilon, 1+\varepsilon]$ is at most $2\exp(-\frac{\varepsilon^2 T p^*_S}{3})$. If we choose T large enough such that this expression is at most $\frac{\delta}{2n+1}$, then we can union bound over the $2n+1$ sets in $\mathcal{S}$ to achieve the desired result. We precisely need T to be at least $3\ln(\frac{4n+2}{\delta})\varepsilon^{-2}\frac{1}{p^*_S}$ for all $S \in \mathcal{S}$. But from (5.3) and the definition of $p_{\min}$, every p^*_S is at least $p_{\min}^{n-1}(2p_{\min})\frac{p_{\min}}{1-p_{\min}}$ (realized when S is of the form $[n] \setminus \{i\}$), completing the proof of Theorem D.2.19. □

D.2.9 Supplement to Section D.2.6

Computing $F_m^{\min}$ and $F_m^{\max}$ given $\varepsilon_{\mathsf{Unif}}$

We can without loss generality re-normalize d to 1 and consider the interval $[0,1]$ instead of the interval $[P_i - d, P_i]$. We have independent random variables $x_1, \ldots, x_m$ on $[0,1]$ each satisfying $f(y) \geq 1 - \varepsilon_{\mathsf{Unif}}$ for all $y \in [0,1]$, and our goal is to find the minimum and maximum values for $\Pr[x_1 + \ldots + x_m \leq 1]$.

$\Pr[x_1 + \ldots + x_m \leq 1]$ is minimized when each random variable has a point mass of size $1 - \varepsilon_{\mathsf{Unif}}$ at 1, and the remaining $1 - \varepsilon_{\mathsf{Unif}}$ mass distributed uniformly over $[0,1]$. In this case, $x_1 + \ldots + x_m$ only has a chance to not exceed 1 if no x_i takes value 1, which occurs with probability $(1 - \varepsilon_{\mathsf{Unif}})^m$, by independence. Conditioned on this occurring, the probability that $\Pr[x_1 + \ldots + x_m \leq 1]$ is $\frac{1}{m!}$, by Lemma 5.2.6. Therefore, we can set $F_m^{\min}$ to $(1 - \varepsilon_{\mathsf{Unif}})^m \cdot \frac{1}{m!}$.

$\Pr[x_1 + \ldots + x_m \leq 1]$ is maximized when each random variable has a point mass of size $1 - \varepsilon_{\mathsf{Unif}}$ at 0 (and the remaining $1 - \varepsilon_{\mathsf{Unif}}$ mass distributed uniformly over $[0,1]$). To compute $\Pr[x_1 + \ldots + x_m \leq 1]$, we condition on the number of random variables taking on value 0. Let Y denote this random variable, which is binomially distributed, consisting of m trials of probability $\varepsilon_{\mathsf{Unif}}$. Conditioned on $Y = \ell$, $\Pr[x_1 + \ldots + x_m \leq 1] = \frac{1}{(m-\ell)!}$, by

Lemma 5.2.6. Therefore,

$$\Pr[x_1 + \ldots + x_m \leq 1] = \sum_{\ell=0}^{m} \binom{m}{\ell} \varepsilon_{\mathsf{Unif}}^{\ell} (1 - \varepsilon_{\mathsf{Unif}})^{1-\ell} \cdot \frac{1}{(m-\ell)!},$$

so we can set $F_m^{\max}$ to this value.

Pre-iteration for Improved Error Bounds

We note the following about the computation of the numbers in Tables D.1–D.3:

- For the combinations in Table D.2 with arrows, we used the pre-iteration technique below, with 3 iterations, to obtain improved bounds.
- For the combinations in Table D.3 with arrows, we used the pre-iteration technique below, with 73 iterations, to obtain improved bounds.
- In the numbers displayed in Tables D.1–D.3, C never had to be redefined according to Definition D.2.8.
- For the combinations in Table D.3, we displayed the improved error bounds when $F_{[n]}^* = F_n$ (see next subsubsection).

We explain how to obtain better error bounds by doing pre-iteration. The observation is that in the analysis in Section D.2, the initial error bound $B^{(0)}$ is fixed given values of L, C, and $F^{\max}$. However, $\gamma_m(z), \overline{\alpha}_m, \overline{\gamma}_m(z)$ (defined in Definition D.2.10) and $H(z), \overline{H}(z)$ (defined in (D.20)–(D.21)) are all increasing in $B^{(0)}$. Therefore, if we could "restart" the analysis after each iteration k, where we set the new $B^{(0)}$ to be $B^{(k)}$ (which is less than $B^{(0)}$), the induction step in Section D.2.4 would still be applicable with decreased values of $\tilde{\alpha}(z)$ and $\tilde{\gamma}(z)$.

Therefore, we can improve the procedure from Section D.2.5 as follows. We have an initial $B^{(0)}$, and then we can do one iteration of the induction step in Section D.2.4 with $z = B^{(0)}$ to obtain $B^{(1)}$. If $B^{(1)} < B^{(0)}$, then we do another iteration of the induction step where we set the new $B^{(0)}$ (and the new z) to be the old $B^{(1)}$. We can repeat this process indefinitely to get the smallest error bound, but after some number of iterations we stop and apply Theorem D.2.15 (with $z = \tilde{B}$) to get the closed-form error bound. Of course, in Tables D.1–D.3, the values of $B^{(0)}$ and $R^{(0)}$ displayed are still the values before any iterations of the induction step—but the final values of $B^{(\infty)}$ and $R^{(\infty)}$ are smaller as a result of this pre-iteration.

Improved Error Bounds when $F^*_{[n]} = F_n$

(D.9) can be rewritten as

$$\frac{q_i^*}{q_i^{(k+1)}} \cdot \frac{1-q_i^{(k+1)}}{1-q_i^*} = \frac{p_i^*}{\hat{p}_i} \cdot \frac{1-F^*_{[n]}A^*}{1-F^*_{[n]\setminus\{i\}}A_i^*} \cdot \frac{1-F_{n-1}A_i^{(k)}}{1-\frac{1}{n}F_{n-1}a_i^{(k)}A_i^{(k)}}.$$

Consider the third fraction on the RHS of the expression. It is equal to

$$1-F_{n-1}A_i^{(k)} \cdot \frac{1-\frac{1}{n}a_i^{(k)}}{1-\frac{1}{n}F_{n-1}a_i^{(k)}A_i^{(k)}}.$$

Now, since $a_i^{(k)} < 1$, $1-\frac{1}{n}a_i^{(k)} < 1$. Therefore, the preceding expression is maximized when $A_i^{(k)}$ is minimized. For all $j \neq i$, let $\underline{a}_j^{(k)} := \min\{a_j^{(k)}, a_j^*\}$. Recalling that $A_i^{(k)} = \prod_{j\neq i} a_j^{(k)}$, we have

$$\begin{aligned}\frac{q_i^*}{q_i^{(k+1)}} \cdot \frac{1-q_i^{(k+1)}}{1-q_i^*} &\leq \frac{p_i^*}{\hat{p}_i} \cdot \frac{1-F_{n-1}\prod_{j\neq i}\underline{a}_j^{(k)}}{1-F^*_{[n]\setminus\{i\}}A_i^*} \cdot \frac{1-F^*_{[n]}A^*}{1-\frac{1}{n}F_{n-1}a_i^{(k)}\prod_{j\neq i}\underline{a}_j^{(k)}} \\ &\leq \frac{p_i^*}{\hat{p}_i} \cdot \frac{1-F_{n-1}\prod_{j\neq i}\underline{a}_j^{(k)}}{1-F^*_{[n]\setminus\{i\}}A_i^*} \cdot \frac{1-F^*_{[n]}A_i^*a_i^*}{1-F_nA_i^*a_i^{(k)}}.\end{aligned}$$

Applying a similar analysis to the reciprocal of (D.9) where $\overline{a}_j^{(k)} := \max\{a_j^{(k)}, a_j^*\}$, we also have

$$\frac{1-q_i^*}{1-q_i^{(k+1)}} \cdot \frac{q_i^{(k+1)}}{q_i^*} \leq \frac{\hat{p}_i}{p_i^*} \cdot \frac{1-F^*_{[n]\setminus\{i\}}A_i^*}{1-F_{n-1}\prod_{j\neq i}\overline{a}_j^{(k)}} \cdot \frac{1-F_nA_i^*a_i^{(k)}}{1-F^*_{[n]}A_i^*a_i^*}.$$

In the case where $F^*_{[n]} = F_n$, we analyze the expressions $\frac{1-F^*_{[n]}A_i^*a_i^*}{1-F_nA_i^*a_i^{(k)}}$ and $\frac{1-F_nA_i^*a_i^{(k)}}{1-F^*_{[n]}A_i^*a_i^*}$ without using Theorem D.2.9.

$$\begin{aligned}\frac{1-F_nA_i^*a_i^*}{1-F_nA_i^*a_i^{(k)}} &\leq \frac{1-F_nA_i^*a_i^*}{1-F_nA_i^*(a_i^*+\frac{B^{(k)}}{1+B^{(k)}}(1-a_i^*))} \\ &\leq \frac{1}{1-F_nA_i^*(\frac{B^{(k)}}{1+B^{(k)}})}\end{aligned} \tag{D.37}$$

where for the second inequality it can be checked (via taking the derivative) that the RHS of (D.37) is a decreasing function of a_i^* on $[0, 1-C]$.

$$\begin{aligned} \frac{1 - F_n A_i^* a_i^{(k)}}{1 - F_n A_i^* a_i^*} &\leq \frac{1 - F_n A_i^*(a_i^* - B^{(k)}(1 - a_i^*))}{1 - F_n A_i^* a_i^*} \\ &\leq \frac{1}{1 - F_n A_i^*(\frac{B^{(k)}}{1+B^{(k)}})} \end{aligned} \tag{D.38}$$

where for the second inequality it can be checked (via taking the derivative) that the RHS of (D.38) is a decreasing function of a_i^* on $[\frac{B^{(k)}}{1+B^{(k)}}, 1-C]$.

In both cases, we would like to bound

$$\begin{aligned} \frac{1}{1 - F_n A_i^*(\frac{B^{(k)}}{1+B^{(k)}})} &= 1 + B^{(k)} \cdot \frac{F_n(1-C)^{n-1}}{1 + B^{(k)} - F_n(1-C)^{n-1}B^{(k)}} \\ &\leq 1 + B^{(k)} \cdot \frac{F_n(1-C)^{n-1}}{1 + z(1 - F_n(1-C)^{n-1})} \end{aligned}$$

where the inequality follows from the fact that $z \in [0, B^{(k)}]$. Therefore, (D.17) and (D.19) hold with

$$\gamma_n(z) = \overline{\gamma}_n(z) = \frac{F_n(1-C)^{n-1}}{1 + z(1 - F_n(1-C)^{n-1})}. \tag{D.39}$$

Note that $\alpha_n = \overline{\alpha}_n = 0$ when $F_n^{\min} = F_n = F_n^{\max}$.

D.3 Testing our Algorithm on Synthetic Data

In this section we numerically test our algorithm on synthetically-generated data instances, whose true parameters are known.

D.3.1 Instance Generation

For consistency, we follow the set-up from Chu et al. (2011) as closely as possible. We let n be 3, 4, or 5. We use the same two-parameter families of valuation distributions commonly used to model demand—Uniform, Normal, Gumbel, and Lognormal. The ranges of parameters we use for these families, whose justification we leave to Chu et al. (2011), are disclosed in Table D.4.

We make some modifications to allow for heterogeneous variances (since otherwise our estimation problem would be trivial). Also, we need to generate prices $P_1, \ldots, P_n$ and

Table D.4: Ranges of Parameters, derived from Chu et al. (2011).

Gumbel	Marginal distributions are Gumbel, with locations μ chosen uniformly from $[0, 2.5]$, and scales σ chosen uniformly from $[0.5, 1.5]$.
Lognormal	Marginal distributions are Lognormal. Logarithms of valuations are Normally distributed, with means μ chosen uniformly from $[-1.5, 1]$, and standard deviations σ chosen uniformly from $[0.5, 1.5]$.
Normal	Marginal distributions are Normal, with means μ chosen uniformly from $[0, 2.5]$, and standard deviations σ chosen uniformly from $[0.5, 1.5]$.
Uniform	Marginal distributions are Uniform on $[a - b, a + b]$, where mean a is chosen uniformly from $[0, 2.5]$, and b is chosen uniformly from $[1, 3]$.

discount d from which we will learn the valuations.

For each item $i \in [n]$, its individual price P_i is chosen uniformly from $[1.25, 2.5]$. Then the bundle discount d is chosen uniformly from $[0.25, \min_{i=1}^{n}\{P_i\}]$. Note that we choose $P_1, \ldots, P_n, d$ independent of any of the underlying distributions, which ensures that the prices alone do not provide any information about the valuations. However, this results in instances where the probabilities of interest ($\Pr[x_i < P_i - d], \Pr[P_i - d \leq x_i < P_i], \Pr[P_i \leq x_i]$) can be zero. Therefore, we restrict to instances where $p_{\min}$, defined in (D.30), is at least 0.05. Additional numerical experiments with the weaker restriction $p_{\min} \geq 0.01$ can be found in Section D.3.5. Of course, the algorithms we test operate independent of these restrictions, and also do not know the ranges of parameters from Table D.4.

For each combination of n (out of 3 choices) and family of distributions (out of 4 choices), we define an *instance* to consist of n independent valuation distributions $D_1, \ldots, D_n$ from that family, n individual prices $P_1, \ldots, P_n$, and a discount d. This induces true values for the three probabilities of interest for each item i, which we denote by $r_{i,1}, r_{i,2}, r_{i,3}$ for brevity:

1. $r_{i,1} := \Pr_{x_i \sim D_i}[x_i < P_i - d]$
2. $r_{i,2} := \Pr_{x_i \sim D_i}[P_i - d \leq x_i < P_i]$
3. $r_{i,3} := \Pr_{x_i \sim D_i}[P_i \leq x_i]$

We let T denote the total number of customers, who have IID valuations drawn from $D_1, \ldots, D_n$, and vary T over $\{10^3, 10^4, 10^5, 10^6, 10^7\}$. For each $S \subseteq [n]$, an algorithm observes N_S, the number of customers who chose to purchase subset S under prices $P_1, \ldots, P_n$ and discount d. An algorithm's objective is to output an estimate $\tilde{r}_{i,\ell}$ of $r_{i,\ell}$, for all $i \in [n]$ and $\ell \in [3]$.

Each algorithm is evaluated on its Mean Average Percentage Error (MAPE), which is defined

$$\text{MAPE} = \frac{1}{3n}\sum_{i=1}^{n}\sum_{\ell=1}^{3}\frac{|\tilde{r}_{i,\ell} - r_{i,\ell}|}{r_{i,\ell}}.$$

For small errors, MAPE is very close to the errors bounded in Section D.2.7. We use MAPE because it is typically used and easily interpretable.

D.3.2 Algorithm Variants Compared

In this section we specify the algorithm variants we compare. By *algorithm variant* we mean an exact procedure for converting input $P_1, \ldots, P_n, d, \{N_S : S \subseteq [n]\}$ into output $\{\tilde{r}_{i,1}, \tilde{r}_{i,2}, \tilde{r}_{i,3} : i \in [n]\}$.

We specify the parameters we use for our algorithm. We set $F_S = \frac{1}{S!}$ (following Lemma 5.2.6) and $\hat{p}_S = \frac{N_S+1}{T+2^n}$ (following Section D.2.1) for all $S \subseteq [n]$, which also ensures that there is no division by zero in the algorithm.

Regarding the number of iterations, we use the following stopping criterion: on every iteration $k \geq 1$, the algorithm checks whether

$$\frac{1}{n}\sum_{i=1}^{n}|a_i^{(k)} - a_i^{(k-1)}| < 0.001. \tag{D.40}$$

If so, it sets $K := k$ and returns the estimates $\tilde{r}_{i,1} = (1-q_i^{(K)})(1-a_i^{(K)})$, $\tilde{r}_{i,2} = (1-q_i^{(K)})a_i^{(K)}$, $\tilde{r}_{i,3} = q_i^{(K)}$.

Regarding choice of $\mathcal{S}_i$, we test two variants of our algorithm which differ in their choices (one makes the collection $\mathcal{S}_i$ as small as possible; the other makes $\mathcal{S}_i$ as large as possible):

1. LfB-One: Set $\mathcal{S}_i = \{\emptyset\}$ for all $i \in [n]$.

2. LfB-All: Set $\mathcal{S}_i = \{S : S \subsetneq ([n] \setminus \{i\})\}$ for all $i \in [n]$.

Also, we compare two algorithms which receive different input.

First, we test how our algorithm is affected if we do not observe the lost sales $N_\emptyset$ (and thus do not know T). In this case, we estimate $\hat{p}_\emptyset$ as follows:

$$\hat{p}_\emptyset = \left(1 - \frac{1}{n!}\right)\prod_{i=1}^{n}\left(1 + \left(\prod_{j\neq i}\frac{N_{\{i,j\}}+1}{N_{\{j\}}+1}\right)^{\frac{1}{n-1}}\right)^{-1}. \tag{D.41}$$

The motivation for (D.41) is as follows. We know from (5.3) that $p^*_\emptyset = (1-\frac{1}{n!})\prod_{i=1}^n(1-q^*_i)$. Therefore, we can combine estimates of $q^*_1,\dots,q^*_n$ to estimate $p^*_\emptyset$. For each i and $S \subsetneq [n]\setminus\{i\}$, we know that $\frac{N_{S\cup\{i\}}+1}{N_S+1}$ is an estimate of $\frac{q^*_i}{1-q^*_i}$, whose bias is smaller for S of smaller cardinality. However, we cannot use $S=\emptyset$, since we don't observe $N_\emptyset$. Therefore, we take the geometric mean for all S of cardinality 1, other than $\{i\}$. Knowing that $\big(\prod_{j\neq i}\frac{N_{\{i,j\}}+1}{N_{\{j\}}+1}\big)^{\frac{1}{n-1}} = \frac{q^*_i}{1-q^*_i}$, solving for $1-q^*_i$ results in (D.41). Afterward using (D.41), we solve for the unique T which ensures that the probabilities $\hat{p}_S$ sum to 1:

3. LfB-Blind: Set $\hat{p}_\emptyset$ according to (D.41), followed by $T = \frac{1-\hat{p}_\emptyset}{\sum_{S\neq\emptyset}(N_S+1)}$, followed by $\hat{p}_S = \frac{N_S+1}{T}$ for $S\neq\emptyset$. For all $i\in[n]$, use $\mathcal{S}_i = \{S : \emptyset \subsetneq S \subsetneq ([n]\setminus\{i\})\}$ (note that we exclude $\emptyset$ from $\mathcal{S}_i$ since any ratios involving $\hat{p}_\emptyset$ may be less accurate).

Secondly, we test an algorithm which observes individual sales at prices $P_1,\dots,P_n$ with no bundling. Such observations allow for accurate estimates of the $r_{i,2}$'s, but the algorithm must guess the breakdown between $r_{i,0}$ and $r_{i,1}$:

4. NoBndl: Set $\tilde{r}_{i,2} = \frac{N_i}{T}$ for all $i\in[n]$, where $N_i := \sum_{S\ni i} N_S$ is the number of customers who bought item i. Afterward, set $\tilde{r}_{i,0} = \tilde{r}_{i,1} = \frac{1-\tilde{r}_{i,2}}{2}$.

D.3.3 Results

For every combination of n $(3,4,5)$ and demand family (Uniform, Normal, Gumbel, Lognormal), we fix 1000 instances, each consisting of n distributions from that family and experimental prices $P_1,\dots,P_n$ and discount d, all generated randomly and independently in accordance to Section D.3.1. For every T $(10^3,\dots,10^7)$, we compute the median MAPE's (over the 1000 instances) in the estimates returned by each of 8 algorithm variants (defined in Section D.3.2). In Table D.5, we display the average of these median MAPE's over the 4 demand families; we display the families separately in Section D.3.5.

We draw the following conclusions from Table D.5:

- The MAPE is quite high when T is small, because the sales observations are just too noisy. The MAPE quickly drops below 10% as T increases. It converges to errors as small as 1%, despite not knowing the demand family.

- LfB-One, the variant we analyzed in Section D.2, is indeed better than LfB-All in the asymptotic regime. However, LfB-All performs better when $T \leq 10^5$.

Table D.5: Median MAPE's over the 1000 instances, averaged across the 4 demand families. The algorithm variant achieving the smallest MAPE within each rectangle is **bolded**.

n	3					4					5				
T	10^3	10^4	10^5	10^6	10^7	10^3	10^4	10^5	10^6	10^7	10^3	10^4	10^5	10^6	10^7
LfB-One	.249	.083	**.031**	**.016**	**.012**	.351	.129	.041	**.016**	**.008**	.484	.228	**.075**	**.024**	**.010**
LfB-All	**.209**	**.072**	**.031**	.019	.017	**.312**	**.114**	**.040**	.017	.011	.462	**.211**	**.075**	.026	.014
LfB-Blind	.270	.173	.158	.158	.159	.345	.139	.058	.034	.029	.476	.223	.079	.029	.016
NoBndl	.413	.402	.400	.401	.400	.433	.421	.418	.418	.418	**.454**	.443	.440	.439	.439

Table D.6: Average numbers of iterations taken over 20000 experiments (1000 instances for each of 4 demand families and 5 values of T).

n	3	4	5
LfB-One	3.46	2.01	1.32
LfB-All	7.39	6.92	6.40

- For a fixed T, all algorithms perform worse for larger n. Indeed, even though the sales rates of more subsets are observed, these observations are far less accurate. However, for larger n, the disadvantage of LfB-Blind from not observing $N_\emptyset$ is much smaller.

- The option of not bundling does not perform well, because it only obtains one piece of information for each item. However, when the bundle observations are at their sparsest ($T = 10^3, n = 5$), guessing using it is better than over-fitting to bundle sales.

We briefly discuss the number of iterations taken by our algorithm, displayed in Table D.6. The number of iterations decreases with n because the initial estimates improve with n, as implied by the bound in Lemma D.2.6. LfB-All takes more iterations than LfB-One, but nonetheless, the number of iterations taken by our algorithm to satisfy (D.40) is very small and never a concern.

Finally, we further investigate the surprisingly small asymptotic error when the demand family is not Uniform, despite our algorithm performing computations as if demand was Uniform. To isolate this error from the error caused by inaccurate input, we consider larger T, and only LfB-One (the best algorithm variant in the large-T regime). It can be seen from Table D.7 that for the non-Uniform families:

- For $T \leq 10^6$, the MAPE is greatest when $n = 5$. In these cases, the main contributor to the MAPE is inaccurate input, a factor most significant when $n = 5$.

- For $T \geq 10^7$, the MAPE is greatest when $n = 3$. In these cases, the main contributor to the MAPE is non-Uniform demand, a factor most significant when $n = 3$.

Nonetheless, for all values of T, there is a very small increase in MAPE from moving from Uniform to non-Uniform demand, especially when the non-Uniform demand is Gumbel. We provide an explanation of why the error is so small for any *unimodal* family of demand in Section D.3.4.

Table D.7: Median MAPE's over the instances for the LfB-One variant. The largest MAPE within each rectangle is **bolded**.

T	10^4			10^5			10^6			10^7			10^8		
n	3	4	5	3	4	5	3	4	5	3	4	5	3	4	5
Uniform	.077	.126	**.193**	.025	.036	**.061**	.009	.014	**.021**	.005	.006	**.008**	.004	.005	**.006**
Normal	.081	.132	**.237**	.031	.043	**.076**	.016	.015	**.025**	**.012**	.008	.010	**.012**	.007	.007
Gumbel	.075	.111	**.201**	.025	.036	**.066**	.013	.013	**.021**	**.008**	.007	**.008**	**.007**	.005	.005
Lognormal	.100	.148	**.282**	.044	.050	**.096**	.025	.021	**.030**	**.023**	.011	.012	**.012**	.009	.007

D.3.4 Unimodality and Uniformity

We sketch why the error caused by our algorithm performing calculations as if the conditional distributions on $[P_i - d, P_i)$ were uniform is usually insignificant.

First, consider the following example. There are 3 IID Normal valuations with means 1 and standard deviations 1. Let the prices be 2.5 for each item, with a global discount of 2 if all items are purchased. The critical region is $[0.5, 2.5]$, on which a $N(1, 1)$ random variable has much more mass on the lower end than the upper end. Therefore, the approximation in Lemma 5.2.6 is poor, overestimating the valuations in the critical region and the demand overall. Indeed, in this example, the true values of $r_{i,1}, r_{i,2}, r_{i,3}$ for each item are $0.308, 0.625, 0.067$ (respectively), while the estimated values approach $0.265, 0.663, 0.072$ (respectively). Note that $p_{\min} = 0.067$, just above the minimum of 0.05. The MAPE is 8.8%, translating to a maximum additive error of 4.3% in this example.

We argue that this is roughly the most inaccurate the approximation in Lemma 5.2.6 can be. Indeed, suppose the true mass on the critical region was more lopsided than in this example, e.g., it is zero on the upper half of the critical region. However, for any *unimodal* distribution, this implies that the probability $r_{i,3}$ of the valuation lying above the critical region is zero! This is a detectable failure where the experimental price was set too high. If the distribution was not unimodal, then it could theoretically be the case that both $\Pr[P_i - d \leq x_i < P_i - \frac{d}{2}]$ and $\Pr[P_i \leq x_i]$ are large while $\Pr[P_i - \frac{d}{2} \leq x_i < P_i]$ is zero, in which case our conclusion would be highly flawed despite the sampled probabilities being accurate.

However, for unimodal distributions, if it were the case that the mass of $r_{i,2}$ on $[P_i - d, P_i)$ was highly non-uniform, then it must be the case that either $r_{i,1}$ and $r_{i,3}$ is very small, in which case there would be high sampling error in the input probabilities. Therefore, the main contributor to the error is never due to non-uniformity.

D.3.5 Additional Reports

See Tables D.8–D.12.

Table D.8: Median MAPE's over the 1000 instances, averaged across the 4 demand families, with $p_{\min} \geq 0.01$ (instead of $p_{\min} \geq 0.05$).

n	3					4					5				
T	10^3	10^4	10^5	10^6	10^7	10^3	10^4	10^5	10^6	10^7	10^3	10^4	10^5	10^6	10^7
LfB-One	.287	.111	.046	.023	.016	.420	.182	.058	.021	.011	.536	.295	.126	.041	.014
LfB-All	.255	.106	.050	.029	.024	.484	.197	.063	.029	.018	.713	.328	.141	.053	.026
LfB-Blind	.334	.244	.229	.227	.229	.537	.230	.133	.098	.088	.749	.357	.150	.063	.032
NoBndl	.632	.619	.616	.615	.615	.612	.596	.593	.591	.591	.621	.615	.613	.611	.611

Table D.9: Median MAPE's over the 1000 instances with Uniform valuations.

n	3					4					5				
T	10^3	10^4	10^5	10^6	10^7	10^3	10^4	10^5	10^6	10^7	10^3	10^4	10^5	10^6	10^7
LfB-One	.222	.077	.025	.009	.005	.335	.126	.036	.014	.006	.421	.193	.061	.021	.008
LfB-All	.170	.059	.020	.008	.005	.291	.095	.033	.013	.007	.361	.171	.055	.018	.007
LfB-Blind	.225	.156	.141	.140	.142	.322	.107	.044	.026	.022	.369	.183	.057	.018	.008
NoBndl	.360	.351	.347	.349	.348	.412	.404	.402	.402	.402	.378	.370	.368	.367	.367

Table D.10: Median MAPE's over the 1000 instances with Normal valuations.

n	3					4					5				
T	10^3	10^4	10^5	10^6	10^7	10^3	10^4	10^5	10^6	10^7	10^3	10^4	10^5	10^6	10^7
LfB-One	.279	.081	.031	.016	.012	.357	.132	.043	.015	.008	.518	.237	.076	.025	.010
LfB-All	.220	.071	.031	.019	.018	.317	.116	.042	.019	.013	.537	.228	.085	.029	.015
LfB-Blind	.296	.214	.197	.207	.203	.354	.144	.069	.046	.045	.543	.244	.088	.031	.019
NoBndl	.381	.369	.368	.367	.367	.394	.381	.377	.377	.377	.402	.392	.388	.388	.388

Table D.11: Median MAPE's over the 1000 instances with Gumbel valuations.

n	3					4					5				
T	10^3	10^4	10^5	10^6	10^7	10^3	10^4	10^5	10^6	10^7	10^3	10^4	10^5	10^6	10^7
LfB-One	.222	.075	.025	.013	.008	.323	.111	.036	.013	.007	.476	.201	.066	.021	.008
LfB-All	.195	.064	.023	.014	.011	.248	.098	.031	.014	.009	.391	.168	.054	.019	.009
LfB-Blind	.219	.127	.106	.106	.108	.271	.103	.038	.017	.013	.400	.174	.059	.019	.010
NoBndl	.396	.389	.387	.388	.388	.418	.411	.407	.407	.407	.437	.432	.429	.428	.428

Table D.12: Median MAPE's over the 1000 instances with Lognormal valuations.

n	3					4					5				
T	10^3	10^4	10^5	10^6	10^7	10^3	10^4	10^5	10^6	10^7	10^3	10^4	10^5	10^6	10^7
LfB-One	.271	.100	.044	.025	.023	.387	.148	.050	.021	.011	.520	.282	.096	.030	.012
LfB-All	.251	.092	.049	.034	.033	.391	.148	.054	.022	.015	.557	.278	.106	.039	.023
LfB-Blind	.341	.195	.187	.179	.184	.432	.203	.082	.046	.037	.592	.290	.110	.046	.027
NoBndl	.514	.499	.498	.498	.498	.506	.488	.486	.486	.486	.598	.576	.575	.574	.574

D.4 Generalizations

D.4.1 Two-part Tariffs via Complementarity

In this section we show that our learning results also hold for the Two-Part Tariff (TPT) pricing scheme. Recall that in the pricing scheme analyzed throughout the rest of this chapter, which we will refer to as Mixed Bundling (MB) in this section, each item $i \in [n]$ is sold individually at price P_i, and a discount of $d > 0$ is applied if all of the items are purchased. In TPT, there is a tariff $t > 0$ to enter the market, after which each item $i \in [n]$ may be purchased for its individual price P_i.

The following theorem shows that the information contained in TPT sales data is equivalent to the information contained in MB sales data.

Theorem D.4.1. *Let there be n items with (potentially correlated) valuations distributed according to random variable $x = (x_1, \ldots, x_n)$. Given a TPT scheme with individual prices $P_1, \ldots, P_n$ and tariff $t > 0$, let p_S^{TPT} denote the (true) probability that subset S is selected by a customer, for all $S \subseteq [n]$.*

Now, consider a disjoint n-item instance with valuations distributed according to $y = (2P_1 - x_1, \ldots, 2P_n - x_n)$. Given a MB scheme with individual prices $P_1, \ldots, P_n$ and discount $d = t$, let p_S^{MB} denote the probability that subset S is selected by a customer, for all $S \subseteq [n]$.

Then for all $S \subseteq [n]$, $p_S^{\mathsf{MB}} = p_{[n]\setminus S}^{\mathsf{TPT}}$.

Proof. Proof of Theorem D.4.1. We saw from Lemma 5.2.3 that under the original pricing scheme, the customer buys everything if $\sum_{i=1}^{n} \max\{P_i - y_i, 0\} \leq d$, and otherwise buys only the items for which $y_i \geq P_i$. Therefore, $p_{[n]}^{\mathsf{MB}}$ is the probability that $\sum_{i=1}^{n} \max\{P_i - y_i, 0\} \leq d$. Since y_i is distributed according to $2P_i - x_i$ for all $i \in [n]$ and $d = t$, this is equal to the probability that $\sum_{i=1}^{n} \max\{x_i - P_i, 0\} \leq t$. However, it is easy to see that under TPT, the customer buys nothing if $\sum_{i=1}^{n} \max\{x_i - P_i, 0\} \leq t$, and otherwise pays the tariff to buy the items for which $x_i > P_i$ (here we have differentiated between strict and non-strict inequalities in a way to facilitate the proof, but these details are unimportant, since in general we model valuations as continuous and break ties arbitrarily). Therefore, the previous probability is equal to $p_{\emptyset}^{\mathsf{TPT}}$, as desired.

For $S \subsetneq [n]$, p_S^{MB} is the probability that $\sum_{i=1}^{n} \max\{P_i - y_i, 0\} > d$ and $\bigcap_{i \in S}(y_i \geq P_i)$ and $\bigcap_{i \notin S}(y_i < P_i)$. Doing the same substitution, this is equal to the probability that

$\sum_{i=1}^{n} \max\{x_i - P_i, 0\} > t$ and $\bigcap_{i\in S}(x_i \leq P_i)$ and $\bigcap_{i\notin S}(x_i > P_i)$. Since $S \neq [n]$ implies $[n] \setminus S \neq \emptyset$, this is equal to $p^{\mathsf{TPT}}_{[n]\setminus S}$, completing the proof. $\square$

Therefore, given a TPT with prices $P_1, \dots, P_n, t$ and empirical probabilities $\hat{p}^{\mathsf{TPT}}_S$ of each subset S being selected, one can use our MB learning algorithm with individual prices equal to $P_1, \dots, P_n$, discount equal to t, and *complement* selection probabilities $\hat{p}^{\mathsf{MB}}_S = \hat{p}^{\mathsf{TPT}}_{[n]\setminus S}$, to infer n independent distributions. Let these distributions be represented by random variables $y_1, \dots, y_n$. Then the original distributions are represented by the random variables $2P_1 - y_i, \dots, 2P_n - y_n$.

D.4.2 Deducing the Number of No-purchases using Bundle Sales

Throughout the analytical sections of this chapter, we have assumed that the number of no-purchases was observed, so that the empirical probability $\hat{p}_S$ for each subset $S \subseteq [n]$ could be computed. However, suppose instead that we could only count N_S, the number of transactions that contained subset S, for $S \neq \emptyset$.

In Section 5.2.4, when we tested on synthetic data instances, we considered a version of our algorithm which does not observe $N_\emptyset$. We tested a method based on estimating the value of $N_\emptyset$ and thus being able to compute the empirical probabilities. The method performed well numerically, although it was worse for $n = 3$, when there were fewer subsets that could be used to estimate $N_\emptyset$.

In this section, we propose an alternate method which bypasses need to estimate $N_\emptyset$, and instead directly solves the theoretical equations without $\emptyset$, motivated by the following. Observe that in instructions (5.6) and (5.8) of our algorithm in Fig. 5-4, $q_i^{(k)}$ is updated using the *relative* value $\frac{\hat{p}_{S\cup\{i\}}}{\hat{p}_S}$, which is equal to $\frac{N_{S\cup\{i\}}}{N_S}$, i.e. it can still be computed so long as $\emptyset \notin S_i$. Knowing $N_\emptyset$ is only important for updating $a_i^{(k)}$, where we needed the *absolute* value of probability $\hat{p}_{[n]\setminus\{i\}}$ for instruction (5.7). We now propose an alternative for updating $a_i^{(k)}$ from the $q_i^{(k)}$'s, using only known *relative* sales counts. We used this method for computing the numbers in the example in Fig. 5-3.

Recall from Lemma 5.2.4 that:

$$p^*_{\{i\}} = q^*_i \prod_{j\neq i}(1-q^*_j)(1-\frac{1}{(n-1)!}\prod_{j\neq i} a^*_j), \qquad i \in [n];$$

$$\sum_{S\neq\emptyset} p^*_S = 1-\prod_{i=1}^{n}(1-q^*_i)(1-\frac{1}{n!}\prod_{i=1}^{n} a^*_i).$$

For each i, we can divide the former equation by the latter to obtain

$$\frac{N_{\{i\}}}{\sum_{S\neq\emptyset} N_S} = \frac{q^*_i \prod_{j\neq i}(1-q^*_j)(1-\frac{1}{(n-1)!}\prod_{j\neq i} a^*_j)}{1-\prod_{i=1}^{n}(1-q^*_i)(1-\frac{1}{n!}\prod_{i=1}^{n} a^*_i)}, \qquad i \in [n] \tag{D.42}$$

where we have replaced the LHS by an expression in *known* relative sales counts. Recall that our goal is to solve for $a^*_1, \ldots, a^*_n$ using (D.42) where the values of q^*_i are replaced with the algorithm's current values of $q^{(k)}_i$ on iteration k. Therefore, we can treat (D.42) as a system of equations in n variables $a^*_1, \ldots, a^*_n$, with all of the other expressions being known. For brevity, we will let N_i denote $\frac{N_{\{i\}}}{\sum_{S\neq\emptyset} N_S}$, ignore all superscripts, and solve for $a_1, \ldots, a_n$ in the following system:

$$N_i = \frac{q_i \prod_{j\neq i}(1-q_j)(1-\frac{1}{(n-1)!}\prod_{j\neq i} a_j)}{1-\prod_{i=1}^{n}(1-q^*_i)(1-\frac{1}{n!}\prod_{i=1}^{n} a_i)}, \qquad i \in [n]. \tag{D.43}$$

We let A denote $\prod_{i=1}^{n} a_i$ and isolate the $\prod_{j\neq i} a_j$ term in (D.43) for each i:

$$\prod_{j\neq i} a_j = (n-1)!\left(1-\frac{N_i(1-\prod_{i=1}^{n}(1-q^*_i)(1-\frac{1}{n!}A))}{q_i\prod_{j\neq i}(1-q_j)}\right), \qquad i \in [n].$$

Multiplying together for all $i \in [n]$, we obtain

$$A^{n-1} = ((n-1)!)^n \prod_{i=1}^{n}\left(1-\frac{N_i(1-\prod_{i=1}^{n}(1-q^*_i)(1-\frac{1}{n!}A))}{q_i\prod_{j\neq i}(1-q_j)}\right).$$

This is a degree-n polynomial in A. We solve for A numerically, and then it is easy to recover each a_i via dividing A by $\prod_{j\neq i} a_j$.

This allows us to iteratively use $\{q^{(k)}_i : i \in [n]\}$ to derive $\{a^{(k)}_i : i \in [n]\}$, and use $\{a^{(k)}_i : i \in [n]\}$ to derive $\{q^{(k+1)}_i : i \in [n]\}$, for $k = 0, 1, \ldots$, even when no-purchases are not observed.

D.4.3 $n = 2$ case via Explicit Calculations

In this section we perform further calculations, solving the $n = 2$ case using brute-force (as opposed to our iterative algorithm from Section 5.2).

Our objective is to solve the following system of equations (it may be useful to refer to the diagram in Fig. 5-1) in variables T (the total number of customers including no-purchases), q_1, q_2, a_1, and a_2, where N_{12}, N_1, N_2 are known, while $N_\emptyset$ may or may not be known:

$$\begin{aligned}
N_{12} &= Tq_1q_2 + Tq_1(1-q_2)a_2 + T(1-q_1)a_1q_2 + T(1-q_1)(1-q_2)\frac{a_1a_2}{2} \\
N_1 &= Tq_1(1-q_2)(1-a_2) \\
N_2 &= T(1-q_1)(1-a_1)q_2 \\
N_\emptyset &= T(1-q_1)(1-q_2)(1-\frac{a_1a_2}{2})
\end{aligned}$$

Of course, this system for $n = 2$ is underdetermined, because there are 5 variables (and either 3 or 4 equations, depending on whether $N_\emptyset$ is known).

Nonetheless, it is easy to see how to solve the system assuming either $N_\emptyset$ and one of q_1, q_2, a_1, a_2 are known, or two of q_1, q_2, a_1, a_2 are known—any single variable can be isolated from one of the equations; in the end we may have to solve a quadratic formula due to the $\frac{a_1a_2}{2}$ term. As a concrete example, suppose we had $N_{12} = 7, N_1 = 4, N_2 = 2, N_\emptyset = 11$, and knew that $q_2 = \frac{1}{2}$. Summing the four equations, we get that $T = N_{12} + N_1 + N_2 + N_\emptyset = 24$, and the following equations:

$$\frac{4}{24} = q_1(1-\frac{1}{4})(1-a_2) \quad \Longrightarrow \quad \frac{2}{9} = q_1(1-a_2) \tag{D.44}$$

$$\frac{2}{24} = (1-q_1)(1-a_1)\cdot\frac{1}{4} \quad \Longrightarrow \quad \frac{1}{3} = (1-q_1)(1-a_1) \tag{D.45}$$

$$\frac{11}{24} = (1-q_1)(1-\frac{1}{4})(1-\frac{a_1a_2}{2}) \quad \Longrightarrow \quad \frac{11}{18} = (1-q_1)(1-\frac{a_1a_2}{2}) \tag{D.46}$$

We can use the rightmost equation in (D.44) to write $a_2 = 1 - \frac{2}{9q_1} = \frac{9q_1-2}{9q_1}$. Similarly, we can use (D.45) to write $a_1 = 1 - \frac{1}{3-3q_1} = \frac{2-3q_1}{3-3q_1}$. Substituting into (D.46), we obtain

$$\frac{11}{18} = (1-q_1)(1 - \frac{(9q_1-2)(2-3q_1)}{2\cdot 9q_1(3-3q_1)})$$
$$\frac{11}{18} = (1-q_1)(\frac{54q_1 - 54q_1^2 - 18q_1 + 27q_1^2 + 4 - 6q_1}{54q_1(1-q_1)})$$
$$27q_1^2 + 3q_1 - 4 = 0$$

The final quadratic yields $q_1 = \frac{1}{3}$ (since the other solution of $q_1 = -\frac{4}{9}$ is inadmissible), leading to $a_2 = \frac{1}{3}$ and $a_1 = \frac{1}{2}$, completing our example of solving the system.

D.5 Testing our Model on Real Data

In this section we provide the details on how we processed the statistics in Definition 5.3.1.

First we describe the raw data given to us by the online retailer. In this section, we will use i to refer to items and j to refer to bundles. For each i and j, we will let m_{ij} denote the number of copies of item i contained in bundle j, which is a non-negative integer.

For each i and j, we were given its sales transactions (time and price) for 26 weeks starting June 1st, 2016. Note that a sales transaction a for bundle j containing i does not lead to a separate transaction for item i. The last of these 26 weeks is Black Friday week, where we observed a sharp decline in prices. Hereinafter we will refer to the first 25 weeks as "Period 1", and the final week as "Period 2".

Most of the items under consideration were staple home and kitchen items, and the bundles containing them were consistently available. Therefore, the stream of sales was relatively stable before Black Friday. To account for the occasional items and bundles that would incur no sales transactions for an extended period of time, we used the following rule. If an item or bundle had 0 transactions in a given week, then we discounted that week in computing its average sales per week, assuming that it was either unavailable or stocked out.

After this pre-processing, we arrived at the following quantities for each item i and bundle j:

- P_i, P_i': average price for the transactions of item i, during periods $1, 2$, respectively;
- P_j, P_j': average price for the transactions of bundle j, during periods $1, 2$, respectively;

- N_i, N'_i: number of transactions for item i, during periods $1, 2$, respectively;
- N_j, N'_j: number of transactions for bundle j, during periods $1, 2$, respectively.

We now progressively derive the quantities in Table D.13, all based on P_i, P_j, N_i, and N_j. This is the knowledge we have before Black Friday, and we will later relate them to the sales during Black Friday week.

$\mathsf{AvgBundDisc}(i)$, $\mathsf{PctBund}(i)$, and $\mathsf{PctBundPartners}(i)$ are the extant quantities used in Definition 5.3.1. We consider their relationship with the price elasticity $\mathsf{PriceElas}(i)$ of an item i, which we define as

$$\mathsf{PriceElas}(i) = \Big(\frac{N'_i}{N_i/25} - 1\Big) \Big/ \Big(1 - \frac{P'_i}{P_i}\Big). \tag{D.47}$$

Finally, for the graphs displayed in Section 5.3, we eliminated data points according to the following rules.

1. We removed the rare items whose individual sales were negligible either before or after Black Friday. Specifically, we only kept items with $N_i \geq 25$ and $N'_i \geq 10$.

2. We also removed the items that were not offered in bundles. We only kept items with $\mathsf{NumSoldInBund}(i) \geq 25$ and $\mathsf{AvgBundDisc}(i) \geq .025$, where the latter constraint guarantees that the bundles containing item i provided a substantial incentive for buying them.

3. We removed items that were not marked down on Black Friday. Specifically, we only kept items with $P'_i \leq .95 \cdot P_i$.

4. Finally, we removed items whose *bundles* were also heavily marked down during Black Friday, which would muddle their individual price elasticities which we calculated according to equation D.47. That is, the price elasticity is most clear for examples like Table 5.1, where the Black Friday sales of the bundles was 0 because it was not marked down (while its constituent items were marked down, dominating the original bundle discount). Therefore, we only kept items with $N'_i \geq 10 \cdot \sum_j m_{ij} N'_j$, which sold 10 times more copies individually than in bundles during Black Friday week.

Table D.13: The progressive derivation of quantities, which leads to the statistics in Definition 5.3.1

Quantity	Definition	Description
$\mathsf{BundDisc}(j)$	$1 - \frac{P_j}{\sum_i m_{ij} P_i}$	The percentage by which bundle j is discounted, as compared to the sum of the constituent prices. $\mathsf{BundDisc}(j) \in (0,1)$.
$\mathsf{NumSoldInBund}(i)$	$\sum_j m_{ij} N_j$	The total number of copies of item i sold in bundles.
$\mathsf{AvgBundDisc}(i)$	$\frac{\sum_j m_{ij} N_j \mathsf{BundDisc}(j)}{\mathsf{NumSoldInBund}(i)}$	The average percentage by which item i was discounted over all of the times it was sold in a bundle.
$\mathsf{PctBund}(i)$	$\frac{\mathsf{NumSoldInBund}(i)}{N_i + \mathsf{NumSoldInBund}(i)}$	The fraction of all copies of item i sold which came from bundles (as opposed to individually).
$\mathsf{Partners}(i)$	$\bigcup_{j:m_{ij}>0} \{i' : m_{i'j} > 0, i' \neq i\}$	All of the items i' that are bundled with item i, which we call *partners*. (Items i' which were present in less than 10% of the times i was sold in a bundle were discarded.)
$\mathsf{PctBundPartners}(i)$	$\frac{\sum_{i' \in \mathsf{Partners}(i)} \mathsf{PctBund}(i')}{\lvert \mathsf{Partners}(i) \rvert}$	A simple average of the tendency for partners of item i to be sold in bundles (as opposed to individually).

Appendix E

Appendix to Chapter 6

E.1 Proofs from Section 6.2

Proof. Proof of Proposition 6.2.2. By the definition of PBD, the customer can purchase any non-empty subset S of items for the price of $P_0^{\mathsf{PBD}} - \sum_{i \notin S} P_i^{\mathsf{PBD}}$. Of course, the customer can also choose not to make a purchase. Altogether, the class of menus represented by PBD is

$$\Big\{\{(\mathbb{1}_S, P_0^{\mathsf{PBD}} - \sum_{i \notin S} P_i^{\mathsf{PBD}}) : S \neq \emptyset\} \cup \{(0,0)\} : \\ P_1^{\mathsf{PBD}} \geq 0, \dots, P_n^{\mathsf{PBD}} \geq 0, P_0^{\mathsf{PBD}} \geq P_1^{\mathsf{PBD}} + \ldots + P_n^{\mathsf{PBD}}\Big\} \tag{E.1}$$

where $\mathbb{1}_S \in \{0,1\}^n$ is the indicator vector for items belonging to S.

Now, note that (6.2) defines a valid menu within PCUD since for all i, $P_i^{\mathsf{PCUD}} = P_i^{\mathsf{PBD}} +$

$P_0^{\mathsf{PCUD}} \geq P_0^{\mathsf{PCUD}} = P_0^{\mathsf{PBD}} - \sum_{j=1}^n P_j^{\mathsf{PBD}} \geq 0$. The class of menus represented by (6.2) is

$$\begin{aligned}
&\Big\{\{(\mathbb{1}_S, \sum_{i\in S} P_i^{\mathsf{PCUD}} - (|S|-1)P_0^{\mathsf{PCUD}}) : S \neq \emptyset\} \cup \{(0,0)\} : \\
&\qquad P_1^{\mathsf{PBD}} \geq 0, \ldots, P_n^{\mathsf{PBD}} \geq 0, P_0^{\mathsf{PBD}} \geq P_1^{\mathsf{PBD}} + \ldots + P_n^{\mathsf{PBD}}\Big\} \\
= \; &\Big\{\{(\mathbb{1}_S, \sum_{i\in S} P_i^{\mathsf{PBD}} + (P_0^{\mathsf{PBD}} - \sum_{i=1}^n P_i^{\mathsf{PBD}})) : S \neq \emptyset\} \cup \{(0,0)\} : \\
&\qquad P_1^{\mathsf{PBD}} \geq 0, \ldots, P_n^{\mathsf{PBD}} \geq 0, P_0^{\mathsf{PBD}} \geq P_1^{\mathsf{PBD}} + \ldots + P_n^{\mathsf{PBD}}\Big\} \\
= \; &\Big\{\{(\mathbb{1}_S, P_0^{\mathsf{PBD}} - \sum_{i\notin S} P_i^{\mathsf{PBD}}) : S \neq \emptyset\} \cup \{(0,0)\} : \\
&\qquad P_1^{\mathsf{PBD}} \geq 0, \ldots, P_n^{\mathsf{PBD}} \geq 0, P_0^{\mathsf{PBD}} \geq P_1^{\mathsf{PBD}} + \ldots + P_n^{\mathsf{PBD}}\Big\}
\end{aligned}$$

which is identical to (E.1). Furthermore, it is easy to see that the relation defined by (6.2) is a bijection between (E.1) and

$$\Big\{\{(\mathbb{1}_S, \sum_{i\in S} P_i^{\mathsf{PCUD}} - (|S|-1)P_0^{\mathsf{PCUD}}) : S \neq \emptyset\} \cup \{(0,0)\} : P_i^{\mathsf{PCUD}} \geq P_0^{\mathsf{PCUD}} \geq 0 \ \forall i \in [n]\Big\}.$$

Similarly, note that (6.3) defines a valid menu within TP since $P_0^{\mathsf{TP}} = P_0^{\mathsf{PBD}} - \sum_{i=1}^n P_i^{\mathsf{PBD}} \geq 0$, and for all i, $P_i^{\mathsf{TP}} = P_i^{\mathsf{PBD}} \geq 0$. The class of menus represented by (6.3) is

$$\begin{aligned}
&\Big\{\{(\mathbb{1}_S, P_0^{\mathsf{TP}} + \sum_{i\in S} P_i^{\mathsf{TP}}) : S \neq \emptyset\} \cup \{(0,0)\} : \\
&\qquad P_1^{\mathsf{PBD}} \geq 0, \ldots, P_n^{\mathsf{PBD}} \geq 0, P_0^{\mathsf{PBD}} \geq P_1^{\mathsf{PBD}} + \ldots + P_n^{\mathsf{PBD}}\Big\} \\
= \; &\Big\{\{(\mathbb{1}_S, (P_0^{\mathsf{PBD}} - \sum_{i=1}^n P_i^{\mathsf{PBD}}) + \sum_{i\in S} P_i^{\mathsf{PBD}}) : S \neq \emptyset\} \cup \{(0,0)\} : \\
&\qquad P_1^{\mathsf{PBD}} \geq 0, \ldots, P_n^{\mathsf{PBD}} \geq 0, P_0^{\mathsf{PBD}} \geq P_1^{\mathsf{PBD}} + \ldots + P_n^{\mathsf{PBD}}\Big\} \\
= \; &\Big\{\{(\mathbb{1}_S, P_0^{\mathsf{PBD}} - \sum_{i\notin S} P_i^{\mathsf{PBD}}) : S \neq \emptyset\} \cup \{(0,0)\} : \\
&\qquad P_1^{\mathsf{PBD}} \geq 0, \ldots, P_n^{\mathsf{PBD}} \geq 0, P_0^{\mathsf{PBD}} \geq P_1^{\mathsf{PBD}} + \ldots + P_n^{\mathsf{PBD}}\Big\}
\end{aligned}$$

which is identical to (E.1). Furthermore, it is easy to see that the relation defined by (6.3) is a bijection between (E.1) and

$$\Big\{\{(\mathbb{1}_S, P_0^{\mathsf{TP}} + \sum_{i\in S} P_i^{\mathsf{TP}}) : S \neq \emptyset\} \cup \{(0,0)\} : P_0^{\mathsf{TP}} \geq 0, P_1^{\mathsf{TP}} \geq 0, \ldots, P_n^{\mathsf{TP}} \geq 0\Big\}.$$

This completes the proof of Proposition 6.2.2. □

Proof. Proof of Proposition 6.2.4. Consider any valuation vector $x \in \mathbb{R}^n$. First suppose the customer bought the bundle with all the items for P^{PB}. Under the PBD menu, the customer will still buy the bundle, since it is non-negative utility even if she keeps all the items. However, she will choose to return any items i with $x_i < c_i$. Let S denote the set of such items, which is possibly empty.

- The producer surplus under PB is $P^{\mathsf{PB}} - \sum_{i=1}^{n} c_i$. The producer surplus under PBD is $(P^{\mathsf{PB}} - \sum_{i\in S} c_i) - \sum_{i\notin S} c_i$, which is identical.
- The consumer surplus under PB is $\sum_{i=1}^{n} x_i - P^{\mathsf{PB}}$. The consumer surplus under PBD is $\sum_{i\notin S} x_i - (P^{\mathsf{PB}} - \sum_{i\in S} c_i) = \sum_{i=1}^{n} \max\{x_i, c_i\} - P^{\mathsf{PB}}$ which can only be greater than the consumer surplus under PB.
- The deadweight loss is 0 in both cases: under PB every item is transferred, whereas under PBD every item valued above cost is still transferred.
- The overinclusion loss under PB is $\sum_{i\in S}(c_i - x_i) \geq 0$. The overinclusion loss under PBD is 0, since items in S are not transferred.

On the other hand, suppose the customer did not buy the bundle with all the items for P^{PB}.

- The producer surplus under PB is 0. The producer surplus under PBD is either 0 or $P^{\mathsf{PB}} - \sum_{i=1}^{n} c_i$ (if the return option allowed the customer to enter the market), which is non-negative.
- The consumer surplus under PB is 0. The consumer surplus under PBD cannot be negative, since the customer is rational and the no-purchase option is always available.
- The deadweight loss under PB is $\sum_{i:x_i>c_i}(x_i - c_i)$, which is the maximum possible. Therefore, the deadweight loss under PBD cannot be greater.

- The overinclusion loss under PB is 0. The overinclusion loss under PBD is always 0 when $P_i^{\mathsf{PBD}} = c_i$ for all i, since items valued below cost are never transferred.

In both cases, we have proven that the statements in Proposition 6.2.4 hold. $\square$

Proof. Proof of Proposition 6.2.5. The firm's problem is to find a menu along with tie-breaking rules which maximize profit. Note that this is equivalent to finding functions q, s defined on $\mathcal{X}$ which are incentive-compatible, individually rational, and profit-maximizing. Formally, the firm's problem is

$$\begin{array}{lrcll} \max & \multicolumn{4}{l}{\mathbb{E}_{x\sim D}[s(x) - q(x)^T c]} \\ s.t. & q(x)^T x - s(x) & \geq & q(y)^T x - s(y) & \forall x, y \in \mathcal{X} \\ & q(x)^T x - s(x) & \geq & 0 & \forall x \in \mathcal{X} \end{array}$$

which can be rewritten as

$$\begin{array}{lrcll} \max & \multicolumn{4}{l}{\mathbb{E}_{x\sim D}[s(x) - q(x)^T c]} \\ s.t. & q(x)^T(x - c) - (s(x) - q(x)^T c) & \geq & q(y)^T(x - c) - (s(y) - q(y)^T c) & \forall x, y \in \mathcal{X} \\ & q(x)^T(x - c) - (s(x) - q(x)^T c) & \geq & 0 & \forall x \in \mathcal{X} \end{array}$$

Now, define $x' := x - c$, $y' := y - c$, $q'(x) := q(x + c)$, and $s'(x) := s(x + c) - q(x + c)^T c$. Let $\mathcal{X}' := \{x - c : x \in \mathcal{X}\}$, and similarly let D' be the distribution D shifted c_i units downward in dimension i for every $i \in [n]$. We can see that the above is equivalent to

$$\begin{array}{lrcll} \max & \multicolumn{4}{l}{\mathbb{E}_{x'\sim D'}[s'(x')]} \\ s.t. & q'(x')^T x' - s'(x') & \geq & q'(y')^T x' - s'(y') & \forall x', y' \in \mathcal{X}' \\ & q'(x')^T x' - s'(x') & \geq & 0 & \forall x' \in \mathcal{X}' \end{array}$$

which is identical to the original problem without costs on this new distribution D'.

Now suppose there was a restriction on the menu $\mathcal{M} = \{(q^{(1)}, s^{(1)}), (q^{(2)}, s^{(2)}), \ldots\}$ to belong to some class $\mathscr{M}$ in the original problem. The menu after the transformation, $\mathcal{M}'$, looks like $\{(q^{(1)}, s^{(1)} - (q^{(1)})^T c), (q^{(2)}, s^{(2)} - (q^{(2)})^T c), \ldots\}$. Therefore, $\mathcal{M}'$ is restricted to the class

$$\mathscr{M}' := \{\{(q^{(1)}, s^{(1)} - (q^{(1)})^T c), (q^{(2)}, s^{(2)} - (q^{(2)})^T c), \ldots\} : \{(q^{(1)}, s^{(1)}), (q^{(2)}, s^{(2)}), \ldots\} \in \mathscr{M}\}.$$

By assumption that $s - q^T c \geq 0$ for all menu entries, the payments in $\mathcal{M}'$ are non-negative.

$\square$

Throughout this chapter, it will be clear whether we are in the context of the original problem or the transformed problem, and we will omit the superscripts used in the preceding proof.

Remark E.1.1. As a concrete example of this transformation, consider the pricing scheme PBDC. $\mathcal{M}$ is restricted to be of the form $\{(\mathbb{1}_S, P_0^{\mathsf{PBD}} - \mathbb{1}_{[n]\setminus S}^T c) : \emptyset \neq S \subseteq [n]\} \cup \{(0,0)\}$ where $\mathbb{1}_S \in \{0,1\}^n$ is the indicator vector for items belonging to S. Hence $\mathcal{M}'$ is restricted to be of the form

$$\begin{aligned}&\{(\mathbb{1}_S, P_0^{\mathsf{PBD}} - \mathbb{1}_{[n]\setminus S}^T c - \mathbb{1}_S^T c) : \emptyset \neq S \subseteq [n]\} \cup \{(0,0)\}\\ &= \{(\mathbb{1}_S, P_0^{\mathsf{PBD}} - \mathbb{1}_{[n]}^T c) : \emptyset \neq S \subseteq [n]\} \cup \{(0,0)\}.\end{aligned}$$

Put in words, $\mathcal{M}'$ must belong to the class of menus that offer the same price for any non-empty subset of items. The fact that the customer can choose to take a subset of items instead of taking all the items is important, because valuations x'_i can be negative (x'_i is equal to the original valuation x_i subtract the cost c_i).

E.2 Proof of Theorem 6.4.1

We will WOLOG normalize the valuations so that the optimal PC revenue is 1 (we can do this so long as the original optimal revenue was positive; if it was 0 then the statement of the theorem is trivial).

E.2.1 The Core-Tail Decomposition

We use the core-tail decomposition of Babaioff et al. (2014), with the original idea coming from Li and Yuan (2013). We will cut up the domain of the joint distribution and consider the conditional distributions on the smaller subdomains. Below, we introduce the notation for working with these distributions on smaller subdomains. One should get comfortable with the idea that some of the distributions defined could be the null distribution, if they were distributions conditioned on a set of measure 0, or a product over an empty set of distributions. The product of a null distribution with any other distribution is still a null distribution.

Definition E.2.1. We make the following definitions for this appendix.

- For all $i \in [n]$, let r_i denote the optimal revenue earned by selling item i individually (by our normalization, $\sum_{i=1}^{n} r_i = 1$).
- Let D_i^C (the "core" of D_i) denote the conditional distribution of D_i when it lies in the range $(-\infty, 1]$.
- Let D_i^T (the "tail" of D_i) denote the conditional distribution of D_i when it lies in the range $(1, \infty)$.
- Let $p_i := \mathbb{P}_{x_i \sim D_i}[x_i > 1]$, the probability item i lies in its tail.
- Let $A \subseteq [n]$ represent a subset of items, usually the items whose valuations lie in their tails.
- Let $D_A^T := \times_{i \in A} D_i^T$, the product distribution of only items in their tails.
- Let $D_A^C := \times_{i \notin A} D_i^C$, the product distribution of only items in their cores.
- Let $D_A := D_A^C \times D_A^T$, the conditional distribution of D when exactly the subset A of items lie in their tails. Let p_A be the probability this occurs, which is equal to $(\prod_{i \notin A}(1 - p_i))(\prod_{i \in A} p_i)$, by independence.
- Let $x_i^+ := \max\{x_i, 0\}$.
- For any valuation distribution S, let $\text{VAL}^+(S) := \sum_i \mathbb{E}_{x \sim S}[x_i^+]$, which is the expected welfare after the transformation from costs to negative valuations. Note that the sum is only over the admissible i if S is a distribution on a smaller subdomain.
- Let $\text{REV}(S)$ denote the optimal revenue obtainable from valuation distribution S via any Incentive Compatible and Individually Rational mechanism, which could include lotteries.
- Let $\text{SREV}(S)$ denote the optimal revenue of any pricing scheme falling under the class of separate sales (Pure Components).

- Let $\text{BDCREV}(S)$ denote the optimal revenue of any pricing scheme falling under the class of PBDC.

(It is understood that $\text{VAL}^+, \text{REV}, \text{SREV}, \text{BDCREV}$ are 0 when evaluated on the null distribution.)

E.2.2 Lemmas for Negative Valuations

We need to modify the statements of lemmas from Hart and Nisan (2012), Li and Yuan (2013), and Babaioff et al. (2014) to handle negative valuations. While their proofs can be extended to negative valuations in a straight-forward manner, we provide full self-contained proofs here for ease of exposition.

Lemma E.2.2. *(Marginal Mechanism) Let S, S' be (potentially negative) valuation distributions over disjoint sets of items. Then*

$$\text{REV}(S \times S') \leq \text{VAL}^+(S) + \text{REV}(S')$$

The Marginal Mechanism tells us that when selling a group of independent items, we cannot do better than breaking off some items individually, extracting the entire welfare from those items, and selling the remaining items as a group.

Proof. Proof of Lemma E.2.2. Consider the following mechanism for selling to a buyer with valuations drawn from S'. First, sample a value $v \sim S$, and reveal to the buyer these make-believe valuations for the items in S. Then run a mechanism obtaining $\text{REV}(S \times S')$ on this buyer, with the modification that whenever the buyer would have received an item i from the support of S, instead she will receive (or pay) money equal to v_i. By independence, this modified mechanism on the buyer with valuations drawn from S' is IC and IR (a buyer with valuations S' will choose the same menu entry under the modified mechanism as a buyer with valuations $S \times S'$ would have chosen under the original mechanism) and we will obtain $\text{REV}(S \times S')$, but then have to settle for the items in S. The most we stand to lose in the settlement is $\sum_i v_i^+$ (each item i in S is transferred in full whenever $v_i \geq 0$, and not transferred when $v_i < 0$), so this amount is upper bounded in expectation by $\text{VAL}^+(S)$. Therefore, the optimal revenue from S' is at least $\text{REV}(S \times S') - \text{VAL}^+(S)$, completing the proof of the lemma. □

Lemma E.2.3. *(Subdomain Stitching) Let S be a product distribution over valuations, with support $\mathcal{X} \subseteq \mathbb{R}^m$ for some $m \in \mathbb{N}$. Let $\mathcal{X}_1, \ldots, \mathcal{X}_k$ form a partition of $\mathcal{X}$ inducing conditional distributions $S^{(1)}, \ldots, S^{(k)}$, respectively, and let $s_j = \mathbb{P}_{x \sim S}[x \in \mathcal{X}_j]$. Then*

$$\text{Rev}(S) \leq \sum_{j=1}^{k} s_j \text{Rev}(S^{(j)})$$

Intuitively, Subdomain Stitching says that revenue can only increase if we sell to each subdomain separately, since we can use a different mechanism for each subdomain that specializes in extracting the welfare from that customer segment.

Proof. Proof of Lemma E.2.3. Let M be an optimal mechanism obtaining $\text{Rev}(S)$, and for any valuation distribution S', let $\text{Rev}_M(S')$ denote the expected revenue obtained from mechanism M when the buyer's valuation is drawn from S'. Clearly $\text{Rev}(S) = \sum_{j=1}^{k} s_j \text{Rev}_M(S^{(j)})$, and furthermore for all $j \in [k]$, $\text{Rev}_M(S^{(j)}) \leq \text{Rev}(S^{(j)})$ since M is an IC-IR mechanism for selling to $S^{(j)}$, completing the proof of the lemma. $\square$

Lemma E.2.4. *Let S be a product distribution over valuations, with support $\mathcal{X} \subseteq \mathbb{R}^m$ for some $m \in \mathbb{N}$. Let $\mathcal{X}'$ be a subset of $\mathcal{X}$ inducing conditional distribution S', and let $s' = \mathbb{P}_{x \sim S}[x \in \mathcal{X}']$. Then*

$$\text{Rev}(S) \geq s' \text{Rev}(S')$$

While Subdomain Stitching places an upper bound on $\text{Rev}(S)$, Lemma E.2.4 places a lower bound on $\text{Rev}(S)$ based on the optimal revenue of any single subdomain.

Proof. Proof of Lemma E.2.4. Consider an optimal mechanism for S', and extend this to an IC-IR mechanism on S by allowing the buyer to report a value in $\mathcal{X}'$ maximizing her utility. With probability s', the buyer's valuation will actually be drawn from S' and we will obtain revenue $\text{Rev}(S')$; otherwise, we still earn a non-negative revenue, since the mechanism never admits a negative payment. Therefore, the optimal revenue for S is at least $s'\text{Rev}(S')$, completing the proof of the lemma. $\square$

Lemma E.2.5. *Let S be a product distribution over m independent (potentially negative) valuations, for some $m \in \mathbb{N}$. Then*

$$\text{Rev}(S) \leq m \cdot \text{SRev}(S)$$

While selling m items together can definitely be better than selling them separately, this lemma tells us it can be no more than m times better.

Proof. Proof of Lemma E.2.5. We proceed by induction. The statement is trivial when $m = 1$. Now, suppose we have proven the statement for m valuations, and we will prove it for $m + 1$ valuations.

Partition the support $\mathcal{X} \subseteq \mathbb{R}^{m+1}$ of S into $\mathcal{X}_1$ and $\mathcal{X}_2$, where $\mathcal{X}_1 := \{x \in \mathcal{X} : x_1 \geq \max\{x_j, 0\} \ \forall\ j = 2, \ldots, m+1\}$ and $\mathcal{X}_2 := \mathcal{X} \setminus \mathcal{X}_1$. Let s_1 denote the probability a value sampled from S lies in $\mathcal{X}_1$, and let S_1 be its distribution conditioned on this event. Define s_2, S_2 respectively. Subdomain stitching tells us $\textsc{Rev}(S) \leq s_1\textsc{Rev}(S^{(1)}) + s_2\textsc{Rev}(S^{(2)})$. Our goal is to separately show that $s_1\textsc{Rev}(S^{(1)}) \leq (m+1)\textsc{SRev}(S_1)$ and $s_2\textsc{Rev}(S^{(2)}) \leq (m+1)\textsc{SRev}(S_{-1})$.

Now, applying Marginal Mechanism on $S^{(1)}$ and multiplying both sides of the inequality by s_1, we get $s_1\textsc{Rev}(S^{(1)}) \leq s_1\textsc{Val}^+(S_{-1}^{(1)}) + s_1\textsc{Rev}(S_1^{(1)})$. By considering a distribution that samples $v \sim S$ but only outputs v_1, we can use Lemma E.2.4 to show that $s_1\textsc{Rev}(S_1^{(1)}) \leq \textsc{Rev}(S_1)$. To bound $\textsc{Val}^+(S_{-1}^{(1)})$, consider the following mechanism for selling just item 1: sample $v_{-1} \sim S_{-1}$, and set the price to be $\max_{i=2}^{m+1}\{\max\{v_i, 0\}\}$. Since the buyer's valuation is drawn from S_1, by independence, we get a sale with probability exactly s_1. Furthermore, $\max_{i=2}^{m+1}\{\max\{v_i, 0\}\} \geq \frac{1}{m}\sum_{i=2}^{m+1}\max\{v_i, 0\}$, so conditioned on us getting a sale, the expected payment is at least $\frac{1}{m}\textsc{Val}^+(S_{-1}^{(1)})$. We have proven $\textsc{Rev}(S_1) \geq \frac{s_1}{m}\textsc{Val}^+(S_{-1}^{(1)})$, hence $s_1\textsc{Rev}(S^{(1)}) \leq (m+1)\textsc{Rev}(S_1) = (m+1)\textsc{SRev}(S_1)$, as required.

It remains to bound $s_2\textsc{Rev}(S^{(2)})$, and using Marginal Mechanism and Lemma E.2.4 in the same way as before, we obtain that it is no more than $s_2\textsc{Val}^+(S_1^{(2)}) + \textsc{Rev}(S_{-1})$. Consider the following mechanism for selling items $2, \ldots, m+1$: sample $v_1 \sim S_1$, and set the individual price for each item $2, \ldots, m+1$ to be $\max\{v_1, 0\}$. Note that the probability of getting at least one sale is less than s_2, since even when there is some $j = 2, \ldots, m+1$ such that $v_1 < \max\{x_j, 0\}$, it is possible for both v_1, x_j to be negative. However, in this case $\max\{v_1, 0\} = 0$, so not getting a sale is still equivalent to getting at least one sale for $\max\{v_1, 0\}$. Therefore, we can think of it as we get at least one sale with probability s_2, in which case we earn in expectation at least $\textsc{Val}^+(S_1^{(2)})$. We have proven that $s_2\textsc{Val}^+(S_1^{(2)}) \leq \textsc{SRev}(S_{-1})$, and by the induction hypothesis $\textsc{Rev}(S_{-1}) \leq m \cdot \textsc{SRev}(S_{-1})$, so $s_2\textsc{Rev}(S^{(2)}) \leq$

$(m+1)\textsc{SRev}(S_{-1})$.

Putting everything together, we have $\textsc{Rev}(S) \leq (m+1)(\textsc{SRev}(S_1) + \textsc{SRev}(S_{-1})) = (m+1)\textsc{SRev}(S)$, completing the induction and the proof of the lemma. $\square$

Using these lemmas, we decompose the revenue of the initial distribution D in the same way as Babaioff et al. (2014):

$$\begin{aligned}
\textsc{Rev}(D) &\leq \sum_{A\subseteq[n]} p_A \textsc{Rev}(D_A) \\
&\leq \sum_{A\subseteq[n]} p_A \big(\textsc{Val}^+(D_A^C) + \textsc{Rev}(D_A^T)\big) \\
&\leq \sum_{A\subseteq[n]} p_A \textsc{Val}^+(D_\emptyset^C) + \sum_{A\subseteq[n]} p_A \textsc{Rev}(D_A^T) \\
&= \textsc{Val}^+(D_\emptyset^C) + \sum_{A\subseteq[n]} p_A \textsc{Rev}(D_A^T)
\end{aligned}$$

where the first inequality is Subdomain Stitching, the second inequality is Marginal Mechanism, the third inequality is immediate from the definition of D_A^C, and the equality is a consequence of $\sum_{A\subseteq[n]} p_A = 1$.

Now, for all $A \subseteq [n]$ such that $p_A > 0$, Lemma E.2.5 tells us that $\textsc{Rev}(D_A^T) \leq |A|\textsc{SRev}(D_A^T) = |A|\sum_{i\in A}\textsc{SRev}(D_i^T)$. Lemma E.2.4 tells us that $\textsc{SRev}(D_i^T) \leq \frac{r_i}{p_i}$, where $p_i \neq 0$ since $p_A > 0$, so

$$\begin{aligned}
\sum_{A\subseteq[n]} p_A \textsc{Rev}(D_A^T) &\leq \sum_{A\subseteq[n]} p_A |A| \sum_{i\in A} \frac{r_i}{p_i} \\
&= \sum_{i=1}^{n} r_i \sum_{A\ni i} |A| \frac{p_A}{p_i}
\end{aligned}$$

$\sum_{A\ni i} |A|\frac{p_A}{p_i}$ is the expected number of items in their tails conditioned on item i being in its tail, so it is equal to $1 + \sum_{j \neq i} p_j$. Thus

$$\begin{aligned}
\sum_{A \subseteq [n]} p_A \text{REV}(D_A^T) &\leq \sum_{i=1}^{n} r_i \Big(1 + \sum_{j \neq i} p_j\Big) \\
&= 1 + \sum_{j=1}^{n} p_j \sum_{i \neq j} r_i \\
&= 1 + \sum_{j=1}^{n} p_j (1 - r_j)
\end{aligned}$$

We will use τ to denote the quantity $\sum_{i=1}^{n} p_i(1-r_i)$. It is immediate that $\tau \leq \sum_{i=1}^{n} p_i \leq 1$, but we can get a stronger bound for the welfare of the core if we don't immediately apply the inequality $\tau \leq 1$. We have

$$\text{REV}(D) \leq \text{VAL}^+(D_\emptyset^C) + 1 + \tau \tag{E.2}$$

Before we proceed, one final lemma we will need later is:

Lemma E.2.6. *Let Y be a random variable distributed over $[0,1]$ and suppose $y(1 - F(y))$ is upper bounded by some value $v \in [0,1]$. Then* $\text{Var}(Y) \leq 2v$.

Proof. Proof of Lemma E.2.6.

$$\begin{aligned}
\text{Var}(Y) &= \mathbb{E}[Y^2] - \mathbb{E}[Y]^2 \\
&\leq \mathbb{E}[Y^2] \\
&= \int_0^1 \mathbb{P}[Y^2 \geq y] dy \\
&= \int_0^1 \mathbb{P}[Y \geq \sqrt{y}] dy \\
&\leq \int_0^1 \frac{v}{\sqrt{y}} dy \\
&= 2v
\end{aligned}$$

where the second inequality uses the fact that the Myerson rcvcnuc for Y is upper bounded by v. ☐

E.2.3 A Tighter Bound for the Welfare of the Core

The main observation behind our improvement is that for τ to be large (and the above bound to be weak), the tail probabilities must be large. However, we will choose the price of the grand bundle, P_t, to be at most 2, so that whenever 2 or more valuations lie in their tails, the customer is guaranteed to want to buy the bundle (and dispose of items for which her valuation is negative). Thus

$$\begin{aligned}
\mathbb{P}[\sum x_i^+ < P_t] &= p_\emptyset \cdot \mathbb{P}_{x\sim D_\emptyset}[\sum x_i^+ < P_t] + \sum_{|A|=1} p_A \cdot \mathbb{P}_{x\sim D_A}[\sum x_i^+ < P_t] + \sum_{|A|\geq 2} p_A \cdot (0) \\
&\leq \Big(p_\emptyset + \sum_{|A|=1} p_A\Big) \cdot \mathbb{P}_{x\sim D_\emptyset^C}[\sum x_i^+ < P_t] \\
&= \Big(\prod_{i=1}^{n}(1-p_i) + \sum_{i=1}^{n} p_i \prod_{j\neq i}(1-p_j)\Big) \cdot \mathbb{P}_{x\sim D_\emptyset^C}[\sum x_i^+ < P_t] \qquad \text{(E.3)}
\end{aligned}$$

where the inequality comes from the fact that the probability of $\sum x_i^+$ being less than the bundle price is greater conditioned on no items being in the tail, than conditioned on some item being in the tail. We used independence to compute the probabilities in the final expression, which we will bound in the following way:

Lemma E.2.7. *Let $p_1, \ldots, p_n$, $r_1, \ldots, r_n$ be real numbers satisfying $0 \leq p_i \leq r_i$ and $\sum_{i=1}^n r_i = 1$. Let $\tau = \sum_{i=1}^n p_i(1-r_i)$. Then*

$$\prod_{i=1}^{n}(1-p_i) + \sum_{i=1}^{n} p_i \prod_{j\neq i}(1-p_j) \leq \frac{\frac{5}{4}+\tau}{e^\tau}$$

This is the key inequality that enables our improved ratio and its proof requires new analysis. Note that we do indeed have the condition $p_i \leq r_i$ in our case, since by Lemma E.2.4 $r_i \geq p_i \text{REV}(D_i^T)$, and $\text{REV}(D_i^T)$ must be at least 1 when D_i^T is distributed over $(1, \infty)$.

Proof. Proof of Lemma E.2.7. We will first prove

$$\frac{3}{4} \cdot \prod_{i=1}^{n}(1-p_i) + \sum_{i=1}^{n} p_i \prod_{j\neq i}(1-p_j) \leq \frac{1+\tau}{e^\tau} \qquad \text{(E.4)}$$

Assume that $p_i < 1$ for all $i \in [n]$; the lemma is trivially true otherwise because we would have LHS $= 1$ and $\tau = 0$. Since $\tau = \sum_{i=1}^n p_i(1-r_i)$ and $1 - x \le e^{-x}$, it suffices to prove

$$\frac{3}{4} \cdot \prod_{i=1}^n (1-p_i) + \sum_{i=1}^n p_i \prod_{j \ne i} (1-p_j) \le \Big(1 + \sum_{i=1}^n p_i(1-r_i)\Big) \prod_{i=1}^n (1 - p_i(1-r_i))$$

which is equivalent to

$$\frac{3}{4} + \sum_{i=1}^n \frac{p_i}{1-p_i} \le \Big(1 + \sum_{i=1}^n (p_i - p_i r_i)\Big) \prod_{i=1}^n (1 + \frac{p_i r_i}{1-p_i})$$

Observe that the RHS is at least

$$\begin{aligned}
& \Big(1 + \sum_{i=1}^n (p_i - p_i r_i)\Big)\Big(1 + \sum_{i=1}^n \frac{p_i r_i}{1-p_i}\Big) \\
= \;& 1 + \sum_{i=1}^n \frac{(p_i - p_i r_i)(1-p_i) + p_i r_i}{1-p_i} + \Big(\sum_{i=1}^n p_i(1-r_i)\Big)\Big(\sum_{i=1}^n \frac{p_i r_i}{1-p_i}\Big) \\
= \;& 1 + \sum_{i=1}^n \frac{p_i}{1-p_i} - \sum_{i=1}^n \frac{p_i^2(1-r_i)}{1-p_i} + \Big(\sum_{i=1}^n p_i(1-r_i)\Big)\Big(\sum_{i=1}^n \frac{p_i r_i}{1-p_i}\Big) \\
= \;& 1 + \sum_{i=1}^n \frac{p_i}{1-p_i} - \sum_{i=1}^n \frac{p_i^2(1-r_i)^2}{1-p_i} + \sum_{i \ne j} p_i(1-r_i) \cdot \frac{p_j r_j}{1-p_j}
\end{aligned}$$

so it remains to prove

$$\sum_{i=1}^n \frac{p_i^2(1-r_i)^2}{1-p_i} - \sum_{i \ne j} p_i(1-r_i) \cdot \frac{p_j r_j}{1-p_j} \le \frac{1}{4}$$

But $p_i \le r_i$ for all $i \in [n]$, so the LHS is at most $\sum_{i=1}^n p_i^2(1-p_i)$, which can be seen to be at most $\frac{1}{4}$, since $p_i(1-p_i)$ is always at most $\frac{1}{4}$ and $\sum_{i=1}^n p_i \le 1$.

Also, since $\tau \le \sum_{i=1}^n p_i$, $e^{-\tau} \ge \exp(-\sum_{i=1}^n p_i) \ge \prod_{i=1}^n (1-p_i)$. Multiplying by $\frac{1}{4}$ and adding to (E.4), we complete the proof of the lemma. □

E.2.4 Applying Cantelli's Inequality

To bound $\mathbb{P}_{x \sim D_\emptyset^C}[\sum x_i^+ < P_t]$, we want to show that $\sum x_i^+$ concentrates around its mean, where valuation x_i is drawn from its conditional core distribution D_i^C for all $i \in [n]$. Note that $y(1 - F_{x_i}(y))$ is bounded above by r_i for all $y \in [0,1]$; otherwise $\text{SRev}(D_i^C) > r_i \implies \text{SRev}(D_i) > r_i$ which is a contradiction. Hence $y(1 - F_{x_i^+}(y))$ is also bounded above by r_i

and we can invoke Lemma E.2.6 to get $\mathrm{Var}_{x_i \sim D_i^C}(x_i^+) \leq 2r_i$ for all $i \in [n]$. By independence, $\mathrm{Var}_{x \sim D_\emptyset^C}(\sum x_i^+) = \sum_{i=1}^n \mathrm{Var}_{x \sim D_\emptyset^C}(x_i^+) \leq \sum_{i=1}^n 2r_i = 2$ and we have successfully bounded the variance of the quantity we are interested in.

At this point, it is common in the literature to see an application of Chebyshev's inequality (e.g. Bakos and Brynjolfsson (1999); Fang and Norman (2006); Hart and Nisan (2012); Babaioff et al. (2014)). However, since we are only interested in the lower tail, we can actually use Cantelli's one-sided inequality (Lemma 6.3.1), which optimizes a shift parameter to obtain an improved bound for a single tail.

Now, note that $\mathbb{E}_{x \sim D_\emptyset^C}[\sum_{i=1}^n x_i^+] = \textsc{Val}^+(D_\emptyset^C)$ by definition. Also, it will be convenient to write the bundle price as $P_t = \alpha \cdot \textsc{Val}^+(D_\emptyset^C)$, for some $\alpha \in [0,1]$ (we would never want $\alpha > 1$ since then the price would be greater than the mean and it would be impossible to use Cantelli). Then

$$\begin{aligned}
\mathbb{P}_{x \sim D_\emptyset^C}[\textstyle\sum x_i^+ < P_t] &= \mathbb{P}_{x \sim D_\emptyset^C}\Big[\sum_{i=1}^n x_i^+ - \textsc{Val}^+(D_\emptyset^C) < -(1-\alpha)\textsc{Val}^+(D_\emptyset^C)\Big] \\
&\leq \frac{\mathrm{Var}_{x \sim D_\emptyset^C}(\sum x_i^+)}{\mathrm{Var}_{x \sim D_\emptyset^C}(\sum x_i^+) + (1-\alpha)^2 \textsc{Val}^+(D_\emptyset^C)^2} \\
&\leq \frac{2}{2 + (1-\alpha)^2 \textsc{Val}^+(D_\emptyset^C)^2}
\end{aligned}$$

where the first inequality is Cantelli's inequality, and the second inequality comes from our variance bound above. So long as we choose $P_t \leq 2$, we can use (E.3), and combined with Lemma E.2.7 we get

$$\mathbb{P}[\textstyle\sum x_i^+ < P_t] \leq \min\Big\{\dfrac{1.25+\tau}{e^\tau}, 1\Big\} \cdot \dfrac{2}{2 + (1-\alpha)^2 \textsc{Val}^+(D_\emptyset^C)^2}$$

and hence the expected revenue from selling the grand bundle at price $\alpha \cdot \textsc{Val}^+(D_\emptyset^C)$ is at least

$$\alpha \cdot \textsc{Val}^+(D_\emptyset^C) \cdot \Big(1 - \min\Big\{\frac{1.25+\tau}{e^\tau}, 1\Big\} \cdot \frac{2}{2 + (1-\alpha)^2 \textsc{Val}^+(D_\emptyset^C)^2}\Big)$$

Recall from (E.2) that $\textsc{Rev}(D) \leq \textsc{Val}^+(D_\emptyset^C) + 1 + \tau$. While τ could take on any value in $[0,1]$, we can choose the price of the bundle based on τ and $\textsc{Val}^+(D_\emptyset^C)$ by adjusting $\alpha \in [0,1]$.

Case 1. If $\textsc{Val}^+(D_\emptyset^C) \leq 3.2$, then $\textsc{Rev}(D) \leq 3.2 + 1 + 1 = 5.2 \cdot \textsc{SRev}(D)$ is immediate and we can just sell the items individually.

Case 2. If $3.2 < \textsc{Val}^+(D_\emptyset^C) \leq 4$, then we will choose $\alpha = \frac{1}{2}$ which guarantees $P_t \leq 2$. Thus

$$\textsc{BdcRev}(D) \geq \textsc{Val}^+(D_\emptyset^C) \cdot \frac{1}{2}\Big(1 - \min\big\{\frac{1.25+\tau}{e^\tau}, 1\big\} \cdot \frac{2}{2 + (1-\frac{1}{2})^2(3.2)^2}\Big)$$

It can be shown with calculus (or numerically) that:

Proposition E.2.8. *For all $\tau \in [0,1]$, $2\Big(1 - \min\big\{\frac{1.25+\tau}{e^\tau}, 1\big\} \cdot \frac{2}{2+(1-\frac{1}{2})^2(3.2)^2}\Big)^{-1} + (1+\tau) < 5.2$, with the maximum of ≈ 5.1952 occuring at the unique positive τ satisfying $\frac{1.25+\tau}{e^\tau} = 1$.*

Hence $\textsc{Val}^+(D_\emptyset^C) \leq (4.2-\tau)\textsc{BdcRev}(D)$. Substituting into (E.2), we get

$$\begin{aligned}
\textsc{Rev}(D) &\leq (4.2-\tau)\textsc{BdcRev}(D) + (1+\tau)\textsc{SRev}(D) \\
&\leq 5.2 \cdot \max\{\textsc{SRev}(D), \textsc{BdcRev}(D)\}
\end{aligned}$$

as desired.

Case 3. If $4 < \textsc{Val}^+(D_\emptyset^C)$, then we will still choose $\alpha = \frac{1}{2}$. We no longer have $P_t \leq 2$, so we have to use the weaker bound $\mathbb{P}_{x\sim D}[\sum x_i^+ < P_t] \leq \mathbb{P}_{x\sim D_\emptyset^C}[\sum x_i^+ < P_t]$. However, applying Cantelli yields

$$\mathbb{P}_{x\sim D_\emptyset^C}[\textstyle\sum x_i^+ < P_t] \leq \frac{2}{2 + (1-\frac{1}{2})^2(4)^2} = \frac{1}{3}$$

so $\textsc{BdcRev}(D) \geq \textsc{Val}^+(D_\emptyset^C) \cdot \frac{1}{2}(1 - \frac{1}{3})$. We get $\textsc{Rev}(D) \leq 3 \cdot \textsc{BdcRev}(D) + (1+\tau)\textsc{SRev}(D) < 5.2 \cdot \max\{\textsc{SRev}(D), \textsc{BdcRev}(D)\}$, completing the proof of Theorem 6.4.1.

E.3 Proof of Theorem 6.4.3

It is immediate that the optimal revenue from PC is 2ρ, attained by selling individual items at any price in $[1, 2]$. Next, we would like to argue that the optimal revenue from PB is also 2ρ. If we offer the bundle at 2, it is guaranteed to get bought if either valuation realizes to 2 or both valuations realize to a positive number, and won't get bought otherwise. Therefore the revenue is $2(\rho^2 + 2(1-\rho)\frac{\rho}{2}) = 2\rho$.

We can do equally well by offering the bundle at 3, and any other price is inferior.

Lemma E.3.1. *The optimal revenue from PB is 2ρ, attained by setting a bundle price of* 2 *or* 3.

Proof. Proof of Lemma E.3.1. Let z denote the price of the bundle. We will systematically analyze all the cases over $1 \leq z \leq 4$ and show that the maximum revenue of 2ρ is attained at $z = 2$ and $z = 3$.

Case 1. Suppose $1 \leq z \leq 2$. Let us condition on the realization y of the first valuation. If $y = 0$, then we get a sale with probability $\frac{\rho}{z}$. If $y \in [1, z)$, then we get a sale so long as the second valuation realizes to a positive number, which occurs with probability $1 - \rho$. If $y \geq z$, then the first valuation alone is enough to guarantee a bundle sale. The expected revenue is

$$z\left((1-\rho)\frac{\rho}{z} + (\rho - \frac{\rho}{z})\rho + \frac{\rho}{z}\right) = 2\rho + (z-2)\rho^2$$

which is clearly maximized at $z = 2$, in which case the revenue is 2ρ.

Case 2. Suppose $2 < z \leq 3$. Let us condition on the realization y of the first valuation. If $y = 0$, then we have no chance of selling the bundle. If $y \in [1, z-1]$, then we get a sale when the other valuation is at least $z - y$. Since $z - y \in [1, 2]$, the probability of this occurring is $\frac{\rho}{z-y}$. If $y \geq z - 1$, then we get a sale so long as the other valuation realizes to a positive number, which occurs with probability ρ. The total probability of getting a sale is

$$\int_1^{z-1} \frac{\rho}{y^2}\frac{\rho}{z-y}dy + \frac{\rho}{z-1}\rho$$

where the PDF of Y satisfies $f(y) = \frac{\rho}{y^2}$ over $[1, 2)$. Using partial fractions, the antiderivative of $\frac{1}{y^2(z-y)}$ can be computed to be

$$\frac{1}{z}\left(\frac{\ln y - \ln(z-y)}{z} - \frac{1}{y}\right)$$

as demonstrated in the proof of (Hart and Nisan, 2012, lem. 6). Therefore, the definite integral evaluates to

$$\rho^2\left(\frac{2\ln(z-1)}{z^2} + \frac{2}{z} - \frac{1}{z-1}\right)$$

and the expected revenue is

$$z\rho^2\left(\frac{2\ln(z-1)}{z^2} + \frac{2}{z} - \frac{1}{z-1} + \frac{1}{z-1}\right) = 2\rho^2\left(\frac{\ln(z-1)}{z} + 1\right)$$

However, $\frac{\ln(z-1)}{z}$ is a strictly increasing function on $(2,3]$, so this expression is uniquely maximized at $z = 3$ where it equals $2\rho^2(\frac{\ln 2}{3} + 1) = 2\rho$.

Case 3. Suppose $3 \le z \le 4$. Let us condition on the realization y of the first valuation. If $y < z - 2$, then we have no chance of selling the bundle. Otherwise, the probability of getting a sale is $\frac{\rho}{z-y}$, since $z - y \in [1,2]$. The total probability of getting a sale is

$$\int_{z-2}^{2} \frac{\rho}{y^2}\frac{\rho}{z-y} + \frac{\rho}{2}\frac{\rho}{z-2}$$

and the integral evaluates to

$$\rho^2\left(\frac{2\ln 2 - 2\ln(z-2)}{z^2} + \frac{1}{z(z-2)} - \frac{1}{2z}\right)$$

Therefore, the expected revenue is

$$z\rho^2\left(\frac{2\ln 2 - 2\ln(z-2)}{z^2} + \frac{1}{z(z-2)} - \frac{1}{2z} + \frac{1}{2(z-2)}\right) = 2\rho^2\left(\frac{\ln 2 - \ln(z-2)}{z} + \frac{1}{z-2}\right)$$

$\frac{\ln 2 - \ln(z-2)}{z} + \frac{1}{z-2}$ is a strictly decreasing function on $[3,4]$, so this expression is uniquely maximized at $z = 3$. $\square$

Now, consider the strategy of offering either item for 2 or the bundle for the discounted price of 3. Note that if buying the bundle is non-negative utility for the customer, then buying either individual item cannot be higher utility, since the price savings is one and the value of the item lost is at least one (recall that the firm gets to break ties in a way that favors itself). Hence there is no cannibalization of bundle sales from individual sales and we earn revenue at least 2ρ. However, when exactly one valuation realizes to a positive number (in which case we have no chance of selling the bundle), we still have a $\frac{1}{2}$ conditional probability of selling that individual item. Hence the revenue from Mixed Bundling is $2\rho + 2(2(1-\rho)\frac{\rho}{2}) = 2\rho(2-\rho)$.

The relative gain over both the PC revenue and the PB revenue is $2 - \rho = \frac{3+2\ln 2}{3+\ln 2}$, completing the proof of Theorem 6.4.3.

Remark E.3.2. A motivating example for our construction is a small modification of the earlier best-known example from Hart and Nisan (2012): consider a distribution that takes on values $0, 1, 2$ with probabilities $\frac{1}{9}, \frac{4}{9}, \frac{4}{9}$, respectively. Let D be the instance consisting of two independent copies of this distribution. Then it can be shown that the optimal PC revenue is $\frac{16}{9}$ (attained at individual prices 1 or 2), the optimal PB revenue is $\frac{16}{9}$ (attained at bundle price 2 or 3), and the optimal revenue is at least $\frac{160}{81}$ (attained at individual prices 2 and bundle price 3), achieving a ratio of $\frac{9}{10}$. Hart and Nisan (2012) had the probabilities be $\frac{1}{3}, \frac{1}{3}, \frac{1}{3}$ instead, achieving a ratio of $\frac{12}{13}$.

E.4 Example where BSP Performs Poorly

Consider a firm that is bundling a higher-profit-margin, lower-valuation good with a low-profit-margin, high-valuation good. This is a common occurrence, for example when video games are bundled with a console, which we will hereinafter refer to as item 1 and item 2, respectively. Item 1 costs zero to produce and has a valuation uniform on [0,1]; item 2 costs 4.5 to produce and has a valuation uniform on [0,5] and independent from item 1. Most of the welfare comes from the lower-valuation item: the expected welfare for item 1 and item 2 are 0.5 and 0.025, respectively.

The optimal deterministic profit is ≈ 0.265, attained by offering item 1 at 0.51, item 2 at 4.83, and the bundle at the discounted price of 5.13.

The optimal BSP pricing charges 4.83 for a single item and 5.03 for both items, earning only 19% of the deterministic optimum. This example highlights the issue with BSP: it cannot afford to charge a low price for a single item if any item has a high production cost. However, most of the potential profit could be coming from offering lower-valuation items at low prices! Chu et al. (2008) bypass such examples in their numerical experiments, assuming that all items have a low cost compared to its mean valuation.

PBDC offers item 1 at 0.51, item 2 at 5.01, and the bundle at 5.01—which is the right idea and earns 99.1% of the deterministic optimum. Interestingly, even the analytical solution provided by Bhargava (2013), which computes the optimal deterministic pricing when there are two independent uniform distributions and costs, is less effective than PBDC on this example. The solution from Bhargava (2013) only attains 97.5% of the deterministic optimum for this example, because it requires a bit of linear approximation.

Optimal bundling is an intricate problem even in the case of two independent uniform distributions, so a simple pricing heuristic as robust as PBDC is invaluable. In fact, for this example PBDC recommends *Partial Mixed Bundling*, which is a Mixed Bundling scheme where one of the items, in this case item 2 (the high-cost low-welfare item), is never sold individually. This matches the intuition that the seller should add item 1 (the low-cost high-welfare item) to item 2 in order to increase the total amount customer is willing to pay (see Proposition 1 in Bhargava (2013)). BSP, on the other hand, does not perform well: it recommends a Partial Mixed Bundling scheme where item 1 is never sold individually.

Appendix F

Appendix to Chapter 7

F.1 Examples of transforming SK jobs to Markov chains.

1. Consider a job that takes time 5 with probability $\frac{1}{3}$, and time 2 with probability $\frac{2}{3}$ (and cannot be canceled once started). If it finishes, the reward returned is 2, independent of processing time. This can be modeled by fig. F-1 where $r_B = \frac{4}{3}$, $r_E = 2$, and $\mathcal{B} = \{B, C, D, E\}$. Note that instead of placing reward 2 on arc (B, C'), we have equivalently placed reward $\frac{4}{3}$ on node B. A corollary of this reduction is that the following reward structure is equivalent to the original for the objective of maximizing expected reward: a guaranteed reward of $\frac{4}{3}$ after 2 time steps, after which the job may run for another 3 time steps to produce an additional 2 reward.

2. Consider the same job as the previous one, except the reward is 4 if the processing time was 5, while the reward is 1 if the processing time was 2 (the expected reward for finishing is still 2). All we have to change in the reduction is setting $r_B = \frac{2}{3}$ and $r_E = 4$ instead.

3. Consider either of the two jobs above, except cancellation is permitted (presumably on node C, after observing the transition from node B). All we have to change in the reduction is setting $\mathcal{B} = \emptyset$ instead.

4. Consider the job from the second bullet that can be canceled, and furthermore, we find out after 1 time step whether it will realize to the long, high-reward job or the short, low-reward job. This can be modeled by fig. F-2 where $r_{B'} = 1$, $r_E = 4$, and $\mathcal{B} = \emptyset$.

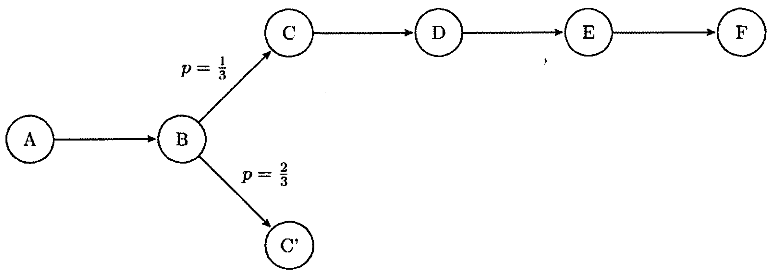


Figure F-1: A Markov chain representing a SK job with correlated rewards

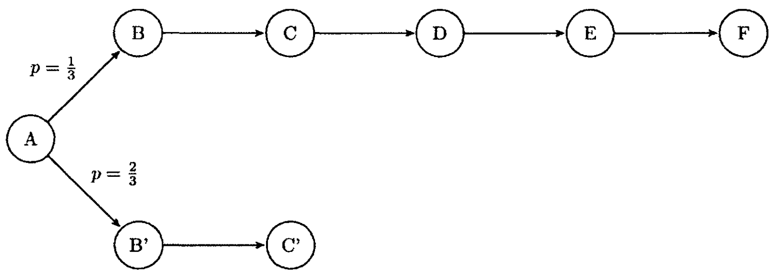


Figure F-2: Another Markov chain representing a SK job with correlated rewards

F.2 Example showing preemption is necessary for uncorrelated SK.

Consider the following example: there are $n = 3$ items, I_1, I_2, I_3. I_1 instantiates to size 6 with probability $\frac{1}{2}$, and size 1 with probability $\frac{1}{2}$. I_2 deterministically instantiates to size 9. I_3 instantiates to size 8 with probability $\frac{1}{2}$, and size 4 with probability $\frac{1}{2}$. I_1, I_2, I_3, if successfully inserted, return rewards of $4, 9, 8$, respectively. We have a knapsack of size 10.

We describe the optimal preempting policy. First we insert I_1. After 1 unit of time, if I_1 completes, we go on to insert I_2, which will deterministically fit. If I_1 doesn't complete, we set it aside and insert I_3 to completion. If it instantiates to size 8, then we cannot get any more reward from other items. If it instantiates to size 4, then we can go back and finish inserting the remaining 5 units of I_1. The expected reward of this policy is $\frac{1}{2}(4+9) + \frac{1}{2}(\frac{1}{2}8 + \frac{1}{2}(8+4)) = 11.5$.

Now we enumerate the policies that can only cancel but not preempt. If we first insert I_2, then the best we can do is try to insert I_1 afterward, getting a total expected reward of 11. Note that any policy never fitting I_2 can obtain reward at most 11, since with probability $\frac{1}{4}$ we cannot fit both I_1 and I_3. This rules out policies that start with I_3, which has no chance of fitting alongside I_2. Remaining are the policies that first insert I_1. If it doesn't complete after 1 unit of time, then we can either settle for the 9 reward of I_2, or finish processing I_1 with the hope of finishing I_3 afterward. However, in this case, I_3 only finishes half the time, so we earn more expected reward by settling for I_2. Therefore, the best we can do after first inserting I_1 is to stop processing it after time 1 (regardless of whether it completes), and process I_2, earning a total expected reward of 11.

We have shown that indeed, for uncorrelated SK, there is a gap between policies that can preempt (which includes canceling) versus policies that can only cancel. It appears that this gap is bounded by a constant, contrary to the gap between policies that can cancel versus policies that cannot cancel (see (Gupta et al., 2011a, appx. A.1)).

F.3 Proofs from Section 7.2.

F.3.1 Proof of Lemma 7.2.5.

Suppose we are given $\{z^a_{\pi,i,t}\}, \{y_{\pi,t}\}$ satisfying (7.2a)-(7.2c), (7.3a)-(7.3c) which imply (7.4). For all $i \in [n]$, $u \in \mathcal{S}_i$, $t \in [B]$, let $s_{u,t} = \sum_{\pi\in\boldsymbol{\mathcal{S}}:\pi_i=u} y_{\pi,t}$, and let $x^a_{u,t} = \sum_{\pi\in\boldsymbol{\mathcal{S}}:\pi_i=u} z^a_{\pi,i,t}$ for each $a \in A$. We aim to show $\{x^a_{u,t}\}, \{s_{u,t}\}$ satisfies (7.10a)-(7.10c), (7.11), (7.12a)-(7.12c) and makes (7.9) the same objective function as (7.1). For convenience, we adopt the notation that $x_{u,t} = \sum_{a\in A} x^a_{u,t}$ and $z_{\pi,i,t} = \sum_{a\in A} z^a_{\pi,i,t}$.

(7.11): $\sum_{u\in\mathcal{S}} x_{u,t} = \sum_{i=1}^n \sum_{u\in\mathcal{S}_i} \sum_{\pi:\pi_i=u} z_{\pi,i,t} = \sum_{\pi\in\boldsymbol{\mathcal{S}}} \sum_{i=1}^n \sum_{u\in\mathcal{S}_i:u=\pi_i} z_{\pi,i,t}$. But there is a unique $u \in \mathcal{S}_i$ such that $u = \pi_i$, so the sum equals $\sum_{\pi\in\boldsymbol{\mathcal{S}}} \sum_{i=1}^n z_{\pi,i,t}$, which is at most 1 by (7.4).

(7.10a): For $u \in \mathcal{S}_i$, $x_{u,t} = \sum_{\pi:\pi_i=u} z_{\pi,i,t}$, and each term in the sum is at most $y_{\pi,t}$ by (7.2a) and (7.2c), hence $x_{u,t} \leq \sum_{\pi:\pi_i=u} y_{\pi,t} = s_{u,t}$.

(7.10b): For $u \in \mathcal{B}$, $x^\alpha_{u,t} = \sum_{\pi:\pi_i=u} z^\alpha_{\pi,i,t}$, and each term in the sum is equal to $y_{\pi,t}$ by (7.2b), hence $x^\alpha_{u,t} = \sum_{\pi:\pi_i=u} y_{\pi,t} = s_{u,t}$.

(7.10c), (7.12a), and (7.12b) are immediate from (7.2c), (7.3a), and (7.3b), respectively.

For (7.12c), fix $t > 1$, $i \in [n]$, and $u \in \mathcal{S}_i$. Sum (7.3c) over $\{\pi : \pi_i = u\}$ to get

$$\begin{aligned}
\sum_{\pi:\pi_i=u} y_{\pi,t} &= \sum_{\pi:\pi_i=u} y_{\pi,t-1} - \sum_{\pi:\pi_i=u} \sum_{j=1}^{n} z_{\pi,j,t-1} + \sum_{\pi:\pi_i=u} \sum_{j=1}^{n} \sum_{(v,a)\in\mathsf{Par}(\pi_j)} z^a_{\pi^v,j,t-1} \cdot p^a_{v,\pi_j} \\
s_{u,t} &= s_{u,t-1} - \sum_{\pi:\pi_i=u} z_{\pi,i,t-1} - \sum_{\pi:\pi_i=u} \sum_{j\neq i} z_{\pi,j,t-1} \\
&\quad + \sum_{\pi:\pi_i=u} \sum_{(v,a)\in\mathsf{Par}(u)} z^a_{\pi^v,i,t-1} \cdot p^a_{v,u} + \sum_{\pi:\pi_i=u} \sum_{j\neq i} \sum_{(v,a)\in\mathsf{Par}(\pi_j)} z^a_{\pi^v,j,t-1} \cdot p^a_{v,\pi_j} \\
s_{u,t} &= s_{u,t-1} - x_{u,t-1} - \sum_{\pi:\pi_i=u} \sum_{j\neq i} z_{\pi,j,t-1} \\
&\quad + \sum_{(v,a)\in\mathsf{Par}(u)} \Big(\sum_{\pi:\pi_i=u} z^a_{\pi^v,i,t-1} \Big) \cdot p^a_{v,u} + \sum_{j\neq i} \sum_{v\in\mathcal{S}_j} \sum_{a\in A} \sum_{\{\pi:\pi_i=u,\mathsf{Par}(\pi_j)\ni(v,a)\}} z^a_{\pi^v,j,t-1} \cdot p^a_{v,\pi_j} \\
s_{u,t} &= s_{u,t-1} - x_{u,t-1} - \sum_{\pi:\pi_i=u} \sum_{j\neq i} z_{\pi,j,t-1} \\
&\quad + \sum_{(v,a)\in\mathsf{Par}(u)} \Big(\sum_{\pi:\pi_i=v} z^a_{\pi,i,t-1} \Big) \cdot p^a_{v,u} + \sum_{j\neq i} \sum_{v\in\mathcal{S}_j} \sum_{a\in A} \sum_{\pi:\pi_i=u,\pi_j=v} z^a_{\pi,j,t-1} \cdot \Big(\sum_{w:p^a_{v,w}>0} p^a_{v,w} \Big) \\
s_{u,t} &= s_{u,t-1} - x_{u,t-1} - \sum_{\pi:\pi_i=u} \sum_{j\neq i} z_{\pi,j,t-1} \\
&\quad + \sum_{(v,a)\in\mathsf{Par}(u)} x^a_{v,t-1} \cdot p^a_{v,u} + \sum_{j\neq i} \sum_{v\in\mathcal{S}_j} \sum_{a\in A} \sum_{\pi:\pi_i=u,\pi_j=v} z^a_{\pi,j,t-1} \cdot (1) \\
s_{u,t} &= s_{u,t-1} - x_{u,t-1} - \sum_{j\neq i} \sum_{\pi:\pi_i=u} z_{\pi,j,t-1} + \sum_{(v,a)\in\mathsf{Par}(u)} x^a_{v,t-1} \cdot p^a_{v,u} + \sum_{j\neq i} \sum_{\pi:\pi_i=u} z_{\pi,j,t-1} \\
s_{u,t} &= s_{u,t-1} - x_{u,t-1} + \sum_{(v,a)\in\mathsf{Par}(u)} x^a_{v,t-1} \cdot p^a_{v,u}
\end{aligned}$$

which is exactly (7.12c).

(7.9): $\sum_{u\in\mathcal{S}} \sum_{a\in A} r^a_u \sum_{t=1}^{B} x^a_{u,t} = \sum_{i=1}^{n} \sum_{u\in\mathcal{S}_i} \sum_{a\in A} r^a_u \sum_{t=1}^{B} \sum_{\pi:\pi_i=u} z^a_{\pi,i,t}$, and by the same manipulation we made for (7.11), this is equal to $\sum_{\pi\in\boldsymbol{\mathcal{S}}} \sum_{i=1}^{n} \sum_{a\in A} r^a_{\pi_i} \sum_{t=1}^{B} z^a_{\pi,i,t}$. Thus (7.9) is the same as (7.1), completing the proof of Lemma 7.2.5.

F.3.2 Proof of Lemma 7.2.6.

Suppose we are given $\{z^a_{\pi,i,t}\}, \{y_{\pi,t}\}$ satisfying (7.6a)-(7.6c), (7.7a)-(7.7e) which imply (7.8). For all $i \in [n]$, $u \in \mathcal{S}_i$, $t \in [B]$, let $s_{u,t} = \sum_{\pi\in\boldsymbol{\mathcal{S}}':\pi_i=u} y_{\pi,t}$, and let $x^a_{u,t} = \sum_{\pi\in\boldsymbol{\mathcal{S}}':\pi_i=u} z^a_{\pi,i,t}$ for each $a \in A$. We aim to show $\{x^a_{u,t}\}, \{s_{u,t}\}$ satisfies (7.10a)-(7.10c), (7.11), (7.13a)-(7.13d) and makes (7.9) the same objective function as (7.5). For convenience, we adopt the notation that $x_{u,t} = \sum_{a\in A} x^a_{u,t}$ and $z_{\pi,i,t} = \sum_{a\in A} z^a_{\pi,i,t}$.

(7.11): $\sum_{u\in\mathcal{S}} x_{u,t} = \sum_{i=1}^{n} \sum_{u\in\mathcal{S}_i} \sum_{\pi:\pi_i=u} z_{\pi,i,t} = \sum_{\pi\in\boldsymbol{\mathcal{S}}'} \sum_{i=1}^{n} \sum_{u\in\mathcal{S}_i:u=\pi_i} z_{\pi,i,t}$. The difference from the previous derivation of (7.11) is that there is only a unique $u \in \mathcal{S}_i$ such that $u = \pi_i$, if $\pi_i \neq \phi_i$. So the sum equals $\sum_{\pi\in\boldsymbol{\mathcal{S}}'} \sum_{i\in I(\pi)} z_{\pi,i,t}$, which is at most 1 by (7.8).

Using this same manipulation, the equivalence of (7.9) and (7.5) follows the same derivation as before. (7.10a) and (7.10b) also follow the same derivations as before; (7.10c), (7.13a), and (7.13b) are immediate. It remains to prove (7.13c) and (7.13d).

(7.13d): Fix $t > 1$, $i \in [n]$, and $u \in \mathcal{S}_i \setminus \{\rho_i\}$. First consider the case where $\mathsf{depth}(u) > 1$. All $\pi \in \boldsymbol{\mathcal{S}}'$ such that $\pi_i = u$ fall under (7.7e), so we can sum over these π to get

$$\begin{aligned}
\sum_{\pi:\pi_i=u} y_{\pi,t} &= \sum_{\pi:\pi_i=u} \sum_{(v,a)\in\mathsf{Par}(u)} z^a_{\pi^v,i,t-1} \cdot p^a_{v,u} \\
s_{u,t} &= \sum_{(v,a)\in\mathsf{Par}(u)} \Big(\sum_{\pi:\pi_i=u} z^a_{\pi^v,i,t-1}\Big) \cdot p^a_{v,u}
\end{aligned}$$

Since $\mathsf{depth}(u) > 1$, $v \neq \rho_i$, so $\{\pi^v : \pi \in \boldsymbol{\mathcal{S}}', \pi_i = u\} = \{\pi : \pi \in \boldsymbol{\mathcal{S}}', \pi_i = v\}$. Hence the RHS of the above equals $\sum_{(v,a)\in\mathsf{Par}(u)} x^a_{v,t-1} \cdot p^a_{v,u}$ which is exactly (7.13d).

For the other case where $\mathsf{depth}(u) = 1$, all $\pi \in \boldsymbol{\mathcal{S}}'$ such that $\pi_i = u$ fall under (7.7d), so we can sum over these π to get

$$\begin{aligned}
\sum_{\pi:\pi_i=u} y_{\pi,t} &= \sum_{\pi:\pi_i=u} \sum_{a:(\rho_i,a)\in\mathsf{Par}(u)} \Big(\sum_{\pi'\in\boldsymbol{\mathcal{P}}(\pi^{\rho_i})} z^a_{\pi',i,t-1}\Big) \cdot p^a_{\rho_i,u} \\
s_{u,t} &= \sum_{a:(\rho_i,a)\in\mathsf{Par}(u)} \Big(\sum_{\pi:\pi_i=u} \sum_{\pi'\in\boldsymbol{\mathcal{P}}(\pi^{\rho_i})} z^a_{\pi',i,t-1}\Big) \cdot p^a_{\rho_i,u} \\
s_{u,t} &= \sum_{a:(\rho_i,a)\in\mathsf{Par}(u)} \Big(\sum_{\pi:\pi_i=\rho_i} z^a_{\pi,i,t-1}\Big) \cdot p^a_{\rho_i,u} \\
s_{u,t} &= \sum_{a:(\rho_i,a)\in\mathsf{Par}(u)} x^a_{\rho_i,t-1} \cdot p^a_{\rho_i,u}
\end{aligned}$$

We explain the third equality. Since $u \neq \rho_i$ implies arm i is the active arm in all of $\{\pi \in \boldsymbol{\mathcal{S}}' : \pi_i = u\}$, this set is equal to $\{\rho_1, \phi_1\} \times \cdots \times u \times \cdots \times \{\rho_n, \phi_n\}$. Thus $\{\pi' \in \boldsymbol{\mathcal{P}}(\pi^{\rho_i}) : \pi \in \boldsymbol{\mathcal{S}}', \pi_i = u\} = \{\pi' \in \boldsymbol{\mathcal{P}}(\pi) : \pi \in \{\rho_1, \phi_1\} \times \cdots \times \rho_i \times \cdots \times \{\rho_n, \phi_n\}\}$. Recall that $\boldsymbol{\mathcal{P}}(\pi)$ is the set of joint nodes that would transition to π with no play. Therefore, this set is equal to $\{\pi' \in \boldsymbol{\mathcal{S}}' : \pi'_i = \rho_i\}$, as desired.

(7.13c): Fix $t > 1$ and $i \in [n]$. Unfortunately, $\pi \in \boldsymbol{\mathcal{S}}'$ such that $\pi_i = \rho_i$ can fall under

(7.7c), (7.7d), or (7.7e). First let's sum over the π falling under (7.7c):

$$\begin{aligned}
\sum_{\pi\notin\boldsymbol{\mathcal{A}}:\pi_i=\rho_i} y_{\pi,t} &= \sum_{\pi\notin\boldsymbol{\mathcal{A}}:\pi_i=\rho_i}\sum_{\pi'\in\boldsymbol{\mathcal{P}}(\pi)}\Big(y_{\pi',t-1}-\sum_{j\in I(\pi')} z_{\pi',j,t-1}\Big)\\
&= \sum_{\pi\in\boldsymbol{\mathcal{S}}':\pi_i=\rho_i}\Big(y_{\pi,t-1}-z_{\pi,i,t-1}-\sum_{j\in I(\pi)\setminus\{i\}} z_{\pi,j,t-1}\Big)\\
&= s_{\rho_i,t-1}-x_{\rho_i,t-1}-\sum_{\pi\in\boldsymbol{\mathcal{S}}':\pi_i=\rho_i}\sum_{j\in I(\pi)\setminus\{i\}} z_{\pi,j,t-1}
\end{aligned}$$

where the second equality requires the same set bijection explained above. Furthermore,

$$\begin{aligned}
\sum_{\pi\in\boldsymbol{\mathcal{S}}':\pi_i=\rho_i}\sum_{j\in I(\pi)\setminus\{i\}} z_{\pi,j,t-1} &= \sum_{k\neq i}\sum_{\pi\in\boldsymbol{\mathcal{A}}_k:\pi_i=\rho_i}\Big(z_{\pi,k,t-1}+\sum_{j\in I(\pi)\setminus\{i,k\}} z_{\pi,j,t-1}\Big)\\
&\quad+\sum_{\pi\notin\boldsymbol{\mathcal{A}}:\pi_i=\rho_i}\sum_{j\in I(\pi)\setminus\{i\}} z_{\pi,j,t-1}\\
&= \sum_{k\neq i}\sum_{\pi\in\boldsymbol{\mathcal{A}}_k:\pi_i=\rho_i}\Big(z_{\pi,k,t-1}+\sum_{j:\pi_j=\rho_j,j\neq i} z_{\pi,j,t-1}\Big)\\
&\quad+\sum_{\pi\notin\boldsymbol{\mathcal{A}}:\pi_i=\rho_i}\sum_{j:\pi_j=\rho_j,j\neq i} z_{\pi,j,t-1}\\
&= \sum_{k\neq i}\sum_{\pi\in\boldsymbol{\mathcal{A}}_k:\pi_i=\rho_i} z_{\pi,k,t-1}+\sum_{\pi\in\boldsymbol{\mathcal{S}}':\pi_i=\rho_i}\sum_{j:\pi_j=\rho_j,j\neq i} z_{\pi,j,t-1}
\end{aligned}$$

Now let's sum over the π falling under (7.7e):

$$\begin{aligned}
\sum_{j\neq i}\sum_{\{\pi:\pi_i=\rho_i,\mathsf{depth}(\pi_j)>1\}} y_{\pi,t} &= \sum_{j\neq i}\sum_{\{\pi:\pi_i=\rho_i,\mathsf{depth}(\pi_j)>1\}}\sum_{(v,a)\in\mathsf{Par}(\pi_j)} z^a_{\pi^v,j,t-1}\cdot p^a_{v,\pi_j}\\
&= \sum_{j\neq i}\sum_{v\in S_j\setminus\{\rho_j\}}\sum_{a\in A}\sum_{\{\pi:\pi_i=\rho_i,\mathsf{Par}(\pi_j)\ni(v,a)\}} z^a_{\pi^v,j,t-1}\cdot p^a_{v,\pi_j}\\
&= \sum_{j\neq i}\sum_{v\in S_j\setminus\{\rho_j\}}\sum_{a\in A}\sum_{\pi:\pi_i=\rho_i,\pi_j=v} z^a_{\pi,j,t-1}\cdot\Big(\sum_{w:p^a_{v,w}>0} p^a_{v,w}\Big)\\
&= \sum_{j\neq i}\sum_{v\in S_j\setminus\{\rho_j\}}\sum_{a\in A}\sum_{\pi:\pi_i=\rho_i,\pi_j=v} z^a_{\pi,j,t-1}\cdot(1)\\
&= \sum_{j\neq i}\sum_{\pi\in\boldsymbol{\mathcal{A}}_j:\pi_i=\rho_i} z_{\pi,j,t-1}
\end{aligned}$$

where the third equality uses the fact that $v\neq\rho_j$ to convert π^v to π. Finally, let's sum over the π falling under (7.7d):

$$\begin{aligned}
&\sum_{j\neq i}\sum_{\{\pi:\pi_i=\rho_i,\mathsf{depth}(\pi_j)=1\}} y_{\pi,t}\\
=&\sum_{j\neq i}\sum_{\{\pi:\pi_i=\rho_i,\mathsf{depth}(\pi_j)=1\}}\sum_{a:(\rho_j,a)\in\mathsf{Par}(\pi_j)}\Big(\sum_{\pi'\in\mathcal{P}(\pi^{\rho_j})} z^a_{\pi',j,t-1}\Big)\cdot p^a_{\rho_j,\pi_j}\\
=&\sum_{j\neq i}\sum_{a\in A}\sum_{\{\pi:\pi_i=\rho_i,\mathsf{Par}(\pi_j)\ni(\rho_j,a)\}}\Big(\sum_{\pi'\in\mathcal{P}(\pi^{\rho_j})} z^a_{\pi',j,t-1}\Big)\cdot p^a_{\rho_j,\pi_j}\\
=&\sum_{j\neq i}\sum_{a\in A}\sum_{\pi:\pi_i=\rho_i,\pi_j=\rho_j} z^a_{\pi,j,t-1}\cdot\Big(\sum_{w:p^a_{\rho_j,w}>0} p^a_{\rho_j,w}\Big)\\
=&\sum_{j\neq i}\sum_{a\in A}\sum_{\pi:\pi_i=\rho_i,\pi_j=\rho_j} z^a_{\pi,j,t-1}\cdot(1)\\
=&\sum_{\pi:\pi_i=\rho_i}\sum_{j:\pi_j=\rho_j,j\neq i} z_{\pi,j,t-1}
\end{aligned}$$

where the third equality requires the same set bijection again. Combining the last four blocks of equations, we get $s_{\rho_i,t}=s_{\rho_i,t-1}-x_{\rho_i,t-1}$ which is exactly (7.13c), completing the proof of Lemma 7.2.6.

F.4 Proofs from Section 7.4.

F.4.1 Proof of Lemma 7.4.1.

Finding the q's is a separate problem for each arm, so we can fix $i\in[n]$. Furthermore, we can fix $u\in\mathcal{S}_i\setminus\{\rho_i\}$; we will specify an algorithm that defines $\{q_{v,b,t',u,a,t}: a\in A, t\in[B],(v,b)\in\mathsf{Par}(u),t'<t\}$ satisfying (7.19a) and (7.19b).

Observe that by substituting (7.10a) into (7.12c), we get $s_{u,t'}\le\sum_{(v,b)\in\mathsf{Par}(u)}x^b_{v,t'-1}\cdot p^b_{v,u}$ for all $t'>1$. Summing over $t'=2,\dots,t$ for an arbitrary $t\in[B]$, and using (7.10a) again on the LHS, we get $\sum_{t'=2}^{t}\sum_{a\in A}x^a_{u,t'}\le\sum_{t'=1}^{t-1}\sum_{(v,b)\in\mathsf{Par}(u)}x^b_{v,t'}\cdot p^b_{v,u}$.

Now, for all $t'=2,\dots,B$ and $a\in A$, initialize $\tilde{x}^a_{t'}:=x^a_{u,t'}$. For all $t'=1,\dots,B-1$ and $(v,b)\in\mathsf{Par}(u)$, initialize $\tilde{x}^b_{v,t'}:=x^b_{v,t'}\cdot p^b_{v,u}$. We are omitting the subscript u because u is fixed. The following $B-1$ inequalities hold:

$$\sum_{t'=2}^{t''}\sum_{a\in A}\tilde{x}^a_{t'}\le\sum_{t'=1}^{t''-1}\sum_{(v,b)\in\mathsf{Par}(u)}\tilde{x}^b_{v,t'}\qquad t''=2,\dots,B \tag{F.1}$$

The algorithm updates the variables $\tilde{x}^a_{t'}$ and $\tilde{x}^b_{v,t'}$ over iterations $t = 2, \dots, B$, but we will inductively show that inequality t'' of (F.1) holds until the end of iteration t''. The algorithm can be described as follows:

Decomposition Algorithm

- Initialize all $q_{v,b,t',u,a,t} := 0$.
- For $t = 2, \dots, B$:
  - While there exists some $a \in A$ such that $\tilde{x}^a_t > 0$:
    1. Choose any non-zero $\tilde{x}^b_{v,t'}$, where $(v, b) \in \mathsf{Par}(u)$, with $t' < t$.
    2. Let $Q = \min\{\tilde{x}^a_t, \tilde{x}^b_{v,t'}\}$.
    3. Set $q_{v,b,t',u,a,t} := \frac{Q}{x^b_{v,t'} \cdot p^b_{v,u}}$.
    4. Subtract Q from both $\tilde{x}^a_t$ and $\tilde{x}^b_{v,t'}$.

Let's consider iteration t of the algorithm. The inequality of (F.1) with $t'' = t$ guarantees that there always exists such a non-zero $\tilde{x}^b_{v,t'}$ in Step 1. In Step 4, Q is subtracted from both the LHS and RHS of all inequalities of (F.1) with $t'' \geq t$, so these inequalities continue to hold. (Q is also subtracted from the RHS of inequalities of (F.1) with $t' < t'' < t$, so these inequalities might cease to hold.) This inductively establishes that all inequalities of (F.1) with $t'' \geq t$ hold during iteration t, and thus Step 1 of the algorithm is well-defined.

Now we show that (7.19b) is satisfied. Suppose on iteration t of the algorithm, we have some $\tilde{x}^a_t > 0$ and $\tilde{x}^b_{v,t'} > 0$ on Step 1. Note that $q_{v,b,t',u,a,t}$ must currently be 0, since if it was already set, then either $\tilde{x}^a_t$ or $\tilde{x}^b_{v,t'}$ would have been reduced to 0. Therefore, in Step 3 we are incrementing the LHS of (7.19b) by $x^b_{v,t'} \cdot p^b_{v,u} \cdot \frac{Q}{x^b_{v,t'} \cdot p^b_{v,u}} = Q$, after which we are subtracting Q from $\tilde{x}^a_t$ in Step 4. Since over iterations $t = 2, \dots, B$, for every $a \in A$, $\tilde{x}^a_t$ gets reduced from $x^a_{u,t}$ to 0, it must be the case that every equation in (7.19b) holds by the end of the algorithm.

For (7.19a), we use a similar argument. Fix some $(v, b) \in \mathsf{Par}(u)$ and $t' \in [B - 1]$. Whenever we add $\frac{Q}{x^b_{v,t'} \cdot p^b_{v,u}}$ to the LHS of (7.19a), we are reducing $\tilde{x}^b_{v,t'}$ by Q. Since $\tilde{x}^b_{v,t'}$ starts at $x^b_{v,t'} \cdot p^b_{v,u}$ and cannot be reduced below 0, the biggest we can make the LHS of (7.19a) is $\frac{x^b_{v,t'} \cdot p^b_{v,u}}{x^b_{v,t'} \cdot p^b_{v,u}} = 1$.

Finally, it is clear that the algorithm takes polynomial time, since every time we loop through Steps 1 to 4 either $\tilde{x}^a_t$ or $\tilde{x}^b_{v,t'}$ goes from non-zero to zero, and there were only a polynomial number of such variables to begin with. Other than the statement for $u \in \mathcal{B}$, this completes the proof of Lemma 7.4.1.

Now, if $u \in \mathcal{B}$, then we can strengthen (F.1). Indeed, substituting (7.10b) (instead of (7.10a)) into (7.12c), we get that all inequalities of (F.1) hold as equality. At the start of iteration $t = 2$, it is the case that $\tilde{x}^\alpha_t = \sum_{(v,b)\in \mathsf{Par}(u)} \tilde{x}^b_{v,t-1}$, and by the end of the iteration, it will be the case that $\tilde{x}^\alpha_t = 0$, and $\tilde{x}^b_{v,t-1} = 0$, $q_{v,b,t-1,u,\alpha,t} = 1$ for all $(v,b) \in \mathsf{Par}(u)$. As a result, the equalities of (F.1) with $t'' > t$ will continue to hold as equality. We can inductively apply this argument to establish that $q_{v,b,t',u,\alpha,t'+1} = 1$ for all $(v,b) \in \mathsf{Par}(u)$ and $t' \in [B-1]$, as desired.

F.4.2 Proof of Lemma 7.4.2.

We make use of the following conjecture of Samuels, which is proven for $n \le 4$ (see Samuels (1966, 1968)):

Conjecture F.4.1. *Let $X_1, \ldots, X_n$ be independent non-negative random variables with respective expectations $\mu_1 \ge \ldots \ge \mu_n$, and let $\lambda > \sum_{i=1}^n \mu_i$. Then $\Pr[\sum_{i=1}^n X_i \ge \lambda]$ is maximized when the X_i's are distributed as follows, for some index $k \in [n]$:*

- *For $i > k$, $X_i = \mu_i$ with probability 1.*
- *For $i \le k$, $X_i = \lambda - \sum_{\ell=k+1}^n \mu_\ell$ with probability $\frac{\mu_i}{\lambda - \sum_{\ell=k+1}^n \mu_\ell}$, and $X_i = 0$ otherwise.*

If we have $\mathbb{E}[Y_i] + \mathbb{E}[Y_j] \le \frac{t}{6}$ for $i \ne j$, then we can treat $Y_i + Y_j$ as a single random variable satisfying $\mathbb{E}[Y_i + Y_j] \le \frac{t}{6}$. By the pigeonhole principle, we can repeat this process until $n \le 3$, since $\sum_{j=1}^m \mathbb{E}[Y_j] \le \frac{t}{3}$. In fact, we assume n is exactly 3 (we can add random variables that take constant value 0 if necessary), so that we can apply Conjecture F.4.1 for $n = 3$, which has been proven to be true. We get that $\Pr[Y_1 + Y_2 + Y_3 \ge \frac{t}{2}]$ cannot exceed the maximum of the following (corresponding to the cases $k = 3, 2, 1$, respectively):

- $1 - (1 - \frac{\mu_1}{\frac{t}{2}})(1 - \frac{\mu_2}{\frac{t}{2}})(1 - \frac{\mu_3}{\frac{t}{2}})$
- $1 - (1 - \frac{\mu_1}{\frac{t}{2} - \mu_3})(1 - \frac{\mu_2}{\frac{t}{2} - \mu_3})$
- $1 - (1 - \frac{\mu_1}{\frac{t}{2} - \mu_2 - \mu_3})$

Now we employ Lemma 4 from Bansal et al. (2012) to bound these quantities:

Lemma F.4.2. *Let r and $p_{\max}$ be positive real values. Consider the problem of maximizing $1-\prod_{i=1}^{t}(1-p_i)$ subject to the constraints $\sum_{i=1}^{t} p_i \leq r$, and $0 \leq p_i \leq p_{\max}$ for all i. Denote the maximum value by $\beta(r, p_{\max})$. Then*

$$\begin{aligned}\beta(r, p_{\max}) &= 1-(1-p_{\max})^{\lfloor \frac{r}{p_{\max}} \rfloor}(1-(r-\lfloor \frac{r}{p_{\max}} \rfloor \cdot p_{\max})) \\ &\leq 1-(1-p_{\max})^{\frac{r}{p_{\max}}}\end{aligned}$$

Recall that $\mu_1, \mu_2, \mu_3 \leq \frac{t}{6}$ and $\mu_1+\mu_2+\mu_3 \leq \frac{t}{3}$.

- In the first case $k=3$, we get $p_{\max}=\frac{1}{3}$ and $r=\frac{2}{3}$, so the quantity is at most $\beta(\frac{2}{3}, \frac{1}{3}) \leq 1-(1-\frac{1}{3})^2=\frac{5}{9}$, as desired.

- In the second case $k=2$, for an arbitrary $\mu_3 \in [0, \frac{t}{6}]$, we get that the quantity is at most $\beta(\frac{\frac{t}{3}-\mu_3}{\frac{t}{2}-\mu_3}, \frac{\frac{t}{6}}{\frac{t}{2}-\mu_3}) \leq 1-\big(1-\frac{\frac{t}{6}}{\frac{t}{2}-\mu_3}\big)^{(\frac{t}{3}-\mu_3)/(\frac{t}{6})}$. It can be checked that the maximum occurs at $\mu_3=0$, so the quantity is at most $\frac{5}{9}$ for any value of $\mu_3 \in [0, \frac{t}{6}]$, as desired.

- In the third case $k=1$, we get that the quantity is at most $\frac{\mu_1}{\frac{t}{2}-(\frac{t}{3}-\mu_1)}=\frac{\mu_1}{\frac{t}{6}+\mu_1}$, which at most $\frac{1}{2}$ over $\mu_1 \in [0, \frac{t}{6}]$, as desired.

Therefore, Conjecture F.4.1 tells us that the maximum value of $\Pr[\sum_{j=1}^{m} Y_j \geq \frac{t}{2}]$ is $\frac{5}{9}$, completing the proof of Lemma 7.4.2.